

# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques

UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

# Planification en Distribution Urbaine:

Optimisation des tournées dans un contexte collaboratif

■ ZAHER AL CHAMI



# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques  
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

THÈSE présentée par  
**ZAHER AL CHAMI**

pour obtenir le  
Grade de Docteur de  
l'Université de Technologie de Belfort-Montbéliard

Spécialité : **Automatique**

**Planification en Distribution Urbaine:**  
Optimisation des tournées dans un contexte collaboratif

Unité de Recherche :  
FEMTO-ST/DISC/OMNI

Soutenue publiquement le 18 juillet 2018 devant le jury composé de :

Président	AZIZ MOUKRIM	Professeur des Universités, UTC
Rapporteur	CHRISTELLE GUÉRET	Professeure des Universités, Université d'Angers
Rapporteur	AMMAR OULAMARA	Professeur des Universités, Université de Lorraine
Examineur	HAOXUN CHEN	Professeur des Universités, UTT
Examineur	GRÉGORY CHIPPEAUX	Directeur de Share and move solutions
Directrice de thèse	MARIE-ANGE MANIER	Maître de Conférences HDR, UTBM
Co-directeur de thèse	HERVÉ MANIER	Maître de Conférences, UTBM



**THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE  
PREPAREE A L'Université de Technologie de Belfort-Montbéliard**

Ecole doctorale n°37

École Doctorale SPIM (Sciences Pour l'Ingénieur et Microtechniques)

Doctorat en Automatique

Par

M. AL CHAMI Zaher

Planification en Distribution Urbaine:

Optimisation des tournées dans un contexte collaboratif

Thèse présentée et soutenue à Belfort, le 18 juillet 2018

Composition du jury :

M., MOUKRIM, Aziz  
Me., GUÉRET, Christelle  
M., OULAMARA, Ammar  
M., CHEN, Haoxun  
M., CHIPPEAUX, Grégory  
Me., MANIER, Marie-Ange  
M., MANIER, Hervé

Professeur des Universités, UTC  
Professeure des Universités, Université d'Angers  
Professeur des Universités, Université de Lorraine  
Professeur des Universités, UTT  
Directeur de Share and move solutions  
Maître de Conférences HDR, UTBM  
Maître de Conférences, UTBM

Président  
Rapporteur  
Rapporteur  
Examineur  
Examineur  
Directrice de thèse  
Co-directeur de thèse

**Titre :** Planification en Distribution Urbaine: Optimisation des tournées dans un contexte collaboratif

**Mots clés :** Transport, Optimisation, Problèmes de tournées

**Résumé :** De nos jours, le transport joue un rôle clé dans la vie des pays modernes, en particulier pour les flux de marchandises. La logistique des flux entre régions, pays et continents a bénéficié d'innovations technologiques et organisationnelles assurant efficacité et efficience. Il n'en a pas été de même à l'échelle urbaine, plus particulièrement dans les centres-villes : la gestion des flux dans un environnement caractérisé par une forte densité démographique n'a pas encore véritablement trouvé son modèle d'organisation. Aujourd'hui, la logistique urbaine ou encore la gestion « du dernier kilomètre » constitue donc un enjeu de premier plan, tant socio politique et environnemental qu'économique. La logistique urbaine est caractérisée par la présence de plusieurs acteurs (chargeurs ou propriétaires de marchandises, clients, transporteurs, autorités publiques, ...) ayant chacun des priorités différentes (réduction de la pollution, amélioration de la qualité de service, minimisation de la distance totale parcourue, ...). Pour relever ces défis, un des leviers possibles consiste à optimiser les tournées de distribution et/ou collecte de marchandises, dans le contexte et sous les contraintes de la ville.

Le but de ce travail de thèse réside alors dans la planification de la distribution des marchandises dans un réseau logistique, abordée sous un angle de collaboration entre les chargeurs. Cette collaboration consiste à regrouper les demandes de divers chargeurs pour optimiser le taux de

chargement des camions et obtenir de meilleurs prix de transport. Ici, la gestion du « dernier kilomètre » s'apparente à ce que l'on identifie dans la littérature comme le Pickup and Delivery Problem (PDP). Dans le cadre de cette thèse, nous nous intéressons à des variantes de ce problème plus adaptées au contexte urbain. Après avoir réalisé un état de l'art sur les problèmes d'optimisation combinatoire autour du transport et les méthodes utilisées pour leur résolution, nous étudions deux nouvelles variantes du problème de collecte et de livraison : le *Selective PDP with Time Windows and Paired Demands* et le *Multi-periods PDP with Time Windows and Paired Demands*. La première permet aux transporteurs de livrer le maximum de clients dans une journée par exemple ; avec la seconde, et en cas d'impossibilité de livraison dans cette période, on détermine la meilleure date de livraison en minimisant la distance parcourue. Chacune d'elles fait l'objet d'une description formelle, d'une modélisation mathématique sous forme de programme linéaire, puis d'une résolution par des méthodes exacte, heuristiques et métaheuristiques, dans des cas mono-objectif et multi-objectifs. La performance de chaque approche a été évaluée par un nombre substantiel de tests sur des instances de différentes tailles issues de la littérature et/ou que nous avons générées. Les avantages et les inconvénients de chaque approche sont analysés, notamment dans le cadre de la collaboration entre chargeurs.

**Title :** Planning in Urban Distribution: Optimizing tours in a collaborative context

**Keywords :** Transport, Optimization, Routing problems

**Abstract :** Nowadays, transportation plays a key role in our modern countries' life, in particular for the goods flows. The logistics of flows between regions, countries and continents have benefited from technological and organizational innovations ensuring efficiency and effectiveness. It has not been the same at the urban scale, especially in city centers: the management of flows in a high population density environment has not yet found its organizational model. Today, urban logistics or "last mile" management is therefore a major issue, both socio-political and environmental as well as economic. Urban logistics is characterized by several actors (shippers or owners of goods, customers, carriers, public authorities, ...) each with different priorities (reduction of pollution, improvement of service quality, minimization of total distance traveled, ...). To overcome these challenges, one possible lever is to optimize the distribution and/or collection of goods in the context and under the constraints of the city.

The goal of this PhD work is then to plan the distribution of goods in a logistics network, approached from a collaboration angle between shippers. This collaboration consists in grouping the demands of several shippers to optimize the loading rate of the trucks and to obtain better

transport prices. Here, managing the "last mile" is similar to what is known in the literature as the Pickup and Delivery Problem (PDP). In this thesis, we are interested in variants of this problem more adapted to the urban context. After having realized a state of the art on the combinatorial optimization problems around the transport and the methods used for their resolution, we study two new variants of the problem of collection and delivery: the *Selective PDP with Windows and Paired Demands* and the *Multi-period PDP with Windows and Paired Demands*. The first allows carriers to deliver the maximum number of customers in a day for example; with the second, and in case of impossibility of delivery in this period, we determine the best delivery date by minimizing the distance traveled. Each of them is the subject of a formal description, of a mathematical modeling in the form of a linear program, then of a resolution by exact methods, heuristics and metaheuristics, in single-objective and multi-objective cases. The performance of each approach was evaluated by a substantial number of tests on instances of different sizes from the literature and / or that we generated. The advantages and drawbacks of each approach are analyzed, in particular in the context of collaboration between shippers.

## Remerciements

À l'issue de ces trois années de thèse, j'aimerais exprimer mes sincères remerciements aux personnes qui ont contribué à l'aboutissement de ce travail.

Je voudrais en tout premier lieu exprimer ma sincère gratitude à mes encadrants de thèse, Mme. Marie-Ange MANIER et M. Hervé MANIER, pour m'avoir permis d'effectuer cette thèse dans les meilleures conditions. Je leur témoigne toute ma reconnaissance pour leur soutien continu, leurs précieux conseils et leurs encouragements tout au long de cette thèse. Ce travail ne serait pas réussi sans leurs idées pertinentes. Je n'aurais pas pu imaginer avoir de meilleurs encadrants pour mon doctorat.

Je tiens à remercier les membres du jury d'avoir accepté de juger ce travail et de me donner des remarques constructives qui vont me permettre d'améliorer le manuscrit.

Je remercie également tous mes collègues de l'équipe OMNI, en particulier : Zhihao, Alexis, Lei et Emna, pour le long temps que nous travaillions ensemble avant les dates limites et pour toutes les discussions enrichissantes au cours des trois dernières années.

Mes sincères remerciements vont aussi à tous mes amis, proches ou éloignés, qui ont su me motiver, m'aider et me soutenir lors des périodes difficiles. En particulier : Mohamad SLEIMAN qui m'a toujours soutenu et encouragé pendant mon travail.

Un grand merci à toute ma famille pour leur soutien inconditionnel. Je remercie mes parents qui m'ont soutenu dans tous les choix que j'ai faits. Il ne faut pas oublier mes frères et ma sœur qui m'ont aidé et encouragé tout au long de cette thèse.

Ce travail n'aurait pas été possible sans le soutien financier de l'agence nationale de la recherche en France obtenu dans le cadre du projet ANR-TCDU (Transport Collaboratif en Distribution Urbaine).

Sachant qu'il est difficile d'établir une liste exhaustive, je tiens à adresser mes derniers remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail et qui ont fait de cette thèse une aventure riche.





# Table des matières

<b>Remerciements</b>	<b>i</b>
<b>Liste des tableaux</b>	<b>vii</b>
<b>Table des figures</b>	<b>ix</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 État de l'art</b>	<b>3</b>
1.1 Logistique urbaine . . . . .	4
1.1.1 Enjeux de la logistique urbaine . . . . .	4
1.1.2 Acteurs principaux . . . . .	8
1.1.3 Projets de recherche . . . . .	9
1.1.4 Projet ANR TCDU . . . . .	10
1.2 Logistique collaborative . . . . .	12
1.2.1 Collaboration entre les chargeurs . . . . .	13
1.2.2 Collaboration entre les transporteurs . . . . .	14
1.3 Problèmes scientifiques liés à la planification des tournées . . . . .	15
1.3.1 Problème du voyageur de commerce . . . . .	16
1.3.2 Vehicle Routing Problem (VRP) . . . . .	18
1.3.3 Team Orienteering Problem (TOP) . . . . .	21
1.3.4 Pickup and Delivery Problem (PDP) . . . . .	23
1.4 Variantes du PDP . . . . .	24
1.5 Méthodes de résolution . . . . .	26
1.5.1 Méthodes exactes . . . . .	27
1.5.2 Méthodes approchées . . . . .	29
1.6 Résolution des variantes du PDP . . . . .	32
1.7 Conclusion . . . . .	34
<b>2 Résolution du SPDPTWPD mono-objectif</b>	<b>35</b>
2.1 Une nouvelle variante de Pickup and Delivery Problem . . . . .	37
2.2 Formulation mathématique . . . . .	39
2.2.1 Notations . . . . .	40
2.2.2 MILP pour le SPDPTWPD . . . . .	40
2.2.3 Comparaison avec le VRP . . . . .	43
2.3 Méthodes exactes . . . . .	45
2.3.1 Modèle initial . . . . .	45
2.3.2 Modèle amélioré . . . . .	46
2.3.3 Résultats expérimentaux sur des problèmes de référence . . . . .	47
2.3.4 Génération d'instances pour le SPDPTWPD . . . . .	50

2.4	Méthodes approchées par regroupements . . . . .	52
2.4.1	Regroupement basé sur la distance euclidienne : . . . . .	53
2.4.2	Regroupement basé sur l'algorithme K-means : . . . . .	54
2.4.3	Procédure de réparation . . . . .	55
2.4.4	Résultats expérimentaux . . . . .	56
2.5	Piste d'amélioration . . . . .	58
2.6	Conclusion . . . . .	61
<b>3</b>	<b>Résolution du SPDPTWPD multi-objectifs</b>	<b>63</b>
3.1	Approche lexicographique . . . . .	65
3.1.1	Approches non Pareto . . . . .	65
3.1.2	Notre approche . . . . .	65
3.1.3	Résultats expérimentaux . . . . .	66
3.2	Algorithme du recuit simulé . . . . .	68
3.2.1	Opérateur PD-Shift . . . . .	70
3.2.2	Opérateur PD-Exchange . . . . .	71
3.2.3	Opérateur PD-Rearrange . . . . .	72
3.2.4	Fonction "comparaison" . . . . .	72
3.2.5	Fonction "optimisation finale" . . . . .	73
3.2.6	Résultats expérimentaux . . . . .	74
3.3	Algorithme génétique hybride . . . . .	76
3.3.1	Représentation de la solution . . . . .	77
3.3.2	Création de la population initiale . . . . .	77
3.3.3	Classement des chromosomes . . . . .	79
3.3.4	Procédure de sélection . . . . .	82
3.3.5	Croisement . . . . .	84
3.3.6	Hybridation avec des méthodes de recherche locale . . . . .	85
3.3.7	Test de dominance . . . . .	87
3.3.8	Algorithme global . . . . .	87
3.3.9	Résultats expérimentaux . . . . .	88
3.4	Conclusion . . . . .	96
<b>4</b>	<b>Résolution du Mu-PDPTWPD</b>	<b>99</b>
4.1	Introduction . . . . .	101
4.2	Formulation mathématique . . . . .	103
4.2.1	Notations . . . . .	103
4.2.2	MILP pour le Mu-PDPTWPD . . . . .	104
4.3	Étude mono-objectif . . . . .	106
4.3.1	Méthode exacte : MILP-Global . . . . .	106
4.3.2	Approches séquentielles . . . . .	107
4.3.3	Résultats expérimentaux . . . . .	111
4.4	Étude multi-objectifs . . . . .	115
4.4.1	Approche lexicographique . . . . .	116
4.4.2	Algorithme génétique hybride combiné avec GRASP . . . . .	117

---

4.4.3	Résultats expérimentaux . . . . .	124
4.5	Analyse de la collaboration entre chargeurs . . . . .	129
4.5.1	Contexte : Transport Collaboratif en Distribution Urbaine . . . . .	129
4.5.2	Collaboration et tournées mono-période . . . . .	129
4.5.3	Collaboration et tournées multi-périodes . . . . .	138
4.5.4	Bilan . . . . .	142
4.6	Conclusion . . . . .	142
<b>Conclusion et perspectives</b>		<b>145</b>
<b>Publications</b>		<b>149</b>
<b>Annexes</b>		<b>151</b>
A.1	Création d'instances pour le SPDPTWPD . . . . .	151
A.2	Création d'instances pour le Mu-PDPTWPD . . . . .	152
A.3	Création des instances utilisées dans la section 4.5 . . . . .	153
<b>Bibliographie</b>		<b>157</b>



# Liste des tableaux

1.1	Quelques projets de recherche . . . . .	10
1.2	Quelques variantes du VRP . . . . .	20
1.3	Quelques variantes du PDP . . . . .	33
2.1	Données de l'exemple . . . . .	44
2.2	Matrice de distances . . . . .	44
2.3	Solution détaillée . . . . .	45
2.4	Comparaison des modèles SM et IM . . . . .	48
2.5	Comparaison avec [Wang <i>et al.</i> 2015] . . . . .	49
2.6	Implémentation de la formulation dans [Wang <i>et al.</i> 2015] . . . . .	50
2.7	Premières instances pour le SPDPTWPD . . . . .	51
2.8	Méthodes de regroupement . . . . .	57
2.9	Nombre des variables et des contraintes . . . . .	60
2.10	Comparaison entre les deux formulations . . . . .	60
3.1	Résultats de l'approche lexicographique . . . . .	67
3.2	Différences entre les deux variantes . . . . .	72
3.3	Résultats de l'algorithme du recuit simulé . . . . .	75
3.4	Analyse des résultats obtenus . . . . .	76
3.5	Calcul des fitness . . . . .	81
3.6	Résultats sur les petites instances . . . . .	90
3.7	Résultats sur les instances moyennes . . . . .	92
3.8	Résultats sur les grandes instances . . . . .	93
3.9	Comparaison entre les trois approches . . . . .	94
3.10	Écart entre HGA et les deux autres approches . . . . .	95
3.11	Analyse des résultats obtenus . . . . .	96
4.1	Différences entre les variantes considérées . . . . .	101
4.2	Récapitulatif des études sur les variantes considérées . . . . .	102
4.3	Résultats expérimentaux pour le Mu-PDPTWPD mono-objectif . . . . .	113
4.4	Analyse des résultats obtenus . . . . .	114
4.5	Impact de la multiplication du nombre de périodes . . . . .	115
4.6	Résultats expérimentaux sur les ensembles 1 et 2 . . . . .	125
4.7	Résultats expérimentaux pour l'ensemble 4 . . . . .	128
4.8	Synthèse de solutions sur les trois périodes . . . . .	137
4.9	Distribution des demandes de Mono-i par période j : cas sans pénalités	138
4.10	Synthèse par demande pour l'instance multi-périodes sans pénalités . . . . .	139
4.11	Synthèse par chargeur pour l'instance multi-périodes sans pénalités . . . . .	139
4.12	Distribution des demandes de Mono-i par période j : cas avec pénalités	140
4.13	Synthèse par demande pour l'instance multi-périodes avec pénalités . . . . .	141

4.14 Synthèse par chargeur pour l'instance multi-périodes avec pénalités .	141
--	-----

# Table des figures

1.1	La logistique urbaine . . . . .	4
1.2	Émissions françaises de gaz à effet de serre depuis 1990 . . . . .	5
1.3	Les espaces logistiques urbains [Pre 2006] . . . . .	7
1.4	Acteurs principaux de la logistique urbaine . . . . .	8
1.5	Projet ANR TCDU . . . . .	11
1.6	Collaboration entre les chargeurs . . . . .	13
1.7	Collaboration entre les transporteurs . . . . .	14
1.8	Instance du TSP . . . . .	16
1.9	Variantes du TSP . . . . .	18
1.10	Exemple de solution du VRP . . . . .	19
1.11	Exemple de solution du TOP . . . . .	21
1.12	Exemple de solution du PDP . . . . .	24
1.13	Classification des variantes du General PDP . . . . .	25
1.14	Transport entre les sites et le dépôt . . . . .	26
1.15	Méthodes de résolution . . . . .	27
1.16	Fonctionnement d'un algorithme génétique . . . . .	31
2.1	Exemple de SPDPTWPD . . . . .	39
2.2	Contraintes additionnelles (C1) . . . . .	46
2.3	Contraintes additionnelles (C2) . . . . .	47
2.4	Contraintes additionnelles (C3) . . . . .	47
2.5	Évolution du temps de résolution de notre IM . . . . .	52
2.6	Exemple de regroupement . . . . .	53
2.7	Exemple d'application de l'algorithme K-means . . . . .	55
2.8	Procédure de résolution complète . . . . .	56
3.1	Approche lexicographique pour le SPDPTWPD . . . . .	66
3.2	Algorithme global utilisé par [Li & Lim 2003] . . . . .	70
3.3	Opérateur PD-Shift . . . . .	71
3.4	Opérateur PD-Exchange . . . . .	71
3.5	Opérateur PD-Rearrange . . . . .	72
3.6	Fonction "comparaison" . . . . .	73
3.7	Fonction "optimisation finale" . . . . .	74
3.8	Représentation d'une solution . . . . .	78
3.9	Attribution des rangs . . . . .	80
3.10	Première roulette de sélection . . . . .	83
3.11	Deuxième roulette de sélection . . . . .	84
3.12	Opérateur de croisement . . . . .	84
3.13	Enfant 1 après réparation . . . . .	85
3.14	Méthode "Improve" . . . . .	86

3.15	Exemple d'application pour la méthode auto-crossover-1 . . . . .	87
3.16	Schéma global de notre algorithme génétique hybride . . . . .	88
3.17	Comparaison des fronts : SPDPTWPD-RC201 . . . . .	91
3.18	Comparaison des fronts : SPDPTWPD-C202 à 2 véhicules . . . . .	91
4.1	Exemple de Mu-PDPTWPD . . . . .	103
4.2	Les approches séquentielles . . . . .	108
4.3	MILP-Séquentiel . . . . .	109
4.4	Lexicographique-Séquentiel . . . . .	110
4.5	HGA-Séquentiel . . . . .	111
4.6	Approche lexicographique pour le Mu-PDPTWPD . . . . .	116
4.7	Exemple de représentation d'une solution . . . . .	117
4.8	Attribution des rangs . . . . .	120
4.9	Procédure de sélection . . . . .	120
4.10	Opérateur de croisement . . . . .	121
4.11	Méthode "HGA-GRASP" . . . . .	123
4.13	Résultats expérimentaux pour l'ensemble 3 . . . . .	127
4.14	Visualisation des demandes pour l'instance Mono-1 . . . . .	130
4.15	Visualisation des tournées pour l'instance Mono-1 . . . . .	130
4.16	Visualisation des demandes pour l'instance Mono-2 . . . . .	133
4.17	Visualisation des tournées pour l'instance Mono-2 . . . . .	133
4.18	Visualisation des demandes pour l'instance Mono-3 . . . . .	135
4.19	Visualisation des tournées pour l'instance Mono-3 . . . . .	135
5.1	Relation entre type de collaboration et variante traitée . . . . .	145
5.2	Principales contributions . . . . .	146
A.1	Exemple d'une instance du SPDPTWPD . . . . .	151
A.2	Exemple d'une instance du Mu-PDPTWPD . . . . .	153
A.3	Exemple d'une instance Mono-x . . . . .	154
A.4	Création des instances multi-périodes . . . . .	154



# Introduction générale

Le transport de marchandises est une activité économique importante et représente un secteur en constant développement. En milieu urbain, une telle activité peut être source de pollution, bruit, congestion et autres perturbations pour les citoyens. La logistique urbaine ou encore la gestion «du dernier kilomètre» constitue donc un enjeu socio-politique et environnemental de premier plan qui impacte directement ou indirectement l'ensemble des acteurs concernés. Dans ce contexte, les objectifs des collectivités, principalement d'ordre fonctionnel et environnemental, sont parfois contradictoires avec ceux des entreprises concernées, plutôt d'ordre économique et organisationnel, d'où la nécessité de concilier les divers intérêts en jeu. Pour cela, plusieurs types d'actions sont envisageables, tels que la gestion de la voirie urbaine au travers de réglementations qui statuent sur les heures de circulation, les conditions de stationnement et les véhicules autorisés. Également, l'intervention directe sur les infrastructures joue un rôle important dans la gestion des flux en milieu urbain. Elle peut se traduire par la création de nouveaux Espaces Logistiques Urbains (ELUs) ou l'amélioration de l'existant. Ces installations géographiquement fixes sont utiles pour le stockage des marchandises et/ou leur transfert (groupage/dégroupage), dans un but de rationalisation du flux des véhicules de transport de biens. Néanmoins ils peuvent à la fois s'avérer coûteux et ne pas être adaptés aux demandes en évolution rapide et constante. Dans ce contexte, la planification des tournées est une alternative organisationnelle intéressante qui permet d'optimiser les déplacements tout en s'adaptant à l'évolution des flux de demandes, à l'évolution des infrastructures et de la législation urbaine.

Les travaux rapportés dans cette thèse s'inscrivent dans cette logique. Ils s'intègrent dans un projet soutenu par l'agence nationale de la recherche (ANR) et intitulé « Transport Collaboratif en Distribution Urbaine » (ANR TCDU). Ce projet ANR-14-CE220017 a été labellisé par le Pôle Véhicule du Futur. Il est effectué conjointement par quatre partenaires, les trois universités françaises de technologie (UTT, UTBM, UTC) et la société Share And Move Solutions. Il vise à proposer des modèles et des méthodes d'optimisation pour la planification collaborative du transport entre chargeurs et transporteurs dans la distribution urbaine, en appliquant les outils de la recherche opérationnelle, les principes de l'enchère combinatoire et la théorie des jeux coopératifs.

Plusieurs types de collaboration peuvent être envisagés mais nos travaux de recherche s'articulent autour de la collaboration entre les chargeurs dont le but est de mutualiser le transport de leurs marchandises. L'amélioration des taux de remplissage des véhicules devraient leur permettre d'obtenir des tarifs de transport plus avantageux. L'acheminement des marchandises des chargeurs vers leurs clients se concrétise par l'élaboration de tournées de véhicules d'un ou plusieurs transporteurs. Il s'agit d'un problème d'optimisation combinatoire bien connu dans la littérature sous le nom Vehicle Routing Problem. De manière plus précise, si nous considérons une tournée

comme la réalisation de plusieurs demandes de transport, celle-ci pourrait être vue comme une succession de trajets directs de fournisseurs vers des clients, en camions complets dans le cas d'un Lane Covering Problem, où la problématique se situe dans le repositionnement des véhicules à vide. Mais dans un contexte urbain, du fait de la proximité géographique des sites à visiter et des faibles quantités transportées, elle s'apparente plutôt à une imbrication de ces trajets directs. Ainsi, pour prendre en compte à la fois certaines contraintes urbaines et les divers types de demandes (collectes et/ou livraisons), nous nous focalisons dans cette thèse sur l'étude d'une variante du VRP qui s'appelle le Pickup and Delivery Problem (PDP). Nos travaux portent plus particulièrement sur deux nouvelles variantes du PDP : la première introduit l'aspect sélectif qui prend en compte le fait qu'un transporteur ne puisse pas honorer toutes les demandes à satisfaire dans une période donnée, et se trouve dans l'obligation de choisir celles qu'il pourra réaliser ; la seconde est une variante multi-périodes, qui repose sur la possibilité que la demande d'un chargeur soit collectée/livrée sur plusieurs périodes (et non sur une période fixe), dans le but de favoriser la collaboration entre chargeurs.

Ce mémoire est divisé en 4 chapitres : Le premier chapitre introduit les notions de la logistique urbaine, au travers de ses enjeux et de quelques-uns des projets réalisés dans ce domaine. Il présente un état de l'art succinct des problèmes scientifiques liés à la planification de tournées, en détaillant quelques variantes du problème de collecte et de livraison. Ce chapitre se termine par une description des principales méthodes utilisées par la communauté scientifique afin de fournir des solutions à ces problèmes.

Les deuxième et troisième chapitres portent sur la modélisation et la résolution du problème de collecte et livraison sélectif, avec fenêtres temporelles et demandes appairées (Selective PDP with Time Windows and Paired Demands ou SPDPTWPD), dans un contexte mono-objectif (chapitre 2) puis multi-objectifs (chapitre 3). Dans les deux cas, une formulation mathématique est tout d'abord proposée sous forme d'un programme linéaire. Ensuite, nous abordons la résolution du problème par diverses méthodes exactes et approchées, heuristiques ou métaheuristiques. Celles-ci sont testées sur des instances de la littérature sur des problèmes proches, ainsi que sur de nouvelles instances que nous avons générées. À partir des résultats obtenus, nous analysons les avantages et les inconvénients de chacune de nos méthodes.

Le quatrième chapitre introduit le problème de collecte et livraison multi-périodes, avec fenêtres temporelles et demandes appairées (Multi-periods PDPTWPD ou Mu-PDPTWPD). Pour traiter cette variante, nous appliquons la même démarche que dans les deux chapitres précédents. De plus, les algorithmes développés nous permettent d'analyser l'intérêt potentiel et les limites d'une collaboration entre chargeurs.

Le rapport se termine par une conclusion qui synthétise les résultats de toutes nos études, met en évidence nos contributions, et ouvre quelques perspectives sur de futurs travaux de recherche.

# État de l'art

*Ce chapitre fournit les éléments de base pour la compréhension du mémoire. Il présente la logistique urbaine, ses enjeux et ses acteurs, ainsi qu'un aperçu de quelques projets de recherche dans ce domaine. Une description plus détaillée est introduite pour le projet ANR TCDU avec une attention particulière sur le sujet de la thèse, et sur les différents types de collaboration que nous pouvons rencontrer dans le contexte de la logistique. Les problèmes scientifiques en relation avec la planification des tournées sont présentés, avec un focus sur la collecte et la livraison qui constitue le problème de base pour les variantes traitées pendant cette thèse. Enfin une synthèse des différentes méthodes de résolution utilisées dans la littérature est donnée, notamment pour les variantes qui nous intéressent.*

## Sommaire

<b>1.1</b>	<b>Logistique urbaine</b>	<b>4</b>
1.1.1	Enjeux de la logistique urbaine	4
1.1.2	Acteurs principaux	8
1.1.3	Projets de recherche	9
1.1.4	Projet ANR TCDU	10
<b>1.2</b>	<b>Logistique collaborative</b>	<b>12</b>
1.2.1	Collaboration entre les chargeurs	13
1.2.2	Collaboration entre les transporteurs	14
<b>1.3</b>	<b>Problèmes scientifiques liés à la planification des tournées</b>	<b>15</b>
1.3.1	Problème du voyageur de commerce	16
1.3.2	Vehicle Routing Problem (VRP)	18
1.3.3	Team Orienteering Problem (TOP)	21
1.3.4	Pickup and Delivery Problem (PDP)	23
<b>1.4</b>	<b>Variantes du PDP</b>	<b>24</b>
<b>1.5</b>	<b>Méthodes de résolution</b>	<b>26</b>
1.5.1	Méthodes exactes	27
1.5.2	Méthodes approchées	29
<b>1.6</b>	<b>Résolution des variantes du PDP</b>	<b>32</b>
<b>1.7</b>	<b>Conclusion</b>	<b>34</b>

## Publications

Al Chami, Z., Manier, H., & Manier, M.-A.. "Modélisation des problèmes de tournées riches en logistique urbaine", *Journées STP du GDR MACS session du Groupe de Travail Bermudes*, Nancy, (9-10 novembre 2017).

## 1.1 Logistique urbaine

La logistique urbaine (ou la distribution urbaine) se définit comme la façon d'acheminer dans les meilleures conditions les flux de marchandises qui entrent, sortent et circulent dans la ville [Patier & Routhier 2009]. Elle implique de multiples acteurs (agents économiques, institutions publiques, habitants, etc...) qui ont parfois des objectifs contradictoires (pour plus de détails, voir la sous-section 1.1.2). En France et contrairement au transport de personnes, le transport des marchandises en ville a longtemps souffert d'un manque de considération du point de vue scientifique et politique [Routhier *et al.* 2002]. C'est la loi sur l'air et l'utilisation rationnelle de l'énergie du 30 décembre 1996 qui a déclenché la prise en compte des marchandises pour la première fois au niveau de l'organisation globale des déplacements dans les grandes villes [Durand *et al.* 2010]. De nos jours, la distribution urbaine représente désormais un axe de recherche qui ne peine plus à rassembler les parties prenantes, privées ou publiques.

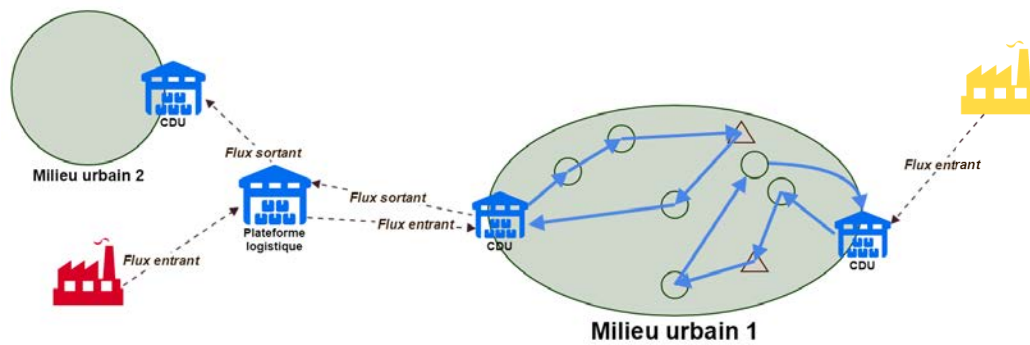


FIGURE 1.1 – La logistique urbaine

La figure 1.1 résume le concept de la logistique urbaine. Les flux entrants de marchandises sont ceux qui arrivent de l'extérieur et seront distribués dans le milieu urbain après le passage par un ou plusieurs centres de distribution urbains (CDUs) si nécessaire [Gonzalez-Feliu 2013]. Par contre, les marchandises collectées en ville seront expédiées vers l'extérieur (les flux sortants) ou vers d'autres sites dans le même milieu urbain (les flux intra-urbains sont représentés par les traits pleins dans la figure 1.1). Dans la suite de cette section, nous définissons les principaux enjeux du transport de marchandises en ville.

### 1.1.1 Enjeux de la logistique urbaine

#### Réduire la congestion et améliorer l'accessibilité :

Dans de nombreuses villes, l'insuffisance des infrastructures nécessaires à la bonne circulation de véhicules cause des problèmes d'accessibilité. Outre la configuration du réseau urbain limitant la taille et donc la capacité des véhicules de transport de marchandises, par exemple en hypercentre, il manque un nombre important de

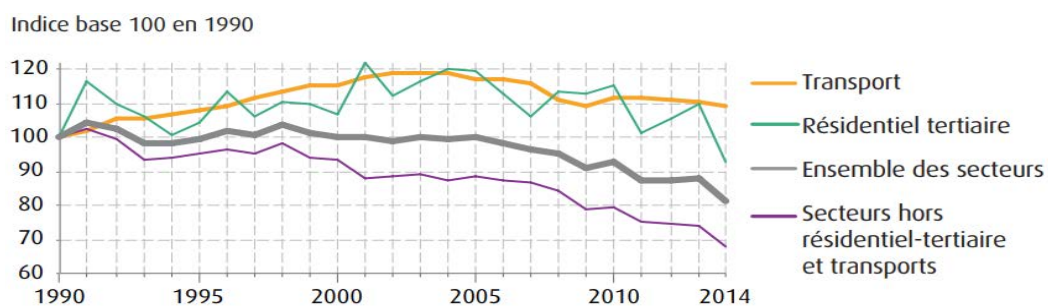
places réservées aux opérations de déchargement et de chargement effectuées par les véhicules qui transportent les marchandises. De plus, les places existantes sont souvent occupées d'une manière illicite par les autres catégories de véhicules, ce qui oblige les véhicules de marchandises à se garer en double file. Un tel comportement génère voire aggrave la congestion du trafic routier et nuit à la sécurité des usagers de l'espace public ainsi qu'à celle des chauffeurs.

De nombreuses communes adoptent des politiques spéciales qui imposent des restrictions d'accès dans des zones déterminées. Ces limitations sont souvent liées aux horaires des opérations, au poids ou à la taille des véhicules mais aussi aux normes de sécurité, et de pollution atmosphériques et sonores... Par exemple, pour des raisons de sécurité, les camions ne sont pas autorisés à collecter le verre aux éco-points situés à proximité d'écoles pendant les horaires d'entrée et sortie des élèves. De même, les camions effectuant des livraisons très tôt le matin ne doivent pas dépasser un certain niveau sonore (60dB, certification PIEK [PIE 2013]).

Par ailleurs la saturation du réseau constitue une difficulté majeure supplémentaire. Elle est principalement causée par l'augmentation du nombre de voitures qui circulent dans les villes.

### Réduire la pollution :

Le transport routier de marchandises, et plus particulièrement celui en milieu urbain, contribue aux émissions de polluants atmosphériques qui impactent directement la qualité de l'air dans les villes. La majorité des opérations dans le cadre de la logistique urbaine se fait avec des petits véhicules qui sont souvent très polluants. Par exemple, d'après la charte en faveur d'une logistique urbaine durable actualisée le 18 septembre 2013, 20% des véhicules en circulation dans Paris sont dédiés au transport de marchandises, ce qui représente environ 1,5 million de mouvements (livraisons et collecte) par semaine.



Source : Gitepa, rapport Secten mai 2015

FIGURE 1.2 – Émissions françaises de gaz à effet de serre depuis 1990

La figure 1.2 donne l'évolution des émissions françaises de gaz à effet de serre (GES) depuis 1990 par secteur d'activité. Nous pouvons remarquer que le transport (de marchandises et de personnes) y est le plus gros contributeur. Dans la distribution

urbaine, le transport de marchandises participe pour une part importante aux émissions de GES (entre 35% et 50%). Il prend également une part importante dans les émissions polluantes d'oxydes d'azote  $NO_x$  (environ 70%) et de particules fines PM10 (32%) [Lal 2015].

Pour remédier à cette situation, ou du moins en limiter les effets, plusieurs solutions peuvent être mises en œuvre, par exemple l'utilisation de véhicules électriques pour le transport des marchandises. En France, La Poste (le groupe français de collecte et de distribution du courrier) dispose d'une flotte composée de 30000 véhicules électriques qui est considérée comme la plus importante flotte de véhicules électriques au monde [Lap 2018].

### Localiser les infrastructures logistiques :

La première infrastructure à laquelle on peut penser pour les mouvements de biens est le réseau routier urbain. Ce réseau est toutefois partagé avec les flux de personnes, ce qui rend la cohabitation parfois difficile en fonction de la densité du trafic et du comportement des usagers.

Dans la suite, nous nous concentrons sur les autres types d'infrastructures urbaines dédiées aux marchandises et que nous divisons en deux catégories : les infrastructures logistiques (entrepôts, plateformes de distribution, etc...) et les infrastructures de transport (gares routières et ferroviaires, ports, aéroports, parkings, etc...). Ces infrastructures ne sont pas forcément adaptées pour les volumes importants de marchandise en transit quotidiennement sur le réseau. Pour résoudre ce problème, l'implantation des nouvelles infrastructures logistiques est indispensable.

Dans un contexte urbain, les infrastructures logistiques sont connues sous le nom Espaces Logistiques Urbains (ELUs). Par définition, les ELUs représentent des interfaces permettant de faciliter les relations entre les envois et les réceptions, entre la voirie et le lieu d'exploitation, entre la ville et sa périphérie. Ils ont vocation à recomposer les flux urbains et ils sont divisées en plusieurs types [Pre 2006] comme illustré dans la figure 1.3 :

1. Le **Zone Logistique Urbaine (ZLU)** : Elle concerne un vaste espace (la ville). Son but est de localiser les professionnels du transport à proximité de la ville pour limiter les mouvements de véhicules. Les marchés d'intérêt national, tel celui de Rungis desservant Paris, font partie des ZLUs.
2. Le **Centre de Distribution Urbain (CDU)** : Il concerne un périmètre plus restreint, souvent l'hyper-centre, un centre historique (La Rochelle, Bristol, Monaco, Padoue...). Il sert à la centralisation des opérations avec délégation de la desserte à un seul opérateur.
3. Le **Point d'Accueil des Véhicules (PAV)** : Il concerne un quartier. Il met à disposition des transporteurs des places de parking réservées et sécurisées pour éviter le stationnement perturbateur durant l'opération de livraison ou enlèvement. Les derniers mètres sont ensuite réalisés à pied.
4. Le **Point d'Accueil des Marchandises (PAM)** : Il représente un point de

dépôt des marchandises en l'absence des clients qui viennent ensuite y récupérer leurs produits. Par exemple, les points relais (presse, magasin...).

5. La **Boîte Logistique Urbaine (BLU)** : Elle permet de déconnecter le transporteur du client. L'idée consiste à déposer les envois dans une "boîte" de type boîte au lettre, consigne dans les gares par exemple, en dehors de la présence du client. La BLU nécessite un outil technologique pour sécuriser et transférer les données en temps réel.

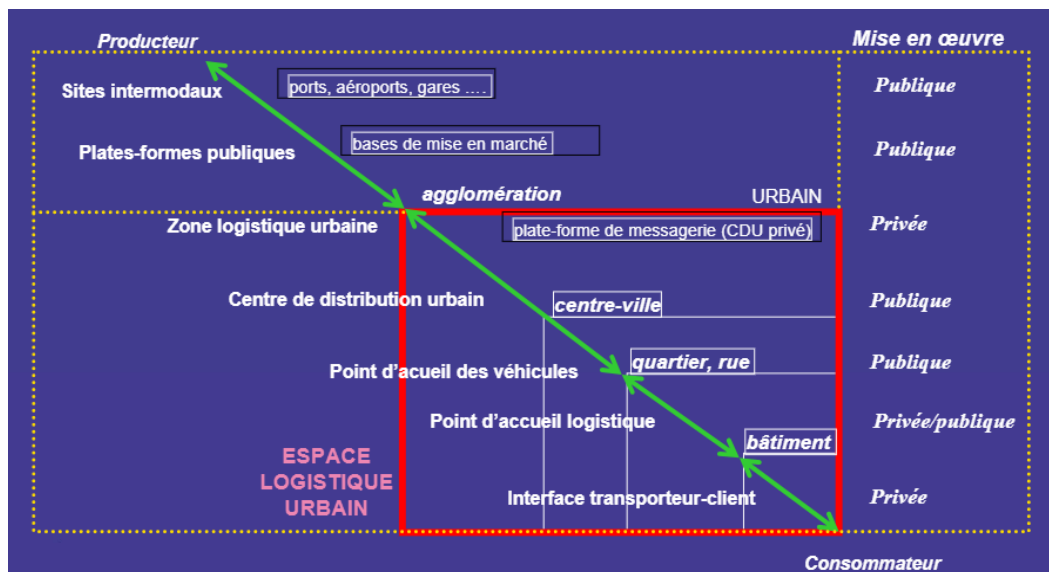


FIGURE 1.3 – Les espaces logistiques urbains [Pre 2006]

Plusieurs problèmes de localisation de ces infrastructures peuvent être rencontrés dans le contexte de la logistique urbaine. En effet, la localisation des nouveaux ELUs est le résultat d'un processus décisionnel qui prend en considération plusieurs critères :

- i) Le potentiel d'activité du transport des marchandises en ville.
- ii) Le prix de l'immobilier et du foncier ainsi que son accessibilité pour mieux capter les flux de marchandises.
- iii) La distance entre les ELUs et les clients finaux.
- iv) La nécessité de sécuriser la marchandise jusqu'à la livraison aux clients finaux.

Pour mes travaux de thèse, je vais utiliser le mot "dépôt" pour représenter un CDU. Le dépôt sera donc considéré comme un lieu de centralisation de flux entrants et sortants dans lequel les marchandises seront déposées. De plus, il représente le point de départ pour une flotte des véhicules afin d'effectuer les opérations de livraison et/ou de collecte. Les véhicules sont supposés revenir au dépôt à la fin de leurs tournées.

### Autres enjeux :

D'autres enjeux peuvent être liés à la logistique urbaine. Nous pouvons citer par exemple la réduction des nuisances sonores liées à cette activité. Ces nuisances sont principalement causées par les véhicules et par les opérations de chargement et de déchargement des marchandises. D'autres formes de nuisances (les odeurs, la gêne physique et les vibrations) affectent quotidiennement la qualité de la vie en ville.

Les enjeux cités dans cette partie n'ont pas la même priorité pour les acteurs de la logistique urbaine que nous présentons dans la sous-section suivante. Par exemple, la réduction de la pollution est plus importante pour les institutionnels que pour les habitants de la ville qui seront intéressés plutôt par la réduction des nuisances sonores.

### 1.1.2 Acteurs principaux

La logistique urbaine englobe plusieurs acteurs interdépendants dont les intérêts sont parfois divergents. Parmi les principaux acteurs, nous pouvons citer :

- (a) Les chargeurs (*en anglais shippers*) : ils désignent les détenteurs du fret et sont à l'origine de la demande de transport. Nous considérons dans les chapitres suivants que les chargeurs sont les fournisseurs des marchandises à transporter, ce qui n'est pas toujours le cas, notamment en transport international.
- (b) Les clients (*en anglais customers*) : les commerçants et les artisans représentent la plus grande part des destinataires dans le contexte urbain. Les particuliers représentent aussi une part importante de destinataires suite aux développements des services de livraison à domicile et du e-commerce.
- (c) Les transporteurs (*en anglais carriers*) : ce sont des entreprises spécialisées dans le transport de marchandises. Ils détiennent et gèrent des flottes de véhicules.
- (d) Les institutionnels : cette catégorie regroupe les décideurs publics qui n'interviennent généralement que par le seul biais de la réglementation et les collectivités qui cherchent à mieux gérer les flux en ville. Leur principal enjeu est de limiter l'impact environnemental et la sujétion de la voirie.
- (e) La population : cette catégorie englobe les habitants et les usagers de la zone urbaine. Ils représentent une grande partie des clients finaux. En même temps, ils représentent les utilisateurs du système de transport.

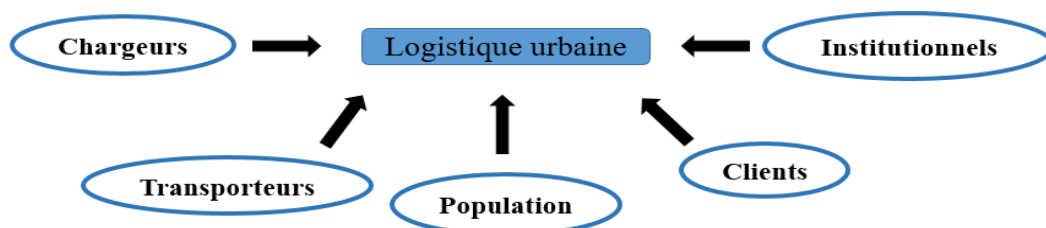


FIGURE 1.4 – Acteurs principaux de la logistique urbaine



La logistique urbaine représente un système complexe dont les intérêts de chaque acteur sont souvent difficiles à concilier. Plus il y a de demandes, plus il y a de trafic. Cela permet aux chargeurs et transporteurs d'accroître leur chiffre d'affaire, mais cela pose problème aux institutionnels et à la population qui souhaitent une amélioration de la fluidité du trafic. Le problème est complexe mais des solutions ont déjà été introduites, en particulier dans le cadre de diverses expérimentations et projets de recherche. La partie suivante résume des projets qui visent à apporter des solutions aux problèmes de la logistique du dernier kilomètre.

### 1.1.3 Projets de recherche

De nombreux projets de recherche ont été financés par différentes organisations nationales et internationales afin d'apporter des solutions aux problèmes liés à la distribution urbaine. Dans cette partie, notre but est d'en fournir une vue globale sur la base de quelques projets récents.

Le tableau 1.1 présente une liste des projets dont quelques uns sont déjà clos ainsi que d'autres qui sont actuellement en cours, avec leurs sources de financement. Les principaux objectifs pour ces projets sont :

- (a) Mesurer l'empreinte environnementale du transport et de la logistique dans les zones urbaines (par exemple : CONCORD, COFRET).
- (b) Proposer des stratégies pour obtenir des mouvements de marchandises plus propres qui économisent de l'énergie dans les zones urbaines (par exemple : C-LIEGE, DELIVER).
- (c) Développer des outils d'aide à la décision pour aider les différents acteurs à collaborer ensemble (par exemple : EVERS, MISC, TCDU).
- (d) Utiliser les nouveautés technologiques et les implémenter dans les systèmes de transport (par exemple : OCAFTCICC, CVIS).

Ces projets ont pour but de répondre aux enjeux de la logistique urbaine déjà abordés. Nous pouvons classer ces projets en deux catégories : la première consiste à trouver les solutions en intervenant directement sur les infrastructures (implantation des nouvelles plateformes, utilisation des nouvelles technologies, etc...). Par contre, la deuxième vise à intervenir sur les flux des marchandises afin d'améliorer les taux de remplissage des moyens de transport (problèmes de planification des tournées). Le présent travail de thèse concerne la seconde catégorie et fait partie du projet ANR TCDU que nous détaillons dans la partie suivante.

Acronyme	Titre complet	Durée	Source de financement	Objectifs
CVIS	Co-operative vehicle - infrastructure Systems	Février 2006 - Janvier 2010	Union européenne	d
BESTFACT	Best Practice Factory for Freight Transport	Septembre 2008 - Février 2011	Union européenne	a et b
SUGAR	Sustainable Urban Goods logistics Achieved by Regional and local policies	Novembre 2008 - Février 2012	Union européenne	c
CITYLOG	Sustainability and efficiency of city logistics	Janvier 2010 - Décembre 2012	Union européenne	d
MODUM	Mutualisation et Optimisation de la distribution Urbaine de Marchandises	Décembre 2010 - Juin 2014	ANR (France)	c et d
DELIVER	Design of electric light vans for environment impact reduction	Novembre 2011 - Octobre 2014	Union européenne	b
CYCLELOGISTICS		Mai 2011 - Avril 2014	Union européenne	b
C-LIEGE	Clean last mile transport and logistics management	Juin 2011 - Novembre 2013	Union européenne	b
COFRET	Carbon footprint of freight transport	Juin 2011 - Novembre 2013	Union européenne	a
CONCOORD	CONsolidation and COORDination in Urban Areas	Janvier 2013 - Décembre 2015	Union européenne en collaboration avec le programme DINALOG (Pays-Bas)	a
LAMILO	Last milelogistics	Janvier 2013 - Mars 2015	Union européenne	d
FREVUE	validating freight electric vehicles in urban Europe	2013 - 2017	Union européenne	a
ANNONA	Aide à la décision pour le développement de schémas logistiques urbains durables	Novembre 2013 - Mai 2018	ANR (France)	c
TCDU	Transport Collaboratif dans la Distribution Urbaine	Octobre 2014 - Mars 2019	ANR (France)	c
MISC	Mobility In Smart Cities	Septembre 2016 - Septembre 2019	Région Franche-Comté (France)	c
EVERS	Ordonnancement et routage des véhicules électriques pour la logistique urbaine : défis et challenges	Janvier 2016 - Janvier 2019	ANR (France)	c
UGMLCAP	Urban Goods Movement and Local Climate Action Plans : Assessing strategies to reduce greenhouse gas emissions from urban freight transportation	Octobre 2017 - Février 2019	Department of Transportation (USA)	a
OCAFTCICC	Operations of Connected and Autonomous Freight Trucks under Congestion and Infrastructure Cost Considerations	Novembre 2017 - Septembre 2022	Department of Transportation (USA)	d

TABLE 1.1 – Quelques projets de recherche

#### 1.1.4 Projet ANR TCDU

Cette thèse s'effectue dans le cadre du projet ANR TCDU (ANR-14-CE22-0017) : Transport Collaboratif dans la Distribution Urbaine. Ce projet est labellisé par le Pôle Véhicule du Futur, et est réalisé conjointement par quatre partenaires, les trois

universités françaises de technologie (UTT, UTBM, UTC) et la société Share And Move Solutions.

Ce projet porte sur le transport collaboratif en ville qui apparait comme un moyen efficace pour consolider davantage les flux logistiques au-delà des plates-formes logistiques dans une ville. Selon cette approche, plusieurs chargeurs ou transporteurs collaborent entre eux en regroupant et en partageant leurs demandes de transport ou en partageant leurs moyens de transport afin d'augmenter les taux de remplissage des véhicules.

En appliquant les outils et méthodes de la recherche opérationnelle, les enchères combinatoires et la théorie des jeux coopératifs, ce projet vise à développer des modèles et des méthodes d'optimisation pour la planification collaborative du transport dans la distribution urbaine. Il étudie la synergie entre la consolidation des flux logistiques à travers les centres de distribution urbains et la consolidation des flux grâce à la collaboration entre chargeurs et transporteurs. Sur la base de ces modèles et méthodes, un prototype de plate-forme Web et un prototype d'outil d'aide à la décision seront développés pour faciliter la collaboration entre chargeurs et transporteurs dans un réseau de collaboration dynamique. Les résultats du projet peuvent aider ces acteurs à réduire leurs coûts logistiques et contribuer au développement durable d'une ville. Le projet est découpé en 5 tâches :

- (a) Tâche 1 : Développement des modèles et des méthodes de planification centralisés pour la collaboration entre chargeurs
- (b) Tâche 2 : Développement des modèles et des méthodes pour les problèmes de tournées avec profits
- (c) Tâche 3 : Utilisation des approches par enchères combinatoire pour la collaboration entre transporteurs
- (d) Tâche 4 : Conception de mécanismes d'affectation coût/profit pour la collaboration entre chargeurs et transporteurs
- (e) Tâche 5 : Élaboration des prototypes de plateforme web et de système d'aide à la décision pour la collaboration entre chargeurs et transporteurs

La figure 1.5 montre la répartition des tâches du projet entre les quatre partenaires.

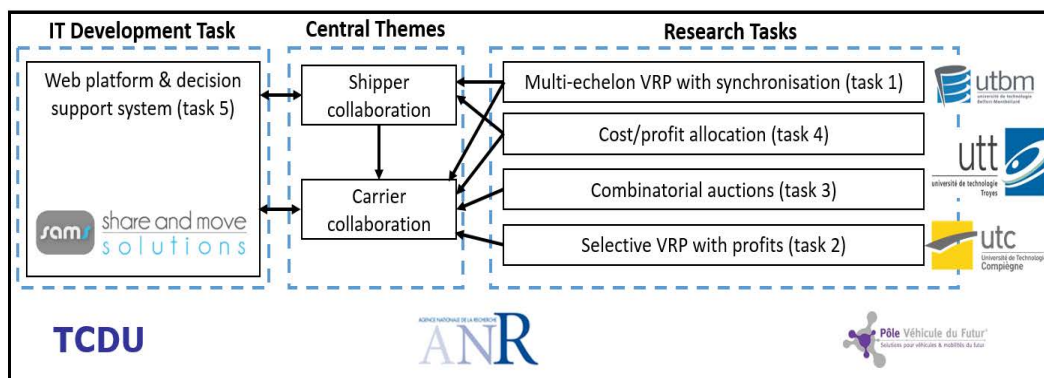


FIGURE 1.5 – Projet ANR TCDU

Le projet s'articule de la façon suivante : les chargeurs déposent sur une plateforme web leurs demandes de transport en précisant le lieu et la date de collecte de telles quantités de marchandises ainsi que le lieu et la date de livraison. A partir de ces informations, le système d'aide à la décision associé à la plateforme regroupe les demandes en plusieurs lots en ayant pour objectif de minimiser la distance totale parcourue dans le but de pouvoir garantir aux chargeurs le meilleur prix (coût). Une fois ce regroupement terminé, les transporteurs accèdent à la plateforme pour choisir les lots qui les intéressent. Un lot peut être choisi alors par plusieurs transporteurs. S'enclenche alors un processus d'enchères en plusieurs tours pour déterminer le transporteur qui remportera le lot. Ce processus permet lui aussi de diminuer le coût pour les chargeurs. Une fois le lot remporté, le transporteur doit assurer la prestation. Pour l'aider dans cette tâche, la plateforme lui permet d'organiser le mieux possible ses tournées. En effet, ce transporteur a non seulement le lot de demandes remportées aux enchères à réaliser mais aussi toutes les autres demandes qu'il a obtenues et qui ne sont pas passées par l'intermédiaire de la plateforme. Cette dernière mettra à disposition des outils qui lui permettront de planifier ses tournées en fonction du nombre de camions dont il dispose.

Plusieurs algorithmes sont développés dans la littérature permettant de déterminer un plan de transport commun optimal ou quasi-optimal entre plusieurs chargeurs tout en minimisant leurs coûts totaux de transport. La plupart des méthodes existantes ne peuvent traiter que les problèmes à un seul échelon, mais lorsque la distribution urbaine passe par des ELUs, nous devons résoudre des problèmes de routage de véhicules multi-échelons plus complexes. En outre, le choix des infrastructures logistiques urbaines pour le passage des flux est une décision supplémentaire à prendre. Ainsi, la tâche 1 est à son tour divisée en deux sous-tâches : la première consiste à traiter les problèmes de localisation des ELUs et de synchronisation des tournées. Par contre, la deuxième qui correspond à mes travaux de thèse vise à traiter les problèmes de planification de tournées en milieu urbain.

Puisque l'idée principale de notre projet est basée sur la notion de logistique collaborative, la section suivante introduit les deux types du transport collaboratif considérés : la collaboration entre les chargeurs et celle entre les transporteurs.

## 1.2 Logistique collaborative

Au cours des dernières années, plusieurs stratégies ont été développées afin d'accroître l'efficacité du transport de marchandises, et plus précisément dans le contexte de la planification et de la gestion de la logistique urbaine [Benjelloun *et al.* 2010]. La collaboration est l'un des domaines d'étude les plus prometteurs et peut se dérouler à plusieurs étapes de la chaîne et à différents niveaux d'interaction [Yearwood 2010]. Elle peut se définir comme un ensemble d'interactions entre deux ou plusieurs acteurs dans le but d'atteindre un ou des objectifs communs et d'en partager les bénéfices [Gonzalez-Feliu & Salanova 2012]. La logistique collaborative est classifiée selon deux grands axes [Salmon 2011] :

- La collaboration verticale qui implique des acteurs à différents niveaux de la chaîne logistique (par exemple, collaboration entre chargeurs et transporteurs)
- La collaboration horizontale qui concerne des acteurs de même niveau (par exemple, entre chargeurs).

La plupart des études sur la logistique collaborative se limitent à la collaboration verticale. Le développement des collaborations horizontales est beaucoup plus récent [Log 2014]. Nos travaux de recherche dans le cadre de projet ANR TCDU visent à étudier la collaboration horizontale entre les chargeurs (la collaboration entre transporteurs étant abordée par les autres partenaires du projet).

### 1.2.1 Collaboration entre les chargeurs

Ce type de collaboration se produit lorsque les demandes de transport de plusieurs chargeurs sont consolidées et desservies par un seul transporteur. En effet, la mutualisation des moyens de transport aide les chargeurs à réaliser des économies en obtenant des tarifs de transport avantageux.

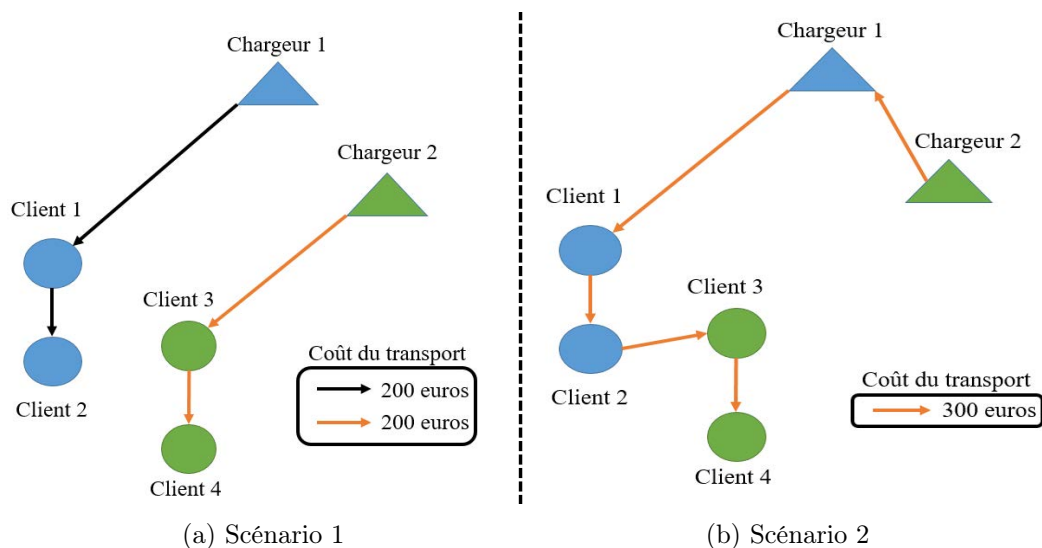


FIGURE 1.6 – Collaboration entre les chargeurs

La figure 1.6 montre un exemple de 2 chargeurs avec 2 demandes de transport pour chaque chargeur. Sans aucune collaboration (scénario 1), le coût demandé par chaque transporteur est de 200 euros. En cas de collaboration (scénario 2), les deux chargeurs économisent 100 euros en total. Les 2 chargeurs ont mis en commun leurs demandes de transport pour n'en faire qu'une et la proposer à un transporteur. Celui-ci a alors pu utiliser un seul camion avec un meilleur taux de remplissage, et une diminution de la distance totale parcourue donc du coût de transport.

Plusieurs obstacles limitent potentiellement une telle collaboration. Le problème principal est de déterminer un plan de transport collaboratif qui regroupe toutes les demandes de transport de sorte que le coût total de transport soit réduit au

minimum. De plus, il faut trouver une façon d'allouer équitablement les économies de coûts générées par la collaboration entre tous ses participants. Cela suppose aussi au préalable un partage d'informations et une coordination entre les chargeurs.

Le nombre de travaux traitant la collaboration des chargeurs est assez restreint. Ce problème a été introduit dans [Ergun *et al.* 2007] comme étant un LCP (Lane Covering Problem) dans lequel chaque envoi régulier entre une origine et une destination est représenté par une liaison. L'objectif du problème est de trouver un ensemble de routes qui couvrent toutes les liaisons (réalisent toutes les demandes) afin que le coût total de transport soit minimisé. Dans un LCP le véhicule ne peut transporter qu'une seule demande à la fois (full truckload) ce qui oblige le point de livraison à être visité directement après le point de collecte pour chaque requête. Quelques études ultérieures ont traité cette collaboration, par exemple : [Dias & Tsuzuki 2010], [Pan *et al.* 2013], etc...

Ce problème est abordé dans la tâche 1 du projet ANR TCDU, mais il doit être adapté à la distribution urbaine où les quantités échangées sont plus faibles. Dans ce contexte, il faut considérer la possibilité qu'un véhicule puisse transporter plus d'une seule demande à la fois (less than truckload). En effet, l'hypothèse d'aller directement de l'origine à la destination d'une demande sans collectes et/ou livraisons intermédiaires semble plutôt restrictive. Pour cela, la section 1.3 sera dédiée à la présentation des problèmes scientifiques associés, liés à l'élaboration des tournées.

### 1.2.2 Collaboration entre les transporteurs

La collaboration entre transporteurs est réalisée en échangeant des demandes de transport afin de minimiser les retours à vide et augmenter les taux de remplissage des véhicules.

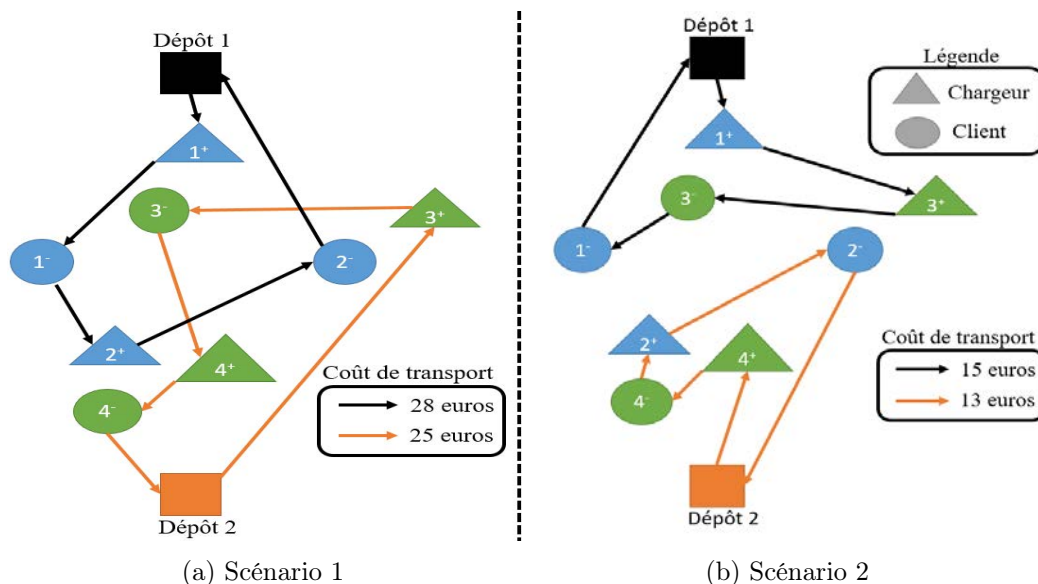


FIGURE 1.7 – Collaboration entre les transporteurs

Dans la figure 1.7, nous représentons le bénéfice potentiel de la collaboration entre les transporteurs, en supposant que nous avons 4 demandes de transport et 2 transporteurs. Un profit de 50 euros est associé à chaque demande. Dans le scénario 1, il n'y a aucune collaboration. Chaque transporteur s'occupe de 2 demandes et gagne 100 euros, mais le coût du transport est élevé à cause des longues distances parcourues. Dans le cas avec collaboration représenté par le scénario 2, l'échange des demandes de transport 2 et 3 a permis à chaque transporteur de minimiser le coût du transport en parcourant des distances beaucoup plus courtes et toujours en gagnant le même profit.

Plusieurs problèmes peuvent limiter la mise en place d'une telle collaboration. La difficulté principale est de trouver la façon optimale d'échanger les demandes de transport entre les transporteurs, de sorte que leurs coûts de transport soient minimisés ou que leurs profits soit maximisés. Il faudra également trouver une répartition équitable des économies ou du gain entre les transporteurs qui collaborent ensemble. Le partage d'informations et la coordination entre les transporteurs sont également indispensables pour assurer la collaboration.

Divers travaux ont déjà traité ce problème. Dans [Dai & Chen 2012], une méthode basée sur la relaxation lagrangienne est développée pour résoudre le problème de planification identifié comme étant une variante du PDP (Pickup and Delivery Problem) dans laquelle il n'existe pas de dépôt et les demandes peuvent être éclatées et servies par plusieurs véhicules. Dans un PDP, une flotte de véhicules, basée dans un dépôt, est utilisée pour satisfaire un ensemble de demandes de transport (collectes et/ou livraisons de marchandises). Deux ans plus tard, un autre modèle a été proposé dans lequel les transporteurs peuvent s'échanger des demandes afin de minimiser les coûts du transport [Wang & Kopfer 2014]. Récemment, [Li *et al.* 2016] ont abordé ce type de collaboration entre transporteurs en introduisant une nouvelle variante du problème dans laquelle chaque transporteur possède deux types de requêtes : celles réservées (transportées obligatoirement par lui-même) et celles non-réservées (qui peuvent être échangées avec les autres transporteurs).

### 1.3 Problèmes scientifiques liés à la planification des tournées

Dans le cadre du projet ANR TCDU, l'aspect collaboratif entre chargeurs doit se traduire par un ensemble de regroupements (lots) des demandes à satisfaire de sorte d'optimiser les coûts de transport. Il s'agit donc de planifier les tournées de véhicules que devraient effectuer les transporteurs. Mais la planification de tournées est un des problèmes d'optimisation combinatoire les plus abordés dans la littérature, pour lequel il existe de nombreuses variantes.

### 1.3.1 Problème du voyageur de commerce

Le problème du voyageur de commerce (ou Traveling Salesman Problem) est le premier problème de tournées étudié dans la littérature. Il consiste en la recherche d'un trajet optimal appelé tournée et permettant à un représentant de commerce de visiter ses clients dans un certain nombre de villes en passant une et une seule fois par toutes les villes. Généralement, les scientifiques cherchent à minimiser le coût de transport qui s'exprime par la distance totale parcourue ou le temps de parcours total.

Au 19<sup>ème</sup> siècle, ce problème a été introduit pour la première fois par un mathématicien d'Irlande nommé Sir William Rowan Hamilton et par le mathématicien britannique nommé Thomas Pennington Kirkman. Une discussion détaillée sur cette première version du problème peut être vue dans un livre intitulé Graph Theory [Biggs *et al.* 1976].

Il s'agit d'un problème d'optimisation NP-difficile. Cela signifie que le temps nécessaire pour trouver une solution optimale augmente de façon exponentielle en fonction de la taille du problème. Celle-ci dépend du nombre de clients à visiter. La figure 1.8 représente un exemple de TSP dans lequel on a une liste de 4 villes (A, B, C et D). Le voyageur doit partir de la ville A et revenir en A après avoir effectué la visite de toutes les autres villes. Les distances entre les villes sont données par le premier graphe (a). Un chemin possible est ABDCA de longueur 14 km (graphe (b)). Ce dernier ne représente pas la solution optimale qui est ACBDA de longueur 7 km donnée par le graphe (c).

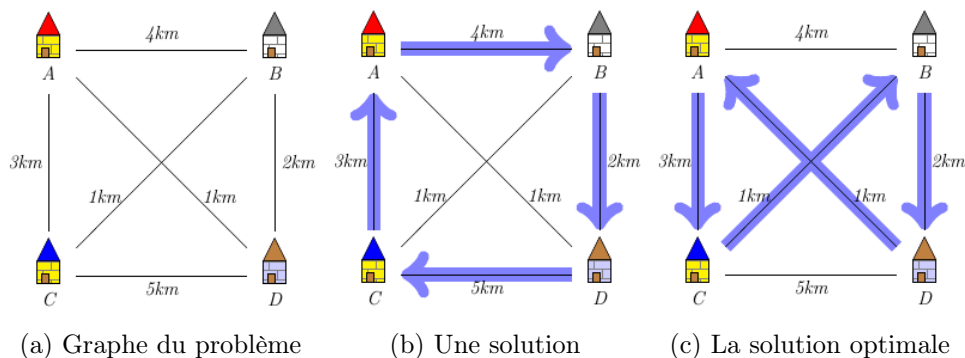


FIGURE 1.8 – Instance du TSP

Ce problème présente de nombreuses applications : d'une part, certains problèmes de séquençement (par exemple : atterrissage d'avions, passage de trains sur une voie...) qui peuvent être exprimés directement sous forme de TSP ; d'autre part, des problèmes de transport, plus complexes mais qui comportent des sous-problèmes de type TSP [Matai *et al.* 2010].

La modélisation mathématique du TSP de base peut s'exprimer comme suit, sous forme d'un programme linéaire à variable booléennes :



**Données :**

- $n$  = nombre total de villes ;
- $N=1,\dots,n$  : ensemble de sommets ;
- $d_{ij}$  = distance entre la ville  $i$  et la ville  $j$ .

**Variables :**

- $x_{ij} = \begin{cases} 1 & \text{Si le voyageur visite la ville } j \text{ directement après la ville } i. \\ 0 & \text{Sinon.} \end{cases}$

**Formulation mathématique :**

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n x_{ij} d_{ij} \quad (1.1)$$

sous contraintes :

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (1.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (1.3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad \text{avec } 2 \leq |S| \leq n - 2 \quad (1.4)$$

La fonction objectif dans ce modèle de base est de minimiser la distance totale parcourue par le voyageur. Les deux premières contraintes sont utilisées pour assurer que chaque ville est visitée exactement une seule fois. La troisième contrainte élimine les sous-tours.  $S$  représente un sous-ensemble de sites possibles. En effet, la tournée devra obligatoirement débuter et se terminer au point de départ. Dans la contrainte (1.4),  $|S|$  est le nombre de sommets inclus dans un sous-tour potentiel composé des clients de l'ensemble  $S$ .

Au cours des années, le modèle traditionnel du TSP a évolué plusieurs fois. Nous pouvons citer quelques variantes de ce problème et quelques études associées :

- (a) sTSP (symmetric TSP) : les distances  $d_{ij}$  et  $d_{ji}$  sont égales [Deng *et al.* 2015] et [Errico *et al.* 2016].
- (b) aTSP (asymmetric TSP) : si les distances  $d_{ij}$  et  $d_{ji}$  sont différentes pour au moins un couple de villes  $i$  et  $j$  [Abdoun *et al.* 2016], [Melkonian 2007].
- (c) STSP (Selective TSP) : l'hypothèse du modèle de base qui oblige le voyageur à visiter toutes les villes est omise. Dans la littérature, nous pouvons rencontrer la même variante sous différentes nomenclatures : OP (Orienteering Problem), PTSP (Prize collecting TSP) et TSPP (TSP with Profits) [Qin *et al.* 2009], [Pedro *et al.* 2013] et [Filippi & Stevanato 2013].
- (d) TSPTW (TSP with Time Windows) : cette variante exige de trouver la tournée optimale visitant un ensemble de villes donné exactement une fois, où chaque

ville doit être visitée dans une fenêtre de temps donnée, par exemple entre 9h et 12h [Papalitsas *et al.* 2015], [Karabulut & Tasgetiren 2014], [Kara *et al.* 2013].

- (e) TSP with multiple time windows : chaque ville possède plusieurs fenêtres temporelles. Et elle doit être visitée dans une des ces fenêtres [Mesquita *et al.* 2013] et [Paulsen *et al.* 2015].
- (f) PTSP (Periodic TSP) : chaque ville doit être visitée plusieurs fois sur un horizon de temps donné [Paletta & Triki 2004].
- (g) Multi-period TSP : dans un horizon de temps divisé en plusieurs périodes, chaque ville doit être visitée une seule fois. Cette visite peut être faite dans n'importe quelle période dans laquelle une fenêtre temporelle existe [Andreatta & Lulli 2008].

La figure 1.9 résume les variantes du TSP que nous avons abordées dans cette section.

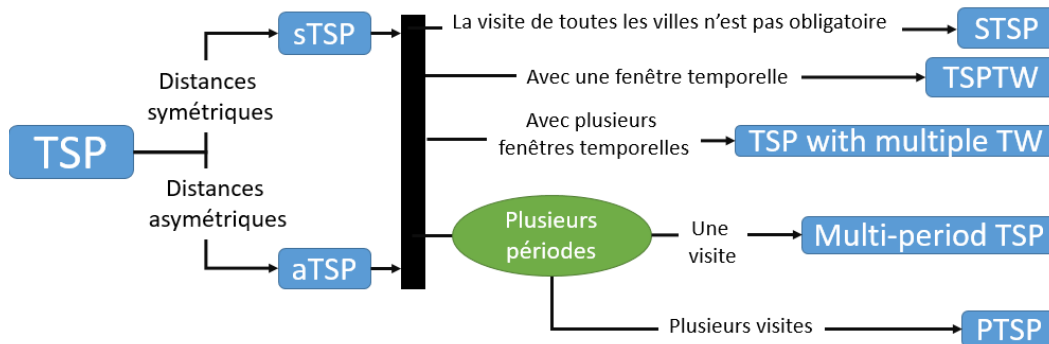


FIGURE 1.9 – Variantes du TSP

### 1.3.2 Vehicle Routing Problem (VRP)

Le VRP (ou problème de tournées de véhicules) est une extension du TSP et fait donc partie de la classe des problèmes NP-difficile. En 1959, ce problème a été introduit pour la première fois sous le nom de "Truck Dispatching Problem" [Dantzig & Ramser 1959]. Actuellement, le VRP représente une classe de problèmes d'optimisation combinatoire et de recherche opérationnelle maintes fois étudiée [Ismail *et al.* 2011]. Le VRP de base est distingué du TSP uniquement par le fait que plusieurs véhicules sont disponibles [Parragh *et al.* 2008a] [Sörensen 2003]. Un VRP classique consiste donc à déterminer les itinéraires optimaux pour un ensemble de véhicules afin de satisfaire les demandes d'un certain nombre de clients à partir d'un dépôt. De multiples applications peuvent être représentées sous forme d'un VRP : tournées de livraisons de marchandises, visites médicales ou commerciales, tournées d'interventions (réparation, maintenance, contrôles), etc...

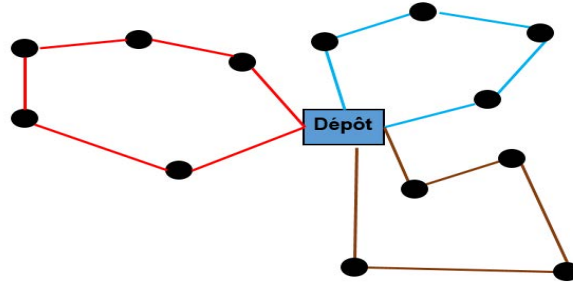


FIGURE 1.10 – Exemple de solution du VRP

La figure 1.10 montre un exemple de solution pour une instance du VRP composé d'un dépôt, treize clients et une flotte composée de trois véhicules.

Le modèle traditionnel du VRP s'exprime par le programme linéaire suivant [Ade-  
wumi & Adeleke 2016] :

**Données :**

- $n$  : nombre de clients
- $D=0$  : dépôt ;
- $N_{\text{œuds}}=1,\dots,n$  : ensemble de clients ;
- $N=D \cup N_{\text{œuds}}$  : ensemble de sommets ;
- $V$  : ensemble de véhicules ;
- $d_{ij}$  = distance entre le client  $i$  et le client  $j$ .

**Variables :**

- $x_{ij}^k = \begin{cases} 1 & \text{Si le site } j \text{ est visité directement après le site } i \text{ par le véhicule } k. \\ 0 & \text{Sinon.} \end{cases}$

**Formulation mathématique :**

$$\text{Minimiser } \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} x_{ij}^k d_{ij} \quad (1.5)$$

sous contraintes :

$$\sum_{j \in N} \sum_{k \in V} x_{ij}^k = 1 \quad \forall i \in \text{Noeuds} \quad (1.6)$$

$$\sum_{i \in N} \sum_{k \in V} x_{ij}^k = 1 \quad \forall j \in \text{Noeuds} \quad (1.7)$$

$$\sum_{i \in N} x_{ij}^k - \sum_{i \in N} x_{ji}^k = 0 \quad \forall j \in \text{Noeuds}, \forall k \in V \quad (1.8)$$

$$\sum_{j \in \text{Noeuds}} x_{0j}^k \leq 1 \quad \forall k \in V \quad (1.9)$$

$$\sum_{i \in \text{Noeuds}} x_{i0}^k \leq 1 \quad \forall k \in V \quad (1.10)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1 \quad \forall S \subseteq \text{Noeuds} \quad \text{avec} \quad |S| \geq 2, \forall k \in V \quad (1.11)$$

La fonction objectif consiste à minimiser la distance. Les contraintes (1.6) et (1.7) assurent que chaque client est visité une seule fois. La conservation du flux est garantie par la contrainte (1.8). Les contraintes (1.9) et (1.10) sont utilisées pour assurer que chaque tournée commence et termine au dépôt. Ainsi, tous les véhicules ne sont pas forcément utilisés. La contrainte (1.11) élimine les sous-tours.

De nombreuses variantes du VRP existent dans la littérature parmi lesquelles : sVRP (symmetric VRP), aVRP (asymmetric VRP), VRPTW (VRP with Time Windows) et SVRP (Selective VRP), qui ont déjà été définies pour le TSP (voir la sous-section 1.3.1). De plus, nous identifions les cas suivants :

- (a) CVRP (Capacitated VRP) : la flotte est composée de véhicules avec une capacité limitée.
- (b) Mono-depot VRP : il existe un seul dépôt.
- (c) Multi-depot VRP : il existe au moins deux dépôts.
- (d) Periodic VRP : [Gaudioso & Paletta 1992], [Yao *et al.* 2013], [Liao *et al.* 2017].
- (e) Multi-period VRP : [Dayarian *et al.* 2015], [Mancini 2016], [Dayarian *et al.* 2016].
- (f) VRP with multiple time windows : [Favaretto *et al.* 2007], [Qiu-ai & Zhen-ping 2012], [Belhaiza *et al.* 2014].

Le tableau 1.2 donne une classification d'un ensemble d'études récentes. Le choix de ces études a été fait de façon à englober les principales variantes du VRP. Pour chacune d'elles, nous indiquons s'il s'agit d'un problème avec un ou plusieurs objectifs. Cette liste n'est pas exhaustive et ne regroupe pas toutes les variantes du VRP parce que le nombre des recherches sur le sujet est très important.

Référence	Symétrique	Asymétrique	Nb de dépôts		Capacité limitée	TW	Sélective	Nb d'objectifs	
			1	>1				1	>1
[Lysgaard 2010], [Xiao <i>et al.</i> 2012], [Ribeiro & Laporte 2012], [Ai & Kachitvichyanukul 2009], [Wang & Lu 2009], [Kuo & Zulvia 2017], [Wei <i>et al.</i> 2017]	X		X		X			X	
[Qureshi <i>et al.</i> 2009]		X	X		X	X		X	
[Ghoseiri & Ghannadpour 2010], [Jiang <i>et al.</i> 2014]	X		X		X	X			X
[Valle <i>et al.</i> 2009]	X		X				X	X	
[Pisinger & Ropke 2007]		X		X	X	X			X
[Liu <i>et al.</i> 2014a]		X		X	X			X	
[Escobar <i>et al.</i> 2014]	X			X	X			X	

TABLE 1.2 – Quelques variantes du VRP

La deux sous-sections suivantes sont dédiées à des variantes plus particulièrement étudiées dans cette thèse, à savoir les tournées sélectives, et les tournées de collecte et livraison.

### 1.3.3 Team Orienteering Problem (TOP)

Le Team Orienteering Problem (TOP) provient d'un jeu de plein air joué dans les zones montagneuses ou boisées. Dans ce jeu, une équipe de plusieurs joueurs tente de collecter autant de points de récompense que possible dans un délai donné. De même, le TOP est le problème où un nombre limité de véhicules sont disponibles pour visiter un ensemble de clients. Le temps de déplacement de chaque véhicule est limité par un quota temporel, un profit est associé à chaque client et ce dernier peut être visité au plus une fois. Le TOP est donc une variante de VRP dans laquelle il est parfois impossible de servir tous les clients, on le trouve aussi dans la littérature sous l'appellation problèmes de tournées sélectives. Le but de ce problème est d'organiser un itinéraire de visites afin de maximiser le profit total collecté.

Le TOP a été mentionné pour la première fois par Butt et Cavalier sous le nom de Multiple Tour Maximum Collection Problem (MTMCP) [Butt & Cavalier 1994].

La figure 1.11 représente un exemple de solution d'une instance du TOP où nous avons un dépôt, treize clients et une flotte de trois véhicules. La solution ne visite que neuf clients.

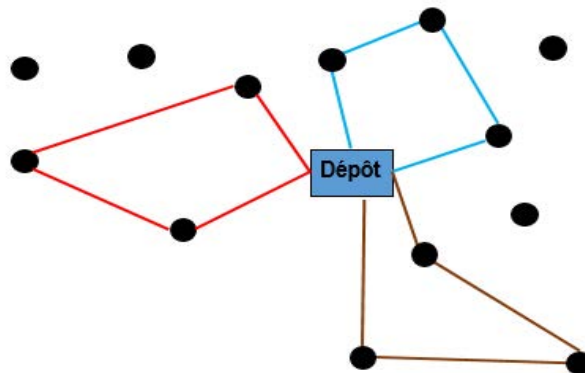


FIGURE 1.11 – Exemple de solution du TOP

Au cours de la dernière décennie, diverses applications ont été modélisées sous la forme d'un TOP dans la logistique, le tourisme et d'autres domaines [Vansteenwegen *et al.* 2011].

Le modèle mathématique du TOP peut être formulé de la façon suivante [El-Hajj 2015] :

**Données :**

- Nœuds : Ensemble de nœuds ;
- $ND = Nœuds \cup d$  : Ensemble de nœuds avec le dépôt de départ  $d$  ;

- $NA = \text{Nœuds} \cup a$  : Ensemble de nœuds avec le dépôt d'arrivée  $a$  ;  
Notons que cette formulation a comme but de modéliser le TOP de base même si le dépôt de départ peut être différent de celui d'arrivée. En effet, cet aspect représente une particularité de ce modèle. Mais si nous considérons que les deux dépôts partagent la même localisation géographique, nous revenons sur le problème de base avec un seul dépôt.
- $V$  : Ensemble de véhicules ;
- $L$  : Limite de coût de voyage
- $P_i$  : Profit associé au nœud  $i$  ;
- $c_{ij}$  : Coût du voyage entre le client  $i$  et le client  $j$ .

**Variables :**

- $x_{ij}^k = \begin{cases} 1 & \text{Si le site } j \text{ est visité directement après le site } i \text{ par le véhicule } k. \\ 0 & \text{Sinon.} \end{cases}$
- $y_i^k = \begin{cases} 1 & \text{Si le site } i \text{ est visité par le véhicule } k. \\ 0 & \text{Sinon.} \end{cases}$

**Formulation mathématique :**

$$\text{Maximiser } \sum_{i \in \text{Noeuds}} \sum_{k \in V} y_i^k P_i \quad (1.12)$$

sous contraintes :

$$\sum_{k \in V} y_i^k \leq 1 \quad \forall i \in \text{Noeuds} \quad (1.13)$$

$$\sum_{j \in NA} x_{dj}^k = \sum_{j \in ND} x_{ja}^k = 1 \quad \forall k \in V \quad (1.14)$$

$$\sum_{i \in NA \setminus \{r\}} x_{ri}^k = \sum_{j \in ND \setminus \{r\}} x_{jr}^k = y_r^k \quad \forall r \in \text{Noeuds}, \forall k \in V \quad (1.15)$$

$$\sum_{i \in ND} \sum_{j \in NA \setminus \{i\}} c_{ij} x_{ij}^k \leq L \quad \forall k \in V \quad (1.16)$$

$$\sum_{(i,j) \in S * S} x_{ij}^k \leq |S| - 1 \quad \forall S \subseteq \text{Noeuds}, |S| \geq 2, \forall k \in V \quad (1.17)$$

La fonction objectif (1.12) vise à maximiser le profit total collecté. Les contraintes (1.13) imposent que chaque client doit être visité au plus une fois par un seul véhicule. Les contraintes (1.14) garantissent que chaque véhicule commence son chemin du dépôt de départ  $d$  et le termine au dépôt d'arrivée  $a$ . De plus, elle supposent que tous les véhicules vont être utilisés. Les contraintes (1.15) assurent la conservation du flux. Les contraintes (1.16) sont utilisées pour assurer que le coût total du voyage ne dépasse pas  $L$ , alors que les contraintes (1.17) éliminent tous les sous-tours.

Depuis son apparition, ce problème ne cesse d'attirer l'attention des chercheurs qui ont modifié le modèle de base en ajoutant plusieurs contraintes pour modéliser des applications réelles. Ce qui a donné lieu à des nombreuses variantes du TOP [Gunawan *et al.* 2016] :

- (a) OP (Orienteering Problem) : la flotte est composée d'un seul véhicule [Schildt *et al.* 2009], [Rezki & Aghezzaf 2017], [Kobeaga *et al.* 2018].
- (b) OPTW (OP with Time Windows) : [Karbowska-Chilinska & Zabielski 2014], [Gunawan *et al.* 2015], [Verbeeck *et al.* 2017].
- (c) TOPTW (TOP with Time Windows) : [Lin & Vincent 2015], [Lin & Vincent 2017], [Vincent *et al.* 2017].
- (d) Capacitated TOP (CTOP) : [Aghezzaf & El Fahim 2015], [Ben-Said *et al.* 2016], [Park *et al.* 2017].
- (e) Multi-period OP : [Tricoire *et al.* 2010].
- (f) TOP with multiple time windows : [Souffriau *et al.* 2013], [Lin & Vincent 2015].

Tous les problèmes scientifiques abordés jusque là ne prennent pas en considération la possibilité de collecter et de livrer des marchandises dans une même tournée. Ainsi, le but de la sous-section suivante est d'aborder une autre catégorie de problèmes traités dans la littérature qui autorise un tel couplage.

#### 1.3.4 Pickup and Delivery Problem (PDP)

Le problème de collecte et livraison ou Pickup and Delivery Problem (PDP) représente une généralisation du VRP auquel sont additionnées de nouvelles contraintes. Le PDP est un problème d'optimisation classique en recherche opérationnelle. Comme le VRP appartient à la classe des problèmes NP-difficile, le PDP est aussi un problème NP-Difficile.

Dans un PDP, un véhicule peut effectuer les deux types d'opérations (livraison de clients et collecte chez des fournisseurs) dans une même tournée, ce qui n'est pas le cas pour tous les autres problèmes mentionnés précédemment.

Le PDP a été introduit pour la première fois sous le nom du GPDP (General PDP) [Savelsbergh & Sol 1995]. Au cours des dernières décennies, il a été étudié de manière intensive. Les chercheurs ont considéré de nombreuses variantes du PDP qui tiennent compte de différents types de contraintes. Pour plus de détails, voir le livre [Toth & Vigo 2002] et un certain nombre de revues de la littérature [Savelsbergh & Sol 1995], [Parragh *et al.* 2008b], [Parragh *et al.* 2008c].

Dans l'exemple de solution représenté par la figure 1.12, la flotte est composée de 3 véhicules. Il existe 4 fournisseurs représentés par des triangles et 9 clients représentés par des cercles. Nous pouvons remarquer que dans une même tournée, le véhicule peut visiter les fournisseurs et les clients.

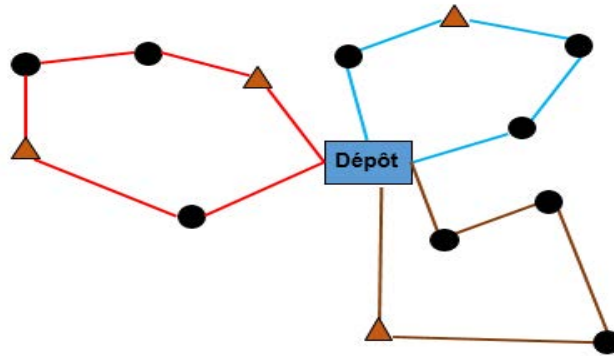


FIGURE 1.12 – Exemple de solution du PDP

La modélisation mathématique du PDP dans sa forme basique est identique à celle du VRP (voir la sous-section 1.3.2). Nos travaux de thèse s'articulent autour de deux variantes spécifiques du PDP. Pour cela, nous allons détailler dans la section suivante, les principales variantes de ce problème.

## 1.4 Variantes du PDP

Dans cette section, nous classifions les nombreuses variantes du problème de collecte et de livraison qui existent dans la littérature. Cette classification repose sur le fait qu'il existe trois types de flux : le premier relie le dépôt avec un ou plusieurs clients, le deuxième est le flux entre un ou plusieurs fournisseurs et le dépôt. Le troisième type connecte les fournisseurs et leurs clients. Ces variantes peuvent donc être classées en deux catégories, elles-mêmes décomposées en plusieurs sous-catégories (voir figure 1.13) :

- Transport entre les sites et le dépôt : Cette première catégorie porte sur le transport des marchandises du dépôt vers les clients et des fournisseurs vers le dépôt.
- Transport entre les fournisseurs et les clients : La deuxième catégorie fait référence à toutes les variantes où les marchandises sont transportées entre les fournisseurs et les clients.

La première catégorie est divisée en 4 sous-catégories, illustrées par la figure 1.14 :

- VRP with Clustered Backhauls (VRPCB) : tous les clients du dépôt (linehaul customers) doivent être visités dans un premier temps puis tous les fournisseurs du dépôt (backhaul customers) [Belloso *et al.* 2015], [Bortfeldt *et al.* 2015], [Belloso *et al.* 2017].
- VRP with Mixed linehauls and Backhauls (VRPMB) : Les séquences de visites mixtes sont autorisées [Ropke & Pisinger 2006b], [Pinto *et al.* 2015], [Belloso *et al.* 2017].
- VRP with Divisible Delivery and Pickup (VRPDDP) : chaque site peut jouer à la fois le rôle d'un fournisseur et d'un client. Ainsi, il n'est pas obligatoire que



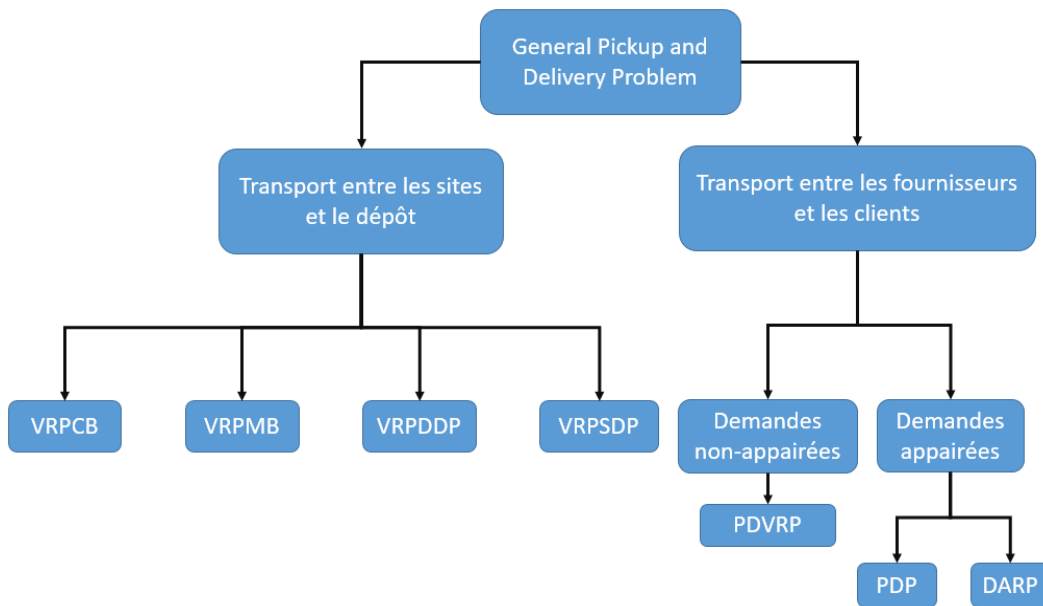


FIGURE 1.13 – Classification des variantes du General PDP

chaque client soit seulement visité une seule fois. Au contraire, deux visites, une pour la livraison et une pour la collecte sont possibles [Mitra 2005], [Nagy *et al.* 2013], [Polat 2017].

- VRP with Simultaneous Delivery and Pickup (VRPSDP) : Chaque site ne peut être visité qu'une seule fois. Si un site est à la fois fournisseur et client, la livraison et la collecte auront lieu en même temps [Goksal *et al.* 2013], [Wang *et al.* 2015], [Avci & Topaloglu 2016].

Dans le cas du transport entre les fournisseurs et les clients, on distingue deux sous-catégories :

- Demands non-appairées : Ce cas se réfère à des situations où les points de ramassage et de livraison sont non appairés. Autrement dit un type de produit identique, tel que l'argent, est considéré. Ainsi, chaque unité ramassée peut être utilisée pour répondre à la demande de n'importe quel client. Dans cette sous-catégorie, la seule variante de la littérature est le "Pickup and Delivery VRP (PDVRP)" [Shi *et al.* 2009], [Chen *et al.* 2014], [Xu *et al.* 2017].
- Demands appairées (pour Paired Demands ou PD) : Cette deuxième sous-catégorie tient compte des demandes de transport appairées. Une demande appairée relie un fournisseur à son client. Elle est donc associée à un point d'origine et un point de destination. Nous pouvons retrouver dans la littérature : le PDP classique qui prend en considération le transport de marchandises (voir les sous-sections suivantes) ainsi que le Dial-A-Ride problem (DARP) qui traite du transport de passagers [Rahmani *et al.* 2016], [Masmoudi *et al.* 2017], [Molenbruch *et al.* 2017].

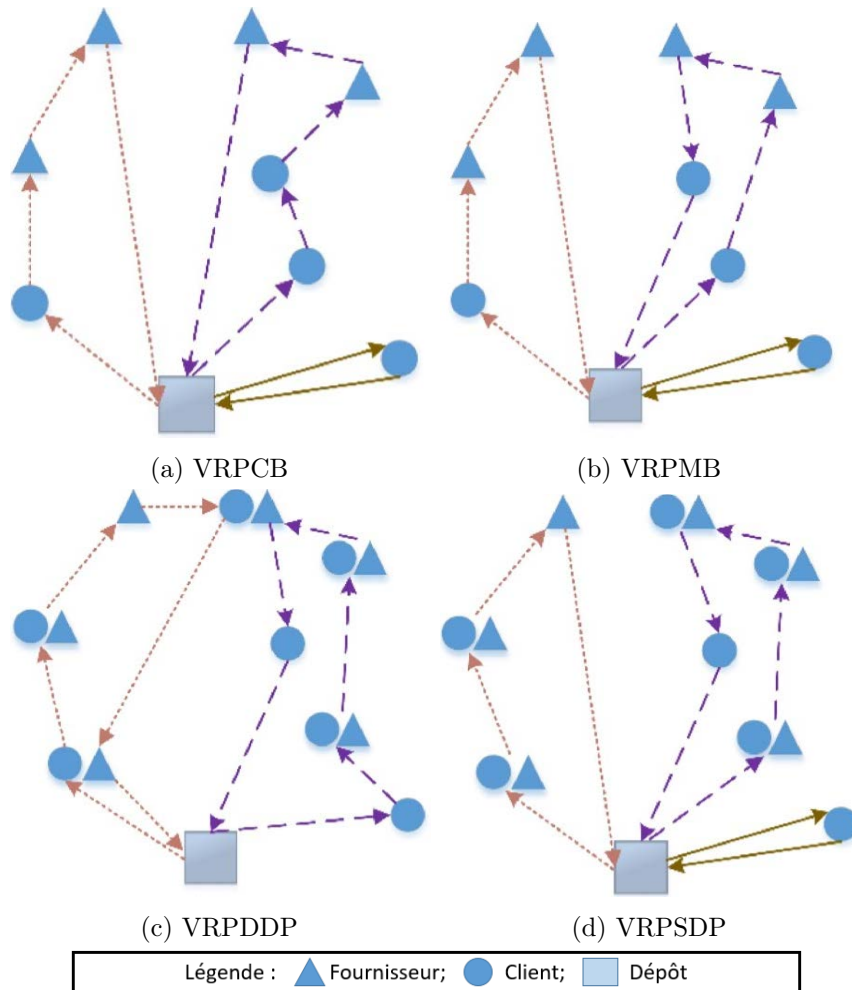


FIGURE 1.14 – Transport entre les sites et le dépôt

Dans la suite, nous donnons un aperçu des méthodes classiquement utilisées pour résoudre des problèmes de planification des tournées.

## 1.5 Méthodes de résolution

Au fil des années, les chercheurs ont proposé des méthodes de résolution pour toutes sortes de problèmes rencontrés dans notre vie quotidienne notamment les problèmes de planification des tournées. Compte tenu de la complexité de ces derniers, de grands efforts ont été réalisés pour améliorer les performances des méthodes de résolution en termes de qualité de la solution proposée et/ou du temps de calcul nécessaire.

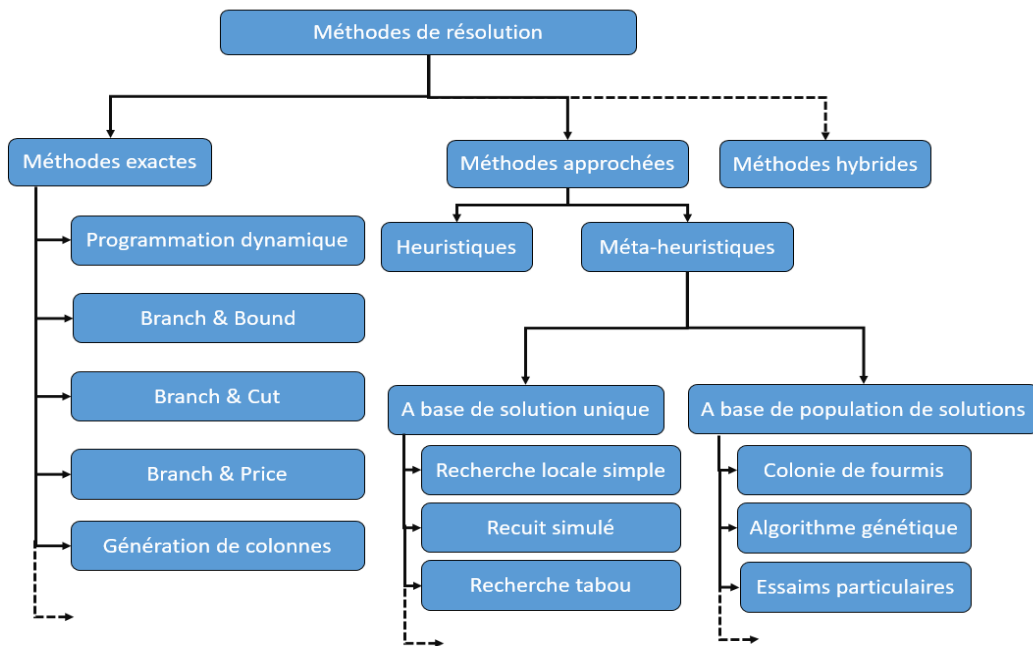


FIGURE 1.15 – Méthodes de résolution

Ainsi, on observe une grande variété et des différences remarquables au niveau du principe et des performances. Cette variété et ces différences ont permis de regrouper les méthodes de résolution en deux classes principales :

- Les méthodes exactes : elles permettent d'obtenir une solution dont l'optimalité est garantie, mais elles sont très gourmandes en termes de temps de résolution surtout si le problème traité est complexe et pour des instances de grande taille.
- Les méthodes approchées : elles permettent d'obtenir rapidement (en un temps de calcul raisonnable) des solutions de bonne qualité mais sans garantie d'optimalité.

La combinaison des méthodes qui appartiennent à ces deux classes a donné naissance à une troisième classe : les méthodes hybrides (figure 1.15).

### 1.5.1 Méthodes exactes

Plusieurs méthodes exactes existent dans la littérature pour résoudre les problèmes d'optimisation. Nous citerons dans cette sous-section les plus utilisées pour résoudre les variantes du problème de collecte et de livraison.

- **Branch & Bound** : La méthode de séparation et évaluation a été introduite pour la première fois par Land et Doig [Land & Doig 1960]. L'idée générale consiste à représenter le problème sous forme d'un arbre, où chaque nœud correspond à une solution partielle. Dans cette méthode, le principe est d'éliminer précocement des sous-arbres partant d'un nœud dans lesquels l'absence de la solution optimale recherchée est certaine.

Par exemple, dans un problème de minimisation, nous pouvons accélérer le

processus de recherche en éliminant un ensemble des sous-arbres partant d'un nœud si pour l'objectif considéré, la borne inférieure pour cet ensemble est plus grande que la valeur de la meilleure solution trouvée jusqu'à présent. Cela implique que les solutions partielles de cet ensemble ne seront pas générées et la méthode devient généralement plus rapide que l'énumération complète (voir [Carrabs *et al.* 2007]).

- Cutting Plane : La méthode de coupes planes est mentionnée pour la première fois par [Schrijver 1998]. Elle est utilisée pour résoudre des problèmes d'optimisation qui peuvent être représentés sous la forme d'un programme linéaire en nombres entiers mixtes (MILP pour Mixed Integer Linear Programming). Par définition, le MILP est un modèle d'optimisation où la fonction objectif, les contraintes de base sont toutes représentées par des expressions linéaires et des variables mixtes (entières et réelles). L'idée principale de cette méthode est d'ajouter des contraintes supplémentaires afin de minimiser l'espace de recherche original sans exclure la solution entière optimale.
- Branch & Cut : Cette méthode introduite par [Padberg & Rinaldi 1991] est une combinaison entre la méthode "Branch & Bound" et celle des coupes planes. Elle consiste à évaluer les nœuds de l'arbre à l'aide de la méthode des coupes planes. Cette méthode a été utilisée pour résoudre des variantes du PDP, par exemple dans [Ropke *et al.* 2007].
- Génération de colonnes : Cette méthode repose sur la décomposition de Dantzig-Wolfe [Dantzig & Wolfe 1960]. Elle est utilisée pour résoudre des programmes linéaires de grande taille. Comme la plupart des variables sont nulles dans la solution optimale, seul un sous-ensemble de variables doit être considéré en théorie lors de la résolution du problème. L'idée centrale de cette méthode est d'éviter de représenter explicitement toutes les variables (colonnes) du programme linéaire (appelé problème maître). Elle consiste donc à résoudre un problème initialisé avec un petit sous-ensemble de colonnes (appelé problème réduit). Ce dernier est plus simple à résoudre. Mais si l'ensemble de ses variables ne contient pas celles qui donne la solution optimale du problème maître, un mécanisme sera utilisé pour générer de nouvelles variables susceptibles d'améliorer la solution actuelle. L'efficacité de cette méthode dépend principalement des techniques utilisées pour générer des colonnes à chaque itération (exemple d'utilisation pour les variantes du PDP : [Venkateshan & Mathur 2011]).
- Branch & Price : Cette méthode combine la technique classique Branch & Bound avec celle de la génération de colonnes appliquée sur chaque nœud de l'arbre de recherche. Elle a été initialement proposée par [Johnson 1989]. Plusieurs exemples d'utilisation de cet algorithme existent dans la littérature (voir [Bettinelli *et al.* 2014]).
- Programmation dynamique : Cette technique est utilisée pour résoudre un problème complexe en le décomposant en une collection de sous-problèmes plus simples. Ce concept a été introduit par [Bellman 1952]. Cette méthode

consiste à résoudre une seule fois chacun de ces sous-problèmes et à stocker leurs solutions. La prochaine fois que le même sous-problème survient, au lieu de recalculer sa solution, il suffit de rechercher la solution précédemment calculée, ce qui économise du temps de calcul. La technique de stockage des solutions des sous-problèmes est appelée mémorisation. Les algorithmes de programmation dynamique sont souvent utilisés pour résoudre des problèmes de tournées (voir [Righini & Salani 2009]).

### 1.5.2 Méthodes approchées

Bien que les méthodes exactes puissent produire des solutions optimales, le temps de résolution pour obtenir ces solutions est en général très important. Ainsi, nous avons souvent recours à des méthodes approchées qui visent à trouver une solution de bonne qualité avec un temps de calcul relativement raisonnable. Dans cette sous-section, nous présentons les principales méthodes approchées utilisées pour la résolution des problèmes de tournées de véhicules dans la littérature.

- Les heuristiques : Ce sont des stratégies qui permettent de se déplacer intelligemment dans l'espace de recherche afin d'obtenir rapidement une solution réalisable à un problème donné. Généralement, elles sont dépendantes du problème à résoudre et ne peuvent donc être généralisées. Elles sont réparties principalement en deux types : les heuristiques de construction qui génèrent une solution itérativement (par exemple les méthodes gloutonnes) et les heuristiques de descente qui cherchent un optimum local à partir d'une solution donnée. Ces méthodes ont été utilisées pour traiter des variantes spécifiques du problème de collecte et de livraison (voir [Ropke & Pisinger 2006a] [Nowak *et al.* 2008], [Benavent *et al.* 2015]).
- Les méta-heuristiques : Ce sont des méthodes génériques qui permettent de résoudre une large gamme de problèmes d'optimisation. La majorité de ces méthodes est inspirée des systèmes naturels. Il existe un grand nombre de méta-heuristiques dans la littérature mais elles sont généralement classées en fonction du nombre de solutions qu'elles manipulent simultanément.

#### 1.5.2.1 Les méta-heuristiques à base de solution unique

Les méthodes appartenant à cette catégorie ne manipulent qu'une seule solution à la fois. En fait, elles essaient itérativement d'améliorer la qualité de cette solution en se basant sur la notion de voisinage. Autrement dit, elles construisent un chemin dans l'espace de recherche en tentant de se diriger vers un optimum global. De plus, la qualité de la solution finale, fournie par ces méthodes, dépend fortement de celle de la solution initiale. Parmi les nombreuses méthodes de cette catégorie, nous décrivons ci-après les méthodes de recuit simulé et recherche tabou à titre d'exemple.

- Recuit simulé (simulated annealing) : Cette méthode a été introduite par [Kirkpatrick *et al.* 1983]. Elle utilise une stratégie spéciale qui consiste à accepter de dégrader la solution afin d'éviter les optima locaux. Le concept est

basé sur un ancien processus utilisé par les métallurgistes. L'idée principale de cette méthode est de débiter la résolution avec une solution initiale générée aléatoirement caractérisée par une énergie et une température initiales. A chaque itération, un changement élémentaire est effectué sur la solution. Ce dernier fait varier l'énergie du système. Si cette variation est négative, elle est acceptée car cela signifie que la nouvelle solution améliore la fonction objectif, et permet de diminuer l'énergie du système. Si la solution trouvée est moins bonne que la précédente alors elle sera acceptée avec une probabilité donnée. La probabilité d'accepter des solutions de mauvaise qualité est directement proportionnelle à la température de cette dernière. Plusieurs approches sont utilisées pour ajuster la température tout au long de la recherche (pour plus d'informations voir [Van Laarhoven & Aarts 1987]). Cette méthode est largement utilisée pour résoudre des problèmes d'optimisation (exemple : [Li & Lim 2003]).

- Recherche Tabou (Tabu Search) : Cette méthode, créée par [Glover 1986], représente une combinaison entre la recherche locale classique avec un ensemble de techniques qui guident la recherche de manière intelligente. Dans cet algorithme, une liste tabou contenant les dernières solutions visitées est mise en œuvre. Elle ne donne pas la possibilité à une solution déjà trouvée d'être acceptée car elle est déjà stockée dans la liste ce qui permet d'éviter les cycles et d'explorer de nouvelles solutions voisines. La recherche tabou a été appliquée avec succès à la résolution de différents problèmes de transport [Nanry & Barnes 2000] [Li & Lim 2003].

### 1.5.2.2 Les méta-heuristiques à base de population de solutions

L'idée commune de toutes les méthodes appartenant à cette catégorie consiste à manipuler un ensemble de solutions en même temps au lieu d'une solution unique afin de renforcer la diversité de la recherche. Dans la littérature, une grande variété de ces méthodes a été proposée mais nous allons à nouveau décrire deux d'entre elles dans cette sous-section : les algorithmes génétiques et les essais particuliers.

- Algorithme génétique : Cette méthode s'inspire des mécanismes de la sélection naturelle et de la génétique. Ces algorithmes ont été adaptés pour la première fois aux problèmes d'optimisation par [Holland 1973]. Le vocabulaire utilisé dans le cadre de cette méta-heuristique est le même que celui de la génétique et de la théorie de l'évolution. On utilise donc les termes suivants : individu (solution potentielle), population (ensemble de solutions), génotype (une représentation de la solution), gène (une partie du génotype), sélection, croisement, mutation, parent, enfant, etc.

L'idée principale est extrêmement simple : on commence la résolution avec une population initiale composée d'individus aléatoirement générés. Ensuite, on évalue la performance de chaque individu. Puis, on génère une nouvelle population en appliquant des opérateurs spécifiques aux solutions de la population courante : la sélection, le croisement et la mutation. Seuls les individus

les mieux adaptés à notre objectif sont supposés survivre pour la génération suivante. Finalement, on répète ce processus jusqu'à ce qu'on trouve une solution satisfaisante (la figure 1.16).

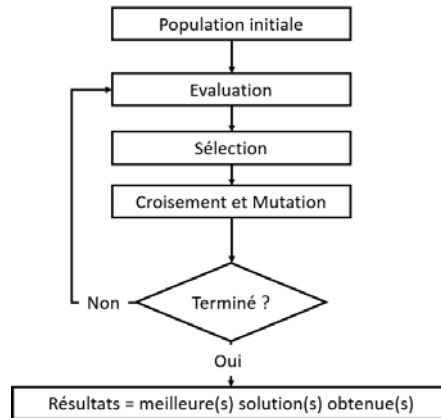


FIGURE 1.16 – Fonctionnement d'un algorithme génétique

De nombreux travaux ont déjà utilisé cet algorithme pour résoudre plusieurs variantes en relation avec le transport de marchandises [Pankratz 2005] [Liao & Ting 2010] [Liao & Ting 2012].

L'algorithme mémétique introduit par [Moscato *et al.* 1989] est considéré comme une hybridation entre l'algorithme génétique classique et les techniques de recherche locale. Ces deux dernières décennies, cet algorithme a connu une investigation remarquable (par exemple : [Nagata & Kobayashi 2010] [Ting & Liao 2013] [Liao & Ting 2013]).

- Optimisation par essais particuliers ou Particle Swarm Optimization (PSO) : Cette méthode a été proposée par [Kennedy & Eberhart 1995]. Elle est inspirée du comportement social des animaux évoluant en essaim. PSO partage de nombreuses similitudes avec les algorithmes génétiques (GA). La méthode est initialisée avec une population de solutions aléatoires. Cependant, contrairement à GA, PSO n'a pas d'opérateurs d'évolution tels que le croisement et la mutation. Dans PSO, les solutions potentielles, appelées particules, sont dotées d'une mémoire afin de sauvegarder le meilleur point par lequel elle est déjà passée en parcourant l'espace de recherche. De plus, chaque particule est informée du meilleur point connu par ses voisins. Cette information va aider les particules à modifier leurs trajectoires afin de converger vers la meilleure solution. Cette méthode permet la résolution d'un large panel de problèmes d'optimisation (exemple : [Zou *et al.* 2013]).

Après l'introduction globale sur toutes les variantes du General Pickup and Delivery Problem (GPDP), nous avons présenté dans cette section, quelques méthodes d'optimisation utilisées pour résoudre les problèmes de planification des tournées. Nous allons nous concentrer dans la suite sur un aperçu des études récentes qui traitent des variantes du PDP classique.

## 1.6 Résolution des variantes du PDP

Une des premières méthodes approchées utilisée pour résoudre le PDP avec demandes appairées et fenêtres temporelles (PDPTWPD) a été proposée par [Nanry & Barnes 2000]. Cette méthode est basée sur une recherche tabou dans laquelle trois opérateurs sont proposés pour explorer le voisinage d'une solution. Le premier opérateur déplace une paire d'une tournée vers une autre, le deuxième consiste à échanger deux paires de deux différentes tournées et le troisième change la position d'une paire dans une même tournée. [Li & Lim 2003] ont également développé une recherche tabou qui optimise plusieurs objectifs en reprenant les trois opérateurs proposés par [Nanry & Barnes 2000]. Par contre, ils ont décidé d'améliorer la solution à chaque itération en utilisant une heuristique de type recuit simulé. Dans la même étude, les auteurs ont généré 56 instances de 100 clients à partir des instances classiques du VRP. Une des premières méthodes exactes pour résoudre le PDPTWPD a été proposée par [Lu & Dessouky 2004]. Elle est basée sur un algorithme de type "Branch & cut" et elle traite le cas de plusieurs dépôts.

Un an plus tard, [Pankratz 2005] a élaboré un algorithme génétique. Le temps de calcul pour quelques instances a été amélioré en utilisant cet algorithme. En 2006, une heuristique constructive a été développée par [Lu & Dessouky 2006] pour résoudre le PDPTWPD dans un contexte multi-objectifs. La même année, [Ropke & Pisinger 2006a] ont proposé une heuristique de type ALNS (Adaptive Large Neighborhood Search) pour résoudre le PDPTWPD avec plusieurs dépôts. De plus, ils ont généré de nouvelles instances pour le PDPTWPD. Une méthode exacte de type "Branch & Bound" pour résoudre le PDPPD a été proposée par [Carrabs *et al.* 2007]. Cette méthode traite deux types de chargements : LIFO ou FIFO. Une méthode de type "Branch & Cut" a été introduite par [Ropke *et al.* 2007]. Puis elle est améliorée par [Ropke & Cordeau 2009] en utilisant une méthode de type "Branch & Cut & Price". Une autre heuristique est développée par [Nowak *et al.* 2008]. Cette méthode traitent une variante dans laquelle les fenêtres temporelles ne sont pas prises en considération et dans laquelle chaque requête peut être divisée et traitée par plusieurs véhicules. Un algorithme mémétique, présenté par [Nagata & Kobayashi 2010], donne de meilleurs résultats que [Ropke & Pisinger 2006a] sur plusieurs instances mais avec des temps de calcul plus longs. Une variante sélective du PDP (SPDP) a été résolue par [Liao & Ting 2010] en utilisant un algorithme génétique. Dans cette variante, les auteurs ont considéré une flotte composée d'un seul véhicule et ils ont résolu le problème en optimisant un seul objectif. Pour cette même variante, un algorithme génétique hybride a été présenté par [Liao & Ting 2012]. Une méthode exacte basée sur un algorithme de génération de colonnes a été développée par [Venkateshan & Mathur 2011] pour résoudre le PDPTWPD avec la possibilité de visiter chaque site plusieurs fois. Toujours dans le même contexte mono-objectif, [Ting & Liao 2013] ont proposé un algorithme mémétique pour la résolution du SPDP. Pendant la même année, une méta-heuristique de type "PSO" est développée par [Zou *et al.* 2013] pour résoudre le PDPTWPD avec plusieurs objectifs. Dans ce même contexte multi-objectifs, un autre algorithme mémétique est présenté par [Liao & Ting 2013] pour



résoudre le SPDP.

Une méthode de type "Branch & Price" a été proposée par [Bettinelli *et al.* 2014] pour résoudre une variante de PDPTWPD avec plusieurs dépôts. Une méthode heuristique pour résoudre le PDPTWPD avec contraintes de chargement de type LIFO a été proposée par [Benavent *et al.* 2015]. Ces contraintes imposent que la dernière requête chargée dans un véhicule doit être la première à être déchargée. Récemment, une nouvelle formulation mathématique utilisant deux indices a été proposée par [Furtado *et al.* 2017] pour résoudre le PDPTWPD. Cette nouvelle modélisation suppose que la flotte est composée des véhicules homogènes.

Ces divers travaux sont résumés dans le tableau 1.3. La première colonne rappelle la référence de chaque étude. La deuxième colonne indique le nombre de dépôts considérés dans le problème. Le nombre de véhicules disponibles ainsi que le fait d'avoir une capacité limitée ou illimitée sont donnés dans la troisième et la quatrième colonne respectivement. Les cinquième, sixième et septième colonnes indiquent respectivement si les fenêtres temporelles, les demandes appairées et l'aspect sélectif sont pris en compte ou non. La huitième colonne donne le nombre d'objectifs pour chaque variante. Enfin la dernière colonne indique la méthode de résolution utilisée.

Variante	Référence	Nb de dépôts		Nb de véhicules		Capacité limitée	TW	Demandes appairées	Sélective	Nb d'objectives		Méthode de résolution
		1	>1	1	>1					1	>1	
PDPTWPD	[Nanry & Barnes 2000]	X			X	X	X	X		X		Recherche tabou
	[Li & Lim 2003]	X			X	X	X	X			X	Recherche tabou et recuit simulé
	[Lu & Dessouky 2004]		X		X	X	X	X		X		Branch & Cut
	[Pankratz 2005]	X			X	X	X	X		X		Algorithme génétique
	[Lu & Dessouky 2006]	X			X	X	X	X			X	Heuristique
	[Ropke & Pisinger 2006a]		X		X	X	X	X		X		ALNS
	[Ropke <i>et al.</i> 2007]	X			X	X	X	X		X		Branch & Cut
	[Ropke & Cordeau 2009]	X			X	X	X	X		X		Branch & Cut & Price
	[Nagata & Kobayashi 2010]	X			X	X	X	X			X	Algorithme mémétique
	[Venkateshan & Mathur 2011]		X		X	X	X	X		X		Génération de colonnes
	[Zou <i>et al.</i> 2013]	X			X	X	X	X			X	PSO
	[Bettinelli <i>et al.</i> 2014]		X		X	X	X	X		X		Branch & Price
	[Benavent <i>et al.</i> 2015]	X			X	X	X	X		X		Heuristique
[Furtado <i>et al.</i> 2017]		X		X	X	X	X		X		MILP	
PDPPD	[Carrabs <i>et al.</i> 2007]	X	X		X			X		X		Branch & Bound
	[Nowak <i>et al.</i> 2008]	X			X	X		X		X		Heuristique
SPDP	[Liao & Tmg 2010]	X	X		X				X	X		Algorithme génétique
	[Liao & Tmg 2012]	X	X		X				X	X		Algorithme génétique hybride
	[Tmg & Liao 2013]	X	X		X				X	X		Algorithme mémétique
	[Liao & Tmg 2013]	X	X						X		X	Algorithme mémétique

TABLE 1.3 – Quelques variantes du PDP

Nous pouvons conclure à partir de cet état de l'art que le PDP et ses variantes attirent toujours l'attention de la communauté scientifique. Plusieurs aspects du problème ont été fortement traités dans la littérature (les demandes appairées, les fenêtres temporelles, la flotte avec plusieurs véhicules). Par contre, il y a d'autres caractéristiques qui sont moins considérées (le PDP avec plusieurs dépôts, l'aspect sélectif et les problèmes multi-objectifs). Dans le cadre de ma thèse, deux nouvelles variantes qui englobent la majorité des aspects abordés dans le tableau 1.3 sont introduites : la première consiste à traiter un problème sélectif et la deuxième est une variante multi-périodes. Nous avons choisi ces deux variantes parce qu'elles représentent deux formes possibles de la collaboration entre les chargeurs.

## 1.7 Conclusion

Ce premier chapitre nous a permis de poser les bases des travaux décrits dans le reste du mémoire. Nous y avons défini les principales notions de la logistique urbaine, ses enjeux parfois contradictoires, avec un focus sur quelques projets récents dans le domaine. Nous avons décrit succinctement le projet dans lequel nos travaux s'inscrivent, reposant sur le levier de la collaboration/mutualisation, notamment entre chargeurs.

Nous avons introduit le problème de planification de tournées traduisant cette collaboration, et dont nous avons décrit les variantes correspondant au cadre de notre étude, s'articulant autour du Pickup and Delivery Problem. Après avoir rappelé les concepts fondamentaux des méthodes de recherche opérationnelle permettant de traiter les problèmes d'optimisation considérés, nous avons terminé par un état de l'art dédié à la résolution des différentes variantes du PDP et donnant un aperçu des études récemment publiées sur le sujet.

Cette synthèse montre qu'il reste encore plusieurs variantes du PDP peu voire non-traitées par la communauté scientifique. Parmi ces dernières, deux nouvelles variantes ont fait l'objet de nos travaux de recherche. Elles seront introduites et résolues dans les chapitres suivants.

# Résolution du SPDPTWPD mono-objectif

*Dans les problèmes de collecte et de livraison (PDP), le but est de transporter des marchandises depuis les points de ramassage aux lieux de livraison, en utilisant un ensemble de véhicules, tout en respectant les différentes contraintes. Dans ce chapitre, nous étudions une nouvelle variante du PDP. Cette variante est le Selective Pickup and Delivery Problem with Time Windows and Paired Demands (SPDPTWPD) que nous modélisons puis résolvons à l'aide de méthodes exactes et approchées.*

## Sommaire

<b>2.1</b>	<b>Une nouvelle variante de Pickup and Delivery Problem</b>	<b>37</b>
<b>2.2</b>	<b>Formulation mathématique</b>	<b>39</b>
2.2.1	Notations	40
2.2.2	MILP pour le SPDPTWPD	40
2.2.3	Comparaison avec le VRP	43
<b>2.3</b>	<b>Méthodes exactes</b>	<b>45</b>
2.3.1	Modèle initial	45
2.3.2	Modèle amélioré	46
2.3.3	Résultats expérimentaux sur des problèmes de référence	47
2.3.4	Génération d'instances pour le SPDPTWPD	50
<b>2.4</b>	<b>Méthodes approchées par regroupements</b>	<b>52</b>
2.4.1	Regroupement basé sur la distance euclidienne :	53
2.4.2	Regroupement basé sur l'algorithme K-means :	54
2.4.3	Procédure de réparation	55
2.4.4	Résultats expérimentaux	56
<b>2.5</b>	<b>Piste d'amélioration</b>	<b>58</b>
<b>2.6</b>	<b>Conclusion</b>	<b>61</b>

## Publications

### Revue nationale :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "A new compact two-indexes formulation for a Pickup and Delivery Problem", *Revue Recherche Opérationnelle de OpenScience*, numéro spécial "Jeunes Chercheurs du GDR MACS au congrès mondial de l'IFAC WC 2017 - JD MACS 2017". (soumis)

**Conférences internationales :**

1. Al Chami, Z., Manier, H., Manier, M.-A., & Khalil, M.. "Two cluster-based approaches for the Pick-up and Delivery Problem with Time Windows", *5th annual workshop of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog2016)*, Nantes, France (june 6-8, 2016).
2. Al Chami, Z., Manier, H., & Manier, M.-A.. "Exact Model for a Selective Pick up and Delivery Problem with Time Windows", *session poster à 8th IFAC/IEEE/IFORS/IFIP Conference on Manufacturing Modelling Management and Control (MIM 2016)*, Troyes, France (june 28-30, 2016).
3. Al Chami, Z., Manier, H., & Manier, M.-A.. "New model for a variant of Pick up and Delivery Problem", *IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2016)*, pp.1708-1713, Budapest, Hungary (October 9-12, 2016).
4. Al Chami, Z., Manier, H., & Manier, M.-A.. "A new compact two-index formulation for a Pickup and Delivery Problem", *20th World Congress of the International Federation of Automatic Control (IFAC WC'17)*, Toulouse, France (9-14 July 2017).

**Conférence nationale :**

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "Méthode exacte pour la résolution du problème de collecte et de livraison sélectif avec fenêtres temporelles", *Dix-septième Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'16)*, Compiègne, France (10-12 février 2016).

**Nouvelles instances SPDPTWPD :**

Les nouvelles instances du SPDPTWPD générées dans ce chapitre sont accessibles via le lien : [https://drive.google.com/drive/folders/1ic9RwY1qqNhAA\\_69-hnuqZxKXj4eZGRm?usp=sharing](https://drive.google.com/drive/folders/1ic9RwY1qqNhAA_69-hnuqZxKXj4eZGRm?usp=sharing).

## 2.1 Une nouvelle variante de Pickup and Delivery Problem

Motivés par le contexte de la thèse (projet ANR TCDU) et par l'étude de la littérature dans le domaine de la collecte et la livraison de marchandises, nous avons choisi de nous concentrer sur la modélisation et la résolution d'une variante de PDP, que nous appelons SPDPTWPD, pour Selective Pickup and Delivery Problem with Time Windows and Paired Demands. Les caractéristiques de notre problème sont :

(a) **Les éléments :**

- Des demandes de transport : elles désignent un besoin de transporter une quantité de marchandises entre une origine et sa destination précise (demande appairée). Le transporteur gagne un profit qui varie selon la demande honorée.
- Des sites : ils sont classés en deux catégories : les fournisseurs et les clients. À chaque site est associé une fenêtre temporelle, une quantité (à collecter ou à livrer), un temps de service pour le chargement ou déchargement des marchandises et un profit. Nous supposons que la capacité de tous les sites, y compris le dépôt, est infinie.
- Un dépôt : c'est un site particulier qui représente une interface permettant de faciliter la relation entre la ville et l'extérieur (par exemple : un Centre de Distribution Urbaine (CDU) tel que défini au chapitre 1). Ainsi, toutes les marchandises arrivant de l'extérieur sont déchargées au dépôt pour être livrées par la suite aux clients finaux. Il en va de même pour les marchandises arrivant de la ville qui vont être expédiées vers l'extérieur. Le dépôt représente donc un point de départ et d'arrivée des véhicules. L'origine de chaque demande de transport peut être un site fournisseur ou le dépôt. De même, la destination peut être associée à un site client ou encore au dépôt. Dans un contexte urbain, cela génère trois types de flux : le premier relie le dépôt et un client (flux entrant), le deuxième relie un fournisseur et le dépôt (flux sortant) et le dernier fait le lien entre un fournisseur et un client (flux intra-urbain).
- Une flotte de véhicules : Elle est composée d'un nombre limité de véhicules basés sur le dépôt. Chaque véhicule a une vitesse et une capacité. La flotte est soit homogène (tous les véhicules possèdent les mêmes caractéristiques), soit hétérogène (chaque véhicule a ses propres caractéristiques).

(b) **Les contraintes :**

Il faut noter que nous traitons une variante du PDP dans un contexte urbain. En effet, dans un tel contexte, les politiques appliquées par les autorités publiques dans certaines villes génèrent des contraintes additionnelles à respecter. Par exemple : limiter les heures de transport par jour pour réduire le bruit (plages

horaires restreintes pour l'accès à certains sites ou à certaines zones); interdire aux grands véhicules d'accéder à certaines zones (une flotte composée de véhicules avec une capacité limitée). Ainsi, les autorités publiques encouragent le remplissage de véhicules. Pour cela nous mutualisons un véhicule pour satisfaire plusieurs demandes de transport (contraintes de précedence). Nous considérons les contraintes suivantes :

- **Des fenêtres temporelles :**

Une fenêtre temporelle est un intervalle de temps compris entre un horaire d'ouverture et de fermeture. La visite d'un site doit être accomplie pendant sa fenêtre temporelle. Si un véhicule arrive avant l'ouverture du site, il doit attendre le début de sa fenêtre pour commencer le service (chargement ou déchargement des marchandises). Par contre, la visite d'un site après sa fermeture est impossible.

- **Des contraintes de précedence :**

Comme les demandes sont appairées : le véhicule doit visiter le fournisseur avant son client, pendant sa tournée.

- **Des contraintes de capacité :**

À tout moment de sa tournée, chaque véhicule ne peut pas transporter une quantité (ou volume) de marchandises supérieure à sa capacité.

- **La sélection des sites à visiter :**

La visite de tous les sites n'est pas obligatoire puisque nous étudions une variante sélective du PDP. En effet, la satisfaction de toutes les demandes de transport sur une période donnée (par exemple, une journée) est parfois impossible à cause de la capacité limitée des véhicules, de l'existence de fenêtres temporelles et des contraintes de précedence. Dans ce cas nous devons choisir un sous-ensemble de demandes à satisfaire dans cette période donc de sites associés à ces demandes. Le profit associé à chaque demande, et de fait à son origine et à sa destination, est utilisé pour choisir les sites à visiter. Par exemple, un transporteur a un ensemble de demandes à effectuer dans la semaine. S'il planifie les tournées de lundi, les demandes non-satisfaites ce jour-là seront satisfaites les jours suivants sans d'ailleurs avoir la certitude que tout sera livré au cours de cette semaine.

(c) **Les objectifs :**

- La minimisation de la distance : cet objectif vise à minimiser la distance totale parcourue par les véhicules afin de satisfaire les demandes de transport. Notre choix s'est porté sur ce premier objectif parce qu'il est classiquement utilisé dans la littérature. Par contre, son utilisation dans le contexte d'une variante sélective conduit à générer une solution dans laquelle aucun site ne sera visité. Pour cela, nous sommes obligés de le coupler avec un deuxième objectif (voir chapitre 3).

- La maximisation du profit : nous avons mentionné précédemment que la satisfaction de toutes les demandes est parfois impossible. L'utilisation de cet objectif permet de choisir les sites à visiter en maximisant le profit total collecté par les véhicules.

Le SPDPTWPD peut être trouvé dans de nombreuses applications pratiques et réelles comme la livraison de courriers et le commerce électronique. Il se rencontre lorsque des entreprises doivent livrer ou/et collecter des marchandises dans des zones urbaines, mais toutes les demandes ne peuvent pas être satisfaites sur la même période par une flotte donnée de véhicules.

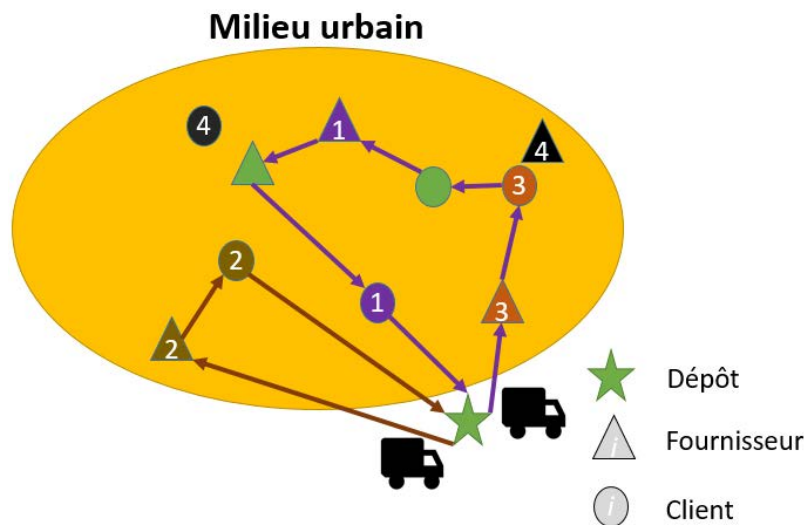


FIGURE 2.1 – Exemple de SPDPTWPD

La figure 2.1 montre un exemple de SPDPTWPD avec un seul dépôt, une flotte de deux véhicules et un ensemble de nœuds composé de cinq fournisseurs et cinq clients. Dans cette figure, l'étoile représente le dépôt. Un triangle est associé à un fournisseur et un cercle à un client. Une demande appairée est identifiée grâce à une couleur commune et un même numéro associés aux paires de nœuds fournisseur et client correspondants. Les nœuds numérotés 4 n'ont pas été sélectionnés et ils ne sont inclus dans aucune tournée. Leur visite devra être planifiée ultérieurement.

Le problème étant maintenant posé, une formulation mathématique pour cette variante en est d'abord proposée. Ensuite, nous décrivons les méthodes de résolution exactes et approchées que nous avons développées pour résoudre la nouvelle variante dans un contexte mono-objectif.

## 2.2 Formulation mathématique

Cette section présente un modèle mathématique qui est une description formelle du SPDPTWPD. Nous fournissons au préalable la notation utilisée pour les données et variables du problème.

### 2.2.1 Notations

Données :

- $C$  : Ensemble des clients,
- $B$  : Ensemble des fournisseurs,
- $V$  : Ensemble des véhicules disponibles,
- $Nodes = B \cup C$  : Ensemble des sites (clients et fournisseurs), aussi appelés nœuds,
- $N = 0 \cup Nodes$  : Ensemble de tous les sites plus le dépôt avec l'identifiant 0,
- $Customer_i$  : Client du fournisseur  $i/i \in B$ ,
- $Supplier_i$  : Fournisseur du client  $i/i \in C$ ,
- $d_{ij}$  : Distance du site  $i$  au site  $j/i, j \in N$ ,
- $P_i$  : Profit associé au site  $i/i \in N$ ,
- $q_i$  : Quantité demandée par le site  $i/i \in N$ 
  - Si  $q_i < 0$  alors  $i$  est un client,
  - Si  $q_i > 0$  alors  $i$  est un fournisseur,
- $[e_i, l_i]$  : Fenêtre temporelle associée au site  $i/i \in N$ . Le site  $i$  peut être visité par un véhicule à n'importe quel moment compris entre la borne inférieure  $e_i$  et la borne supérieure  $l_i$ ,
- $S_i$  : Temps de service au site  $i/i \in N$ . Il correspond au temps de chargement ou de déchargement d'un véhicule,
- $Q^k$  : Capacité maximale du véhicule  $k/k \in V$ ,
- $Speed^k$  : Vitesse moyenne du véhicule  $k/k \in V$ ,
- $M$  : Un très grand nombre.

Variables :

- $D_i^k$  : Date de départ du véhicule  $k$  du site  $i/k \in V$  et  $i \in N$ ,
- $A_i^k$  : Date de début de service du véhicule  $k$  au site  $i/k \in V$  et  $i \in N$ ,
- $Y_i^k$  : Quantité de marchandises dans le véhicule  $k$  après la visite du site  $i/k \in V$  et  $i \in N$ ,
- $X_{ij}^k = \begin{cases} 1 & \text{Si le véhicule } k \text{ visite directement le site } j \text{ après le site } i/k \in V \text{ et} \\ & i, j \in N \\ 0 & \text{Sinon} \end{cases}$

### 2.2.2 MILP pour le SPDPTWPD

Le problème traité peut être modélisé de la façon suivante :

$$\text{Minimiser/Maximiser } F \tag{2.1}$$



Sous les contraintes :

$$\sum_{i \in N} \sum_{k \in V} X_{ij}^k \leq 1 \quad \forall j \in Nodes \quad (2.2)$$

$$\sum_{j \in N} \sum_{k \in V} X_{ij}^k \leq 1 \quad \forall i \in Nodes \quad (2.3)$$

$$\sum_{i \in N} X_{iu}^k - \sum_{j \in N} X_{uj}^k = 0 \quad \forall k \in V; \forall u \in Nodes \quad (2.4)$$

$$\sum_{i \in Nodes} X_{0i}^k = 1 \quad \forall k \in V \quad (2.5)$$

$$\sum_{j \in Nodes} X_{j0}^k = 1 \quad \forall k \in V \quad (2.6)$$

$$Y_0^k = \sum_{i \in C/Supplier_i=0} (-q_i \sum_{j \in N/j \neq i} X_{ji}^k) \quad \forall k \in V \quad (2.7)$$

$$Y_j^k \geq Y_i^k + q_j - M(1 - X_{ij}^k) \quad \forall i, j \in N; \forall k \in V \quad (2.8)$$

$$Y_j^k \leq Y_i^k + q_j + M(1 - X_{ij}^k) \quad \forall i, j \in N; \forall k \in V \quad (2.9)$$

$$Q^k \geq Y_i^k \geq 0 \quad \forall i \in N; \forall k \in V \quad (2.10)$$

$$A_j^k \geq D_i^k + \frac{d_{ij}}{Speed^k} - M(1 - X_{ij}^k) \quad \forall i, j \in N; \forall k \in V \quad (2.11)$$

$$D_i^k \geq A_i^k + S_i - M(1 - X_{ij}^k) \quad \forall i, j \in Nodes; \forall k \in V \quad (2.12)$$

$$D_i^k \leq A_i^k + S_i + M(1 - X_{ij}^k) \quad \forall i, j \in Nodes; \forall k \in V \quad (2.13)$$

$$e_i \sum_{j \in N} X_{ij}^k \leq A_i^k \leq l_i \sum_{j \in N} X_{ij}^k \quad \forall i \in N; \forall k \in V \quad (2.14)$$

$$D_0^k = 0 \quad \forall k \in V \quad (2.15)$$

$$D_f^k \leq A_c^k \quad \forall f \in B; \forall c \in C/Supplier_c = f; \forall k \in V \quad (2.16)$$

$$\sum_{i \in N/i \neq Supplier_c} X_{iSupplier_c}^k - \sum_{j \in N/j \neq c} X_{cj}^k = 0 \quad \forall c \in C/Supplier_c \neq 0; \forall k \in V \quad (2.17)$$

Notre fonction objectif F (2.1) représente les principaux objectifs utilisés dans la littérature, à savoir le profit ou la distance à optimiser :

- La maximisation de profit :  $F1 = \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} P_i X_{ij}^k$
- La minimisation de distance :  $F2 = \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} d_{ij} X_{ij}^k$

Les contraintes (2.2) et (2.3) garantissent que chaque nœud est visité par un seul véhicule et au plus une fois. La continuité du flux est garantie par les contraintes (2.4). Les contraintes (2.5) et (2.6) assurent que chaque tournée commence et termine au dépôt. Les contraintes de capacité des ressources de transport sont respectées

grâce à (2.7), (2.8), (2.9) et (2.10). Les contraintes (2.11), (2.12), (2.13), (2.14) et (2.15) garantissent le respect des fenêtres temporelles. Les contraintes de précédence (2.16) assurent que la date de départ du véhicule d'un fournisseur est inférieure à celle d'arrivée à son client. Les contraintes (2.17) sont utilisées pour assurer que le fournisseur et son client sont visités par le même véhicule. Ces contraintes sont redondantes avec les contraintes (2.16) dans plusieurs cas. Nous détaillons ci-dessous les scénarios possibles, en fonction que le dépôt est (ou n'est pas) fournisseur ou client d'une demande donnée :

1. Si une demande n'est pas appairée avec le dépôt :
  - (a) Si un fournisseur F1 et son client C1 sont choisis dans la solution finale mais desservis par le véhicule k1 et k2 respectivement. D'après les contraintes des fenêtres temporelles, nous aurions :
 
$$D_{F1}^{k1} = \text{Date de départ du véhicule k1 du fournisseur F1}$$

$$D_{F1}^{k2} = 0$$

$$A_{C1}^{k1} = 0$$

$$A_{C1}^{k2} = \text{Date de début de service du véhicule k2 au client C1.}$$
 Les contraintes (2.16) interdisent ce cas.
  - (b) Si le fournisseur F1 est choisi dans la solution finale et desservi par le véhicule k1, mais son client n'est pas sélectionné :
 
$$D_{F1}^{k1} = \text{Date de départ du véhicule k1 du fournisseur F1}$$

$$D_{F1}^{k2} = 0$$

$$A_{C1}^{k1} = 0$$

$$A_{C1}^{k2} = 0.$$
 Les contraintes (2.16) interdisent également ce cas.
  - (c) Si le client C1 est choisi dans la solution finale et desservi par le véhicule k2, mais son fournisseur n'est pas sélectionné :
 
$$D_{F1}^{k1} = 0$$

$$D_{F1}^{k2} = 0$$

$$A_{C1}^{k1} = 0$$

$$A_{C1}^{k2} = \text{Date de début de service du véhicule k2 au client C1.}$$
 Les contraintes (2.16) n'interdisent pas ce cas et donc il est nécessaire d'utiliser les contraintes (2.17) ci-dessus.
2. Si une demande est appairée avec le dépôt :
  - (a) Si un fournisseur F a le dépôt comme client. Dans ce cas, il n'y a aucun problème au niveau du véhicule par rapport à la contrainte de précédence, mais bien entendu la solution doit respecter les autres contraintes imposées par le problème.
  - (b) C'est également le cas, si un client C (associé, à titre d'exemple, avec une quantité égale à -10) a le dépôt comme fournisseur et est desservi par le véhicule k2. Ici, il faut vérifier que la demande du client C soit chargée dans le véhicule k2 et pas dans un véhicule k1. D'après les contraintes (2.7) :

$$\begin{aligned} Y_0^{k1} &= 0 \\ Y_0^{k2} &= 10 \end{aligned}$$

Donc ces contraintes vérifient que les marchandises du client sont bien chargées dans le véhicule correspondant.

C'est l'aspect sélectif de notre problème qui nécessite l'utilisation des contraintes (2.17). Celles-ci ne seront utiles que dans le cas de type (1.c) expliqué ci-dessus, qui ne se rencontre pas si souvent.

Le SPDPTWPD est une extension du PDP donc il appartient à la classe des problèmes NP-difficiles. Dans la littérature, la plupart des modèles de routage qui traitent l'aspect sélectif supposent que chaque site est visité au plus une fois. Ces modèles ne prennent pas en considération le cas où des sites peuvent être visités plusieurs fois. Par exemple, un fournisseur qui livre à la fois plusieurs clients, ou un client qui demande d'être livré par plus d'un seul fournisseur. Pour prendre en compte de tels cas, les sites communs à plusieurs demandes seront dupliqués autant de fois que nécessaire. Nous créons donc un nœud artificiel par demande supplémentaire pour chaque site associé à plus d'une demande. Ce nœud artificiel partage la même localisation géographique que le site d'origine et peut avoir des caractéristiques différentes au niveau de la quantité demandée, de l'identité de son fournisseur ou client... Par conséquent, ces nœuds artificiels sont nécessaires pour garantir que :

- (1) Chaque demande est réalisée seulement entre un fournisseur spécifique et son client.
- (2) Chaque véhicule peut visiter le même site autant de fois que le nombre de demandes associées.

À l'aide de la formulation mathématique utilisée, il n'est pas nécessaire de dupliquer le dépôt puisque les contraintes (2.7) calculent la quantité à charger dans chaque véhicule au cas où, dans sa tournée, celui-ci visite un ou plusieurs clients associés au dépôt.

### 2.2.3 Comparaison avec le VRP

Nous présentons dans cette sous-section une comparaison entre le PDP en général et le VRP afin de montrer la complexité de notre variante. Nous nous focalisons sur les contraintes utilisées pour assurer le respect de capacité. En effet, la seule différence entre les deux problèmes est la possibilité, dans les PDPs, de collecter et de livrer des marchandises dans la même tournée. Par conséquent, nous devons vérifier la capacité résiduelle des véhicules à chaque nœud. Au contraire, dans les VRPs, le véhicule ne fait que de la collecte (ou de la livraison). Il suffit alors de vérifier une seule fois que la somme de toutes les quantités collectées (ou livrées) ne dépasse pas la capacité. Nous n'avons donc besoin que des contraintes (2.18) décrites ci-dessous pour vérifier la limite de capacité :

$$Y_0^k + \sum_{i \in N} \sum_{j \in N} (X_{ij}^k q_j) \leq Q^k \quad k \in V \quad (2.18)$$

Avec :

$$Y_0^k = \begin{cases} \sum_{i \in C / \text{Supplier}_i=0} (-q_i \sum_{j \in N / j \neq i} X_{ji}^k) & k \in V \quad : \text{Tournée de livraison} \\ 0 & \quad : \text{Tournée de collecte} \end{cases}$$

Les contraintes (2.18) sont remplacées par les contraintes (2.7), (2.8), (2.9) et (2.10) dans notre modèle. Nous allons considérer maintenant un exemple simple de notre problème pour prouver que les contraintes (2.18) ne peuvent être utilisées que dans le contexte du VRP. Dans cet exemple, nous avons une flotte composée d'un seul véhicule, un dépôt et six nœuds (trois fournisseurs et trois clients). Le véhicule unique est caractérisé par une capacité maximale de 100 et une vitesse moyenne égale à 1. Le tableau 2.1 fournit les détails de notre exemple. La quantité de marchandises demandée par chaque site est donnée par la deuxième colonne. Les fournisseurs sont les nœuds avec des valeurs de quantité positives et les clients sont ceux avec les valeurs négatives. Ainsi sur la deuxième ligne, on lit que le nœud 1 est le client du dépôt qui doit lui livrer une quantité égale à 30 dans la fenêtre temporelle [30,500]. Le temps de déchargement est 10.

Id du nœud	Quantité demandée	Fournisseur(s)	Client(s)	Fenêtre temporelle associée	Temps du service
Dépôt	50	4 et 6	1 et 2	[0,200]	0
1	-30	Dépôt	-	[30,500]	10
2	-20	Dépôt	-	[40,600]	10
3	-20	5	-	[50,700]	10
4	20	-	Dépôt	[0,20]	10
5	20	-	3	[10,30]	10
6	20	-	Dépôt	[20,50]	10

TABLE 2.1 – Données de l'exemple

Le tableau 2.2 fournit la matrice de distances entre les nœuds.

	Dépôt	1	2	3	4	5	6
Dépôt	0	20	25	30	5	10	15
1	20	0	35	50	10	15	45
2	25	35	0	40	15	20	35
3	30	50	40	0	20	25	25
4	5	10	15	20	0	10	55
5	10	15	20	25	10	0	10
6	15	45	35	25	55	10	0

TABLE 2.2 – Matrice de distances

En utilisant uniquement les contraintes (2.18) pour modéliser la limite de capacité, nous avons obtenu la solution optimale suivante : (Dépôt  $\Rightarrow$  4  $\Rightarrow$  5  $\Rightarrow$  6  $\Rightarrow$  3  $\Rightarrow$  2  $\Rightarrow$  1  $\Rightarrow$  Dépôt). Mais si on vérifie la capacité du véhicule après avoir desservi chaque nœud, on peut observer que la capacité maximale est dépassée après avoir servi le fournisseur 6. Les détails de la solution sont rapportés dans le tableau 2.3. Le véhicule

part du dépôt avec une charge de 50. Il doit collecter une quantité de 20 sur les trois premiers sites visités, cette quantité de 60 s'ajoute au 50 unités initiales. Donc arrivé sur le site 6, le véhicule n'aura plus assez de capacité pour charger la totalité de la demande. Le nombre de contraintes utilisées, dans le cadre du VRP, pour vérifier le respect de la capacité est égal à  $|V|$  sans la nécessité d'utiliser des variables supplémentaires, avec  $|V|$  représentant le nombre de véhicules disponibles. En outre, dans un PDP, le nombre de contraintes utilisées s'élèvent à  $|N| * |V| * (2 * |N| + 1)$  avec la nécessité d'utiliser  $|N| * |V|$  variables supplémentaires, avec  $|N|$  représentant le nombre de sites. Cette différence, entre le VRP et le PDP, en termes de contraintes et de variables explique la complexité de traiter le PDP ainsi que ses variantes.

	Quantité chargée	Quantité déchargée	Quantité de marchandises dans le véhicule au passage	Capacité restante
Dépôt	50	-	50	50
4	20	-	70	30
5	20	-	90	10
6	20	-	<b>110</b>	<b>-10</b>
3	-	-20	90	10
2	-	-20	70	30
1	-	-30	40	60

TABLE 2.3 – Solution détaillée

Dans cette sous-section, nous avons montré que dans toutes les variantes du PDP, la modélisation de la capacité est plus complexe que dans les VRPs et peut nécessiter plusieurs équations/inéquations pour vérifier que la capacité n'est pas dépassée après avoir desservi chaque nœud. Cela vient de la nécessité de vérifier le niveau de charge ou capacité résiduelle à chaque site visité. Une vérification au début ne suffit plus. La faisabilité de la tournée dépend alors en partie de l'ordre de visite des sites.

Dans la partie suivante, nous allons décrire les différentes méthodes de résolution que nous avons utilisées pour notre variante en considérant un seul objectif.

## 2.3 Méthodes exactes

### 2.3.1 Modèle initial

Le modèle de base que nous avons présenté dans la section 2.2 est un programme linéaire mixte en nombres entiers. Nous l'avons d'abord résolu à l'aide d'un solveur linéaire (Gurobi) en prenant la maximisation de profit total collecté comme fonction objectif (F1).

Nous avons choisi de maximiser le profit au lieu de minimiser la distance totale parcourue parce que, dans un problème sélectif, la minimisation de la distance conduit à une solution optimale dans laquelle aucun site n'est desservi (distance = 0).

Cependant, compte tenu de la complexité de notre variante, seules les petites instances peuvent être résolues à l'optimalité dans un temps raisonnable. Donc, pour améliorer la performance du modèle initial, nous proposons cinq contraintes additionnelles comme décrit dans la partie suivante.

### 2.3.2 Modèle amélioré

Pour accélérer la résolution de notre MILP initial, nous ajoutons les contraintes supplémentaires suivantes (réparties en trois catégories) :

- **Incompatibilité temporelle entre deux nœuds i et j :**

$$If(\max(e_i, \frac{d_{0i}}{Speed^k}) + S_i + \frac{d_{ij}}{Speed^k} > l_j) \\ X_{ij}^k = 0 \quad \forall i, j \in N; \forall k \in V \quad (C1)$$

Les contraintes (C1) interdisent la visite d'un site si le véhicule ne peut l'atteindre avant la date limite de visite (borne supérieure de la fenêtre temporelle du site). Par conséquent, si le temps de service du nœud i entraîne un retard dans le service du nœud j en raison de la fenêtre temporelle associée, alors la succession de nœuds (i,j) est incompatible. Donc les véhicules ne peuvent pas voyager entre ces deux nœuds. Le principe de ces contraintes est représenté par la figure 2.2.

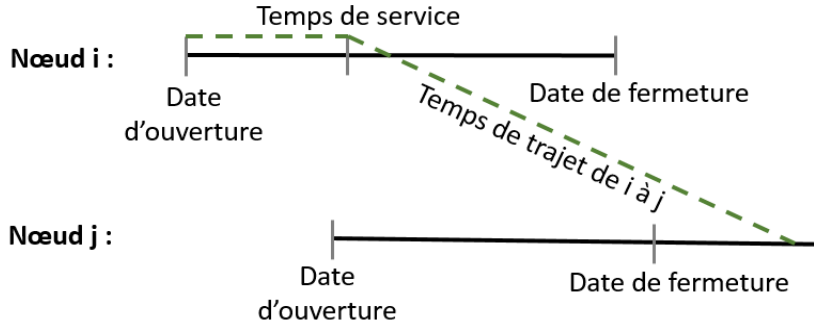


FIGURE 2.2 – Contraintes additionnelles (C1)

Ici l'ordre des sites compte : par exemple la visite de j directement après i peut être irréalisable, mais la visite inverse (i directement après j) peut être faisable.

- **Appairage entre fournisseurs et clients :**

$$\sum_{k \in V} X_{ij}^k = 0 \quad \forall i \in C / j \neq 0 \text{ et } j = \text{Supplier}_i \quad (C2.1)$$

$$\sum_{k \in V} X_{0j}^k = 0 \quad \forall j \in C / \text{Supplier}_j \neq 0 \quad (C2.2)$$

$$\sum_{k \in V} X_{i0}^k = 0 \quad \forall i \in B/ \text{Customer}_i \neq 0 \quad (C2.3)$$

Ces contraintes (voir figure 2.3) signifient que pour toute demande, aucun client ne peut être visité avant son fournisseur. Cette catégorie contient trois contraintes différentes pour englober les trois types de flux traités dans notre variante.



FIGURE 2.3 – Contraintes additionnelles (C2)

- **Limite de capacité de chaque véhicule :**

$$\text{If}(q_i + q_j > Q^k) \quad X_{ij}^k = 0 \quad \forall i, j \in N; \forall k \in V \quad (C3)$$

Les contraintes (C3) garantissent que le véhicule  $k$  ne peut pas visiter le nœud  $j$  directement après le nœud  $i$  si la somme des quantités prélevées sur ces deux nœuds dépasse la capacité du véhicule (figure 2.4). (C3) n'a d'intérêt que pour des sites successifs de type fournisseur. En effet nous traitons un problème de type "less than truckload" :  $q_i \leq Q^k \quad \forall i \in N; \forall k \in V$ . Si 1 des 2 sites (par exemple  $j$ ) est un client alors  $q_j < 0$  donc  $q_i + q_j \leq q_i \leq Q^k$  (contrainte vérifiée de fait et (C3) n'apporte rien).

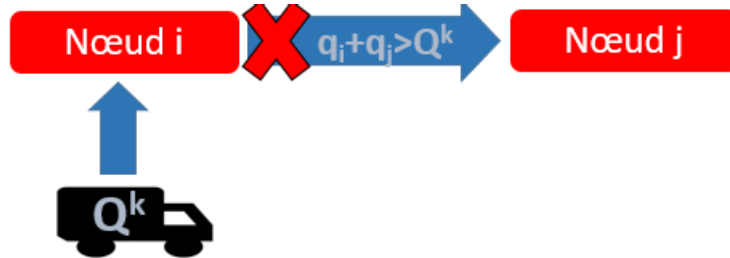


FIGURE 2.4 – Contraintes additionnelles (C3)

### 2.3.3 Résultats expérimentaux sur des problèmes de référence

À notre connaissance, aucune étude antérieure ne traite le SPDPTWPD dans la littérature. Il est donc difficile d'évaluer nos modèles à l'aide de tests comparatifs. Par conséquent, nous avons testé et validé nos deux modèles sur des instances de la littérature pour des problèmes proches et moins contraints. En particulier, nous avons vérifié les composantes de notre modèle initial liées à chaque contrainte spécifique, ainsi que l'efficacité de nos contraintes additionnelles. Les résultats décrits ci-après ont été obtenus en utilisant un solveur commercial Gurobi 6.0.5 et un ordinateur avec un processeur Intel Core i5, 2,80 GHz et 4 Go de RAM.

Pour valider notre modèle et tester l'efficacité de nos contraintes (C1), nous avons comparé nos modèles aux résultats obtenus sur le TOPTW par [El-Hajj 2015]

qui a utilisé une méthode exacte basée sur un MILP et couplée avec une approche heuristique pour fournir des bornes initiales. Dans le cas du TOPTW, la capacité des véhicules est supposée illimitée, alors nous pouvons relâcher les contraintes de capacité (2.7), (2.8), (2.9), (2.10) et (C3). Nous pouvons aussi résoudre ce problème sans supprimer ces contraintes en considérant que  $Q^k = \infty/k \in V$ . De plus, tous les nœuds sont des clients ( $B = \emptyset$ ) donc les contraintes : (2.16), (C2.1), (C2.2) et (C2.3) peuvent aussi être relâchées. Puisque le but du TOPTW est de maximiser le profit total collecté, nous avons utilisé notre fonction objectif F1. Le tableau 2.4 donne un exemple d'instances testées et signale l'écart de temps CPU entre notre modèle simple (SM) et notre modèle amélioré (IM).

Dans un deuxième temps, nous avons validé les contraintes additionnelles (C1) et (C3) sur quelques instances du VRPTW créées par [Solomon 1987]. Notons que dans le VRPTW, il n'y a pas d'aspect sélectif donc nous avons modifié nos contraintes (2.2) et (2.3) de " $\leq 1$ " à " $= 1$ " et nous avons utilisé F2 comme fonction objectif (minimisation de la distance). Dans ce cas, les contraintes (2.16), (C2.1), (C2.2) et (C2.3) sont également relâchées.

Nous avons également comparé nos modèles avec les résultats obtenus par [Li & Lim 2003] sur le PDPTWPD, afin de valider nos modèles pour ce problème, et de montrer l'efficacité de l'ensemble des nos contraintes additionnelles. Ici, nous ne pouvons pas relâcher les contraintes (2.16), (C2.1), (C2.2) et (C2.3) parce que nous avons des demandes appairées entre les fournisseurs et les clients.

Les résultats de ces tests pour quelques instances de la littérature sont présentés dans le tableau 2.4. Ce tableau signale l'écart de temps CPU entre notre modèle initial simple (SM) et notre modèle amélioré (IM). La colonne "Solution optimale" indique suivant les cas soit le meilleur profit, soit la meilleure distance.

Problème	Instance	Solution optimale	Temps CPU (s)		Déviation CPU %
			SM	IM	
TOPTW	C101_m1	320	5.89	5.18	12
	C101_m2	590	1413.93	510.21	64
	R101_m1	198	4.51	2.11	53
	R101_m2	349	3.45	3.12	10
	RC101_m1	219	1802.14	1220.41	32
VRPTW	C101.25	191.3	0.64	0.2	69
	C101.50	362.4	58.98	6.75	89
	C201	589.1	61.51	3.44	94
	C202.25	214.7	30.49	11.61	62
	C202.50	360.2	673.4	144.77	79
PDPTWPD	LC101	828.94	7500	173	98
	LC201	591.56	57.38	4	93
	LC202	591.56	6738	130	98
	LR101	1650.78	> 14400	1220	> 92

TABLE 2.4 – Comparaison des modèles SM et IM



Les deux modèles permettent de résoudre à l'optimum les instances des trois problèmes proches considérés. Le tableau 2.4 montre l'efficacité de chaque contrainte additionnelle car l'écart temporel entre notre SM et IM est d'au moins 10%, il peut atteindre 98% et il est en moyenne de 67.5%. Dans ce tableau, nous n'avons pas fait de comparaison avec le temps CPU des autres méthodes parce que les auteurs ont utilisé des méthodes approchées ou hybrides. En effet, ils ont normalement un temps de résolution plus faible mais l'avantage d'utiliser notre méthode exacte est que nous avons prouvé l'optimalité de certains de leurs résultats car nous avons obtenu les mêmes valeurs que celles rapportées par les auteurs pour toutes les instances testées. Dans ce qui suit, nous comparons notre modèle amélioré (IM) avec une autre étude de la littérature dans laquelle les auteurs utilisent la programmation linéaire pour résoudre de façon exacte leur variante. [Wang *et al.* 2015] ont développé une méthode exacte pour résoudre le PDPTW en utilisant le solveur commercial CPLEX pour les instances de petite taille (10, 25, 50 nœuds) et en utilisant un algorithme de recuit simulé parallèle (noté p-SA) pour les plus grandes instances (100 nœuds). Le tableau 2.5 donne une comparaison entre nos résultats et les leurs.

Instance	Nombre des nœuds	IM		Résultats de [Wang <i>et al.</i> 2015]			
		Solution optimale	CPU (s)	CPLEX		p-SA	
				Solution optimale	CPU (s)	Objectif	CPU (s)
RCdp1001/10	10	<b>348.98</b>	<b>0.56</b>	348.98	1	348.98	1
RCdp1004/10	10	<b>216.69</b>	<b>0.90</b>	216.69	1503	216.69	1
RCdp1007/10	10	<b>310.81</b>	<b>5.17</b>	310.81	25	310.81	1
RCdp2501/25	25	551.05	41.88	<b>551.05</b>	<b>16</b>	552.21	2
RCdp2504/25	25	<b>530.74</b>	<b>4612</b>	738.22*	485660	-	-
RCdp2507/25	25	<b>540.88*</b>	<b>64290</b>	634.20*	439321	-	-
Cdp101	100	<b>848.93</b>	<b>142</b>	-	-	992.88	36
Cdp201	100	<b>591.56</b>	<b>5.21</b>	-	-	591.56	86

\* : Meilleure borne

TABLE 2.5 – Comparaison avec [Wang *et al.* 2015]

Le tableau 2.5 montre que nous trouvons les solutions optimales avec notre IM plus rapidement que la méthode exacte utilisée par [Wang *et al.* 2015]. De plus, nous trouvons des solutions optimales lorsque leur méthode exacte ne fournit que des bornes supérieures après un long temps de résolution (RCdp2504/25) ou lorsque leur p-SA ne trouve pas la solution optimale (RCdp2501/25). Même lorsque notre temps de résolution est également trop long, nous fournissons de meilleures bornes que les leurs (RCdp2507/25). Pour les instances avec 100 nœuds, [Wang *et al.* 2015] n'utilisaient que le p-SA. Dans ce cas, nous arrivons parfois à résoudre des instances à l'optimalité (Cdp101) quand ils n'y arrivent pas.

Problème	IM		formulation de [Wang <i>et al.</i> 2015]	
	Opt	CPU (s)	Opt	CPU (s)
RCdp1001/10	<b>348.98</b>	<b>0.56</b>	348.98	0.86
RCdp1004/10	<b>216.69</b>	<b>0.90</b>	216.69	17
RCdp1007/10	<b>310.81</b>	<b>5.17</b>	310.81	7.75
RCdp2501/25	551.05	41.88	<b>551.05</b>	<b>6.35</b>
RCdp2504/25	<b>530.74</b>	<b>4612</b>	-	-
RCdp2507/25	-	-	-	-

TABLE 2.6 – Implémentation de la formulation dans [Wang *et al.* 2015]

Afin de comparer les temps de calcul des algorithmes et de prouver que notre IM est plus rapide que la méthode exacte utilisée par [Wang *et al.* 2015], nous avons implémenté la même formulation mathématique présentée par les auteurs et nous l'avons résolue en utilisant la même machine et le même solveur (dans notre cas, GUROBI). Le "-" indique que nous n'avons pas résolu à l'optimalité l'instance correspondante après deux heures de résolution. Les résultats obtenus sont rapportés dans le tableau 2.6 et montrent l'efficacité de notre IM par rapport au modèle développé par les auteurs dans presque tous les cas. Dans l'instance RCdp2501/25, la largeur des fenêtres temporelles est très petite par rapport aux autres instances. Et comme le nombre de solutions est assez limité, la résolution de cette instance est plus facile. Cela peut expliquer le fait que notre IM est moins efficace que la formulation des auteurs dans ce cas précis.

### 2.3.4 Génération d'instances pour le SPDPTWPD

Dans la section précédente, nous avons validé nos modèles à travers des tests et comparaisons sur des problèmes proches de la littérature. Pour confirmer l'efficacité de nos approches, nous avons également généré de nouvelles instances traitant le SPDPTWPD. Elles représentent les premières instances qui traitent cette variante dans la littérature. Pour les générer, nous avons utilisé les instances de référence présentées dans [Li & Lim 2003] dans lesquelles nous avons introduit :

1. L'aspect sélectif (il n'est pas obligatoire de visiter tous les nœuds).
2. Les trois types de flux (du dépôt aux clients, des fournisseurs aux clients et des fournisseurs au dépôt).

Les nouvelles instances générées sont divisées en trois classes :

- (a) SPDPTWPD-RXY : tous les sites sont distribués aléatoirement.
- (b) SPDPTWPD-CXY : tous les sites sont regroupés géographiquement.
- (c) SPDPTWPD-RCXY : dans cette classe, les sites sont partiellement groupés et partiellement distribués aléatoirement.

Le nom de chaque instance se termine par "Y" qui vaut : "1" pour les instances dont lesquelles toutes les demandes ont le même profit et "2" pour celles ayant une

distribution aléatoire des profits. Le "X" correspond au nombre de nœuds dans chaque instance. Des informations complémentaires ainsi que les données d'instances sont disponibles en annexe A.1 et à l'adresse : [https://drive.google.com/drive/folders/1ic9RwY1qqNhAA\\_69-hnuqZxKXj4eZGRm?usp=sharing](https://drive.google.com/drive/folders/1ic9RwY1qqNhAA_69-hnuqZxKXj4eZGRm?usp=sharing).

Le tableau 2.7 présente les résultats obtenus par nos modèles. Nous remarquons l'efficacité de notre modèle amélioré (IM) par rapport au modèle initial (SM). Le gain induit par les contraintes additionnelles peut atteindre 94.82% du temps de résolution et il est en moyenne de 56%.

Le temps de résolution dépend du nombre de véhicules, du celui de nœuds et du type de chaque instance, comme on peut les observer sur les figures 2.5. Il peut aussi dépendre d'autres caractéristiques comme la distribution des intervalles de temps... Chaque point des graphes représente la résolution d'une des instances du tableau 2.7.

Problème	Nombre de véhicules	Solution optimale	CPU (s)		Déviation CPU %
			SM	IM	
SPDPTWPD-C201	1	180	0.16	0.11	31.25
SPDPTWPD-C202	1	144	13.21	10.15	23.16
SPDPTWPD-R201	6	190	0.73	0.15	79.45
SPDPTWPD-R202	4	213	10.54	7.2	31.69
SPDPTWPD-RC201	2	160	0.37	0.13	64.86
SPDPTWPD-RC202	2	180	0.76	0.16	78.95
SPDPTWPD-C501	5	480	17.57	7.39	57.94
SPDPTWPD-C502	1	437	22.8	9.57	58.03
SPDPTWPD-R501	13	490	55.7	10.63	80.92
SPDPTWPD-R502	10	576	113.68	13.08	88.49
SPDPTWPD-RC501	4	320	1721.51	1125.42	34.63
SPDPTWPD-RC502	4	420	115.34	87.32	24.29
SPDPTWPD-C1001	6	690	2763.12	911.19	67.02
SPDPTWPD-C1002	7	770	9216.23	477	94.82
SPDPTWPD-R1001	7	500	7155.23	1470.72	79.45
SPDPTWPD-R1002	6	514	917.36	705.26	23.12
SPDPTWPD-RC1001	1	120	384.78	264.12	31.36
SPDPTWPD-RC1002	1	138	225.66	93.56	58.54

TABLE 2.7 – Premières instances pour le SPDPTWPD

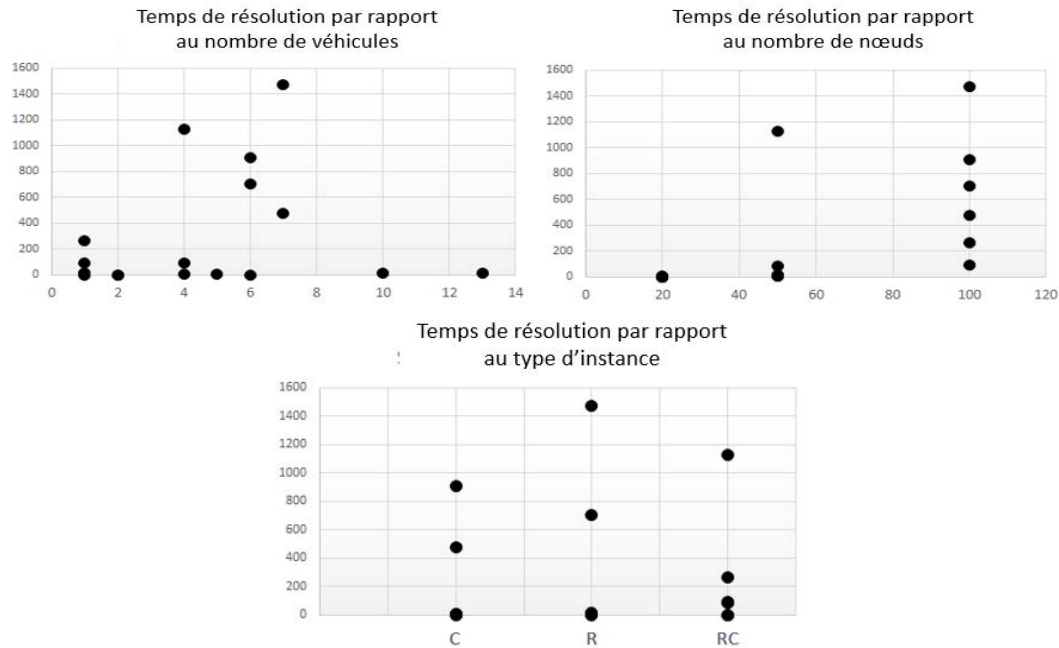


FIGURE 2.5 – Évolution du temps de résolution de notre IM

Tous les tests réalisés ont mis en évidence l'efficacité de nos modèles. Compte tenu de la complexité de notre variante, le temps de résolution reste relativement long. Pour réduire ce dernier, nous avons développé des méthodes approchées décrites dans la partie suivante.

## 2.4 Méthodes approchées par regroupements

Afin de surmonter la limitation des méthodes exactes, nous avons utilisé des approches de regroupements adaptées pour décomposer les grandes instances en un ensemble d'instances de taille plus réduite. L'idée consiste à effectuer pour chaque instance une étape de pré-traitement dans laquelle un regroupement des nœuds est effectué pour obtenir une formulation de problème basée sur des petits groupes (ou clusters). Donc l'optimalité de la solution n'est plus garantie car la somme des optima locaux peut être différente de l'optimum global. La figure 2.6 illustre le principe de regroupement appliqué sur l'exemple présenté précédemment (figure 2.1). Pour ce dernier, nous avons regroupé les nœuds de façon à obtenir deux clusters.

Nous pouvons observer qu'un tel regroupement peut générer des clusters infaisables si un fournisseur et son client sont affectés à deux clusters différents, d'où la nécessité d'utiliser une fonction de réparation qui a comme rôle d'assurer que chaque paire "fournisseur-client" appartient au même cluster.

Plusieurs algorithmes de regroupement ont été déjà utilisés dans la littérature : l'algorithme K-means [He *et al.* 2009], K-medoids [Coene *et al.* 2010], etc... Contrairement à ce type d'algorithmes dans lesquels on doit spécifier K (nombre de clusters), les

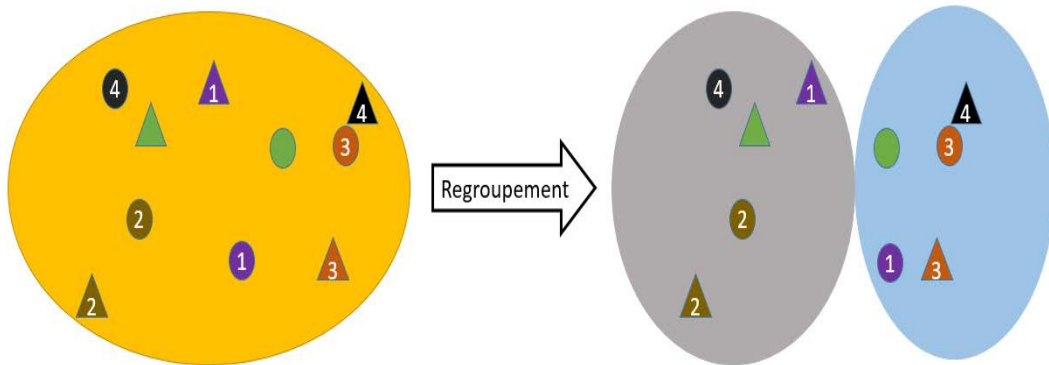


FIGURE 2.6 – Exemple de regroupement

méthodes de regroupement basées sur la densité ne nécessitent pas de fixer le nombre de clusters souhaité [Chen *et al.* 2017].

Dans cette partie, deux algorithmes de regroupement ont été développés : le premier est basé sur la distance euclidienne entre les différents nœuds et le deuxième est basé sur l’algorithme K-means.

#### 2.4.1 Regroupement basé sur la distance euclidienne :

L’algorithme proposé utilisant la distance euclidienne entre les nœuds est appelé Algorithme 1. Il utilise un paramètre rayon, noté *Cluster\_radius* et fixé à priori. L’algorithme 1 part d’un nœud choisi aléatoirement, et l’affecte par défaut au premier cluster. Nous mettons dans le même groupe tous les autres sites qui ont une distance euclidienne inférieure au rayon du cluster déjà fixé. Puis, nous prenons un autre nœud non-affecté et nous répétons le même processus avec un nouveau cluster. Nous arrêtons l’algorithme lorsque tous les nœuds sont affectés.

---

**Algorithme 1** Regroupement basé sur la distance euclidienne

---

**Input** : Nodes = Ensemble des nœuds, Cluster\_radius ;

**Output** : Clustered[Nodes] = Ensembles des nœuds regroupés ;

Clustered[Nodes] = 0 ;

Id = 1 ;

**for** each  $i$  in Nodes **do**

**if** Clustered[i]==0 **then**

        Clustered[i]==Id ;

**for** each  $j$  in Nodes **do**

**if** Clustered[j]==0 and  $i \neq j$  and  $d_{ij} \leq \text{Cluster\_radius}$  **then**

                Clustered[j]=Clustered[i] ;

**end**

**end**

        Id++ ;

**end**

**end**

---

Dans Algorithme 1, nous ne connaissons pas à l'avance le nombre de clusters que nous allons obtenir à la fin du regroupement, même si ce dernier est lié à la valeur choisie pour la variable Cluster\_radius. Dans le pire cas, le nombre de clusters est égal au nombre de nœuds. De plus, l'ordre dans lequel nous affectons les nœuds peut impacter les clusters obtenus à la fin.

### 2.4.2 Regroupement basé sur l'algorithme K-means :

L'algorithme de partitionnement K-means (ou K-moyennes en français) est une méthode très connue et elle est même considérée parmi les algorithmes de regroupement les plus utilisés dans la littérature [Jain 2010].

Étant donné un ensemble des nœuds et un entier K, l'objectif de l'algorithme est de diviser ces nœuds en K clusters de façon à minimiser la distance entre les points à l'intérieur de chaque partition. Autrement dit, le but est de minimiser la distance entre les nœuds et le barycentre de chaque cluster. Dans cet algorithme, nous commençons par donner une valeur à K qui représente le nombre de groupes, puis nous plaçons K points de manière aléatoire. Ces derniers représentent les barycentres des clusters initiaux. Après cela, nous attribuons chaque nœud au cluster ayant le barycentre le plus proche. Lorsque tous les nœuds ont été affectés, nous recalculons les positions des K barycentres. Pour un cluster donné, les coordonnées du barycentre sont respectivement la moyenne des abscisses et celle des ordonnées de ses points. L'algorithme s'arrête si aucun site n'a été réaffecté. Sinon, le processus se répète depuis le début avec les nouveaux barycentres (Algorithme 2).

**Algorithme 2** Regroupement basé sur K-means**Input** : Nodes = Ensemble des nœuds, K ;**Output** : Clustered[Nodes] = Ensembles des nœuds regroupés ;

Choisir K barycentres aléatoirement ;

**for** each  $i$  in Nodes **do**

Clustered[i]=identifiant du Cluster ayant un barycentre  $bar\_cluster$  tel que la distance entre  $i$  et  $bar\_cluster$  est minimal ;

**end**

Recalculer les barycentres de chaque cluster ;

Itérer l'algorithme jusqu'à ce que les objets ne changent plus de cluster ;

La figure 2.7 montre un exemple d'application de l'algorithme avec  $K = 2$ .

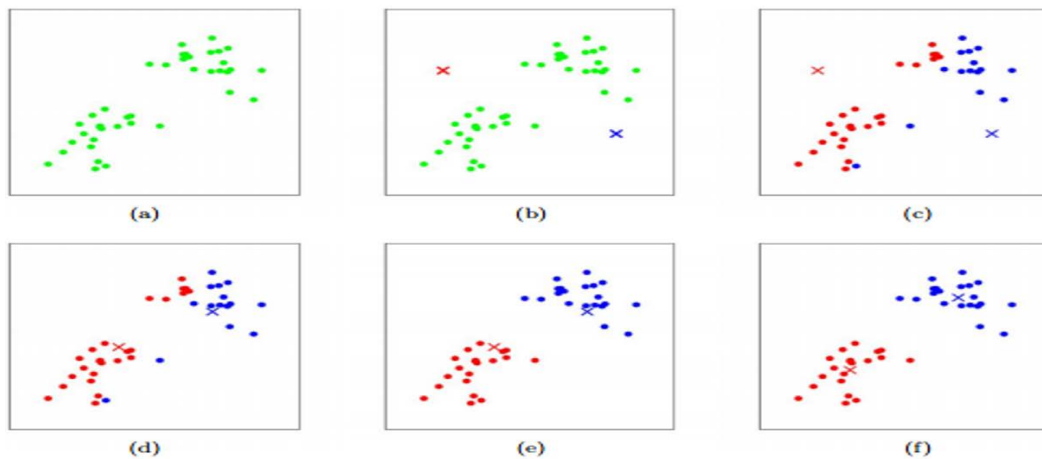


FIGURE 2.7 – Exemple d'application de l'algorithme K-means

Contrairement au premier algorithme de regroupement, le K-means nous donne la possibilité de maîtriser le nombre des clusters que nous souhaitons obtenir à la fin de la résolution. De plus, l'ordre d'affectation des nœuds n'a ici aucun impact sur les K clusters obtenus à la fin. Par contre, le regroupement final dépend du tirage initial des barycentres.

**2.4.3 Procédure de réparation**

Après avoir mis en œuvre une des deux méthodes de regroupement, nous utilisons une fonction de vérification afin de nous assurer que chaque paire fournisseur-client appartient au même cluster. Si nous constatons qu'il y a un cluster infaisable, nous utilisons une méthode de réparation simple qui consiste à déplacer chaque fournisseur de son cluster vers celui de son client. Enfin, nous commençons par résoudre un cluster choisi aléatoirement en utilisant tous les véhicules disponibles. Après cette résolution, les véhicules non-utilisés vont servir pour résoudre un autre cluster et ainsi de suite. L'algorithme 3 exprime cette procédure de réparation.

**Algorithme 3** Procédure de réparation**Input** : Clustered[Nodes];**Output** : Clustered[Nodes];**for** each  $i$  in  $C$  **do**    **if** Clustered[ $i$ ] est différent du Clustered[Supplier $_i$ ] **then**        Clustered[Supplier $_i$ ]=Clustered[ $i$ ]    **end****end**

La figure 2.8 illustre la procédure de résolution complète.

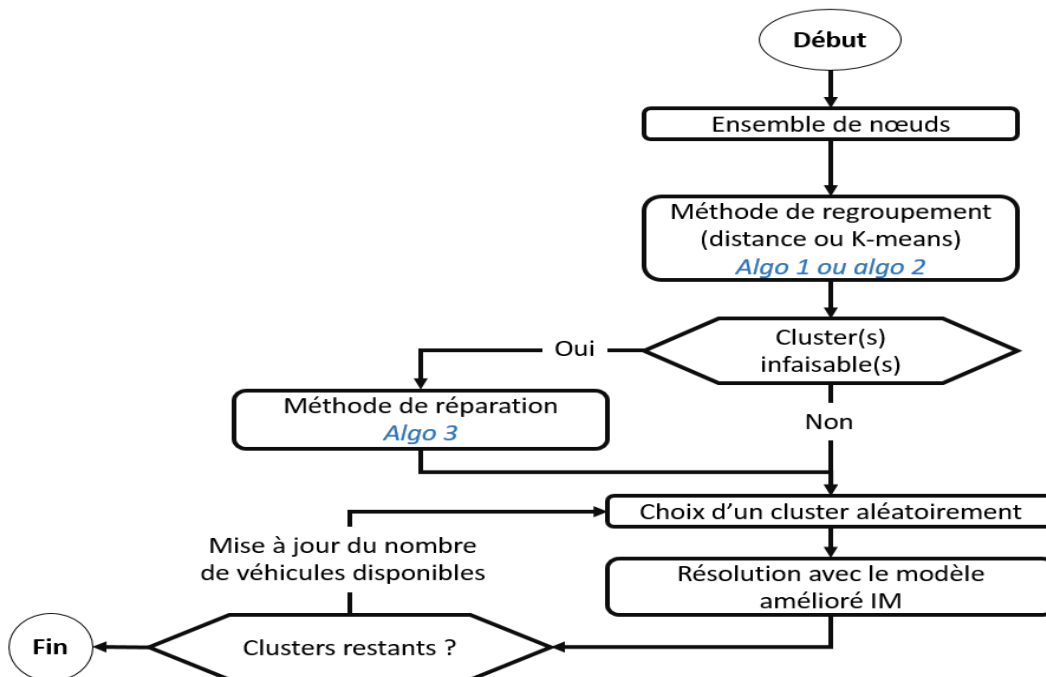


FIGURE 2.8 – Procédure de résolution complète

**2.4.4 Résultats expérimentaux**

Nous avons appliqué notre algorithme de résolution avec chacune des deux méthodes sur quatre instances de référence créées par [Li & Lim 2003] avec 106 sites. Nous avons choisi ces instances car elles sont difficiles à résoudre en utilisant des méthodes exactes. Le tableau 2.8 présente les solutions fournies par les deux méthodes de regroupement. Les solutions avec "\*" représentent des solutions faisables obtenues avec notre IM après deux heures de résolution. Le but de nos approches par regroupement est de décomposer une instance en plusieurs instances plus petites, d'où la nécessité d'obtenir au moins deux clusters. Les valeurs de "Cluster\_radius" et "K" pour chaque instance sont calculées suite à plusieurs tests effectués afin d'obtenir la meilleure solution possible. Les résultats obtenus avec l'algorithme K-



means sont meilleurs que ceux obtenus par le regroupement basé sur la distance euclidienne. De plus, pour l'instance LC101, nous avons retrouvé la solution optimale avec l'algorithme K-means dans un temps de résolution plus petit que celui de notre IM. Avec ce même algorithme, nous avons trouvé pour les instances LC108 et LRC103 des meilleures solutions par rapport à celles fournis après deux heures de résolution de notre modèle exact IM.

Instance	Regroupement basé sur la distance euclidienne				Regroupement basé sur l'algorithme K-means			
	Cluster_radius	Distance	CPU (s)	Nombre de sites visités	K	Distance	CPU (s)	Nombre de sites visités
LC101	39.58	829.49	138	102	8	<b>828.94</b>	<b>7.72</b>	<b>106</b>
LR101	13.95	1943.12	250	104	3	1730.99	244.12	106
LC108	49.85	976.41	384	100	8	<b>828.94</b>	<b>322</b>	<b>106</b>
LRC103	37.55	1272.09	562	106	9	<b>1258,74</b>	<b>479</b>	<b>106</b>

Instance	IM		
	Distance	CPU (s)	Nombre de sites visités
LC101	828.94	173	106
LR101	<b>1650.78</b>	<b>1220</b>	<b>106</b>
LC108	863.29*	7200	106
LRC103	1272.09*	7200	106

TABLE 2.8 – Méthodes de regroupement

Bien que ces méthodes peuvent décomposer les grandes instances et donc faciliter leurs résolutions, elles présentent certains inconvénients répartis en plusieurs niveaux, qui pourraient chacun constituer des perspectives d'amélioration potentielles :

- Le regroupement lui même est basé uniquement sur la distance (la localisation géographique des sites). Il ne prend pas donc en considération les autres critères du problème (la distribution des fenêtres temporelles, les quantités demandées, etc...).
- La fonction de réparation utilisée peut causer une déformation des clusters déjà créés par le regroupement.
- Arrivé à un cluster donné, si nous avons utilisé tous les véhicules disponibles, la visite de tous les sites appartenant à ce cluster ne peut pas se faire (par exemple, l'instance LC108 résolue avec l'algorithme 1). Donc nous ne pouvons pas garantir la visite du même nombre de sites que si nous traitons l'instance sans regroupement.

Néanmoins, notre algorithme heuristique basé sur K-means, dans sa version actuelle, est d'ores et déjà capable de concurrencer de manière satisfaisante notre méthode exacte, tant en termes de temps de résolution que d'objectif.

## 2.5 Piste d'amélioration

L'ajout des contraintes additionnelles expliquées précédemment dans notre modèle initial(SM) nous a permis de réduire substantiellement le temps nécessaire à la résolution exacte de notre variante. Les deux modèles (simple et amélioré) sont basés sur une formulation mathématique à 3 indices dans laquelle la variable possédant le plus grand nombre d'indices est  $X_{ij}^k$ . Dans cette partie, une nouvelle modélisation est proposée dans laquelle nous avons diminué le nombre d'indices de 3 à 2. Pour cela, nous considérons que la flotte est composée strictement de véhicules homogènes (même capacité et même vitesse moyenne). De plus, la procédure (expliquée dans la sous-section 2.2.2) visant à dupliquer les sites est également appliquée au dépôt dans cette nouvelle formulation.

Supposons que nous avons  $n$  demandes de transport. Les sites  $\{0\}$  et  $\{m+m'+2n+1\}$  correspondent respectivement aux dépôts initial et final ayant la même localisation géographique avec  $m$  et  $m'$  représentant le nombre de demandes ayant respectivement le dépôt initial comme fournisseur et le dépôt final comme client. L'ensemble  $B'=\{1,\dots,m,m+1,\dots,m+n\}$  représente le nouvel ensemble des fournisseurs qui englobe le sous-ensemble  $\{1,\dots,m\}$  associé aux dépôts initiaux artificiels et  $C'=\{m+n+1,\dots,m+2n,m+2n+1,\dots,m+m'+2n\}$  représente celui des clients. Le sous-ensemble  $\{m+2n+1,\dots,m+m'+2n\}$  est associé aux dépôts finaux artificiels. Nous supposons également que nous avons les ensembles suivants :  $Nodes = \{B' \cup C'\}$ ,  $N'=\{0 \cup B' \cup C'\}$  et  $N''=\{B' \cup C' \cup m+m'+2n+1\}$ . Sur cette base, la nouvelle formulation basée sur deux indices est la suivante :

$$\text{Maximize } \sum_{i \in N} \sum_{j \in N} P_i X_{i,j} \quad (2.19)$$

Subject to :

$$\sum_{i \in N'} X_{i,j} \leq 1 \quad \forall j \in Nodes \quad (2.20)$$

$$\sum_{j \in N''} X_{i,j} \leq 1 \quad \forall i \in Nodes \quad (2.21)$$

$$\sum_{i \in N} X_{i,u} - \sum_{j \in N} X_{u,j} = 0 \quad \forall u \in Nodes \quad (2.22)$$

$$\sum_{i=0}^m \sum_{j=m+1}^{m+m'+2n} X_{i,j} \leq V \quad (2.23)$$

$$\sum_{i=1}^{m+2n} \sum_{j=m+2n+1}^{m+m'+2n+1} X_{i,j} \leq V \quad (2.24)$$

$$\sum_{j \in Nodes} X_{j,i} = 0 \quad \forall i \in \{0, \dots, m\} \quad (2.25)$$

$$\sum_{j \in Nodes} X_{i,j} = 0 \quad \forall i \in \{m+2n+1, \dots, m+m'+2n+1\} \quad (2.26)$$

$$Y_j \geq Y_i + q_j - M(1 - X_{i,j}) \quad \forall i, j \in N \quad (2.27)$$

$$\max\{0, q_i\} \leq Y_i \leq \min\{Q, Q + q_i\} \quad \forall i \in N \quad (2.28)$$

$$A_{Customer_i} \geq A_i + S_i + \frac{d_{i,Customer_i}}{Speed} \quad \forall i \in F' \quad (2.29)$$

$$A_j \geq A_i + S_i + \frac{d_{i,j}}{Speed} - M(1 - X_{i,j}) \quad \forall i, j \in N \quad (2.30)$$

$$e_i \leq A_i \leq l_i \quad \forall i \in N \quad (2.31)$$

$$\sum_{u \in N} X_{i,u} - \sum_{u \in N} X_{Customer_i,u} = 0 \quad \forall i \in F' \quad (2.32)$$

$$First_i = First_{Customer_i} \quad \forall i \in F' \quad (2.33)$$

$$First_j \geq j.X_{0,j} \quad \forall j \in Nodes \quad (2.34)$$

$$First_j \leq j.X_{0,j} - M(X_{0,j} - 1) \quad \forall j \in Nodes \quad (2.35)$$

$$First_j \geq First_i + M(X_{i,j} - 1) \quad \forall i, j \in Nodes \quad (2.36)$$

$$First_j \leq First_i + M(1 - X_{i,j}) \quad \forall i, j \in Nodes \quad (2.37)$$

La nouvelle variable  $First_i$  donne le premier nœud servi après le nœud 0 dans la tournée visitant le nœud  $i$ . La fonction objectif (2.19) maximise le profit total. Les contraintes (2.20) et (2.21) assurent que chaque nœud peut être visité au plus une fois. La conservation du flux est garantie par les contraintes (2.22). Les contraintes (2.23) et (2.24) vérifient que nous ne dépassons pas le nombre limité de véhicules disponibles. Les contraintes (2.25) et (2.26) garantissent que chaque tournée doit commencer au dépôt initial et terminer au dépôt final. Les contraintes de capacité sont assurées par (2.27) et (2.28). Les contraintes (2.29), (2.30) et (2.31) vérifient le respect des fenêtres de temps. Les contraintes (2.32) assurent que si le fournisseur est visité, son client doit également être visité. Les contraintes (2.33) garantissent que chaque fournisseur et son client doivent se trouver dans la même tournée. Les contraintes (2.34) et (2.35) sont là pour confirmer que le premier nœud de la route qui visite  $j$  est  $j$  si  $j$  est le successeur direct du nœud 0. Les contraintes (2.36) et (2.37) sont utilisées pour assurer que les nœuds  $i$  et  $j$  partagent le même nœud initial si  $j$  est visité directement après  $i$ .

Ce modèle avec deux indices peut également être amélioré en utilisant les contraintes additionnelles initialisées dans la sous-section précédente. Ces dernières nécessitent quelques modifications pour s'adapter à la nouvelle formulation.

Nous avons testé cette nouvelle modélisation du problème sur un ensemble des instances précédemment créées. Le tableau 2.9 montre le nombre de variables (Var.) et de contraintes (Cont.) pour chaque instance dans quatre cas différents : formulation avec trois indices (SM et IM) et formulation avec deux indices (SM et IM). La première colonne donne le nom de l'instance et la deuxième donne le nombre de véhicules (NV). Comme nous pouvons le voir dans ce tableau, la nouvelle formulation

à deux indices a un nombre de variables et de contraintes plus petit que la formulation à trois indices (sauf pour le SPDPTWPD-C201 parce que nous avons une flotte composée d'un seul véhicule).

Problème	NV	Formulation avec 3 indices				Formulation avec 2 indices			
		SM		IM		SM		IM	
		Var.	Cont.	Var.	Cont.	Var.	Cont.	Var.	Cont.
SPDPTWPD-C201	1	484	1402	484	2277	866	3069	866	3305
SPDPTWPD-R201	6	2899	8107	2899	11187	866	3069	866	3374
SPDPTWPD-RC201	2	967	2743	967	4059	866	3069	866	3360
SPDPTWPD-C501	5	13516	39406	13516	55005	3536	13314	3536	14781
SPDPTWPD-R501	13	35140	102214	35140	138621	3536	13314	3536	15191
SPDPTWPD-RC501	4	10813	31555	10813	44553	3536	13314	3536	15133

TABLE 2.9 – Nombre des variables et des contraintes

Le tableau 2.10 donne le temps de résolution (en secondes) pour chaque instance en utilisant les deux formulations proposées. Il montre que les meilleurs temps sont obtenus avec la formulation à deux indices, sauf pour la première instance (SPDPTWPD-C201) où le nombre des variables et des contraintes en utilisant la nouvelle formulation sont plus grandes que l'ancienne. Nous déduisons de nos tests que la formulation proposée à deux indices devient plus efficace pour les problèmes avec plusieurs véhicules. De plus, nous remarquons que pour chaque formulation, l'IM obtient de meilleurs résultats que SM ce qui montre encore une fois l'efficacité de nos contraintes additionnelles.

Problème	Formulation avec 3 indices		Formulation avec 2 indices	
	SM	IM	SM	IM
SPDPTWPD-C201	0.16	0.11	1.2	1
SPDPTWPD-R201	0.73	0.15	0.14	0,02
SPDPTWPD-RC201	0.37	0.13	0.12	0.09
SPDPTWPD-C501	17.57	7.39	6.3	4.81
SPDPTWPD-R501	55.7	10.63	10.54	8.73
SPDPTWPD-RC501	1721.51	1125.42	59.7	44.87

TABLE 2.10 – Comparaison entre les deux formulations

En utilisant le modèle présenté dans cette partie, nous avons gagné au niveau du temps de résolution pour la plupart des instances parce que le nombre des variables et celui des contraintes sont plus petits que dans la modélisation à 3 indices. Mais le plus gros inconvénient de cette formulation réside dans l'obligation de considérer une flotte homogène de véhicules, ce qui n'est pas toujours le cas dans les applications réelles. Pour cette raison, nous conservons dans les chapitres suivants la formulation initiale à 3 indices.

## 2.6 Conclusion

Ce chapitre était consacré à la résolution d'une nouvelle variante du PDP qui est le Selective Pickup and Delivery Problem with Time Windows and Paired Demands (SPDPTWPD). Cette variante peut être trouvée dans diverses applications réelles lorsque les moyens de transport sont incapables de satisfaire toutes les demandes dans une période donnée (par exemple : une journée ou une demi-journée) à cause des nombreuses contraintes imposées. Nous avons introduit cette variante en donnant une description détaillée de ses éléments, ainsi que des contraintes et des éventuels objectifs associés.

Une formulation mathématique du SPDPTWPD a été présentée sous la forme d'un programme linéaire mixte en nombres entiers. Nous avons ensuite illustré la complexité de la modélisation des PDPs par rapport aux VRPs à partir d'un exemple simple.

Le modèle initial proposé dans un contexte mono-objectif a été amélioré par des contraintes additionnelles dans le but de réduire le temps de résolution. Ces deux modèles, résolus de manière exacte, ont fait l'objet de plusieurs tests sur des instances de référence et sur des nouvelles instances du SPDPTWPD. Ensuite, des méthodes approchées basées sur la notion de regroupement des nœuds ont été introduites afin de réduire encore le temps de résolution.

Les méthodes exactes présentées dans ce chapitre sont gourmandes en termes de temps de résolution. Mais les résultats obtenus ont eu au moins le mérite de nous fournir une base de référence pour valider nos méthodes approchées, soit sous forme de valeur optimale pour les petites instances, soit sous forme de borne. Les méthodes par regroupement présentent elles aussi certains inconvénients. D'autres pistes pourraient être explorées dans le futur, tant au niveau de méthodes exactes (Branch & Bound, génération de colonnes,...) que celles approchées (prise en compte de nouveaux critères de regroupement...). L'ensemble des méthodes précédemment présentées traitent la variante en se limitant à un seul objectif. Dans la suite, nous étendons notre étude à la résolution du SPDPTWPD dans un contexte multi-objectifs.



# Résolution du SPDPTWPD multi-objectifs

*Dans ce chapitre, nous étudions la variante introduite dans le chapitre précédent mais cette fois-ci dans un contexte où nous avons plusieurs objectifs. Pour résoudre le Selective Pickup and Delivery Problem with Time Windows and Paired Demands (SPDPTWPD), nous proposons une approche lexicographique ainsi que deux métaheuristiques. La première est basée sur une approche de type recuit simulé combinée avec une recherche Tabou et la deuxième est basée sur un algorithme génétique hybride. Les performances de ces trois méthodes sont comparées et discutées sur la base de diverses instances du problème.*

## Sommaire

<b>3.1</b>	<b>Approche lexicographique</b>	<b>65</b>
3.1.1	Approches non Pareto	65
3.1.2	Notre approche	65
3.1.3	Résultats expérimentaux	66
<b>3.2</b>	<b>Algorithme du recuit simulé</b>	<b>68</b>
3.2.1	Opérateur PD-Shift	70
3.2.2	Opérateur PD-Exchange	71
3.2.3	Opérateur PD-Rearrange	72
3.2.4	Fonction "comparaison"	72
3.2.5	Fonction "optimisation finale"	73
3.2.6	Résultats expérimentaux	74
<b>3.3</b>	<b>Algorithme génétique hybride</b>	<b>76</b>
3.3.1	Représentation de la solution	77
3.3.2	Création de la population initiale	77
3.3.3	Classement des chromosomes	79
3.3.4	Procédure de sélection	82
3.3.5	Croisement	84
3.3.6	Hybridation avec des méthodes de recherche locale	85
3.3.7	Test de dominance	87
3.3.8	Algorithme global	87
3.3.9	Résultats expérimentaux	88
<b>3.4</b>	<b>Conclusion</b>	<b>96</b>

## Publications

### Revue internationale :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "A lexicographic approach for the bi-objective selective pickup and delivery problem with time windows and paired demands", *Annals of Operations Research*, pp.1-19, doi :10.1007/s10479-017-2500-9.

### Conférences internationales :

1. Al Chami, Z., Manier, H., Manier, M.-A., & Fitouri, C.. "A hybrid genetic algorithm to solve a multi-objective Pickup and Delivery Problem", *20th World Congress of the International Federation of Automatic Control (IFAC WC'17)*, IFAC-PapersOnLine, Volume 50, Issue 1, pp. 14656-14661, Toulouse, France (9-14 July 2017).
2. Al Chami, Z., El Flity, H., Manier, H., & Manier M.-A.. "A New Metaheuristic to Solve a Selective Pickup and Delivery Problem", *4th International Conference on Logistics Operations Management (GOL'18)*, Le Havre, France (April 10-12, 2018).

### Conférence nationale :

1. Al Chami, Z., Manier, H., Manier, M.-A.. "Un algorithme génétique amélioré pour un problème de ramassage et de livraison", *Colloque FUTUR-MOB'17 : Préparer la transition vers la mobilité autonome*, Montbéliard, France (5-7 septembre 2017).



## 3.1 Approche lexicographique

### 3.1.1 Approches non Pareto

Différentes stratégies existent dans la littérature et permettent de prendre en compte la présence de plusieurs objectifs [Jozefowicz 2013]. Elles sont divisées en deux catégories :

1. les approches Pareto : elles consistent à utiliser la notion de dominance. Elles sont liées, dans la majorité des cas, à des métaheuristiques à base de population de solutions.
2. les approches non Pareto : elles ne se basent pas sur la notion de dominance et sont réparties en deux sous-catégories : les approches scalaires et les approches non-scalaires.

Le principe des méthodes scalaires est de revenir à un problème mono-objectif via un ensemble de paramètres [Haimès 1971]. Nous citons, par exemple, la somme pondérée qui combine les différents objectifs du problème en une seule fonction. Le décideur attribue donc à chaque objectif un coefficient qui représente son importance. La méthode  $\varepsilon$ -contrainte consiste à optimiser une seule fonction objectif tandis que les autres objectifs sont sujets à des contraintes. La difficulté des approches scalaires réside dans le fait que le décideur doit choisir le poids de chaque objectif ou la valeur de  $\varepsilon$  pour chaque fonction objectif.

Les méthodes non-scalaires ne transforment pas le problème multi-objectifs en un problème mono-objectif, mais utilisent des opérateurs qui traitent séparément les différents objectifs [Ehrgott 2005]. L'approche lexicographique représente une des méthodes non-scalaires dans lesquelles aucune valeur (poids ou  $\varepsilon$ ) n'est à fixer par le décideur. Par contre, la difficulté de ces méthodes est qu'un ordre des objectifs doit être établi. Dans la suite de cette section, nous avons choisi d'utiliser l'approche lexicographique parce que l'ordre d'importance des deux objectifs considérés est facile à établir.

### 3.1.2 Notre approche

Comme nous l'avons précédemment mentionné, l'approche lexicographique consiste à établir au préalable un ordre d'importance entre les objectifs. Ensuite, l'optimum est obtenu en considérant tout d'abord la fonction objectif la plus importante puis la deuxième et ainsi de suite.

Dans notre approche, le profit collecté est d'abord maximisé puis la distance est minimisée. Nous n'avons pas commencé avec la minimisation de distance comme première fonction objectif car dans un problème sélectif, la solution qui minimise la distance est celle où aucun site n'est visité, ce qui n'est pas acceptable.

La figure 3.1 montre les étapes de notre méthode qui consiste à résoudre deux fois séquentiellement notre modèle MILP. En effet, nous commençons par maximiser le profit en utilisant le MILP présenté dans la sous-section 2.2.2. Nous obtenons une valeur optimale *Profit\** à la fin de cette première résolution. Ensuite, nous

résolvons la distance en résolvant le même MILP auquel nous ajoutons une contrainte supplémentaire pour imposer une valeur du profit égale à  $Profit^*$ . Dans ce modèle, le fonction objectif considérée est (F2), soit la minimisation de la distance.

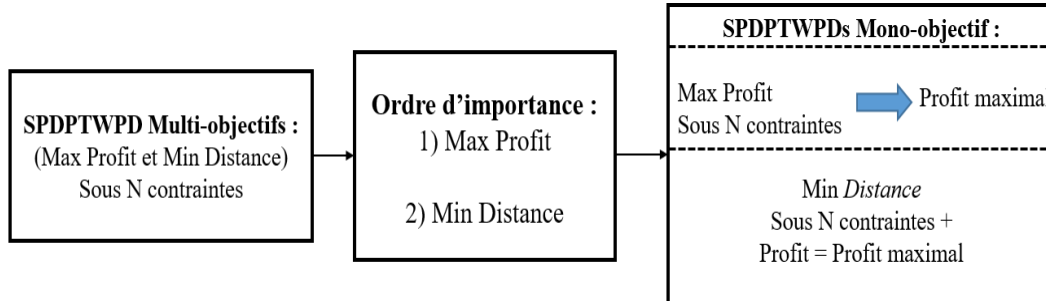


FIGURE 3.1 – Approche lexicographique pour le SPDPTWPD

### 3.1.3 Résultats expérimentaux

Nous avons appliqué cette méthode lexicographique aux nouvelles instances que nous avons générées pour le SPDPTWPD en section 2.3.4. Les résultats ont été obtenus en utilisant un solveur commercial Gurobi 6.0.5 et un ordinateur avec un processeur Intel Core i5, 2.80 GHz et 4 Go de RAM.

Le tableau 3.1 montre qu'avec la méthode lexicographique, nous pouvons obtenir deux solutions à l'issue des deux étapes de résolution. Les solutions obtenues à la fin de l'étape 2 dominent logiquement celles de l'étape 1 dans les cas où nous avons un gain de distance supérieur à 0%. De plus, ce gain peut atteindre jusqu'à 38% dans certains cas, ce qui représente une très bonne amélioration parce que la minimisation de la distance induit celle du coût total du transport. Lorsque le gain de distance est égal à 0%, cela signifie que la même solution a été obtenue à l'issue de chaque étape de résolution. Enfin le temps CPU représenté dans le tableau 3.1 est la somme du temps nécessaire pour les deux étapes de résolution.

Pour évaluer l'impact du nombre de véhicules sur les solutions obtenues, nous avons fait varier ce nombre pour quelques instances (SPDPTWPD-C202, SPDPTWPD-C502, etc...). Avec un seul véhicule et comme les profits associés aux sites sont différents pour les instances testées ici, nous pouvons en déduire qu'il y a une très forte probabilité que les solutions de même profit global visitent le même sous-ensemble de sites. Dans ce cas, les sources de gain sur la distance ( $d_2^*$  par rapport à  $d_1$ ) proviennent d'une modification de l'ordre de ces visites. Nous observons également que l'ajout d'un véhicule permet de visiter de nouveaux sites, ce qui implique à la fois une augmentation du profit et de la distance parcourue. De plus, cet ajout apporte une flexibilité supplémentaire liée à l'affectation des sites aux véhicules. Cela explique l'augmentation du temps de résolution (par exemple, pour l'instance SPDPTWPD-RC1002 traitée, le temps de résolution est 70 fois plus important qu'avec un véhicule en moins ce qui n'est pas négligeable) et doit permettre une meilleure répartition géographique par véhicule, pour un meilleur gain en termes de

Problème	Nombre de véhicules	Étape 1		Étape 2		CPU (s)	Distance gain (%)
		$P_1^*$	$d_1$	$d_2^*$			
SPDPTWPD-C201	1	180	364.93	275.19	0.30	25	
SPDPTWPD-C202	1	144	174.17	167.99	19.88	4	
SPDPTWPD-C202	2	217	250.87	212.94	52.15	15	
SPDPTWPD-R201	6	190	460.09	398.25	0.35	13	
SPDPTWPD-R202	4	213	406.73	378.95	12.20	7	
SPDPTWPD-RC201	2	160	233.2	233.2	0.33	0	
SPDPTWPD-RC202	2	180	235.49	235.09	0.52	0	
SPDPTWPD-C501	5	480	428.39	363.28	9.67	15	
SPDPTWPD-C502	1	437	440.42	431.25	28.31	2	
SPDPTWPD-C502	2	656	850.31	525.87	204.44	38	
SPDPTWPD-R501	13	490	1082.5	1075.77	14.86	1	
SPDPTWPD-R502	10	576	835.83	834.92	17.24	0	
SPDPTWPD-RC501	4	320	498.21	481.37	1629.02	3	
SPDPTWPD-RC502	4	420	527.76	527.76	151.37	0	
SPDPTWPD-RC502	5	500	691.94	686.69	1352.38	1	
SPDPTWPD-RC502	6	562	826.72	758.004	4505.16	8	
SPDPTWPD-C1001	6	690	507.55	437.04	928.93	14	
SPDPTWPD-C1002	7	770	649.79	542.15	621.54	17	
SPDPTWPD-R1001	7	500	758.47	639.64	3540.04	16	
SPDPTWPD-R1002	6	514	698.5	679.44	1391.96	3	
SPDPTWPD-RC1001	1	120	140.35	140.35	382.23	0	
SPDPTWPD-RC1002	1	138	128.44	125.68	103	2	
SPDPTWPD-RC1002	2	252	282.9	274.49	7000	3	

TABLE 3.1 – Résultats de l'approche lexicographique

distance.

Le principal avantage de cette méthode est qu'elle fournit une solution ayant un profit optimal (maximal) avec la moindre distance totale nécessaire pour obtenir ce profit. Mais son inconvénient réside dans l'obligation de lancer deux fois l'exécution de notre programme linéaire (autant de fois qu'il y a d'objectifs). Donc le temps nécessaire pour obtenir la solution est assez long.

Pour pallier le problème lié au temps de résolution évoqué précédemment, nous avons orienté nos recherches vers les méthodes approchées. Nous avons indiqué dans la section 1.5.2 que les métaheuristiques sont divisées en deux catégories : les métaheuristiques à base de solution unique et à base de population de solutions. Donc nous avons décidé d'implémenter dans la suite une méthode de chaque catégorie. Dans la partie suivante, nous présentons une méthode approchée de type recuit simulé, avec comme but de trouver la même solution que notre approche lexicographique.

### 3.2 Algorithme du recuit simulé

Nous avons tout d'abord développé un algorithme combinant un recuit simulé et une recherche tabou. Il est basé sur la métaheuristique de [Li & Lim 2003] permettant de résoudre le PDPTWPD, et que nous avons adaptée afin de résoudre notre variante. Dans l'algorithme de base, Li et Lim ont considéré un nombre illimité de véhicules. Les objectifs de leur méthode sont définis en respectant l'ordre de priorité suivant : (1) la minimisation du nombre de véhicules ; (2) la minimisation du coût total du voyage ; (3) la minimisation de la durée totale (somme du temps d'attente, du temps de déplacement et du temps de service) ; (4) la minimisation du temps d'attente total pour commencer le service.

Dans la phase d'initialisation, les premières tournées sont obtenues en utilisant l'heuristique d'insertion proposée par [Solomon 1987]. Cette heuristique développée pour le VRPTW est adaptée par [Li & Lim 2003] (Algorithme 4). Dans la suite, les auteurs ont utilisé un algorithme de recuit simulé qui diffère de son implémentation traditionnelle. Au lieu d'effectuer la procédure de recuit simulé sur la solution probabiliste jusqu'à ce que la procédure se termine, ils obligent la procédure de recuit simulé à redémarrer à partir de la meilleure solution actuelle après plusieurs itérations sans amélioration. Le nombre d'itérations possibles sans amélioration est fixé par le paramètre MSNI. L'algorithme 5 décrit la procédure développée par les auteurs. L'algorithme 6 rappelle la procédure de recuit simulé classique.

---

#### Algorithme 4 Algorithme d'insertion adapté

---

**Input** : Ensemble de nœuds ;

**Output** : Solution initiale ;

Id=1 ;

**while** *tous les nœuds ne sont pas affectés* **do**

Initialise la route R(Id) avec la paire de sites  $(B_i, C_i)$  ayant la plus grande distance combinée  $(DC(B_i, C_i)) / (DC(B_i, C_i)) = \text{distance}(0, B_i) + \text{distance}(B_i, C_i) + \text{distance}(C_i, 0)$  ;

**Fonction** *Calculer\_meilleure\_insertion*(Route(Id)) :

**for** *chacune de paires restantes* **do**

| Calculer la meilleure position d'insertion réalisables dans Route(Id) ;

**end**

**if** *aucune insertion possible* **then**

| Ajouter Route(Id) à la solution initiale ;

| Incrémenter Id et revenir vers le début de boucle "While" ;

**else**

| Insérer dans Route(Id) la paire qui provoque l'augmentation minimale en termes de distance ;

| Appeler la fonction *Calculer\_meilleure\_insertion*(Route(Id)) ;

**end**

**end**

---

---

**Algorithme 5** Algorithme hybride recuit simulé et recherche tabou [Li & Lim 2003]

**Input** : Solution S ;  
**Output** : Solution S\_Finale\_temp ;  
S\_Finale\_temp  $\leftarrow$  Applique les trois opérateurs successivement sur S (PD-Shift, puis PD-Exchange, puis PD-Rearrange) ;  
S = S\_Finale\_temp ;  
Itérations\_sans\_amélioration = 0 ;  
**if** *Itérations\_sans\_amélioration*  $\leq$  *MSNI* **then**  
    S\_temp  $\leftarrow$  Choisir une solution voisine aléatoire de S non-enregistrée dans la liste tabou en appliquant la procédure de recuit simulé classique ;  
    Enregistrer S\_temp dans la liste tabou ;  
    S\_temp  $\leftarrow$  Applique les trois opérateurs sur S\_temp ;  
    **if** *S\_temp est meilleure que S\_Finale\_temp* **then**  
        S\_Finale\_temp = S\_temp ; Itérations\_sans\_amélioration = 0 ;  
    **else**  
        Itérations\_sans\_amélioration = Itérations\_sans\_amélioration + 1 ;  
    **end**  
    S = S\_temp ;  
**end**

---

**Algorithme 6** Algorithme de recuit simulé classique

**Input** : T : température de recuit global,  $\sigma$  : rapport de refroidissement de T, Solutions finale\_solution et nouvelle\_solution ;  
**Output** : Solution finale\_solution ;  
**Répéter jusqu'à atteindre le critère d'arrêt** :  
Générer nouvelle\_solution ;  
**if** *nouvelle\_solution est meilleure que finale\_solution* **then**  
    finale\_solution = nouvelle\_solution ;  
**else**  
    **if** *Probabilité(finale\_solution, nouvelle\_solution, T)  $\geq$  nombre aléatoire entre 0 et 1*  
        **then**  
            finale\_solution = nouvelle\_solution ;  
            Mettre à jour la température de recuit globale :  $T = \sigma * T$  ;  
        **end**  
**end**  
**end**

---

L'algorithme global est schématisé par la figure 3.2. Il se termine après K itérations sans amélioration. Les solutions visitées sont enregistrées dans une liste tabou pour éviter les cycles.

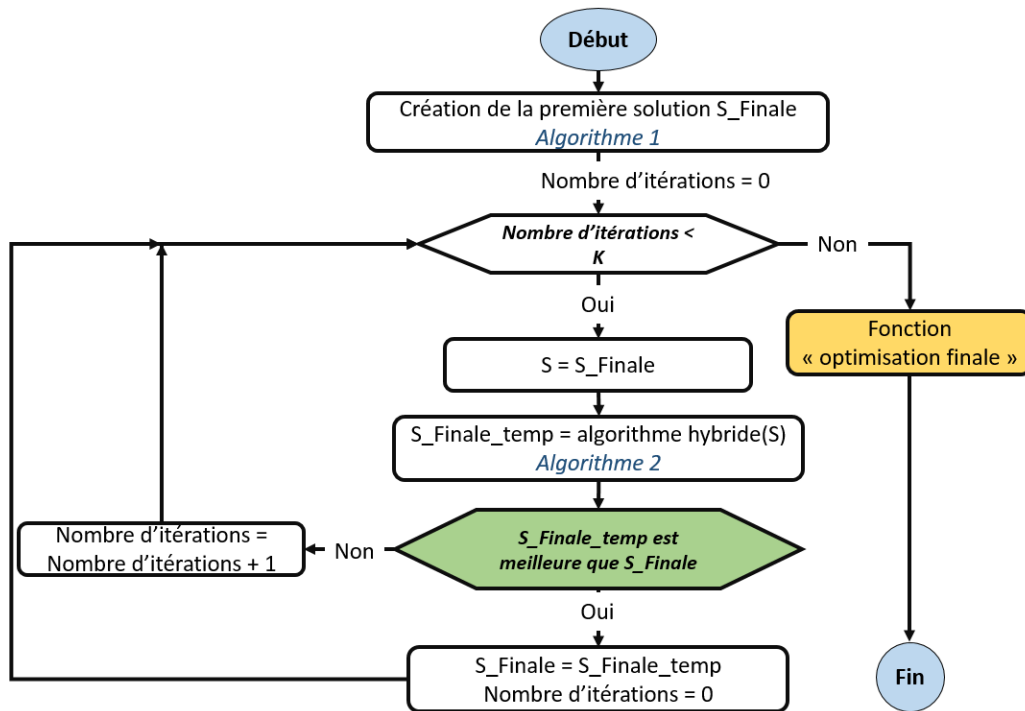


FIGURE 3.2 – Algorithme global utilisé par [Li &amp; Lim 2003]

Comme nous l'avons mentionné, la métaheuristique développée repose principalement sur trois opérateurs de permutation décrits ci-après et utilisés pour plusieurs itérations permettant de restructurer les tournées dans le but d'optimiser les objectifs précédemment évoqués et d'obtenir la meilleure solution possible.

### 3.2.1 Opérateur PD-Shift

Cet opérateur déplace une paire fournisseur-client d'une tournée vers une autre. Dans la figure 3.3, l'opérateur PD-Shift est utilisé pour déplacer la paire  $B_i$ - $C_i$  de la tournée  $T_i$  à la tournée  $T_j$ . Tout d'abord l'opérateur PD-Shift supprime  $B_i$  et  $C_i$  de la route d'origine, puis les insère à une position réalisable dans la route  $T_j$ , sous réserve des contraintes imposées par le SPDPTWPD. Nous testons toutes les possibilités d'insertion jusqu'à en trouver une faisable. Les déplacements impossibles sont interdits. Si aucune insertion faisable n'existe, on passe à l'opérateur suivant.

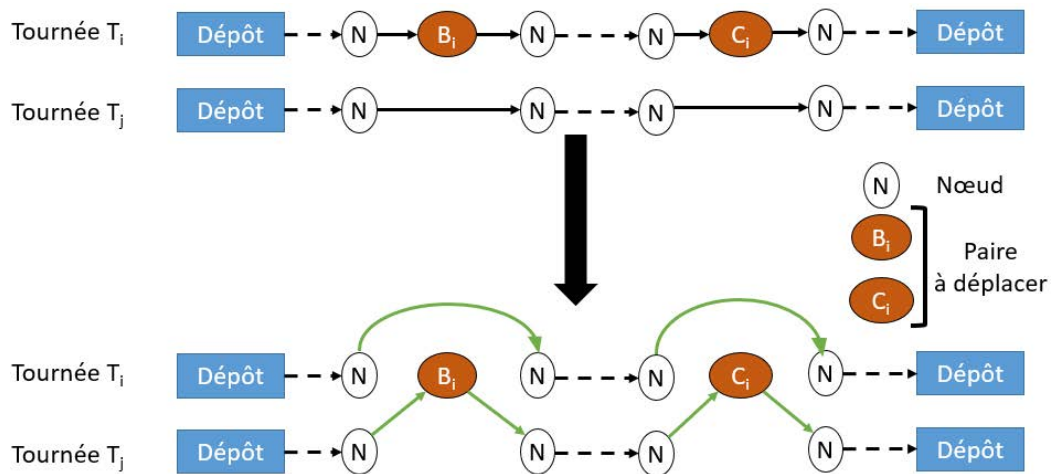


FIGURE 3.3 – Opérateur PD-Shift

### 3.2.2 Opérateur PD-Exchange

L'opérateur PD-Exchange échange des paires entre deux tournées. En d'autres termes, il déplace une paire de chaque tournée dans l'autre. Le nombre d'échanges testés est fixé a priori. Par exemple, dans la figure 3.4, la paire  $B_i-C_i$  est à l'origine dans  $T_i$ , et  $B_j-C_j$  appartient à la tournée  $T_j$ . L'opérateur PD-Exchange supprime d'abord  $B_i-C_i$  et  $B_j-C_j$ , puis les insère respectivement à une position possible dans les routes  $T_j$  et  $T_i$ . Concernant la méthode d'insertion, nous appliquons la même procédure que celle expliquée pour le PD-Shift. Les échanges qui ne respectent pas les contraintes du problème ne sont pas autorisés.

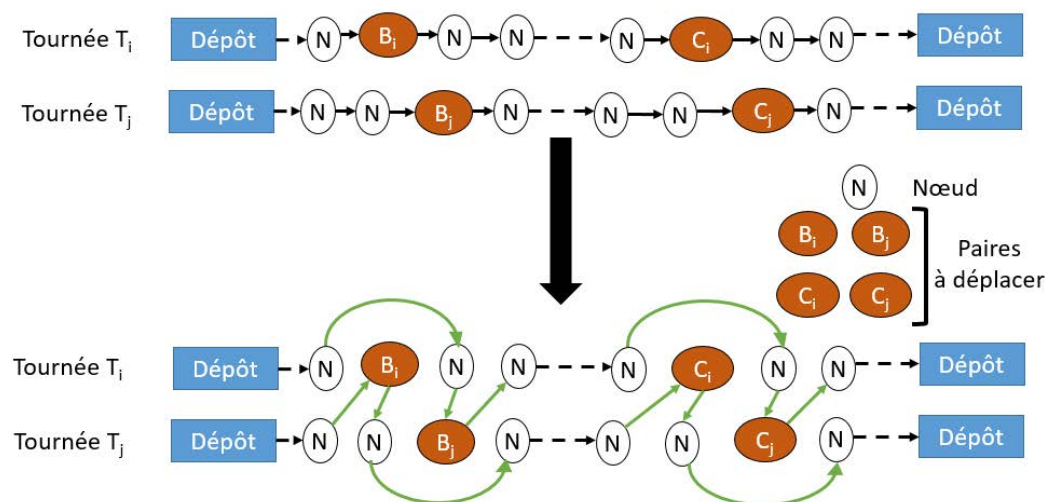


FIGURE 3.4 – Opérateur PD-Exchange

### 3.2.3 Opérateur PD-Rearrange

L'opérateur PD-Rearrange repositionne les paires dans une même tournée. Son objectif est de déplacer une paire appartenant à une tournée donnée vers la meilleure position qui optimise les objectifs du problème. Dans la figure 3.5, l'opérateur PD-Rearrange supprime la paire  $B_i-C_i$  qui est dans la route  $T_i$ , puis l'insère à une autre position possible dans  $T_i$ . Les insertions impossibles sont également interdites.

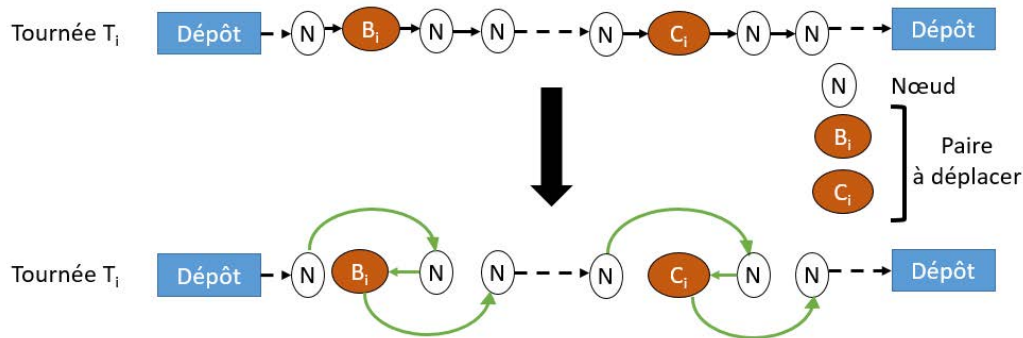


FIGURE 3.5 – Opérateur PD-Rearrange

### 3.2.4 Fonction "comparaison"

Comme nous l'avons déjà mentionné, nous avons développé une extension de cette métaheuristique afin de l'adapter à notre variante. Le tableau 3.2 résume les différences entre les deux variantes. Nous avons appliqué la stratégie introduite en section 2.5 et qui consiste à dupliquer le dépôt afin de traiter les trois types de flux.

Critère	[Li & Lim 2003]	Al Chami
Aspect sélectif	Non	Oui
Nombre de véhicules	Illimité	limité
Types de flux	1	3
Liste d'objectifs	(1) la minimisation du nombre de véhicules (2) la minimisation du coût total du voyage (3) la minimisation de la durée totale (4) la minimisation du temps d'attente total	(1) La maximisation du profit total collecté (2) La minimisation de la distance totale parcourue

TABLE 3.2 – Différences entre les deux variantes

Contrairement à notre cas où nous utilisons un nombre limité de véhicules, le nombre de véhicules utilisé par [Li & Lim 2003] est illimité. Nous avons donc besoin d'une nouvelle fonction pour transformer une solution infaisable (avec un nombre



de véhicules dépassant celui disponible) en une solution faisable. Dans le but de respecter les fonctions objectifs, le principe de la fonction "comparaison" est de retirer de la solution les tournées ayant le profit minimal parmi l'ensemble des routes, jusqu'à avoir une solution avec le nombre de tournées égal au nombre de véhicules disponibles.

Dans l'exemple de la figure 3.6, nous avons deux solutions S1 et S2 avec  $m$  et  $n$  tournées respectivement. L'objectif est de comparer ces deux solutions entre elles pour garder la meilleure solution. Étant donné que la flotte est composée de  $V$  véhicules, la fonction "comparaison" nous donne en sortie deux solutions S' et S'' avec  $V$  tournées si  $m$  et  $n$  sont plus grands que  $V$ . Si une de deux solutions a un nombre de tournées plus petit que  $V$ , nous gardons la même solution en sortie.

Pour les solutions obtenues, la meilleure est celle avec le profit le plus élevé. Si les deux solutions ont le même profit, la meilleure est celle qui a la plus faible distance. Si les deux solutions ont un nombre de routes différent après application de la fonction "comparaison" autrement dit si  $m < V$  et/ou  $n < V$ , nous avons décidé de considérer que la meilleure est celle qui a besoin du plus petit nombre de véhicules, en ignorant le profit et la distance. En effet, des tests nous ont montré que, la plupart du temps, cette stratégie conduit à de meilleures solutions à la fin. À l'issue de cette procédure, parmi les deux solutions comparées, nous conservons la solution S1 (respectivement S2) si S' (respectivement S'') est meilleure que S'' (respectivement S'). S' et S'' ne sont que des solutions virtuelles intermédiaires nous permettant de faire notre choix entre S1 et S2.

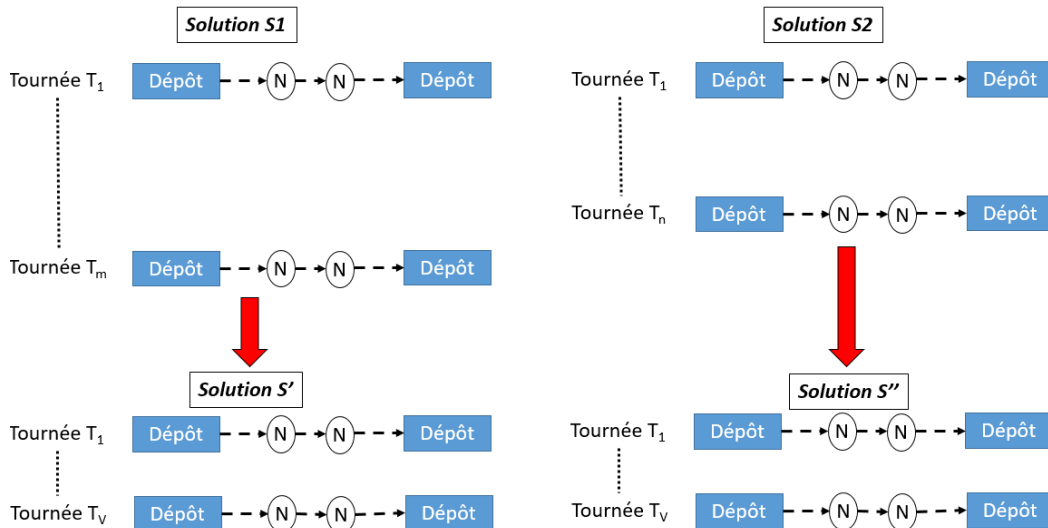


FIGURE 3.6 – Fonction "comparaison"

### 3.2.5 Fonction "optimisation finale"

À la fin de la résolution, c'est à dire lorsque le nombre d'itérations est atteint, la solution fournie par la méthode hybride peut contenir un nombre de tournées plus grand que le nombre de véhicules disponibles. Dans ce cas, nous avons créé

une nouvelle fonction "optimisation finale" qui supprime dans un premier temps les tournées ayant les plus faibles profits, comme nous l'avons fait dans la phase de virtualisation de la fonction "comparaison". Dans un deuxième temps, la fonction "optimisation finale" met en oeuvre une heuristique d'insertion de nœuds dans les routes restantes, ces nœuds étant choisis au sein des tournées supprimées. Ici, nous cherchons à nouveau la première insertion faisable. Pour chaque demande composée d'une paire des nœuds, si l'insertion ne respecte pas les contraintes du problème, nous essayons de l'échanger avec une autre paire aléatoirement choisie dans les routes restantes, et ayant un profit inférieur. Le principe de cette fonction est schématisé dans la figure 3.7.

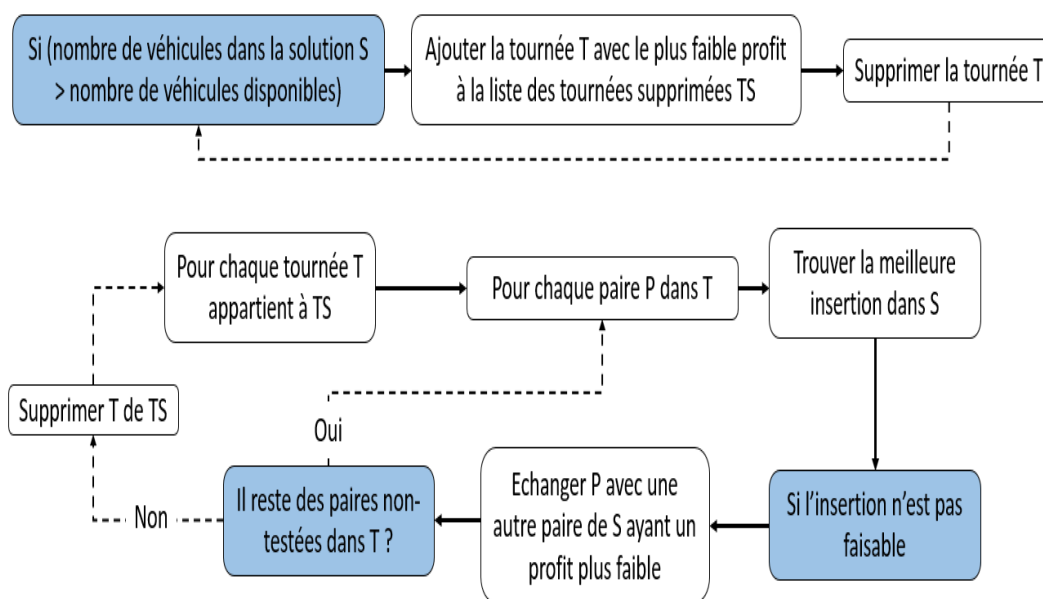


FIGURE 3.7 – Fonction "optimisation finale"

### 3.2.6 Résultats expérimentaux

Le tableau 3.3 donne un aperçu des performances de notre métaheuristique pour les instances de référence du SPDPTWPD. Comme nous l'avons mentionné précédemment, chaque instance est appelée SPDPTWPDxyz, x est le type d'instance (R : les nœuds sont distribués aléatoirement, C : les nœuds sont totalement groupés et RC : les nœuds sont partiellement groupés et partiellement répartis aléatoirement), y est le nombre de nœuds (20, 50 et 100) et i identifie la distribution du profit (1 : les nœuds ont le même profit et 2 : chaque site a sa propre valeur de profit).

Nous avons comparé les performances de cette méthode avec notre approche lexicographique, en termes d'objectifs et de temps CPU. Le tableau 3.3 montre les résultats pour chaque méthode, ainsi que les écarts obtenus pour chaque critère. Nous en faisons l'analyse suivante :

- Dans 60.87% des cas (14 sur 23 instances), la nouvelle métaheuristique fournit

les mêmes résultats (même profit optimal et même distance) que l'approche lexicographique, par exemple pour l'instance SPDPTWPD-C20.

- Pour 17.39% des instances testées, la métaheuristique donne un profit moins intéressant que l'approche lexicographique, mais avec une distance plus faible. Dans le cas des instances du groupe 1 (même profit pour tous les sites), telles que SPDPTWPD-RC1001, un profit plus faible s'interprète par un nombre de sites visités moindre, et il paraît logique d'obtenir alors une distance plus faible. Mais cela ne garantit en rien que cette distance soit optimale pour ce profit. Ce raisonnement est moins évident à tenir pour les instances du groupe 2 où les profits sont différents selon les sites. En effet, dans ce cas, un profit global plus faible ne signifie pas obligatoirement moins de sites visités.
- Dans 21.74% des instances testées, notre métaheuristique donne des solutions moins intéressantes que l'approche lexicographique, à la fois en termes de profit et de distance, comme pour l'instance SPDPTWPD-RC502 à 6 véhicules.

Instance	Nombre de véhicules	Approche lexicographique (AL)			Recuit simulé (RS)			Écart(RS AL) en %	
		Profit	Distance	CPU (s)	Profit	Distance	CPU (s)	Profit	Distance
SPDPTWPD-C201	1	180	275.19	0.3	180	275.19	11	0	0
SPDPTWPD-C202	1	144	167.99	19.88	144	167.99	7	0	0
SPDPTWPD-C202	2	217	212.94	52.15	217	212.94	8	0	0
SPDPTWPD-R201	6	190	398.25	0.35	190	398.25	6	0	0
SPDPTWPD-R202	4	213	378.95	12.2	213	378.95	6	0	0
SPDPTWPD-RC201	2	160	233.2	0.33	160	233.2	6	0	0
SPDPTWPD-RC202	2	180	235.09	0.52	180	235.09	6	0	0
SPDPTWPD-C501	5	480	363.28	9.67	480	363.28	57	0	0
SPDPTWPD-C502	1	437	431.25	28.31	436	386.08	433	-0.23	-10.47
SPDPTWPD-C502	2	656	525.87	204.44	656	525.87	401	0	0
SPDPTWPD-R501	13	490	1075.77	14.86	490	1075.77	21	0	0
SPDPTWPD-R502	10	576	834.92	17.24	576	834.92	20	0	0
SPDPTWPD-RC501	4	320	481.37	1629.02	310	489.4	28	-3.13	1.67
SPDPTWPD-RC502	4	420	527.76	151.37	400	518.14	26	-4.76	-1.82
SPDPTWPD-RC502	5	500	686.69	1352.38	480	677.07	27	-4	-1.4
SPDPTWPD-RC502	6	562	758	4505.16	547	801.64	28	-2.67	5.76
SPDPTWPD-C1001	6	690	437.04	928.93	680	469.58	290	-1.45	7.45
SPDPTWPD-C1002	7	770	542.15	621.54	760	575.08	233	-1.3	6.07
SPDPTWPD-R1001	7	500	639.64	3540.04	500	639.64	138	0	0
SPDPTWPD-R1002	6	514	679.44	1391.96	512	710.02	157	-0.39	4.5
SPDPTWPD-RC1001	1	120	140.35	382.23	110	127.22	151	-8.33	-9.36
SPDPTWPD-RC1002	1	138	125.68	103	138	125.68	109	0	0
SPDPTWPD-RC1002	2	252	274.49	7000	252	274.49	145	0	0

TABLE 3.3 – Résultats de l'algorithme du recuit simulé

L'écart entre les deux méthodes est calculé en utilisant l'équation suivante :

$$Ecart(X|Y) = \frac{100 * (Valeur_X - Valeur_Y)}{Valeur_Y} \quad (3.1)$$

Dans le tableau 3.4, nous analysons les résultats obtenus par notre métaheuristique (RS) par rapport à l'approche lexicographique (AL). Les valeurs avec "\*" sont calculées en se basant sur les valeurs absolues de l'écart. Nous remarquons que pour les petites instances avec 20 sites, la métaheuristique fournit les mêmes solutions que l'approche lexicographique avec un temps CPU moyenne un peu plus faible. Et puisque l'écart-type est petit, nous considérons que notre RS est plus stable. Pour les autres instances et bien que nous ayons une dégradation en termes de profit d'une moyenne de 1.64%, la métaheuristique fournit des solutions avec parfois des meilleures distances que celles obtenues avec l'AL. Contrairement aux petites instances, la différence en termes de temps de résolution devient plus grande en augmentant la taille du problème. Et la performance de notre RS dépasse largement celle de l'approche lexicographique. Prenant toutes les instances testées, la dégradation est de 1.14% en moyenne pour le profit et la différence absolue est de 2.11% en moyenne, cela implique que les bonnes qualités de résultats obtenus.

Instances	Temps CPU				Ecart(RS AL)			
	Moyenne		Ecart-type		Moyenne		Ecart-type	
	AL	RS	AL	RS	Profit	Distance*	Profit	Distance*
<b>20 sites</b>	12.25	<b>7.14</b>	19.21	<b>1.86</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>50 sites</b>	879.16	<b>115.67</b>	1495.36	<b>171.37</b>	-1.64	2.35	1.98	3.56
<b>100 sites</b>	1995.39	<b>174.71</b>	2483.43	<b>63.39</b>	-1.64	3.91	3.01	3.94
<b>Toutes</b>	955.04	<b>100.61</b>	1767.71	<b>128.05</b>	-1.14	2.11	2.12	3.41

TABLE 3.4 – Analyse des résultats obtenus

Les résultats obtenus montrent l'efficacité de la métaheuristique hybride proposée. Elle permet d'obtenir des résultats identiques ou proches de ceux obtenus par la méthode exacte, avec des temps de calcul raisonnables.

L'inconvénient de cette méthode approchée réside dans le fait que nous obtenons au final une solution unique. Tandis que dans les problèmes multi-objectifs, le décideur a besoin parfois d'un ensemble de solutions qui reflète les meilleurs compromis entre les objectifs traités pour faciliter son choix. Pour répondre à un tel besoin, nous proposons dans la suite un nouvel algorithme, cette fois basé sur une métaheuristique manipulant une population de solutions.

### 3.3 Algorithme génétique hybride

Cette méthode est basée sur un algorithme génétique hybridé avec des algorithmes de recherche locale permettant de minimiser la distance totale et de maximiser le profit collecté. La définition ainsi que les principes de base d'un algorithme génétique sont déjà introduits dans le premier chapitre (section 1.5.2.2). L'algorithme 7 rappelle l'implémentation classique d'un algorithme génétique.

---

**Algorithme 7** Algorithme génétique classique

---

Générer une population initiale de solutions ;

**while** *critère d'arrêt n'est pas satisfait* **do**

    Évaluer la population à l'aide des opérateurs d'évaluation ;

    Choisir deux parents avec une méthode de sélection ;

    Appliquer les opérateurs de croisement et de mutation afin de générer de nouveaux enfants ;

    Mettre à jour la population ;

**end**

---

### 3.3.1 Représentation de la solution

Un des facteurs les plus importants dans les algorithmes génétiques est la façon avec laquelle les solutions sont codées. Dans la littérature, deux principaux types de codage existent :

- Le codage binaire : il consiste à coder la solution selon une chaîne de bits (qui peuvent prendre les valeurs 0 ou 1). Ce type de codage a été utilisé dans les premiers algorithmes génétiques de la littérature (par exemple, [Holland 1973]). Mais il entraîne certains problèmes. En effet, il est souvent peu naturel et plus compliqué d'utiliser ce codage par rapport à un problème donné. De plus, la marge donnée par ce type de représentation est trop limitée, si nous codons avec 5 bits, nous pouvons représenter  $2^5 = 32$  valeurs différentes seulement. Un autre inconvénient d'utiliser ce codage indirect réside dans la nécessité d'un effort de calcul supplémentaire pour décoder les solutions.
- Le codage de permutation : la solution est codée avec une séquence de valeurs. Ce type de codage est plus naturel que celui introduit ci-avant puisque la structure du problème est conservée dans le codage. Notons que [Wright 1991] est la première étude dans laquelle une stratégie pour utiliser ce codage a été proposée. Le codage par permutation ou codage direct est très spécifique aux problèmes dans lesquels on cherche un ordre optimal de la séquence (par exemple, de tâches à effectuer, des trajets entre villes, etc...). Compte tenu des inconvénients du codage indirect, nous avons décidé d'utiliser le codage de permutation dans ce travail.

Chaque solution est représentée sous la forme d'un chromosome (une séquence de gènes). Chaque gène représente une tournée de véhicule qui commence à partir du dépôt, dessert un ensemble de nœuds et revient, à la fin, au dépôt. Un exemple de ce type de représentation est donné dans la figure 3.8 pour une flotte de 3 véhicules et un dépôt noté 0.

### 3.3.2 Création de la population initiale

La population initiale est créée par des permutations entre les nœuds (Algorithme 8). Ces permutations sont aléatoires, mais respectent la contrainte d'appariage

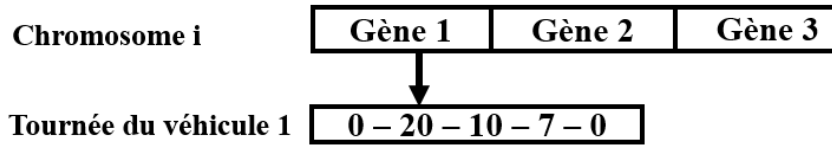


FIGURE 3.8 – Représentation d’une solution

fournisseur-client ainsi que les autres contraintes du problème. Avec cette procédure, notre population initiale est composée uniquement de solutions réalisables.

---

**Algorithme 8** Création de la population initiale
 

---

**Input :** Problème P, Taille maximale de la population initiale Max;

**Output :** Ensemble des solutions initiales  $S_{initial}$ ;

```

while Nombre_de_solutions_générées ≤ Max do
  Véhicule_actuel ← 0;
  while Véhicule_actuel ≤ P.Nombre_des_véhicules do
    List ← Tri aléatoire des nœuds;
    i ← 0;
    while i ≤ P.Nombre_des_nœuds do
      N ← List[i];
      if N est non-affecté et [N ∈ B ou [N ∈ C et SupplierN = 0]] then
        Ajouter N à la tournée R;
        if CustomerN ≠ 0 then
          | Ajouter CustomerN à la tournée R;
        end
        if Tournée R est infaisable then
          | Supprimer N et/ou CustomerN de la tournée R
        else
          | N et/ou CustomerN sont affectés
        end
      end
      i ← i + 1;
    end
    R ← Recherche_locale(R);
    Ajouter R au Chromosome C; Véhicule_actuel ← Véhicule_actuel + 1;
  end
  Insérer Chromosome C dans  $S_{initial}$ ;
  Nombre_des_solutions_générées ← Nombre_des_solutions_générées + 1;
end

```

---

Comme mentionné dans Algorithme 8, chaque gène généré (associé à une tournée) est amélioré par une méthode de recherche locale puis inséré dans le chromosome correspondant. Cette procédure de recherche locale est explicitée par Algorithme 9.

**Algorithme 9** Procédure de recherche locale

---

**Input** : Tournée R1 ; **Output** : Tournée R\_finale ;  
Tournée R2 ;  
R\_finale  $\leftarrow$  R1 ;  
**for** *CP permutations possibles de R1* **do**  
| R2  $\leftarrow$  Permutation\_suivante de R1 ;  
| **if** *R2 est faisable et Distance(R2) < Distance(R1)* **then**  
| | R\_finale  $\leftarrow$  R2 ;  
| **end**  
**end**  
retourner R\_finale ;

---

La procédure de recherche locale de l'Algorithme 9 teste CP permutations possibles des nœuds sur la tournée considérée. Dans chaque permutation, deux nœuds choisis aléatoirement sont échangés. Ensuite, nous évaluons la nouvelle tournée. Si elle est encore faisable (respect de toutes les contraintes) et si nous obtenons une distance plus petite, nous remplaçons la tournée précédente par la nouvelle. Sinon, nous gardons la route précédente.

**3.3.3 Classement des chromosomes**

Afin de classer les chromosomes dans la population, nous allons appliquer une procédure d'évaluation en deux étapes : l'attribution des rangs et le calcul des fitness.

**3.3.3.1 Attribution des rangs**

Pour la population courante, le rang de chaque chromosome créé est déterminé à l'aide de la méthode NSGA (Non-dominated Sorting Genetic Algorithm) introduite pour la première fois par [Srinivas & Deb 1994]. Elle se base sur une classification de la population en fonction du degré de dominance au sens de Pareto de chaque individu.

Supposons que nous traitons un problème avec un ensemble de  $F$  objectifs de minimisation, une solution  $x$  domine une solution  $y$  si et seulement si :

- $f_i(x) \leq f_i(y) \quad \forall i \in F$
- $f_i(x) < f_i(y) \quad \exists i \in F$

Cela signifie que  $x$  est meilleure que  $y$  pour au moins un critère, et au moins aussi bonne que  $y$  pour tous les autres critères. Dans notre cas, une solution  $x$  domine une solution  $y$  si et seulement si :

- Distance( $x$ )  $\leq$  Distance( $y$ ) et Profit( $x$ )  $\geq$  Profit( $y$ )
- Distance( $x$ ) < Distance( $y$ ) ou Profit( $x$ ) > Profit( $y$ )

De plus, pour une population  $P$  de solutions, une solution  $x$  est dite Pareto optimale ou non-dominée si et seulement si :

- $\nexists y \in P / y$  domine  $x$

Dans cette approche, pour chaque solution  $i$ , un rang  $Rank_i$  est attribué. La méthode NSGA est basée sur plusieurs couches de classification. Premièrement, les solutions non-dominées se voient attribuer un rang égal à 1. Ensuite, ce groupe de solutions non-dominées est retiré de la population et une autre couche de chromosomes non-dominés est considérée dans l'ensemble des solutions restantes. En d'autres termes, le reste de la population est re-classifié et dans cette étape, les nouveaux chromosomes non-dominés sont associés avec un rang égal à 2. Le processus continue jusqu'à ce qu'un rang soit affecté à toutes les solutions de la population. Les étapes utilisées dans notre algorithme de classement sont résumées comme suit :

1. Les solutions non-dominées sont affectées avec un rang égal à  $Rank = 1$ .
2. Les solutions non dominées sont temporairement retirées de la population.
3. Une autre couche de chromosomes non-dominés est considérée et affectée avec un  $Rank = Rank + 1$ .
4. Nous répétons les étapes (2) et (3) jusqu'à ce que tous les chromosomes aient un rang.

	Distance	Profit	Rank
<b>Chromosome 1</b>	180	400	1
<b>Chromosome 2</b>	170	320	1
<b>Chromosome 3</b>	160	250	1
<b>Chromosome 4</b>	170	280	2
<b>Chromosome 5</b>	160	220	2
<b>Chromosome 6</b>	160	180	3

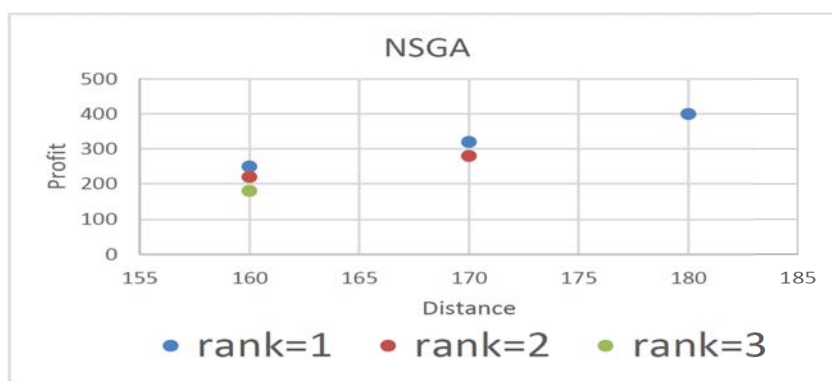


FIGURE 3.9 – Attribution des rangs

La figure 3.9 montre un exemple de classement utilisant l'approche NSGA pour une population composée de six chromosomes.



## 3.3.3.2 Calcul des fitness

Avant l'étape de sélection et afin de maintenir la diversité, nous utilisons une méthode de partage (Sharing method) qui attribue une valeur de fitness (force) aux individus déjà classés.

L'idée de base de la méthode de partage est de pénaliser les solutions dans les zones surpeuplées, réduisant ainsi la probabilité de leur survie à la génération suivante. Donc cette méthode augmente la probabilité de sélection des solutions dans les zones non peuplées ce qui peut augmenter la diversification de notre algorithme. Par la suite, cette méthode nous aide à répartir la population sur le front de Pareto (pour plus de détails, voir [Goldberg *et al.* 1987]). Par définition, le front de Pareto est composé de l'ensemble des solution non-dominées

Dans cette méthode, nous calculons tout d'abord un paramètre de densité  $\Phi(dist_{i,j})$  pour chaque solution  $i$  selon l'équation (3.2).

$$\Phi(dist_{ij}) = \begin{cases} 1 - \left(\frac{dist_{i,j}}{\sigma_{sh}}\right)^\alpha & : dist_{i,j} < \sigma_{sh} \\ 0 & : Otherwise \end{cases} \quad (3.2)$$

Où  $j \in Population_i$ , et  $Population_i$  est l'ensemble des solutions possédant un rang égal à  $Rank_i$ . Dans cette équation,  $dist_{i,j}$  représente la distance entre deux solutions  $i$  et  $j$  obtenue en faisant la somme des différences entre les deux objectifs (F1) et (F2). Elle est calculée avec l'équation (3.3) :

$$dist_{i,j} = |Distance(i) - Distance(j)| + |Profit(i) - Profit(j)| \quad (3.3)$$

Les paramètres  $\alpha$  et  $\sigma_{sh}$  sont utilisés pour contrôler la méthode de partage.  $\sigma_{sh}$  représente le seuil de différence (aussi la distance de coupure ou le rayon de niche). De plus,  $\alpha$  est utilisé pour régulariser la forme de la fonction de partage (typiquement  $\alpha = 1$ ) [Sareni & Krahenbuhl 1998]. Nous utilisons l'équation (3.4) afin de calculer une valeur de fitness  $Fitness_i$  pour chaque solution.

$$Fitness_i = \frac{Rank_i}{\sum_{j \in Population_i} \Phi(dist_{i,j})} \quad (3.4)$$

	Distance	Profit	Rank	$\Sigma_{\text{densité}}$	Valeur fitness
<b>Chromosome 1</b>	180	400	1	1,455	0,688
<b>Chromosome 2</b>	170	320	1	1,970	0,508
<b>Chromosome 3</b>	160	250	1	1,515	0,660
	Distance	Profit	Rank	$\Sigma_{\text{densité}}$	Valeur fitness
<b>Chromosome 4</b>	170	280	2	1,576	1,269
<b>Chromosome 5</b>	160	220	2	1,576	1,269
	Distance	Profit	Rank	$\Sigma_{\text{densité}}$	Valeur fitness
<b>Chromosome 6</b>	160	180	3	1	3

TABLE 3.5 – Calcul des fitness

Le tableau 3.5 montrent un exemple d'application de la méthode de partage après le classement avec NSGA. Nous utilisons le même exemple que dans la figure 3.9. Ici, nous avons pris  $\alpha=1$  et  $\sigma_{sh}=165$ .

### 3.3.4 Procédure de sélection

Après avoir généré et évalué une population donnée, nous sélectionnons les parents pour le croisement. Nous proposons une procédure de sélection basée sur deux roulettes consécutives. Elle vise à garder l'aspect aléatoire de cette étape tout en donnant une plus grande chance aux meilleures solutions d'être sélectionnées. Dans la première étape de sélection, nous utilisons une roulette où la part affectée à chaque rang est proportionnelle au nombre d'individus possédant ce rang. Le but de cette roulette est de choisir le rang du futur parent parmi tous les rangs possibles. Avec l'équation (3.5), nous déterminons la valeur  $Chance_i$  où  $Number_i$  représente le nombre de solutions de rang égal à  $i$  et  $Rank_{MAX}$  est le plus grand rang de la population. Nous utilisons la variable  $Chance_i$  dans l'équation (3.6) pour calculer la probabilité de sélection du rang  $Rank_i$ ,  $Prob_i$ .

$$Chance_i = Number_i \cdot (Rank_{MAX} - Rank_i + 1) \quad (3.5)$$

$$Prob_i = \frac{Chance_i}{\sum_{j=1}^{Rank_{MAX}} Chance_j} \quad (3.6)$$

Ainsi, pour un nombre égal de solutions pour deux rangs différents  $Rank_i$  et  $Rank_j$  (avec  $Rank_i < Rank_j$ , et  $Number_i = Number_j$ ), les solutions de rang plus faible  $Rank_i$  (donc meilleurs à priori) auront plus de chance d'être sélectionnées que celles de rang plus élevé  $Rank_j$ . À contrario, un rang élevé peut avoir une forte probabilité de choix si un grand nombre de solutions a ce rang.

Enfin, une sélection aléatoire est faite en appliquant les étapes suivantes :

- On génère un nombre aléatoire  $Roulette1$  entre 0 et 1.
- On initialise les variables suivantes :  $i=1$  et  $Plage\_Roulette1=0$ .
- $Plage\_Roulette1=Plage\_Roulette1 + Prob_i$
- Si  $Roulette1$  est entre 0 et  $Plage\_Roulette1$ , alors le rang du futur parent égal à  $i$ . Sinon, on incrémente  $i$  de 1 et on revient à l'étape précédente.

La figure 3.10 représente la première roulette de sélection pour l'exemple précédemment mentionné. Si par exemple on a généré un nombre aléatoire  $Roulette1$  égal à 0.65 donc le rang du futur parent égal à 2.

i	Chance <sub>i</sub>	Prob <sub>i</sub>
1	9	0.64
2	4	0.29
3	1	0.07

$$\sum_{j=1}^3 \text{Chance}_j = 14$$

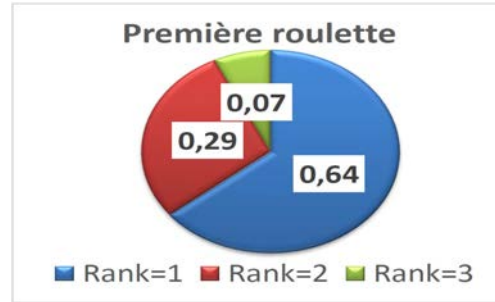


FIGURE 3.10 – Première roulette de sélection

La première roulette fournit donc une valeur de rang que nous utilisons dans la deuxième étape de sélection. Celle-ci consiste à sélectionner le parent final parmi les solutions candidates associées au rang obtenu par la première roulette. Comme nous l'avons abordé précédemment, la fonction de partage utilisée donne une fitness plus élevée aux chromosomes qui se trouvent dans les zones les moins denses. Nous utilisons ces valeurs fitness dans la deuxième roulette pour donner une plus grande chance aux chromosomes les plus "isolés" d'être sélectionnés. L'équation (3.7) définit comment nous calculons la probabilité finale  $FProb_i$  pour chaque solution candidate.

$$FProb_i = \frac{Fitness_i}{\sum_{j \in Population_i} Fitness_j} \quad (3.7)$$

Enfin, une sélection aléatoire est faite en appliquant les étapes suivantes :

- La deuxième roulette utilisée dépend du rang sélectionné en première phase de sélection.
- On génère un nombre aléatoire  $Roulette2$  entre 0 et 1.
- On initialise les variables suivantes :  $i$  égale à l'identifiant du première chromosome associé à cette roulette et  $Plage\_Roulette2=0$ .
- $Plage\_Roulette2=Plage\_Roulette2 + FProb_i$
- Si  $Roulette2$  est entre 0 et  $Plage\_Roulette2$ , alors le chromosome sélectionné est celui avec l'identifiant  $i$ . Sinon, on incrémente  $i$  de 1 et on revient à l'étape précédente.

La figure 3.11 représente les trois cas possibles pour la deuxième roulette de sélection, toujours pour le même exemple. Supposons qu'en utilisant la première roulette, nous ayons obtenu un rang égal à 2. Autrement dit le futur parent doit posséder ce rang. Ce cas correspond au schéma au centre de la figure 3.11. Si le nombre aléatoire  $Roulette2$  est égal à 0.7, cela implique que le chromosome 5 est sélectionné.

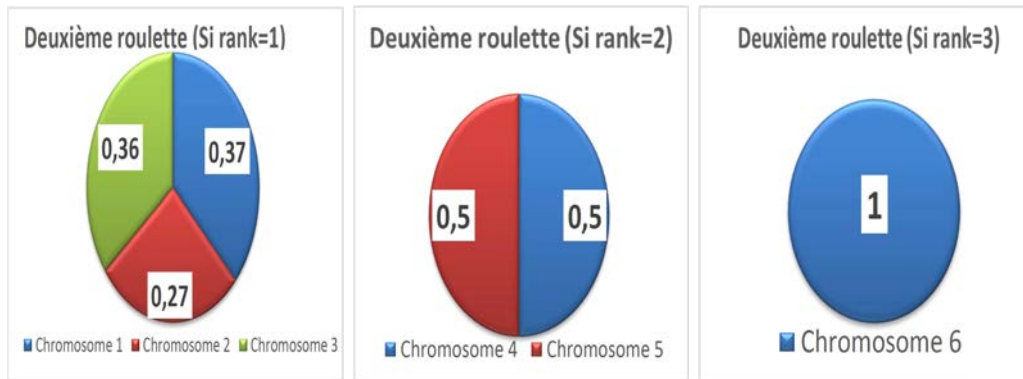


FIGURE 3.11 – Deuxième roulette de sélection

### 3.3.5 Croisement

Une fois que nous avons sélectionné deux parents en utilisant notre procédure de sélection, nous appliquons un opérateur de croisement afin d'obtenir de nouveaux enfants. L'idée est que chaque enfant hérite des bons gènes de ses parents. L'opérateur de croisement compare 2 à 2 les gènes en même position  $i$  de chaque parent. Il recopie en même position  $i$  de l'enfant le meilleur de ces deux gènes (meilleur au sens du plus fort profit pour l'enfant 1 et de la plus faible distance pour l'enfant 2).

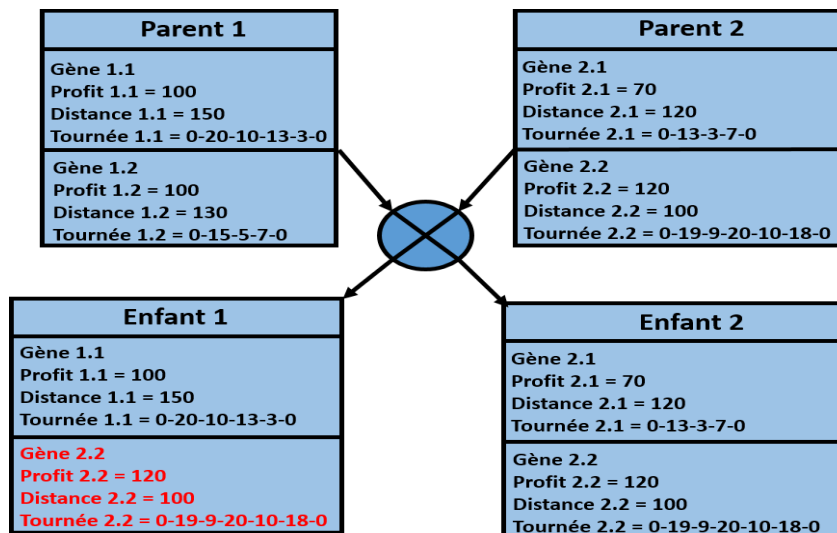


FIGURE 3.12 – Opérateur de croisement

Dans l'exemple de la figure 3.12, on note que l'enfant 2 est la copie conforme du parent 2 car  $\text{Distance } 2.1 < \text{Distance } 1.1$  et  $\text{Distance } 2.2 < \text{Distance } 1.2$ . Par contre, l'enfant 1 hérite du premier gène du parent 1 et du deuxième gène du parent 2 sur le critère du profit.

Dans notre exemple, l'enfant 2 représente une solution réalisable. Mais ce n'est pas

le cas pour l'enfant 1 qui représente une solution irréalisable car certains sites sont visités deux fois. Pour résoudre ce problème, nous avons implémenté un algorithme de réparation que nous appliquons directement après l'opérateur de croisement pour nous assurer que les enfants obtenus respectent toutes les contraintes de notre variante. Cet algorithme consiste à supprimer les doublons de la solution. Il choisit aléatoirement un des gènes (donc une des tournées) contenant des sites redondants et supprime ces sites. Ensuite, nous utilisons notre méthode de recherche locale décrite dans Algorithme 9 afin d'améliorer, si possible, ce gène. La figure 3.13 représente le résultat de la réparation sur l'enfant 1. L'algorithme a ici supprimé les sites redondants 10 et 20 dans le deuxième gène, puis il a amélioré par recherche locale la distance totale du gène 2.2, en réorganisant la visite des sites restants.

<b>Enfant 1</b>	
<b>Gène 1.1</b>	
<b>Profit 1.1 = 100</b>	
<b>Distance 1.1 = 150</b>	
<b>Tournée 1.1 = 0-20-10-13-3-0</b>	
<b>Gène 2.2</b>	
<b>Profit 2.2 = 100</b>	
<b>Distance 2.2 = 80</b>	
<b>Tournée 2.2 = 0-19-18-9-0</b>	

FIGURE 3.13 – Enfant 1 après réparation

### 3.3.6 Hybridation avec des méthodes de recherche locale

Pour améliorer la qualité des enfants obtenus, nous avons testé trois méthodes de recherche locale, utilisées comme des procédures de mutation :

- (a) La méthode "Improve" consiste à améliorer le premier enfant en essayant d'augmenter son profit total.

L'idée de cette méthode est de choisir aléatoirement une demande, composée d'une paire de sites ou d'un site couplé avec le dépôt, parmi l'ensemble des demandes non-satisfaites. Ensuite, nous appliquons une heuristique d'insertion de cette demande dans les gènes de l'enfant 1. Une fois que nous avons trouvé une position d'insertion faisable (on s'arrête à la première trouvée), nous passons à une autre demande non-honorée. Pour limiter les temps d'exécution, nous avons choisi d'arrêter cette procédure lorsque nous détectons qu'aucune position d'insertion n'est faisable. Par ailleurs, nous n'appliquons "Improve" qu'à l'enfant 1, parce qu'il représente pour nous le descendant caractérisé par le profit le plus élevé possible. Dans la figure 3.14, nous illustrons le concept de cette méthode en considérant l'enfant 1 obtenu dans l'exemple de la figure 3.13 et en essayant d'insérer une demande non-satisfaite, composée du site 12 couplé avec le dépôt, dans une de ses tournées (étape 1). Puisque une insertion faisable existe (étape 2), une autre demande non-honorée est testée. Et comme aucune insertion faisable n'est trouvée pour cette dernière (étape 3), la méthode s'arrête.

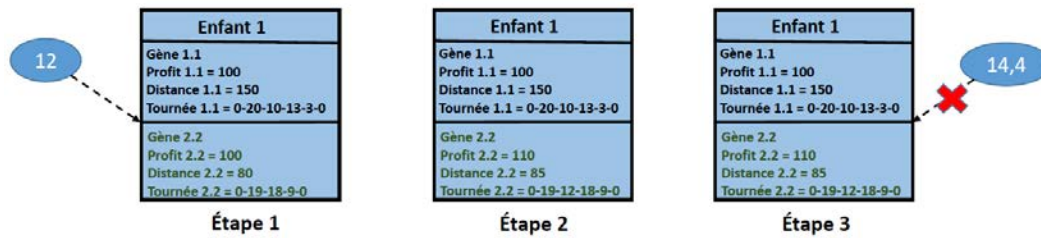


FIGURE 3.14 – Méthode "Improve"

- (b) Les méthodes «auto-crossover-1» et «auto-crossover-2» (Algorithme 10 avec  $x$  respectivement égal à 1 ou 2) consistent à déplacer respectivement une et deux demandes d'un gène vers un autre dans le but de diminuer la distance totale de la solution. Ces deux méthodes sont appliquées sur les deux descendants, enfants 1 et 2.

---

**Algorithme 10** La méthode "auto-crossover-x"
 

---

**Input :** Solution  $S$ ;

**Output :** Solution  $S$ ;

$S\_temp = S$ ;

**for**  $i$  varie de 1 à  $S\_temp.Nombre\_des\_tournées$  **do**

  Enlever aléatoirement  $x$  demande(s) de la tournée  $i$ ;

**for**  $j$  varie de 1 à  $S\_temp.Nombre\_des\_tournées$  **do**

**if**  $i \neq j$  **then**

      Tester d'insérer les  $x$  demande(s) dans la meilleure position de la tournée  $j$  telle que  $j$  reste faisable;

**if**  $Distance(S\_temp) < Distance(S)$  **then**

        |  $S = S\_temp$ ;

**else**

        |  $S\_temp = S$ ;

**end**

$j = j + 1$ ;

**end**

**end**

**end**

retourner  $S$ ;

---

Selon la méthode utilisée, une demande (composée d'une paire de sites ou d'un site couplé avec le dépôt) ou deux demandes sont déplacées. La figure 3.15 représente l'application de la méthode "auto-crossover-1" sur l'enfant 2 obtenu dans l'exemple de la figure 3.13. L'étape 1 représente l'état de l'enfant avant l'application de la méthode. Une demande est alors choisie aléatoirement dans le premier gène (étape 2). La meilleure insertion faisable de cette demande dans le deuxième gène est représentée par l'étape 3, nous remarquons que la distance totale après cette insertion a augmenté. Nous reprenons donc l'état initial de

l'enfant 2 (étape 4). Une autre demande est choisie dans le deuxième gène (étape 5). Mais cette fois-ci, l'insertion de cette demande dans le premier gène a diminué la distance totale de la solution donc nous remplaçons l'état initial d'enfant 2 par son nouvel état.

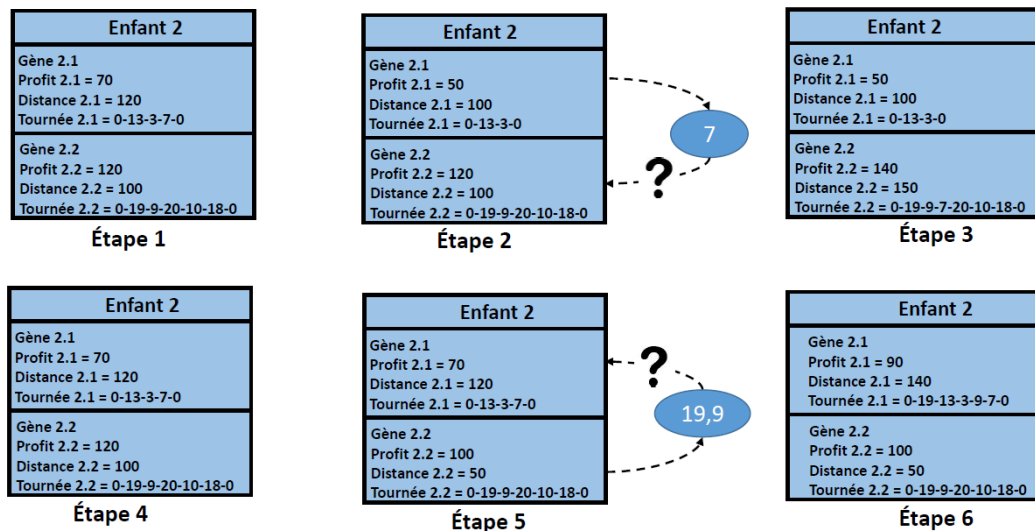


FIGURE 3.15 – Exemple d'application pour la méthode auto-crossover-1

Le résultat souhaité en appliquant les méthodes "autocrossover-x" est de diminuer la distance totale de chaque enfant.

### 3.3.7 Test de dominance

Avant de valider définitivement un enfant, après les phases de génération, réparation et amélioration, nous évaluons sa propriété de dominance : s'il domine au moins un de ses parents, nous le gardons et il sera inséré dans la population. Sinon, il sera détruit. Dans l'exemple des figures 3.12 et 3.13, l'enfant 1 domine le parent 1 donc il sera conservé et inséré dans la nouvelle population. Au contraire, l'enfant 2 a échoué au test de dominance parce qu'il ne domine aucun de ses parents, il sera donc supprimé. Après la génération de la nouvelle population, nous augmentons notre indice d'itérations de 1. Ce processus est répété jusqu'à ce que le nombre maximal d'itérations ait été atteint.

### 3.3.8 Algorithme global

La figure 3.16 résume toutes les étapes de notre algorithme génétique hybride.

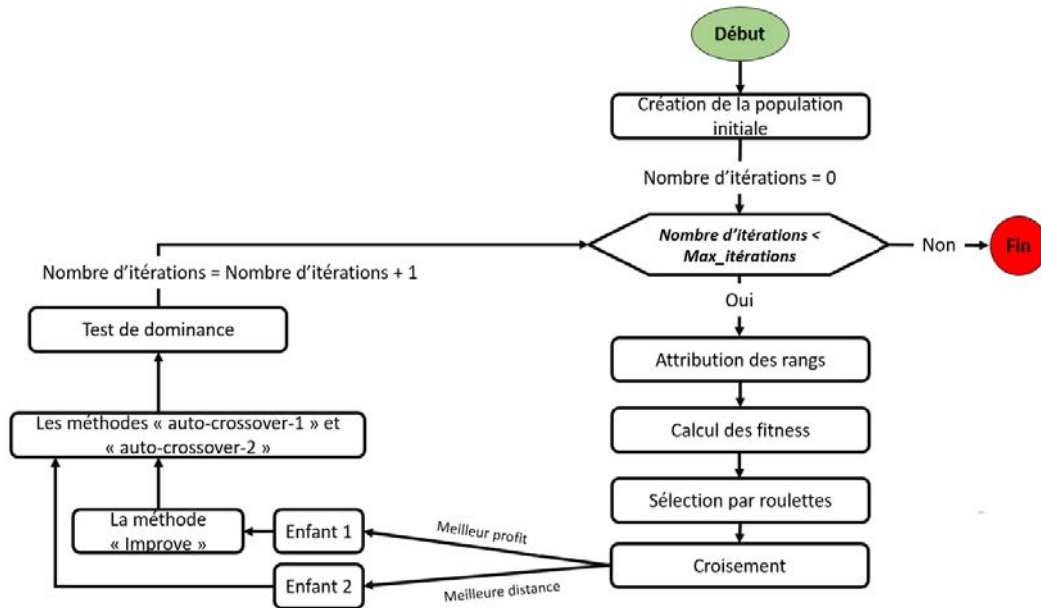


FIGURE 3.16 – Schéma global de notre algorithme génétique hybride

### 3.3.9 Résultats expérimentaux

Dans cette partie, nous donnons quelques résultats expérimentaux obtenus avec notre algorithme génétique hybride (HGA). Notre environnement expérimental est le Windows 10 et un ordinateur avec un processeur Intel Core i5, 2.80 GHz et 4 Go de RAM. L'algorithme a été codé en C++. Comme dans la plupart des métaheuristiques, l'ajustement des paramètres est l'une des étapes les plus difficiles étape lors de la mise en œuvre. Malheureusement, il n'y a pas de règle de paramétrage standard car les paramètres dépendent fortement de la structure même du problème d'optimisation. Les valeurs des paramètres utilisés ont été choisies après plusieurs tests. La configuration retenue est la suivante :  $Max = 200$ ,  $CP = 15$ , maximum nombre d'itérations = 10,  $\alpha = 1$  et  $\sigma_{sh} = 150$ . En effet, la diminution de la taille de la population initiale  $Max$  va limiter les performances de l'algorithme. En d'autres termes, la population initiale devient moins diversifiée. Par conséquent, l'algorithme converge généralement vers des minima locaux. D'un autre côté, avoir plus de 200 individus initiaux peut augmenter la diversité dans la première population. Cependant, cela représentera une dépense supplémentaire de temps de calcul et, dans la plupart des cas, la même qualité de solutions calculées finales. Les deux variables CP et maximum nombre d'itérations sont en relation proportionnelle avec le temps de résolution. De plus, si nous décidons d'incrémenter ou de décrémenter largement les paramètres  $\alpha$  et  $\sigma_{sh}$ , nous diminuons la chance d'avoir des solutions bien réparties sur le front Pareto. Nous avons utilisé les mêmes instances du SPDPTWPD précédemment générées. Afin de vérifier la stabilité de notre méthode HGA, nous avons effectué 10 exécutions par instance et les résultats sont presque les mêmes dans toutes les exécutions. En moyenne, nous avons retrouvé les mêmes résultats 8



fois environ sur 10 lancements.

Dans un premier temps, nous avons testé notre approche sur les petites instances avec 20 nœuds et un nombre de véhicules de 1 à 6. Les résultats sont illustrés dans le tableau 3.6 . Celui-ci donne pour chaque instance les valeurs (Profit, Distance) des solutions non dominées (de l'espace de recherche exploré) obtenues sur l'ensemble des 10 exécutions.

Problème	Nombre de véhicules	Temps HGA(s)	HGA		LP		Écart (HGA LP) en %
			Profit	Distance	Distance*	Temps	
SPDPTWPD-C201	1	17	180	275.19	275.19	0.3	0
			170	272.59	272.59	0.45	0
			160	201.53	201.53	0.92	0
			150	199.07	198.92	3.2	0.08
			140	156.15	151.96	3.5	2.76
			130	146.71	141.52	3.7	3.67
			100	58.72	58.72	2.1	0
			70	50.6	50.6	1.7	0
			40	40.56	40.56	2.4	0
			20	33.36	33.36	1.4	0
SPDPTWPD-C202	1	13.12	144	167.99	167.99	19.88	0
			132	106.06	104.45	20.65	1.54
			124	102.27	101.75	19.62	0.51
			107	94.26	94.26	18.4	0
			83	59.42	59.42	17.24	0
			44	46.62	46.62	17.08	0
			25	40.22	40.22	16.83	0
SPDPTWPD-C202	2	15.47	217	212.94	212.94	52.15	0
			190	149.19	149.19	48.13	0
			132	106.06	104.45	49.22	1.54
			117	101.80	95.88	54.4	6.17
			107	94.26	94.26	52.18	0
			83	59.42	59.42	47.12	0
			44	46.62	46.62	45.13	0
25	40.22	40.22	44.41	0			
SPDPTWPD-R201	6	12.45	190	398.25	398.25	0.35	0
			180	379.40	379.40	0.42	0
			170	327.66	327.66	0.47	0
			160	310.96	310.96	0.38	0
			150	259.21	259.21	0.35	0
			140	242.93	242.93	0.31	0
			100	158.45	158.45	0.29	0
			70	104.26	104.26	0.29	0
			20	30.54	30.54	0.28	0
SPDPTWPD-R202	4	13.01	213	384.24	378.959	12.2	1.39
			190	326.57	326.57	13.07	0
			120	191.87	187.95	12.95	2.09
			81	129	129	12.59	0
			46	41.78	41.78	12.45	0
			14	26.38	26.38	12.17	0
			160	233.2	233.2	0.33	0
SPDPTWPD-RC201	2	12.39	150	225.37	224.22	0.37	0.51
			140	214.25	212.72	0.36	0.72
			130	207.04	203.73	0.34	1.62
			120	193.5	193.5	0.31	0
			110	190.02	189.76	0.3	0.14
			80	116.46	116.46	0.28	0
			60	96.25	96.25	0.27	0
			40	80.7	80.7	0.25	0

Problème	Nombre de véhicules	Temps HGA (s)	HGA		LP		Écart (HGA LP) en %
			Profit	Distance	Distance*	Temps	
SPDPTWPD-RC202	2	12.82	180	239.66	235.09	0.52	1.94
			156	203.22	202.86	0.56	0.18
			107	115.48	115.48	0.57	0
			85	98.93	98.93	0.51	0
			45	81.3	81.3	0.49	0
			10	32.24	32.24	0.47	0

TABLE 3.6 – Résultats sur les petites instances

La méthode exacte abordée dans la partie 2.3.2 est notée LP dans le tableau 3.6. Elle sert à vérifier si chaque solution obtenue avec notre HGA, ayant un profit total  $Profit$  et une distance totale  $Distance$ , est sur le front de Pareto. Pour ce faire, nous résolvons notre modèle mathématique en utilisant le solveur commercial CPLEX avec la minimisation de distance (F2) comme fonction objectif. Outre les contraintes de notre modèle (voir la sous-section 2.2), nous avons ajouté l'équation (3.8) suivante :

$$\sum_{i \in N} \sum_{j \in N} \sum_{k \in V} P_i X_{ij}^k \geq Profit \quad (3.8)$$

À l'aide de notre méthode exacte, nous obtenons  $Distance^*$  qui représente la solution optimale fournie par notre méthode exacte pour un  $Profit$  donné. Puis nous calculons l'écart entre cette valeur et  $Distance$  obtenue par notre HGA en utilisant l'équation 3.1. Nous répétons ces étapes pour chaque solution obtenue par notre métaheuristique. Le tableau 3.6 montre que nous avons de bonnes solutions car dans le pire des cas, nous avons un écart de 6.17% parmi toutes les solutions testées. En moyenne, l'écart est de 0.45%. Les solutions pour lesquelles l'écart est égal à 0% appartiennent au front de Pareto.

Le temps de calcul de notre HGA pour chacune des petites instances ne dépasse pas 17 secondes. Par contre, le temps nécessaire pour trouver les solutions non-dominées en utilisant notre méthode exacte varie entre quelques secondes (par exemple, SPDPTWPD-RC201) et quelques minutes (par exemple, SPDPTWPD-C202 avec 2 véhicules).

Nous avons choisi deux instances avec 20 sites (celles qui nécessitent le plus faible et le plus élevé temps de calcul) et nous avons généré le front de Pareto complet en testant toutes les possibilités de profit. Le but de ces tests était de comparer le front obtenu à l'aide de notre HGA et le front complet ainsi que le temps de résolution nécessaire pour avoir chacune de ces courbes (figures 3.17 et 3.18). Pour les instances de groupe 1 (le profit est fixe pour tous les sites), le nombre maximal de solutions non-dominées est égal au nombre de sites plus un (la solution avec un profit et une distances nuls). Par contre, pour les instances de groupe 2 (le profit est différent pour chaque site), l'obtention du front de Pareto complet nécessite de tester toutes les combinaisons des profits possibles. Les deux comparaisons montrent que d'une part les solutions fournies par notre HGA sont réparties sur le front de Pareto grâce à la fonction de partage que nous utilisons, ce qui donne au décideur un ensemble représentatif des compromis possibles. D'autre part, nous remarquons que dans la

figure 3.17, notre métaheuristique fournit toutes les solutions avec les profits élevés. Cela est dû au fait que dans le groupe 1, le nombre de combinaisons de sites pour avoir un profit donné est plus élevé que dans les instances appartenant au deuxième groupe.

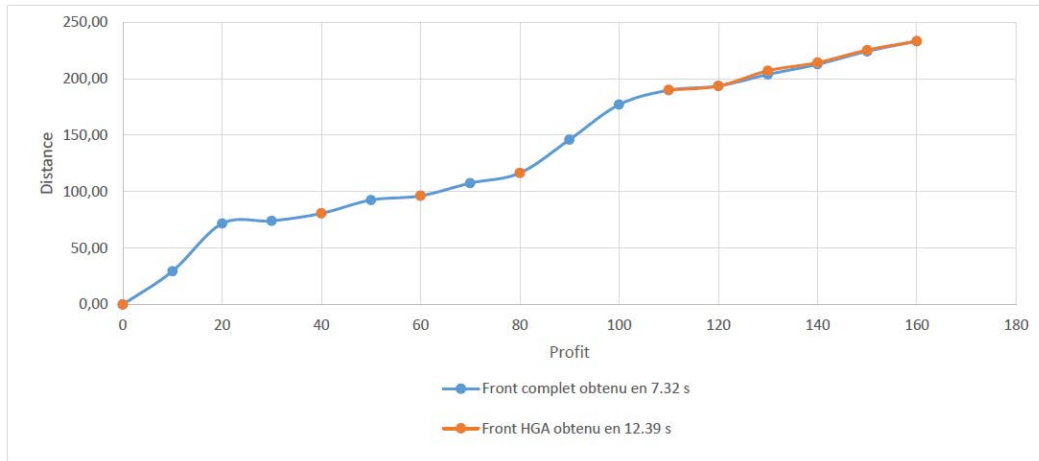


FIGURE 3.17 – Comparaison des fronts : SPDPTWPD-RC201

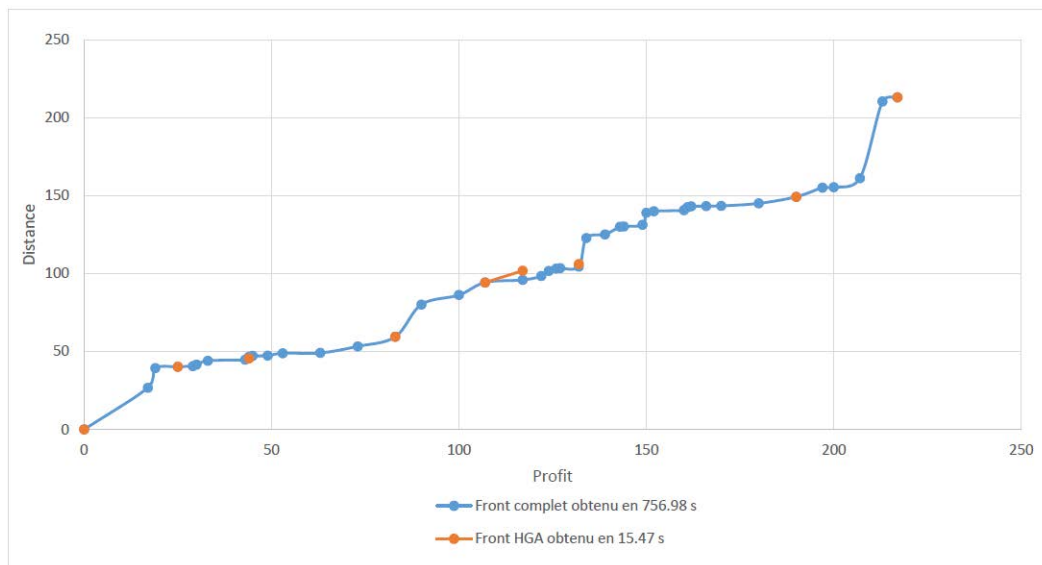


FIGURE 3.18 – Comparaison des fronts : SPDPTWPD-C202 à 2 véhicules

Nous avons également testé notre approche sur des instances de taille moyenne (50 nœuds) et un nombre de véhicules de 1 à 13. Les résultats obtenus et le temps de calcul pour chaque méthode ainsi que l'écart sont donnés par le tableau 3.7. Pour ces instances, la méthode LP n'était parfois pas capable de donner des solutions optimales dans un délai raisonnable ( $> 2$  heures), alors une solution avec le symbole \* représente la meilleure solution faisable obtenue après deux heures de résolution.

Problème	Nombre de véhicules	Temps HGA(s)	HGA		LP		Écart (HGA LP) en %
			Profit	Distance	Distance*	Temps	
SPDPTWPD-C501	5	63.87	480	370.06	363.28	9.67	1.87
			460	360.13	356.75	22.17	0.59
			380	259.52	259.52	27.75	0
			260	152.01	152.01	32.42	0
			170	97.69	97.69	30.15	0
			50	38.41	38.41	29.14	0
SPDPTWPD-C502	1	442.15	437	431.25	431.25	28.31	0
			389	183.11	179.55	144.12	1.98
			272	147.06	144.58	138.15	1.72
			184	103.60	103.60	137.29	0
			83	56.89	56.89	34.85	0
SPDPTWPD-C502	2	484.71	656	525.87	525.87	204.44	0
			591	299.13	295.25	315.16	1.31
			538	272.25	272.25	321.12	0
			459	226.78	226.78	302.49	0
			327	163.18	159.77	285.72	2.13
			214	110.9	108.29	263.99	2.41
			138	87.62	84.06	250.16	4.24
			83	56.89	56.89	247.12	0
SPDPTWPD-R501	13	32.87	490	1076.57	1075.77	14.86	0
			450	932.02	932.02	18.32	0
			370	719.6	719.6	19.47	0
			260	461.92	461.92	18.26	0
			160	256.48	256.48	17.77	0
			70	103.15	103.15	15.08	0
SPDPTWPD-R502	10	22.13	576	834.92	834.92	17.24	0
			564	818.91	780.59	2520	4.91
			380	500.03	476.01*	7200	5.05
			260	321.86	293.31	1250	9.73
			160	180.55	180.55	1187	0
			50	61.05	61.05	202	0
SPDPTWPD-RC501	4	31.14	320	481.37	481.37	1629.02	0
			280	421.49	421.49	2943.12	0
			170	282.87	282.87	2820.73	0
			100	174.51	174.51	2534.99	0
			40	80.66	80.66	2058.12	0
SPDPTWPD-RC502	4	27.48	420	527.76	527.76	151.37	0
			313	395.26	369.09	247.88	7.09
			274	353.2	325.52	266.13	8.5
			165	200.88	200.88	195.54	0
SPDPTWPD-RC502	5	29.01	116	128.10	128.10	128.22	0
			500	689.51	686.69	151.37	0.41
			432	548.87	544.29	342.19	0.48
			345	422.63	418.05	489.07	1.1
			238	283.35	278.1	512.99	1.89
			116	128.10	128.10	423.65	0
SPDPTWPD-RC502	6	29.98	57	87.71	87.71	312.36	0
			557	739.61	739.61	5120.15	0
			509	673.84	670.92	4952.98	0.44
			447	558.43	558.43*	7200	0
			390	470.72	470.72*	7200	0
			307	365.6	362.81	4223.42	0.77
			252	301.05	301.05	3112.88	0
			161	193.19	193.19	2568.08	0
			97	109.26	109.26	2184.59	0
15	63.24	63.24	1523.74	0			

TABLE 3.7 – Résultats sur les instances moyennes

En augmentant la taille des instances, la complexité augmente ce qui explique que l'écart maximal entre notre HGA et la méthode LP est 9.73% avec une moyenne

de 1.01%. Pour ce deuxième ensemble d'instances, nous remarquons que notre métaheuristique est plus rapide que la méthode exacte. Nous avons testé l'instance SPDPTWPD-RC502 avec trois nombres de véhicules (4,5 et 6) afin d'analyser l'impact de ce dernier sur les résultats obtenus. Nous remarquons que le temps de résolution augmente, et que le nombre de solutions obtenues augmente aussi. Ces augmentations peuvent être vérifiées par le fait que l'espace de solutions augmente en incrémentant le nombre de véhicules. Enfin, le tableau 3.8 montre les résultats obtenus sur l'ensemble des grandes instances (avec 100 nœuds) et un nombre de véhicules de 1 à 7.

Problème	Nombre de véhicules	Temps HGA(s)	HGA		LP		Écart (HGA LP) en %
			Profit	Distance	Distance*	Temps	
SPDPTWPD-C1001	6	310.79	690	437.04	437.04	928.93	0
			580	305.03	305.03	1125.03	0
			420	214.04	214.04	1556.42	0
			330	161.7	161.7	1674.48	0
			240	110.2	110.2	1324.12	0
			100	46.44	46.44	922.76	0
SPDPTWPD-C1002	7	256.57	730	600.32	535.21*	7200	12.17
			630	388.48	394.43*	7200	-1.51
			540	295.08	295.08*	7200	0
			420	217.31	217.31*	7200	0
			340	164.91	164.91	3562.18	0
			210	101.84	101.84	2475.09	0
SPDPTWPD-R1001	7	152.95	100	48.33	48.33	1178.12	0
			490	635.12	598.49*	7200	6.12
			460	599.3	552.3*	7200	8.51
			320	353.22	350.59	5485.12	0.75
			240	246.66	246.66	4100.25	0
			110	108.05	108.05	3562.18	0
SPDPTWPD-R1002	6	183.95	514	679.44	679.44	1391.96	0
			490	582.54	560.19*	7200	3.99
			372	327.65	327.65	6230.56	0
			284	225.59	225.59	4058.33	0
			170	126.75	126.75	3026.17	0
			90	66.16	66.16	1674.12	0
SPDPTWPD-RC1001	1	169.08	120	140.35	140.35	382.23	0
			100	105.04	105.04	593.17	0
			70	82.71	82.71	432.08	0
			40	55.12	55.12	228.93	0
			20	32.88	32.88	3026.17	0
			138	125.68	125.68	103	0
SPDPTWPD-RC1002	1	112.54	96	99.17	99.17	168.12	0
			80	95.07	95.07	122.74	0
			56	89.86	89.86	107.09	0
			25	34.78	34.78	104.74	0
			252	274.49	274.49	7000	0
SPDPTWPD-RC1002	2	166.24	214	222.17	237.74*	7200	-6.55
			179	175.87	175.87	6150.2	0
			111	110.81	110.81	4230.09	0
			66	84.61	84.61	1963.55	0
			25	34.78	34.78	287.63	0
			25	34.78	34.78	287.63	0

TABLE 3.8 – Résultats sur les grandes instances

Dans le tableau 3.8, nous trouvons des valeurs d'écart négatives. En effet, elles montrent que la solution trouvée par notre HGA est meilleure que celle fournie par la méthode exacte LP à l'issue de deux heures de résolution. Pour cet ensemble

d'instances, l'écart maximal est de 12.17% avec une moyenne de 0.58%.

Une autre comparaison est faite mais cette fois-ci entre les trois méthodes abordées dans ce chapitre (tableau 3.9). La deuxième colonne (NV) donne le nombre de véhicules. La colonne NSV représente le nombre de sites visités dans la solution fournie par chacune de trois approches. Ici, nous avons pris uniquement la solution avec le meilleur profit fournie par notre HGA.

Instance	NV	Approche lexicographique (AL)				Recuit simulé (RS)				HGA			
		NSV	Profit	Distance	Temps	NSV	Profit	Distance	Temps	NSV	Profit	Distance	Temps
SPDPTWPD-C201	1	18	180	275.19	0.3	18	180	275.19	11	18	180	275.19	17
SPDPTWPD-C202	1	11	144	167.99	19.88	11	144	167.99	7	11	144	167.99	13.12
SPDPTWPD-C202	2	20	217	212.94	52.15	20	217	212.94	8	20	217	212.94	15.47
SPDPTWPD-R201	4	19	190	398.25	0.35	19	190	398.25	6	19	190	398.25	12.45
SPDPTWPD-R202	6	19	213	378.95	12.2	19	213	378.95	6	19	213	384.24	13.01
SPDPTWPD-RC201	2	16	160	233.2	0.33	16	160	233.2	6	16	160	233.209	12.39
SPDPTWPD-RC202	2	16	180	235.09	0.52	16	180	235.09	6	16	180	239.66	12.82
SPDPTWPD-C501	5	48	480	363.28	9.67	48	480	363.28	57	48	480	370.06	63.87
SPDPTWPD-C502	1	30	437	431.25	28.31	31	436	386.08	433	30	437	431.25	442.15
SPDPTWPD-C502	2	50	656	525.87	204.44	50	656	525.87	401	50	656	525.87	484.71
SPDPTWPD-R501	13	49	490	1075.77	14.86	49	490	1075.77	21	49	490	1076.57	32.87
SPDPTWPD-R502	10	41	576	834.92	17.24	41	576	834.92	20	41	576	834.92	22.13
SPDPTWPD-RC501	4	32	320	481.37	1629.02	31	310	489.4	28	32	320	481.37	31.14
SPDPTWPD-RC502	4	29	420	527.76	151.37	27	400	518.14	26	29	420	527.76	27.48
SPDPTWPD-RC502	5	35	500	686.69	1352.38	33	480	677.07	27	35	500	689.51	29.01
SPDPTWPD-RC502	6	39	562	758	4505.16	38	547	801.64	28	39	557	739.61	29.98
SPDPTWPD-C1001	6	69	690	437.04	928.93	68	680	469.58	290	69	690	437.04	310.79
SPDPTWPD-C1002	7	77	770	542.15	621.54	76	760	575.08	233	73	730	600.32	256.57
SPDPTWPD-R1001	7	50	500	639.64	3540.04	50	500	639.64	138	49	490	635.12	152.95
SPDPTWPD-R1002	6	40	514	679.44	1391.96	40	512	710.02	157	40	514	679.44	183.95
SPDPTWPD-RC1001	1	12	120	140.35	382.23	11	110	127.22	151	12	120	140.35	169.08
SPDPTWPD-RC1002	1	10	138	125.68	103	10	138	125.68	109	10	138	125.68	112.54
SPDPTWPD-RC1002	2	18	252	274.49	7000	18	252	274.49	145	18	252	274.49	166.24

TABLE 3.9 – Comparaison entre les trois approches

L'algorithme génétique est un peu plus gourmand que l'approche de recuit simulé en termes de temps. Si nous nous focalisons sur le nombre de sites visités, nous remarquons que deux solutions équivalentes en termes de nombre de sites visités peuvent donner lieu à des valeurs différentes en termes de profit et de distance (SPDPTWPD-R1002). De plus, une solution qui consiste à visiter plus de sites n'a pas nécessairement un profit plus élevé (SPDPTWPD-C502 à 1 véhicule). Le tableau 3.10 visualise les écarts (calculés avec l'équation (3.1)), en termes du profit et de la distance, entre notre HGA et les deux autres approches. Dans un premier temps, nous nous concentrons sur l'écart entre HGA et AL. Dans 65.21% des cas (contre 60.87% pour RS), l'algorithme génétique fournit les mêmes résultats que l'approche

lexicographique et pour 8.69% des instances testées (contre 17.39% pour RS), notre HGA fournit des solutions avec un profit moins intéressant que l'AL mais avec une distance plus faible. Pour le reste des instances testées qui représentent 26.08% (contre 21.74% pour RS), les solutions obtenues par notre HGA sont dominées par les solutions de l'AL. En analysant les deux dernières colonnes dans le tableau 3.10, nous observons plusieurs cas possibles. Si les écarts sont nuls, en termes de profit et de distance, alors les deux méthodes fournissent la même solution. Un écart positif dans la case profit (respectivement distance) implique que notre HGA obtient un profit (respectivement une distance) plus élevé(e) que RS. À contrario, un écart négatif implique une diminution.

Instance	Nombre de véhicules	Ecart(HGA AL) en %		Ecart(HGA RS) en %	
		Profit	Distance	Profit	Distance
SPDPTWPD-C201	1	0	0	0	0
SPDPTWPD-C202	1	0	0	0	0
SPDPTWPD-C202	2	0	0	0	0
SPDPTWPD-R201	4	0	0	0	0
SPDPTWPD-R202	6	0	1.40	0	1.40
SPDPTWPD-RC201	2	0	0	0	0
SPDPTWPD-RC202	2	0	1.94	0	1.94
SPDPTWPD-C501	5	0	1.87	0	1.87
SPDPTWPD-C502	1	0	0	0.23	11.70
SPDPTWPD-C502	2	0	0	0	0
SPDPTWPD-R501	13	0	0.07	0	0.07
SPDPTWPD-R502	10	0	0	0	0
SPDPTWPD-RC501	4	0	0	3.23	-1.64
SPDPTWPD-RC502	4	0	0	5	1.86
SPDPTWPD-RC502	5	0	0.41	4.17	1.84
SPDPTWPD-RC502	6	-0.89	-2.43	1.83	-7.74
SPDPTWPD-C1001	6	0	0	1.47	-6.93
SPDPTWPD-C1002	7	-5.19	10.73	-3.95	4.39
SPDPTWPD-R1001	7	-2	-0.71	-2	-0.71
SPDPTWPD-R1002	6	0	0	0.39	-4.31
SPDPTWPD-RC1001	1	0	0	9.09	10.32
SPDPTWPD-RC1002	1	0	0	0	0
SPDPTWPD-RC1002	2	0	0	0	0

TABLE 3.10 – Écarts entre HGA et les deux autres approches

Le tableau 3.11 montre la moyenne et l'écart-type en termes de temps CPU et d'objectifs pour les deux approches HGA et AL. Comparant ce tableau avec celui qui donne la même analyse mais entre AL et RS (tableau 3.4), nous remarquons que la méthode RS (100.61 secondes) est un peu plus rapide que notre HGA (113.55 secondes) mais les deux méthodes restent largement plus rapides que l'approche lexicographique. Par contre, notre HGA dépasse le RS par rapport à la qualité des solutions fournies avec un écart moyen pour le profit de 0.58 (contre -1.14) couplé avec un écart-type moyen de 1.15 (contre 2.12) et un écart moyen absolu pour la distance de 0.58 (contre 2.11) avec un écart-type de 2.37 (contre 3.41).

Instances	Temps CPU				Ecart(HGA AL)		Ecart(HGA AL)	
	Moyenne		Ecart-type		Moyenne		Ecart-type	
	AL	HGA	AL	HGA	Profit	Distance*	Profit	Distance*
<b>20 sites</b>	12.25	13.75	19.21	1.77	0	0.48	0	0.83
<b>50 sites</b>	879.16	129.26	1495.36	190.13	-0.10	-0.01	0.30	1.09
<b>100 sites</b>	1995.39	193.16	2483.43	67.53	-1.03	1.43	1.98	4.11
<b>Toutes</b>	955.04	113.55	1767.71	140.27	-0.35	0.58	1.15	2.37

TABLE 3.11 – Analyse des résultats obtenus

Outre le faible temps de résolution (moins élevé que AL et un peu au-dessus de RS) et la bonne qualité des solutions obtenues par rapport aux deux autres approches abordées dans ce chapitre, notre algorithme génétique hybride fournit un ensemble des solutions non-dominées (dans l'espace de recherche exploré). Les solutions fournies par notre HGA représentent des compromis entre les objectifs traités et facilitent donc le choix du décideur en lui proposant une multitude de possibilités. Le seul inconvénient est que le HGA ne donne pas de garantie que les solutions obtenues sont Pareto-optimales, même si nous avons pu observer qu'elles sont effectivement proches ou sur ce front.

### 3.4 Conclusion

Ce chapitre était consacré à la résolution du Selective Pickup and Delivery Problem with Time Windows and Paired Demands (SPDPTWPD) en tenant compte de deux objectifs afin de se rapprocher des diverses applications réelles dans lesquelles plusieurs objectifs sont présents.

Trois méthodes multi-objectifs sont introduites pour traiter notre variante. Une approche lexicographique est d'abord utilisée pour la résoudre. Ensuite, deux méta-heuristiques ont été introduites. La première est basée sur une approche de type recuit simulé combinée avec une recherche Tabou et la deuxième est basée sur un algorithme génétique hybride. Ce dernier combine des avantages que nous ne retrouvons pas simultanément dans les deux autres approches : la rapidité (le temps de résolution est proche de celui de l'algorithme de type recuit simulé) et la qualité des solutions (un écart très faible par rapport à l'approche lexicographique). Contrairement aux deux autres approches, l'algorithme génétique est capable de générer un ensemble de solutions à l'issue d'une seule exécution. Pour toutes ces raisons, nous pouvons considérer que notre algorithme génétique hybride représente la meilleure méthode introduite dans ce chapitre.

Dans la réalité, toutes les demandes doivent être satisfaites. Or la variante considérée dans les chapitres 2 et 3 laisse la possibilité qu'un ensemble de demandes ne soient pas honorées à la fin de la période considérée. Cela implique que les demandes restantes devront être planifiées au cours de la période suivante ou celles d'après. Pour nous, considérer ce problème dans sa globalité revient à traiter une



autre variante du PDP dans laquelle nous définissons plusieurs périodes et nous supposons que la totalité des demandes doivent être satisfaites sur l'ensemble des périodes disponibles. Le but du dernier chapitre sera donc de résoudre cette variante multi-périodes du PDP.



# Résolution du Mu-PDPTWPD

*Le but de ce chapitre est de traiter une nouvelle variante du PDP dans laquelle les demandes sont planifiées sur un horizon composé de plusieurs périodes. Cette variante est le Multi-period Pickup and Delivery Problem with Time Windows and Paired Demands (Mu-PDPTWPD) que nous modélisons puis traitons dans un contexte mono-objectif avec une méthode exacte et trois approches séquentielles. Ensuite, une approche lexicographique ainsi qu'une métaheuristique basée sur un algorithme génétique combiné avec une méthode de type GRASP sont introduites pour résoudre le Mu-PDPTWPD dans un contexte multi-objectifs. Pour montrer les performances de chacune de nos approches, nous menons divers tests sur plusieurs jeux d'instances du problème. Ce chapitre se termine par des tests et une discussion sur la pertinence des variantes étudiées et méthodes développées dans la thèse, dans le cadre d'une planification collaborative.*

## Sommaire

<b>4.1</b>	<b>Introduction</b>	<b>101</b>
<b>4.2</b>	<b>Formulation mathématique</b>	<b>103</b>
4.2.1	Notations	103
4.2.2	MILP pour le Mu-PDPTWPD	104
<b>4.3</b>	<b>Étude mono-objectif</b>	<b>106</b>
4.3.1	Méthode exacte : MILP-Global	106
4.3.2	Approches séquentielles	107
4.3.3	Résultats expérimentaux	111
<b>4.4</b>	<b>Étude multi-objectifs</b>	<b>115</b>
4.4.1	Approche lexicographique	116
4.4.2	Algorithme génétique hybride combiné avec GRASP	117
4.4.3	Résultats expérimentaux	124
<b>4.5</b>	<b>Analyse de la collaboration entre chargeurs</b>	<b>129</b>
4.5.1	Contexte : Transport Collaboratif en Distribution Urbaine	129
4.5.2	Collaboration et tournées mono-période	129
4.5.3	Collaboration et tournées multi-périodes	138
4.5.4	Bilan	142
<b>4.6</b>	<b>Conclusion</b>	<b>142</b>

## Publications

### Revue internationale :

1. Al Chami, Z., Manier, H., Manier, M.-A., & Chebib, E.. "An advanced GRASP-HGA combination to solve a multi-period Pickup and Delivery Problem",

*Expert Systems with Applications*, Volume 105, 2018, Pages 262-272, doi : 10.1016/j.eswa.2018.03.040.

2. Al Chami, Z., Manier, H., & Manier, M.-A.. "Multi-Period Pickup and Delivery Problem with Time Windows and Paired Demands", *Computers & Industrial Engineering*. (soumis)

#### Conférence internationale :

1. Manier, H., Manier, M.-A., & Al Chami, Z.. "Shippers' collaboration in city logistics", *8th IFAC/IEEE/IFORS/IFIP Conference on Manufacturing Modelling Management and Control (MIM 2016)*, IFAC-PapersOnLine, Volume 49, Issue 12, pp. 1880-1885, Troyes, France (june 28-30, 2016).

#### Conférence nationale :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "Différentes approches pour la résolution d'un problème multipériodes de collecte et de livraison". *Dix-neuvième Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'18)*, Lorient, France (21-23 février 2018).

#### Nouvelles instances multi-périodes :

Les nouvelles instances générées dans ce chapitre sont accessibles via ce lien : <https://drive.google.com/drive/folders/1c7cZAvEDtNDj1oVK1Af0HcWPnMj2EkjF>.

## 4.1 Introduction

Au chapitre 2, nous avons introduit une nouvelle variante de problème de tournées qui est le SPDPTWPD. Elle nécessite de choisir un sous-ensemble des sites à visiter quand il est impossible de satisfaire toutes les demandes sur une période donnée. Pour réussir à livrer toutes les demandes, il est alors nécessaire de considérer plusieurs périodes successives où pour chaque période l'on essaiera de satisfaire les demandes non encore honorées dans les périodes précédentes. Mais livrer sur la période suivante peut signifier être en retard, c'est pourquoi pour éviter le retard il est possible de définir pour une demande l'ensemble des périodes pendant lesquelles il est possible de livrer celle-ci, avec un système de pénalités pour indiquer une certaine priorité entre ces périodes.

Les études citées aux chapitres précédents (dédiées à la résolution du PDP et ses variantes) traitent le problème dans un contexte mono-période. Néanmoins des chercheurs ont abordé l'aspect multi-périodes mais sur des problèmes différents du PDP. De plus, plusieurs travaux ont traité les problèmes de tournées périodiques (où un site doit être visité plusieurs fois sur un ensemble de périodes) et les problèmes avec fenêtres temporelles multiples (où dans une période donnée, un site associé avec plusieurs fenêtres temporelles doit être visité une seule fois). Le tableau 4.1 résume les différences entre ces trois types de variantes. Le tableau 4.2 présente une liste non exhaustive de ces études. Nous remarquons que les tournées périodiques et multi-périodes sont moins étudiées que les problèmes avec fenêtres temporelles multiples.

Aspect	Nombre de périodes	Nombre de visites pour chaque site	Nombre de fenêtres temporelles pour chaque site
Fenêtres temporelles multiples	Une période	Une seule visite	Plusieurs fenêtres
Multi-périodes	Plusieurs périodes	Une seule visite	Une seule fenêtre par période acceptée
Périodique	Plusieurs périodes	Plusieurs visites	Une seule fenêtre par période acceptée

TABLE 4.1 – Différences entre les variantes considérées

Référence	Problème de routage	Multi-périodes	Fenêtres temporelles multiples	Périodique
[Gaudioso & Paletta 1992]	VRP			X
[Favaretto <i>et al.</i> 2007]	VRP		X	
[Tricoire <i>et al.</i> 2010]	OP	X	X	
[Qiu-ai & Zhen-ping 2012]	VRP		X	
[Mesquita <i>et al.</i> 2013]	TSP		X	
[Yao <i>et al.</i> 2013]	VRP			X
[Souffriau <i>et al.</i> 2013]	TOP		X	
[Belhaiza <i>et al.</i> 2014]	VRP		X	
[Liu <i>et al.</i> 2014b]	PDP			X
[Dayarian <i>et al.</i> 2015]	VRP	X		
[Paulsen <i>et al.</i> 2015]	TSP		X	
[Lin & Vincent 2015]	TOP		X	
[Mancini 2016]	VRP	X		
[Dayarian <i>et al.</i> 2016]	VRP	X		
[Liao <i>et al.</i> 2017]	VRP			X

TABLE 4.2 – Récapitulatif des études sur les variantes considérées

Dans ce chapitre, nous introduisons une nouvelle variante du PDP qui, à notre connaissance, n'a pas encore été traitée dans la littérature. Il s'agit du problème de collecte et livraison multi-périodes avec fenêtres temporelles et demandes appairées, soit le Mu-PDPTWPD (pour Multi-period Pickup and Delivery Problem with Time Windows and Paired Demands). L'aspect multi-périodes est ajouté au problème que nous avons précédemment étudié, dans le but de mieux correspondre aux applications réelles dans lesquelles tous les sites doivent être visités une fois pendant un ensemble des périodes.

Dans cette nouvelle variante, il existe plusieurs périodes au cours desquelles les véhicules effectuent les opérations de livraison ou de ramassage. Une période peut être définie comme un jour ou tout autre intervalle de temps. Chaque site peut accepter d'être visité en une ou plusieurs périodes avant la période prévue de sa visite. Toutefois, une visite anticipée entraîne des coûts supplémentaires qui seront pris en compte sous forme de pénalités. Ce même principe est appliqué sur les périodes ultérieures, avec également un jeu de pénalités. Par rapport au problème de routage mono-période, la variante multi-périodes apporte un défi supplémentaire : l'affectation d'une période pour chaque site à visiter.

Le Mu-PDPTWPD appartient également à la classe des problèmes NP-difficiles puisqu'il représente une extension du PDP. Ses caractéristiques sont les suivantes :

- Une flotte de véhicules hétérogènes à capacité limitée est basée au dépôt.
- Un site peut accepter d'être visité pendant une ou plusieurs périodes. D'où la nécessité de définir pour ce site une fenêtre de temps par période acceptée. Il doit être visité au cours d'une de ces période(s).
- L'ensemble des demandes de transport doit être satisfait sur toutes les périodes à condition que chaque demande soit satisfaite dans une période donnée (nous ne prenons pas en compte la possibilité de diviser une demande sur plusieurs périodes).

- (d) Le dépôt est une plate-forme logistique située en périphérie de la ville, où tous les véhicules commencent et terminent leurs tournées. En outre, il reçoit des marchandises de différents fournisseurs situés en dehors de la ville qui seront ensuite livrées dans la zone urbaine. Il reçoit également des marchandises de différents fournisseurs situés à l'intérieur de la ville afin de les envoyer dans d'autres villes. Ainsi, le dépôt peut représenter l'origine ou la destination pour une ou plusieurs demandes de transport. Sinon, l'origine sera un autre fournisseur et la destination sera un autre client situés à l'intérieur de la ville.
- (e) Les contraintes de précédence imposent que chaque client soit visité après son fournisseur à la fois par le même véhicule et au cours de la même période.

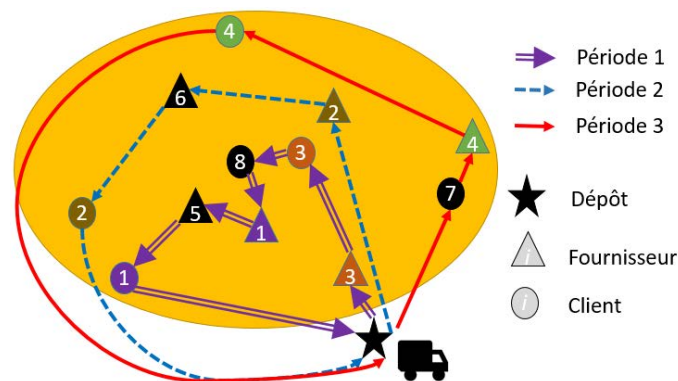


FIGURE 4.1 – Exemple de Mu-PDPTWPD

La figure 4.1 montre un exemple de Mu-PDPTWPD avec trois périodes, un dépôt unique représenté par une étoile noire, un véhicule unique et un ensemble de nœuds composé de six clients (cercles) et de six fournisseurs (triangles). Une demande de transport donnée est identifiée grâce à un numéro et une couleur associés aux nœuds fournisseur et client correspondants. Dans la figure 4.1, nous pouvons voir des nœuds qui correspondent à des clients ou des fournisseurs associés au dépôt. Ici, les cercles noirs représentent les clients ayant le dépôt en tant que fournisseur et les triangles noirs représentent les fournisseurs dont le client est le dépôt.

Le reste de ce chapitre sera organisé comme suit : dans la deuxième section, nous proposons une formulation mathématique du Mu-PDPTWPD. Nous étudions cette variante dans un contexte mono-objectif dans la troisième section. Le cas multi-objectifs fera l'objet de la quatrième section. La cinquième section resituera l'apport potentiel des deux variantes étudiées dans le contexte de la collaboration entre chargeurs.

## 4.2 Formulation mathématique

### 4.2.1 Notations

Le Mu-PDPTWPD est caractérisé par les paramètres suivants :

Données :

- P : Ensemble des périodes,
- C : Ensemble des clients,
- B : Ensemble des fournisseurs,
- Nodes = C ∪ B : Ensemble des nœuds,
- N = Nodes ∪ 0 : Ensemble de tous les sites plus le dépôt avec l'identifiant 0,
- $Supplier_i$  : Fournisseur du client i,
- $Customer_i$  : Client du fournisseur i,
- $q_i$  : Quantité demandée par le site i
  - Si  $q_i > 0$  alors  $i \in B$ ,
  - Si  $q_i < 0$  alors  $i \in C$ ,
- $[e_{(i,t)}, l_{(i,t)}]$  : Fenêtre temporelle associée au site i dans la période t,
- $ST_i$  : Temps du service au site i,
- $Penalty_{i,t}$  : Pénalité associée à la visite du site i pendant la période t,
- M : Très grand nombre,
- V : Ensemble des véhicules disponibles,
- $Q_k$  : Capacité maximale du véhicule k,
- $Speed_k$  : Vitesse moyenne du véhicule k,
- $d_{ij}$  : Distance du site i au site j.

Variables :

- $X_{i,j,k,t} = \begin{cases} 1 & \text{Si le site j est directement visité après le site i par le véhicule k} \\ & \text{pendant la période t} \\ 0 & \text{Sinon} \end{cases}$
- $A_{i,k,t}$  : Date de début de service du véhicule k au site i pendant la période t,
- $D_{i,k,t}$  : Date de départ du véhicule k du site i pendant la période t,
- $Y_{i,k,t}$  : Quantité de marchandises dans le véhicule k visitant le site i pendant la période t,
- $E_{i,t} = \begin{cases} 1 & \text{Si le site i est visité pendant la période t} \\ 0 & \text{Sinon} \end{cases}$

#### 4.2.2 MILP pour le Mu-PDPTWPD

La formulation mathématique du problème étudié peut être la suivante :

$$\text{Minimize/Maximize } F \quad (4.1)$$

Sous contraintes :

$$\sum_{i \in N} \sum_{k \in V} \sum_{t \in P} X_{i,j,k,t} = 1 \quad j \in Nodes \quad (4.2)$$



$$\sum_{j \in N} \sum_{k \in V} \sum_{t \in P} X_{i,j,k,t} = 1 \quad i \in Nodes \quad (4.3)$$

$$\sum_{i \in N} X_{i,u,k,t} - \sum_{i \in N} X_{u,i,k,t} = 0 \quad k \in V, u \in Nodes, t \in P \quad (4.4)$$

$$\sum_{i \in Nodes} X_{0,i,k,t} \leq 1 \quad k \in V, t \in P \quad (4.5)$$

$$\sum_{i \in N} \sum_{j \in N} X_{i,j,k,t} \leq M * \sum_{i \in Nodes} X_{0,i,k,t} \quad k \in V, t \in P \quad (4.6)$$

$$\sum_{j \in Nodes} X_{j,0,k,t} = \sum_{i \in Nodes} X_{0,i,k,t} \quad k \in V, t \in P \quad (4.7)$$

$$D_{0,k,t} = 0 \quad k \in V, t \in P \quad (4.8)$$

$$D_{i,k,t} \geq A_{i,k,t} + S_i - M * (1 - \sum_{j \in N} X_{i,j,k,t}) \quad i \in Nodes, k \in V, t \in P \quad (4.9)$$

$$D_{i,k,t} - M * (1 - \sum_{j \in N} X_{i,j,k,t}) \leq A_{i,k,t} + S_i \quad i \in Nodes, k \in V, t \in P \quad (4.10)$$

$$D_{i,k,t} + \frac{d_{ij}}{Speed_k} - M * (1 - X_{i,j,k,t}) \leq A_{j,k,t} \quad i \in N, j \in N, k \in V, t \in P \quad (4.11)$$

$$D_{i,k,t} \leq M * \sum_{j \in N} X_{i,j,k,t} \quad i \in Nodes, k \in V, t \in P \quad (4.12)$$

$$A_{i,k,t} \leq M * \sum_{j \in N} X_{j,i,k,t} \quad i \in Nodes, k \in V, t \in P \quad (4.13)$$

$$A_{0,k,t} \leq l_{(0,t)} \quad k \in V, t \in P \quad (4.14)$$

$$\sum_{t \in P} E_{i,t} = 1 \quad i \in Nodes \quad (4.15)$$

$$E_{i,t} = \sum_{j \in N} \sum_{k \in V} X_{j,i,k,t} \quad i \in Nodes, t \in P \quad (4.16)$$

$$\sum_{k \in V} A_{i,k,t} - M * (1 - E_{i,t}) \leq l_{(i,t)} \quad i \in Nodes, t \in P \quad (4.17)$$

$$e_{(i,t)} \leq \sum_{k \in V} A_{i,k,t} + M * (1 - E_{i,t}) \quad i \in Nodes, t \in P \quad (4.18)$$

$$Y_{0,k,t} = \sum_{i \in C / Supplier_i=0} (-q_i * \sum_{j \in N} X_{j,i,k,t}) \quad k \in V, t \in P \quad (4.19)$$

$$Y_{i,k,t} + q_j - M * (1 - X_{i,j,k,t}) \leq Y_{j,k,t} \quad i \in N, j \in Nodes, k \in V, t \in P \quad (4.20)$$

$$Y_{j,k,t} - q_j - M * (1 - X_{i,j,k,t}) \leq Y_{i,k,t} \quad i \in N, j \in Nodes, k \in V, t \in P \quad (4.21)$$

$$Y_{i,k,t} \leq M * \sum_{j \in N} X_{j,i,k,t} \quad i \in N, k \in V, t \in P \quad (4.22)$$

$$0 \leq Y_{i,k,t} \leq Q_k \quad i \in N, k \in V, t \in P \quad (4.23)$$

$$\sum_{u \in N} X_{Supplier_i, u, k, t} = \sum_{u \in N} X_{u, i, k, t} \quad i \in C / Supplier_i \in B, k \in V, t \in P \quad (4.24)$$

$$D_{Supplier_i, k, t} \leq A_{i, k, t} \quad i \in C, k \in V, t \in P \quad (4.25)$$

L'équation (4.1) représente notre fonction objectif. Les contraintes (4.2) et (4.3) garantissent que chaque nœud doit être visité une fois pendant l'ensemble des périodes. La continuité du flux pour chaque véhicule et pour chaque période est assurée par les contraintes (4.4). Les contraintes (4.5), (4.6) et (4.7) garantissent que chaque tournée doit commencer et finir à un dépôt. Les contraintes (4.8) assurent que chaque route de véhicule commence sur chaque période à l'instant 0. Les contraintes (4.9), (4.10), (4.11), (4.12) et (4.13) sont utilisées pour mettre à jour la date de début de service et celle de départ. Les contraintes (4.14) assurent que tous les véhicules doivent terminer leurs déplacements avant la fermeture du dépôt. Les contraintes (4.15) et (4.16) permettent que chaque nœud soit visité pendant une seule période. La conformité des fenêtres temporelles est assurée par les contraintes (4.17) et (4.18). Les contraintes (4.19), (4.20), (4.21), (4.22) et (4.23) garantissent que la capacité propre à chaque véhicule est respectée. Les contraintes (4.24) et (4.25) garantissent que chaque véhicule doit visiter un fournisseur avant son client.

### 4.3 Étude mono-objectif

Le but de cette section est d'étudier le Mu-PDPTWPD dans un contexte mono-objectif. Pour cela, nous introduisons deux stratégies. La première consiste à traiter le problème dans son ensemble. La deuxième consiste à considérer le problème étudié comme une séquence de SPDPTWPD mono-période.

#### 4.3.1 Méthode exacte : MILP-Global

En nous basant sur le MILP présenté dans la section 4.2, notre méthode exacte consiste à implémenter ce modèle en utilisant le solveur commercial CPLEX 12.4. Puisque toutes les demandes doivent être satisfaites, le profit est directement obtenu par la somme des profits de tous les sites à visiter. Et si toutes les demandes ne sont pas satisfaites, alors le modèle linéaire n'a pas de solutions. Par conséquent, la fonction objectif utilisée ici ne peut être que la minimisation de la distance parcourue sur l'ensemble des périodes (F3) :  $\sum_{i \in N} \sum_{j \in N} \sum_{k \in V} \sum_{t \in P} d_{ij} X_{i,j,k,t}$ .

Dans le but d'améliorer les performances de notre méthode exacte, nous avons ajouté les cinq contraintes additionnelles présentées dans la sous-section 2.3.2 en les adaptant à notre nouveau contexte (existence de plusieurs périodes). De plus, nous avons utilisé une sixième contrainte additionnelle, associée au manque de fenêtre temporelle pendant une période :

$$Si e_{(i,t)} = l_{(i,t)} \Rightarrow E_{i,t} = 0 \quad \forall i \in Nodes, \forall t \in P \quad (C4)$$

Les contraintes (C4) expriment que la visite d'un site pendant une période donnée est impossible s'il n'y a pas une fenêtre temporelle associée à ce site pendant cette période.

Ce MILP (que l'on notera MILP-Global) aborde le problème dans sa globalité. Par conséquent, la complexité en est très élevée. CPLEX ne peut résoudre que de très petites instances à l'optimalité dans un délai raisonnable.

Pour pallier l'inconvénient lié au temps de résolution, nous avons élaboré différentes approches associées à la seconde stratégie évoquée pour résoudre le Mu-PDPTWPD.

### 4.3.2 Approches séquentielles

Considérant la limitation présentée par notre méthode exacte, nous avons développé et testé trois approches séquentielles basées sur la deuxième stratégie déjà mentionnée. Avec la première stratégie, en plus de la complexité liée au SPDPTWPD, on rajoute le problème d'affectation de la demande à une période. Pour la résoudre de façon optimale, il faut donc envisager toutes les combinaisons possibles d'affectation des demandes aux périodes, ce qui par exemple dans le cas où les  $n$  demandes peuvent être réalisées sur deux périodes implique d'envisager  $2^n$  PDPTWPD à résoudre. Pour éviter cette explosion combinatoire, il est alors possible d'appréhender le problème différemment.

Le principe de base adopté pour nos trois approches séquentielles consiste à décomposer le problème multi-périodes en plusieurs sous-problèmes sélectifs mono-période qui vont être résolus de manière séquentielle. Il s'agit de méthodes approchées qui fournissent des solutions dont l'optimalité n'est pas garantie (la somme d'optimaux locaux ne donne pas forcément un optimum global).

À une période  $n$  (appelée période actuelle), l'ensemble des demandes considérées est composé des demandes répondant aux caractéristiques suivantes :

- non satisfaites aux  $n-1$  périodes précédentes.
- pouvant être réalisées dans la période actuelle.

Le SPDPTWPD actuel à résoudre dépend alors des résultats des SPDPTWPD résolus aux  $n-1$  périodes précédentes puisqu'ils influent sur l'ensemble des demandes à considérer en période  $n$ . Par ailleurs, au cours de cette période actuelle, les véhicules ne seront pas toujours en mesure de répondre à toutes les demandes considérées en raison des contraintes à respecter. Cela explique que nous devons résoudre un problème sélectif. Enfin, après avoir résolu séquentiellement tous les problèmes sélectifs mono-période, la solution finale sera réalisable si toutes les demandes sont satisfaites. Notons cependant que, de manière générale, l'inconvénient du traitement séquentiel du problème multi-périodes est que l'obtention d'une solution faisable n'est pas garantie, même si cette solution existe. En termes de complexité, la résolution d'une variante multi-périodes avec  $T$  périodes est équivalente à la résolution de  $T^n$  fois la variante mono-période (si les  $n$  demandes peuvent être satisfaites à n'importe quelle période).

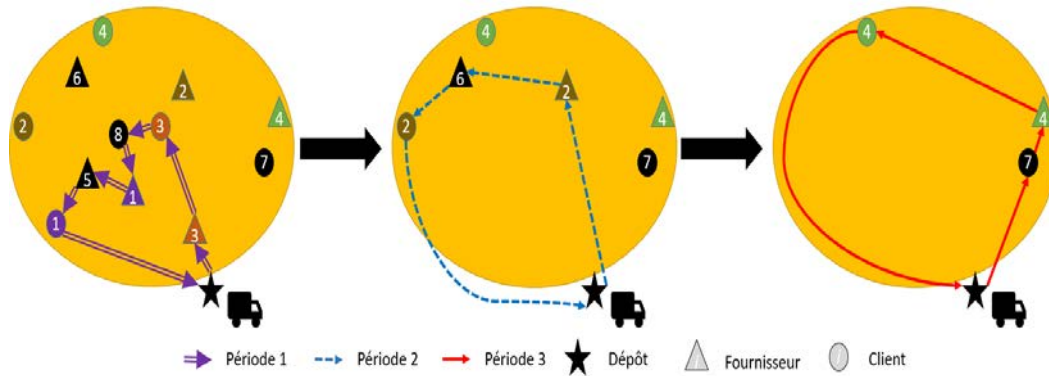


FIGURE 4.2 – Les approches séquentielles

La figure 4.2 illustre le même problème que dans la figure 4.1, mais étudiée comme trois problèmes sélectifs séquentiels mono-période (Mo-SPDPTWPD). Dans la première période, les nœuds numérotés 2, 4, 6 et 7 n'ont pas été sélectionnés, soit parce qu'ils n'ont pas accepté d'être visités dans cette période, soit parce que le véhicule ne peut pas visiter ces nœuds après avoir servi les autres. Dans la deuxième période, nous remarquons que les nœuds servis dans la première période ont disparu et nous pouvons maintenant choisir entre les nœuds restants et ainsi de suite. Dans cet exemple, nous avons résolu la séquence et nous avons obtenu une solution optimale car nous avons les mêmes routes que dans la figure 4.1, ce qui n'est pas toujours le cas.

#### 4.3.2.1 Approche MILP-Séquentiel

La figure 4.3 représente notre première approche séquentielle, notée MILP-Séquentiel, qui consiste à traiter chronologiquement chaque période. Nous commençons donc avec la première période et exécutons notre MILP amélioré développé pour le Mo-SPDPTWPD en sous-section 4.3.1 pour obtenir la solution de profit optimal sur cette période. Après cela, nous calculons la distance obtenue afin d'ajouter sa valeur à la distance finale et nous insérons les sites visités dans la liste des sites servis (ces nœuds ne peuvent plus être visités sur les périodes suivantes). Finalement, nous résolvons le problème pour la période suivante et nous répétons ce processus pour toutes les périodes, tant que toutes les demandes ne sont pas satisfaites.

Dans le Mu-PDPTWPD global, l'objectif considéré est la minimisation de la distance totale. Mais ici nous ne pouvons pas utiliser la minimisation de distance ou F2 comme fonction objectif sur chaque période en raison de l'aspect sélectif de chaque problème mono-période. En effet, dans un problème sélectif, la distance minimale est zéro avec aucun site visité. Par conséquent, nous utilisons la maximisation de profit ou F1 comme fonction objectif.

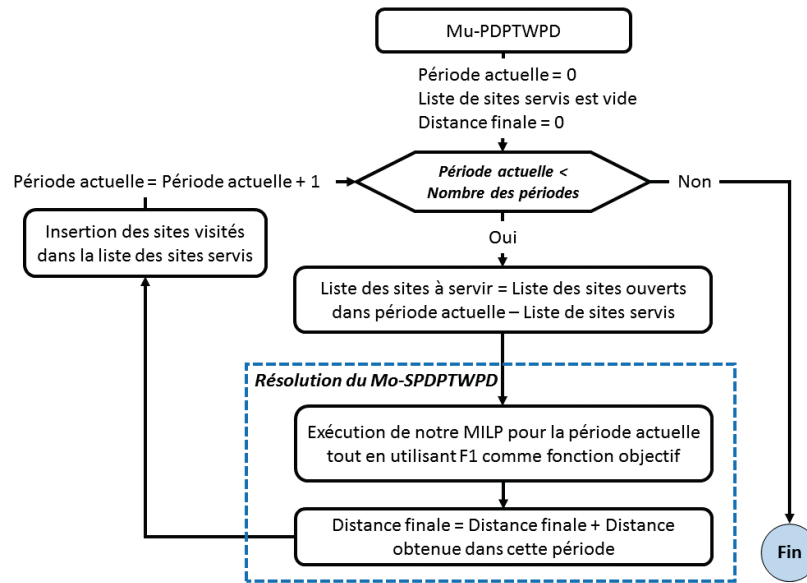


FIGURE 4.3 – MILP-Séquentiel

Avec notre approche MILP-séquentiel, la distance obtenue peut ne pas être optimale, ni à chaque période ni globalement. Pour remédier à cet inconvénient et améliorer les performances de notre stratégie séquentielle, nous développons dans la suite deux méthodes séquentielles en nous basant sur une résolution multi-objectifs pour chaque problème mono-période.

#### 4.3.2.2 Approche Lexicographique-Séquentiel

La deuxième approche séquentielle notée Lexicographique-Séquentiel consiste à résoudre chaque problème Mo-SPDPTWPD en utilisant l'approche lexicographique développée dans le chapitre précédent. Le fait d'utiliser une approche multi-objectifs qui minimise la distance après la maximisation du profit nous permet d'obtenir une meilleure distance à chaque période et donc nous pourrions nous rapprocher de la meilleure distance globale. La figure 4.4 montre le principe de cette approche.

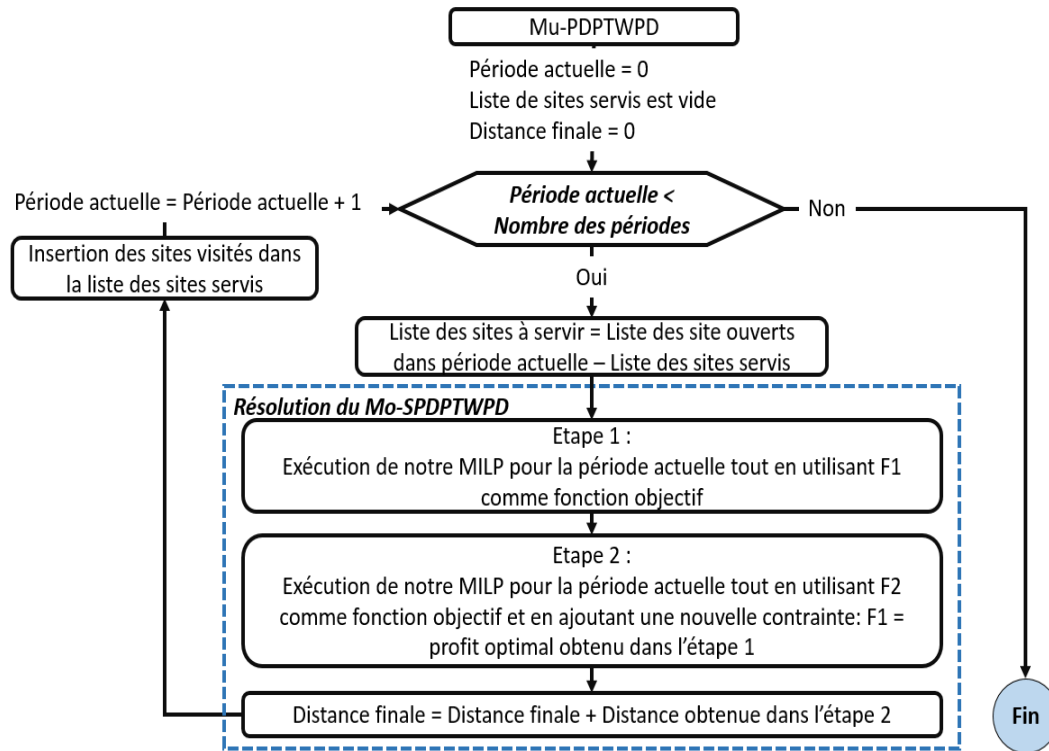


FIGURE 4.4 – Lexicographique-Séquentiel

Bien que cette deuxième approche améliore la valeur de la distance finale obtenue par rapport à la première approche, elle présente encore un gros inconvénient. Ce dernier réside dans le fait que le nombre des MILPs qui doivent être exécutés est multiplié par deux, ce qui implique un temps de résolution plus élevé.

#### 4.3.2.3 Approche HGA-Séquentiel

La dernière approche séquentielle est notée HGA-Séquentiel (pour Hybrid Genetic Algorithm séquentiel). Le but de cette approche est de traiter chaque SPDPTWPD mono-période dans un contexte multi-objectifs tout en trouvant rapidement des solutions pour chaque période.

Pour cela, nous allons utiliser notre algorithme génétique hybride développé en sous-section 3.3. Cependant, il y a deux choix à faire. Le premier est lié aux réglages des paramètres pour assurer le bon fonctionnement de notre algorithme génétique. Concernant ce premier choix, nous avons testé plusieurs combinaisons des paramètres. Enfin notre choix s'est arrêté sur les valeurs montrées dans la figure 4.5, soit  $\text{Max} = 200$ ,  $\text{CP} = 15$ , maximum nombre d'itérations = 10,  $\alpha = 1$  et  $\sigma_{sh} = 150$  (ces valeurs ont été discutées aussi au chapitre 3). Le deuxième choix à faire est de sélectionner la solution avec laquelle nous allons passer à la période suivante vu que l'algorithme génétique nous donne un ensemble de solutions à la fin de la résolution. Ici, nous avons décidé de choisir la solution non-dominée avec le plus grand profit parce que

elle représente généralement la solution avec le plus grand nombre de demandes satisfaites. Notre but est de trouver une solution faisable à notre Mu-PDPTWPD, c'est à dire que toutes les demandes ont été réalisées. On espère ainsi par ce choix augmenter la probabilité d'obtenir une solution faisable. La figure 4.5 décrit les différentes étapes de notre approche "HGA-Séquentiel".

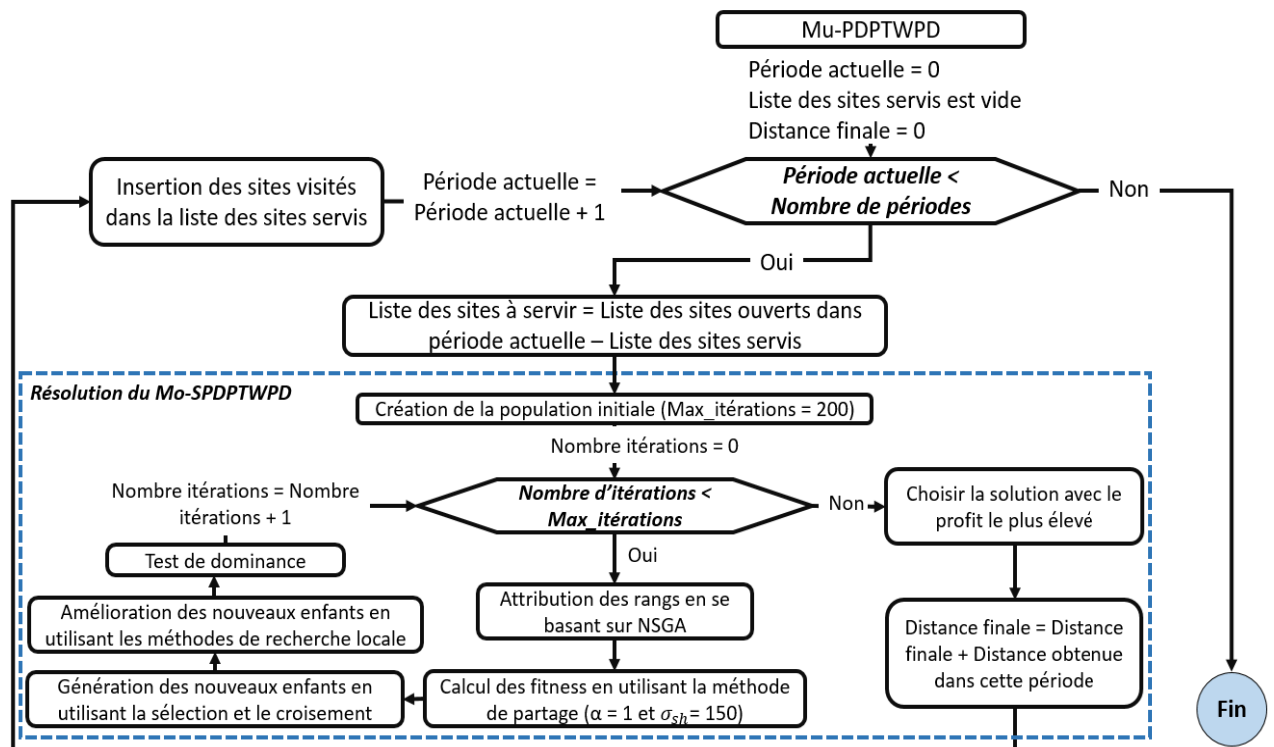


FIGURE 4.5 – HGA-Séquentiel

### 4.3.3 Résultats expérimentaux

Dans cette sous-section, nous donnons les résultats expérimentaux obtenus en exécutant les quatre méthodes que nous avons développées pour résoudre le Mu-PDPTWPD : MILP-Global, MILP-Séquentiel, Lexicographique-Séquentiel et HGA-Séquentiel. Pour les trois premières, nous utilisons le solveur commercial CPLEX 12.4. Pour le HGA-Séquentiel, nous avons codé les algorithmes en C++. Tous les tests ont été effectués en utilisant un ordinateur avec un processeur Intel Core i7, 2.80 GHz et 16 Go de RAM avec Windows 10 comme système d'exploitation. Les tests ont été réalisés sur de nouvelles instances que nous avons générées car, à notre connaissance, aucune instance dédiée au problème considéré n'est disponible dans la littérature. Nous proposons deux ensembles de nouvelles instances :

1. Ensemble 1 : composé de 30 instances générées aléatoirement avec un nombre fixe de périodes égal à 10. La particularité de cet ensemble est qu'un site peut choisir la ou les périodes dans lesquelles il peut être visité. Cela signifie que

pour chaque site, il peut y avoir plusieurs fenêtres temporelles associées mais au plus une fenêtre par période. Dans cet ensemble, le même profit est associé à toutes les demandes.

2. Ensemble 2 : composé de 18 instances générées à partir d'instances introduites dans [Li & Lim 2003] que nous avons adaptées au Mu-PDPTWPD. Dans cet ensemble, tous les sites acceptent d'être visités à n'importe quelle période, ce qui signifie que pour chaque site il y a exactement une fenêtre temporelle associée par période. De plus, les profits peuvent être différents suivant la demande. Notons que pour l'ensemble 2, nous avons déterminé le nombre de périodes nécessaires pour chaque instance en utilisant le nombre minimum de périodes requises par nos trois approches séquentielles (voir colonne 2 du tableau 4.3), en répétant la procédure de résolution jusqu'à la visite de tous les sites.

Chaque instance est étiquetée **S-x-y-z-i**, x est le numéro d'ensemble, y est le type d'instance (R : les nœuds sont distribués aléatoirement, C : les nœuds sont totalement groupés et RC : les nœuds sont partiellement groupés et partiellement distribués aléatoirement), z est le nombre de nœuds et i identifie le numéro de l'instance. Des informations complémentaires ainsi que les données d'instances sont disponibles en annexe A.2 et à l'adresse : <https://drive.google.com/drive/folders/1c7cZAvEDtNDj1oVK1Af0HcWPnMj2EkjF>.

Le tableau 4.3 montre les résultats pour les approches considérées. Pour chacune, nous fournissons la distance totale obtenue et le temps CPU en secondes. Le nombre de périodes (NP) est représenté par la deuxième colonne. Pour les instances avec 50 nœuds, le MILP-Global a résolu à l'optimalité seulement une instance sur dix de l'ensemble 1 et trois instances sur six de l'ensemble 2. Les solutions avec le "\*" et "\*\*" correspondent respectivement aux meilleures solutions faisables et meilleures bornes obtenues par l'approche MILP-Global après deux heures de résolution. Le terme "Time limit" signifie que l'approche n'a pas réussi à trouver une solution faisable (même pas une borne réelle) pour les instances correspondantes après deux heures de fonctionnement.

La différence entre le nombre d'instances résolues peut s'expliquer par le fait que le nombre de périodes dans l'ensemble 2 peut être plus petit que celui de l'ensemble 1 (qui est fixé à 10). Pour l'instance S1-R-20-5, le MILP-Séquentiel et le Lexicographique-Séquentiel nous donnent des solutions inacceptables qui consistent à ne visiter que 19 sites sur les 20 (apparaissant en gris dans le tableau). Sauf dans ce cas, tous les résultats présentés dans le tableau 4.3 font référence à des solutions réalisables.



Instance	NP	MILP-Global		MILP-Séquentiel		Lexicographique-Séquentiel		HGA-Séquentiel	
		Distance	Temps CPU (s)	Distance	Temps CPU (s)	Distance	Temps CPU (s)	Distance	Temps CPU (s)
S1-R-10-1	10	910.659	3.2	991.456	1.34	910.659	2.21	910.659	0.96
S1-R-10-2	10	657.043	5.67	930.628	2.21	930.628	3.45	930.628	0.46
S1-R-10-3	10	821.459	6.98	821.459	3.14	821.549	4.16	821.459	1.05
S1-R-10-4	10	912.905	6.45	1039.51	2.19	1039.51	3.23	1039.51	1.24
S1-R-10-5	10	843.879	6.37	914.891	2.13	914.891	3.15	914.891	1.21
S1-R-10-6	10	930.679	6.87	1011.39	2.57	1011.39	3.72	1011.39	1.45
S1-R-10-7	10	897.632	6.49	1031.62	2.25	1031.62	3.31	1031.62	1.34
S1-R-10-8	10	759.548	7.13	1026.88	2.36	955.985	3.4	955.985	1.51
S1-R-10-9	10	938.955	7.04	1015.72	2.33	1015.72	3.51	1015.72	1.57
S1-R-10-10	10	803.484	7.17	893.013	2.41	893.013	3.53	893.013	1.62
S1-R-20-1	10	1814.512	10.73	1986.65	7.23	1934.25	9.13	1903.12	9.53
S1-R-20-2	10	1706.712	8.95	2016.43	4.25	1855.51	7.21	1855.51	6.15
S1-R-20-3	10	1829.728	6.35	2137.83	4.23	2010.9	5.32	2010.9	4.12
S1-R-20-4	10	1493.822	11	1830.17	5.13	1573.07	8.97	1589.89	5.72
S1-R-20-5	10	1667.67	9.33	1781.38	8.76	1580.91	15.34	1784.01	7.29
S1-R-20-6	10	1643.006	11.64	1993.01	6.21	1724.93	9.32	1724.93	8.2
S1-R-20-7	10	1547.684	6.76	1929.81	6.12	1870.6	14.52	1889.09	5.27
S1-R-20-8	10	1635.755	5.82	1857.75	3.71	1672.39	5.03	1672.39	3.52
S1-R-20-9	10	1605.324	6.74	1938.1	3.51	1807.07	5.4	1815.19	4.16
S1-R-20-10	10	1524.8	6.1	1719.11	3.2	1696.7	5.7	1696.7	2.23
S1-R-50-1	10	3242.77*	7200	4967.02	173.122	3814.23	612.84	3834.33	76.249
S1-R-50-2	10	3217.59*	7200	4497.41	250.266	3626.87	701.44	3672.24	85.661
S1-R-50-3	10	3397.16*	7200	4612.08	117.089	4029.62	436.43	4029.62	58.269
S1-R-50-4	10	3385.38*	7200	4262.57	135.515	3734.91	472.85	3758.32	60.311
S1-R-50-5	10	3129.47*	7200	4589.05	164.35	3748.56	542.24	3748.56	70.534
S1-R-50-6	10	3169.07*	7200	4528.73	128.641	3410.08	452.23	3426.08	56.874
S1-R-50-7	10	3201.02*	7200	4516.64	180.295	3474.27	637.15	3520.86	78.214
S1-R-50-8	10	3169.13*	7200	4539.33	107.972	3474.69	422.72	3494.81	80.561
S1-R-50-9	10	2996.748	4259.32	4267.47	129.265	3627.01	467.68	3686.43	58.721
S1-R-50-10	10	3286.35*	7200	4288.83	114.74	3783.69	435.29	3796.12	62.968
S2-C-10-1	1	58.7243	3.16	58.7243	1.19	58.7243	2.53	58.7243	1.1
S2-C-10-2	1	59.4222	17.18	63.8216	10.12	59.4222	16.42	59.4222	4.6
S2-R-10-1	5	270.925	2.13	290.643	0.87	290.643	1.58	290.643	0.09
S2-R-10-2	3	226.993	13.42	294.437	5.13	294.437	11.71	294.437	3.14
S2-RC-10-1	2	193.085	2.04	193.085	0.45	193.085	1.65	193.085	0.23
S2-RC-10-2	2	195.147	3.21	195.147	0.93	195.147	1.97	195.147	0.76
S2-C-20-1	2	204.177	5.13	484.5	2.3	356.868	3.7	347.429	2.15
S2-C-20-2	2	212.947	42.12	232.838	21.88	221.31	41.75	249.7233	16.23
S2-R-20-1	2	449.999	4.81	548.814	1.9	479.247	2.74	455.168	2.15
S2-R-20-2	2	400.301	140.51	491.133	14.53	451.07	29.27	406.317	13.89
S2-RC-20-1	2	323.981	3.63	399.931	1.31	323.981	2.59	323.981	2.21
S2-RC-20-2	2	327.786	5.12	405.616	1.87	327.786	3.54	327.786	2.54
S2-C-50-1	2	435.339	192.13	523.752	55.452	435.339	167.458	610.244	14.78
S2-C-50-2	2	525.871	305.49	1034.05	206.12	1025.55	258.889	909.847	26.58
S2-R-50-1	2	1125.31	1024.17	1158.24	131.505	1156.77	864.67	1138.93	181.328
S2-R-50-2	2	1122.95*	7200	Time limit	Time limit	Time limit	Time limit	1208.04	81.08
S2-RC-50-1	2	699.59**	7200	Time limit	Time limit	Time limit	Time limit	920.245	71.372
S2-RC-50-2	3	677.21**	7200	Time limit	Time limit	Time limit	Time limit	969.247	163.378

TABLE 4.3 – Résultats expérimentaux pour le Mu-PDPTWPD mono-objectif

Pour comparer les quatre approches proposées, le tableau 4.4 synthétise l'écart moyen de distance et de temps entre nos trois approches séquentielles et l'approche globale. Il fournit également les écarts-types associés. Ces écarts sont calculés sur la base des 46 instances résolues en utilisant l'approche globale dont 36 instances résolues à l'optimalité. Nous n'avons pas pris en compte les cas où les approches MILP-Séquentiel et Lexicographique-Séquentiel ne fournissent pas des solutions réalisables (S1-R-20-5, S2-R-50-2, S2-RC-50-1 et S2-RC-50-2). De plus, ce tableau donne le nombre total d'instances résolues pour chaque approche sur les 48 instances testées.

Approche	Écart moyen de distance (%)	Écart-type moyen de distance (%)	Écart moyen de temps (%)	Écart-type moyen de temps (%)	Nombre d'instances résolues sur 48
MILP-Global	0	0	0	0	46
MILP-Séquentiel	25.34	25.08	-66.31	22.35	44
Lexicographique-Séquentiel	13.38	18.21	-40.26	38.97	44
HGA-Séquentiel	13.78	16.28	-72.85	24.74	48

TABLE 4.4 – Analyse des résultats obtenus

Le tableau 4.4 donne une idée claire des avantages et des inconvénients de l'utilisation de chaque approche. Le MILP-Global peut être considéré comme la meilleure approche car il fournit une solution optimale pour le Mu-PDPTWPD mais le nombre d'instances résolues à l'optimalité met en évidence ses limites surtout pour les instances avec 50 nœuds. Les trois approches séquentielles ne fournissent pas de solutions optimales et cela peut être justifié par le fait que la somme des solutions locales optimales peut ne pas donner la solution globale optimale. Mais nous pouvons observer que nous minimisons le temps de résolution. Nous remarquons aussi que l'écart moyen obtenu en utilisant le Lexicographique-Séquentiel et le HGA-Séquentiel est presque égal pour la distance. De plus, l'écart-type utilisant HGA-Séquentiel est plus petit que celui avec Lexicographique-Séquentiel, ce qui signifie que les valeurs de distance sont moins dispersées autour de la moyenne. En outre, le HGA-Séquentiel peut résoudre toutes les instances en très peu de temps. Ainsi, nous pouvons le considérer comme étant meilleur que les autres approches séquentielles.

Maintenant, nous étudions l'effet du nombre de périodes sur les résultats obtenus. Dans ce but, nous utilisons les instances de l'ensemble 2 où le nombre de périodes n'est pas une constante de notre problème. Comme nous l'avons mentionné précédemment, le nombre de périodes utilisées pour obtenir la solution dans le tableau 4.3 est égal au nombre de périodes nécessaires pour visiter tous les nœuds et donné en résolvant le problème en utilisant les approches séquentielles. Le tableau 4.5 montre l'impact sur la distance finale obtenue en multipliant par deux le nombre de périodes. Pour calculer les écarts, nous utilisons l'équation (3.1) précédemment donnée. Il est clair que si nous augmentons le nombre de périodes, nous pouvons minimiser la distance

totale parcourue. Le gain peut être jusqu'à 23.63% et en moyenne de 2.74%. Cette augmentation affecte également le temps de résolution et augmente la complexité de l'instance.

Instance	MILP-Global				Écarts (%)	
	Nombre de périodes * 1	Temps CPU (s)	Nombre de périodes * 2	Temps CPU (s)	Distance	Temps CPU
S2-C-10-1	58.7243	3.16	58.7243	3.87	0	18.35
S2-C-10-2	59.4222	17.18	59.4222	18.07	0	4.93
S2-R-10-1	270.925	2.13	270.925	3.19	0	33.23
S2-R-10-2	226.993	13.42	223.048	15.21	-1.73	11.77
S2-RC-10-1	193.085	2.04	193.085	2.97	0	31.31
S2-RC-10-2	195.147	3.21	195.147	4.12	0	22.09
S2-C-20-1	204.177	5.13	195.863	6.98	-4.07	26.50
S2-C-20-2	212.947	42.12	182.11	64.41	-14.48	34.61
S2-R-20-1	449.999	4.81	449.999	5.68	0	15.32
S2-R-20-2	400.301	140.51	400.301	460.03	0	69.46
S2-RC-20-1	323.981	3.63	323.981	4.98	0	27.11
S2-RC-20-2	327.786	5.12	327.786	7.84	0	34.69
S2-C-50-1	435.339	192.13	435.339	221.74	0	13.35
S2-C-50-2	525.871	305.49	401.594	341.51	-23.63	10.55
S2-R-50-1	1125.31	1024.17	1125.31	1257.21	0	18.54
S2-R-50-2	1122.95*	7200	1122.95*	7200	0	0
S2-RC-50-1	699.59**	7200	613.25**	7200	-	-
S2-RC-50-2	677.21**	7200	583.96**	7200	-	-

TABLE 4.5 – Impact de la multiplication du nombre de périodes

Dans cette partie, nous avons présenté quatre approches pour résoudre le mono-objectif Mu-PDPTWPD. Trois de ces méthodes abordent le problème sous la forme d'une séquence de Mo-SPDPTWPDs. L'écart constaté entre les solutions fournies par nos approches séquentielles et les solutions optimales s'explique par la stratégie elle-même qui consiste à résoudre chaque période sans avoir un aperçu sur la globalité du problème.

#### 4.4 Étude multi-objectifs

Cette partie sera consacrée à la résolution du Mu-PDPTWPD en prenant en considération plusieurs objectifs. Nous introduisons un deuxième objectif lié au respect des livraisons pendant les périodes prévues. La pénalité en cas de livraison anticipée peut correspondre au coût de stockage entre la date de livraison réelle et la date de livraison demandée. Si une demande est livrée en retard, une pénalité de retard est également payée. Notre premier objectif reste la minimisation de la distance totale parcourue pendant l'ensemble des périodes (F3). De plus, nous considérons la minimisation de la pénalité comme deuxième objectif. Ce dernier est représenté par

l'équation suivante :  $F4 = \sum_{i \in Nodes} \sum_{t \in P} E_{i,t} Penalty_{i,t}$ .

Dans la suite de cette section, nous introduisons deux approches différentes pour résoudre le Mu-PDPTWPD bi-objectifs.

#### 4.4.1 Approche lexicographique

Cette approche est déjà utilisée dans le chapitre précédent pour résoudre le SPDPTWPD. Pour ce dernier, l'ordre d'importance des objectifs était facile à définir puisque nous étions obligés de maximiser le profit collecté (F1) avant de minimiser la distance (F2). Ce n'est pas le cas du Mu-PDPTWPD qui n'est pas un problème sélectif. Donc, nous pouvons imaginer n'importe quel ordre d'importance. Et puisque nous allons considérer deux objectifs F3 et F4, deux ordres sont possibles : le premier vise à minimiser la distance puis la pénalité et le deuxième consiste à commencer par minimiser la pénalité avant de passer à la minimisation de la distance.

Chacune de ces deux configurations fournit une solution qui appartient au front de Pareto. La figure 4.6 schématise l'approche lexicographique avec les deux ordres que nous pourrions utiliser pour résoudre notre variante.

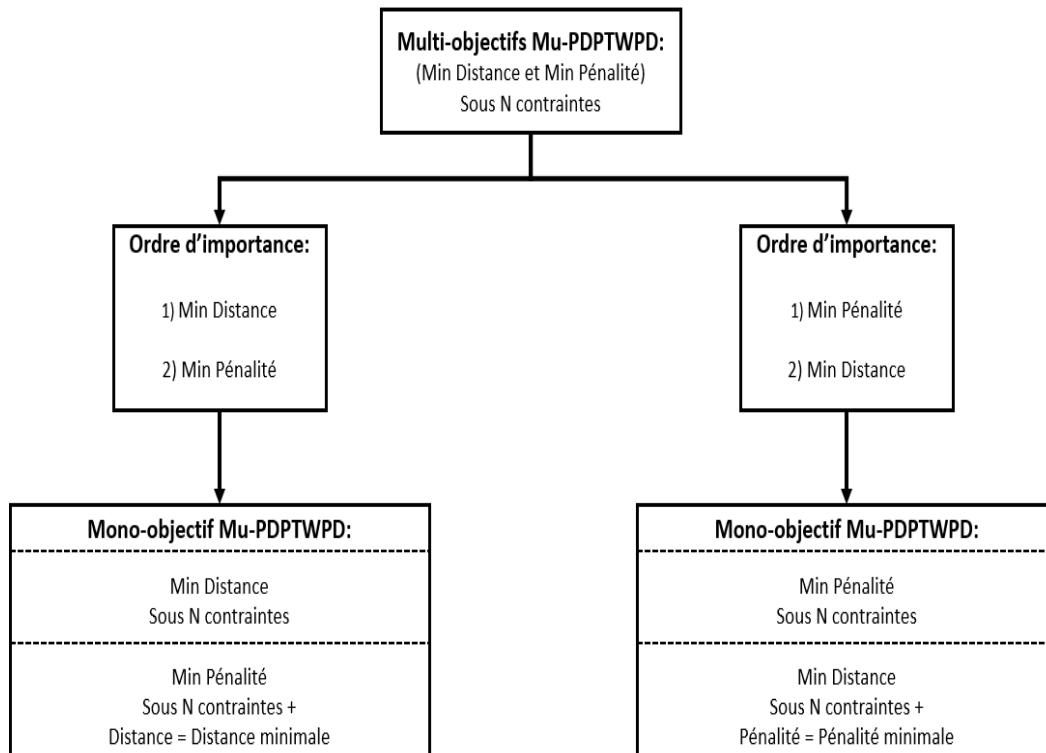


FIGURE 4.6 – Approche lexicographique pour le Mu-PDPTWPD

Un intérêt de cette approche est qu'elle garantit que la solution obtenue suivant un de ces deux ordres est non-dominée. Ses principaux inconvénients sont : le temps

de résolution élevé et le fait de n'obtenir que deux solutions non-dominées. Dans la sous-section suivante, nous introduisons une autre méthode dans le but est de surmonter les limites de notre approche lexicographique.

#### 4.4.2 Algorithme génétique hybride combiné avec GRASP

Notre deuxième méthode pour traiter le Mu-PDPTWPD dans un contexte multi-objectifs est basée sur une combinaison entre un algorithme génétique hybride et une méthode de type GRASP (Greedy Randomized Adaptive Search Procedure). La méthode GRASP est une approche méta-heuristique. Elle est introduite pour la première fois dans [Feo & Resende 1995]. Elle est divisée généralement en deux phases exécutées l'une à la suite de l'autre. La première est une phase de construction d'une solution. La deuxième peut être considérée comme une phase d'application d'une recherche locale sur la solution déjà construite.

Dans la suite de cette partie, nous détaillons notre méthode qui sera notée "HGA-GRASP". En effet, notre approche est divisée en deux étapes principales. Dans la première, nous appliquons la GRASP pour générer des solutions initiales réalisables. Dans la deuxième étape, l'algorithme génétique hybride (HGA) améliore les solutions générées initialement.

##### 4.4.2.1 Représentation de la solution :

Chaque solution est représentée par un individu. Le codage utilisé dans cette méthode consiste à diviser chaque solution en plusieurs chromosomes. Chacun est associé à une période spécifique et composé de plusieurs gènes où chacun correspond à une tournée.

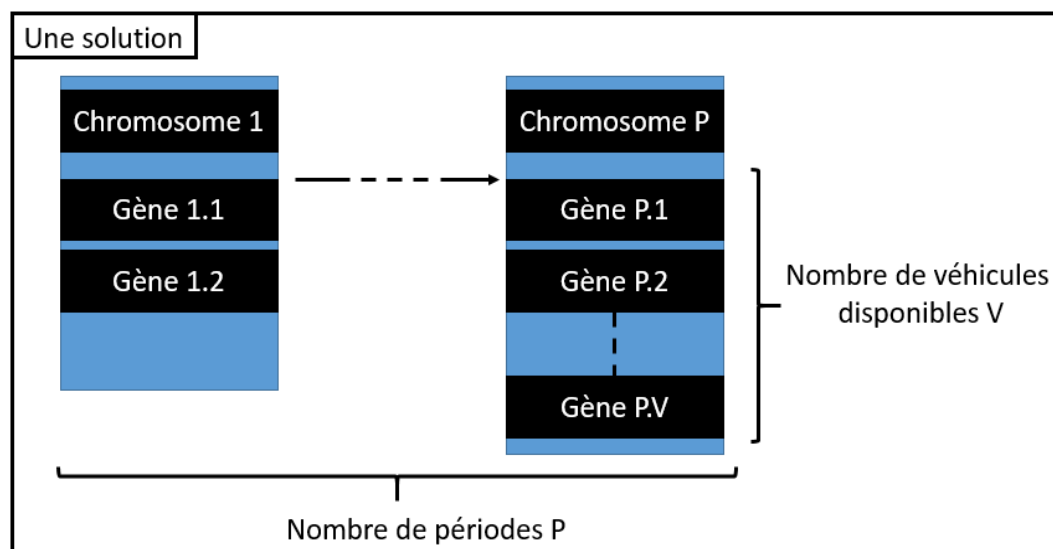


FIGURE 4.7 – Exemple de représentation d'une solution

Étant donné un ensemble de  $P$  périodes et une flotte composée de  $V$  véhicules, la

figure 4.7 donne un exemple de représentation d'une solution. Elle est composée de  $P$  chromosomes, un par période. Pour chacun d'entre eux, le nombre de gènes (c'est à dire le nombre de tournées) peut varier entre 0 et  $V$  parce que l'utilisation de tous les véhicules disponibles n'est pas obligatoire.

#### 4.4.2.2 Création de la population initiale :

La population initiale est construite en utilisant une approche GRASP afin de produire de bons individus mélangeant l'aspect aléatoire et une démarche guidée.

---

#### Algorithme 11 Algorithme GRASP

---

**Input :**  $K$ ,  $Max$ ,  $MNIWI$ ;

**Output :** Ensemble des solutions  $Sol_{init}$ ;

```

for  $i=1$  jusqu'au  $Max$  do
  Nœud de réf  $\leftarrow$  Dépôt;
  Solution  $Sol$ ;
   $Sol.id \leftarrow i$ ;
  for  $p=1$  jusqu'au  $P$  do
    Chromosome  $C$ ;
    for  $v=1$  jusqu'au  $V$  do
      Gene  $G$ ;
       $j \leftarrow 0$ ;
       $N \leftarrow$  Sélectionner un nœud parmi les  $K$  plus proches du nœud de réf;
      if  $N$  n'est pas affecté et [ $N \in B$  ou [ $N \in C$  et  $Supplier_N=0$ ]] then
        Ajouter  $N$  à  $G$ ;
        if  $Customer_N \neq 0$  then
          | Add  $Customer_N$  to  $G$ ;
        end
        if  $G$  est infaisable then
          | Supprimer  $N$  (et  $Customer_N$ ) de  $G$ ;
          | if  $j \leq MNIWI$  then
          | |  $j \leftarrow j + 1$ ; Revenir en arrière et choisir un autre  $N$ ;
          | else
          | | Quitter la boucle courante;
          | end
        else
          |  $j \leftarrow 0$ ;  $N$  (et  $Customer_N$ ) sont affectés;
          | Nœud de réf  $\leftarrow N$ ;  $G \leftarrow$  Appliquer la recherche locale sur  $G$ ;
        end
      end
    Ajouter le gène  $G$  au chromosome  $C$ ;
  end
  Ajouter le Chromosome  $C$  à la solution  $S$ ;
end
  Insérer la solution  $Sol$  dans l'ensemble  $Sol_{init}$ ;
end

```

---

Dans le GRASP que nous avons développé, nous générons une première route de la première période et nous choisissons le dépôt comme nœud de référence. Ensuite, nous choisissons un site au hasard parmi les  $K$  sites les plus proches du nœud de référence pour l'insérer dans la route (avec son site appairé s'il est différent du dépôt). Après une insertion réalisable, nous considérons le site inséré comme le nouveau nœud de référence et nous appliquons une méthode de recherche locale pour améliorer le gène actuel. L'algorithme de notre procédure de recherche locale a déjà été présenté dans le chapitre précédent (algorithme 9). Elle consiste donc à minimiser la distance d'une tournée donnée en testant un nombre donné de permutations, en respectant toutes les contraintes du problème. Comme le montre l'algorithme 11, nous répétons la procédure jusqu'à ce que nous ayons exécuté le nombre maximum d'itérations sans amélioration (MNIWI). Une fois le premier gène construit, nous passons au suivant et ainsi de suite jusqu'à la construction du premier chromosome. Ensuite, nous passons à la période suivante avec toutes les demandes non affectées et nous répétons ce processus à toutes les périodes pour obtenir une solution. Enfin, la méthode GRASP est itérée  $Max$  fois afin de générer notre population initiale.

#### 4.4.2.3 Attribution des rangs :

Chaque solution créée est évaluée en utilisant une procédure de classement. En effet, nous avons utilisé la méthode NSGA afin d'associer un rang  $Rank_i$  à chaque solution  $i$ . Le mécanisme de travail pour la méthode NSGA a été déjà abordé dans le chapitre précédent (partie 3.3.3.1). La figure 4.8 montre un exemple de classement pour une population composée de six solutions classées selon la méthode NSGA.

	Distance	Pénalité	Rank
<b>Solution 1</b>	170	30	1
<b>Solution 2</b>	180	25	1
<b>Solution 3</b>	150	40	1
<b>Solution 4</b>	170	60	2
<b>Solution 5</b>	150	70	2
<b>Solution 6</b>	180	90	3

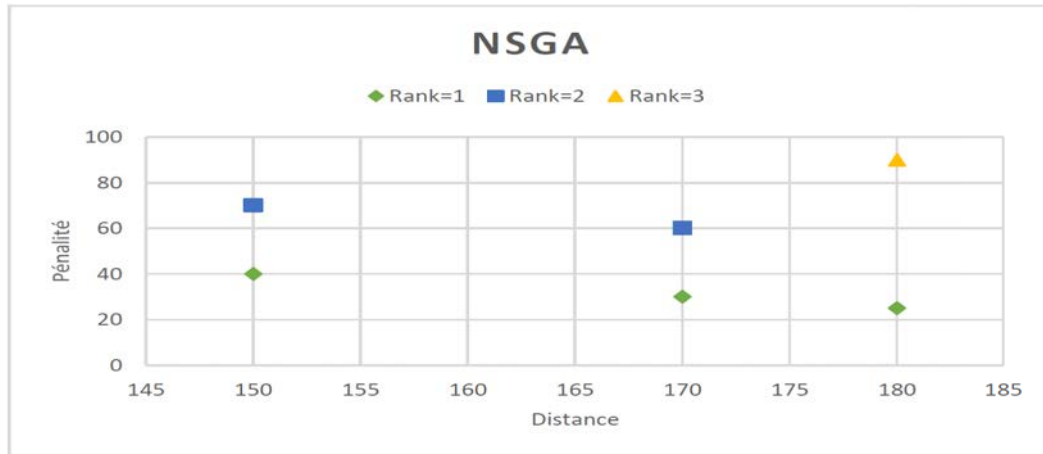


FIGURE 4.8 – Attribution des rangs

#### 4.4.2.4 Procédure de sélection :

Après avoir généré et évalué une population donnée, l'étape suivante consiste à sélectionner deux parents sur lesquels l'opérateur de croisement sera appliqué. Pour ce faire, nous avons utilisé une procédure de sélection basée sur le principe de deux tournois consécutifs. Chacun des parents est alors obtenu avec un tournoi entre deux solutions choisies au hasard parmi la population. Le gagnant est la solution avec le rang le plus bas. Ce type de sélection donne aussi une chance aux individus faibles (ayant un rang élevé) d'être sélectionnés, ce qui peut être considéré comme un mécanisme de diversification. La figure 4.9 représente notre procédure de sélection.

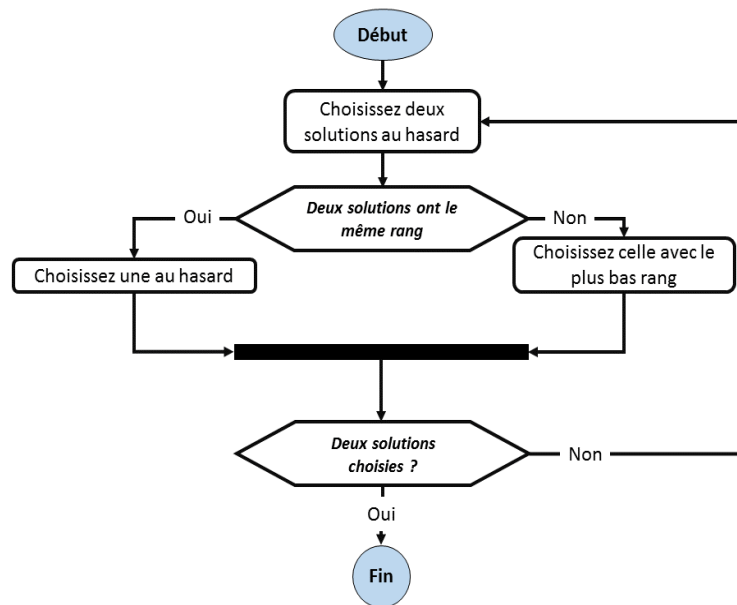


FIGURE 4.9 – Procédure de sélection



4.4.2.5 Croisement :

L'opérateur de croisement est appliqué sur les deux parents sélectionnés (Parent1 et Parent2) grâce à notre procédure de sélection afin de reproduire des nouveaux descendants (Enfant1 et Enfant2).

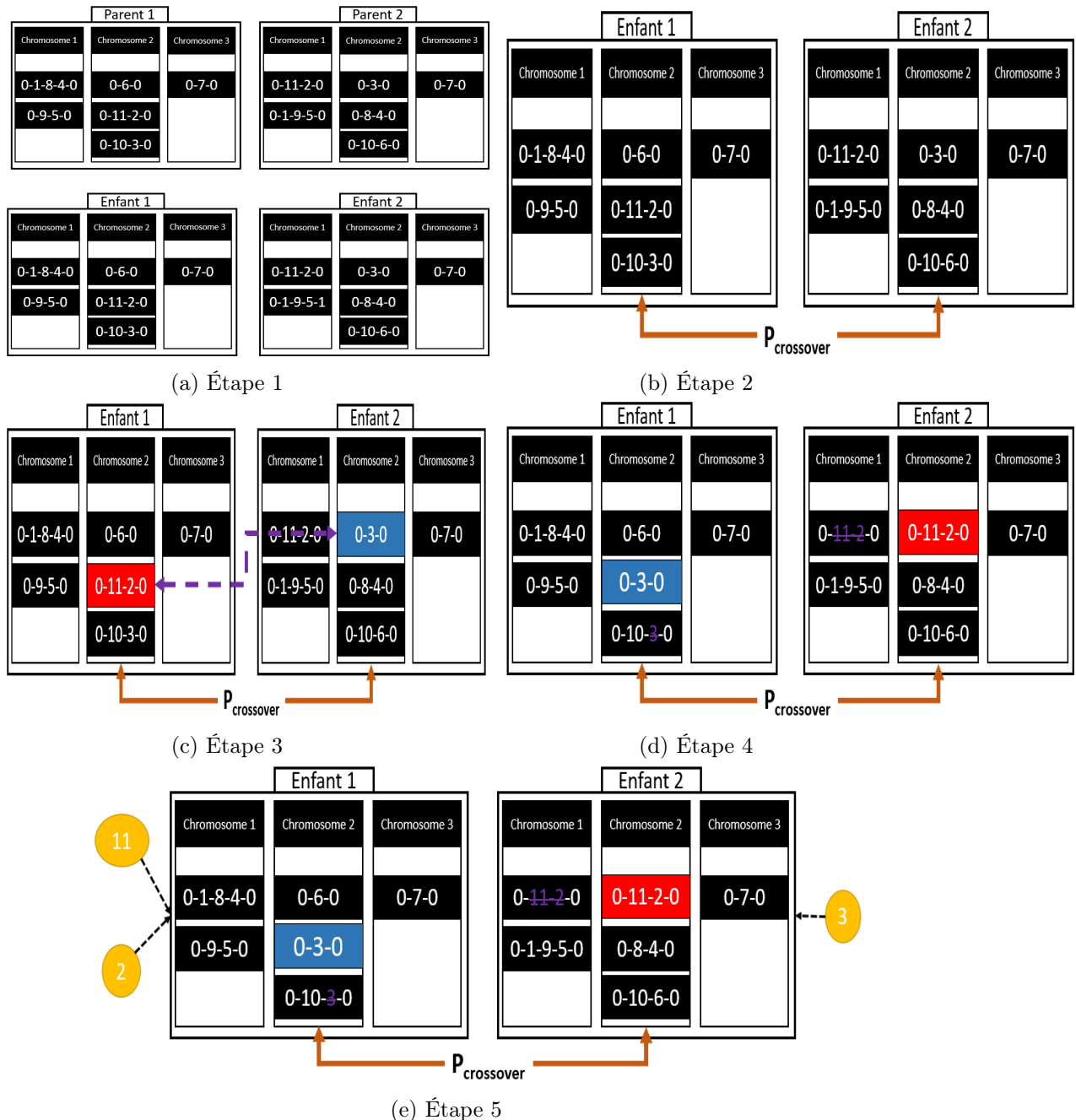


FIGURE 4.10 – Opérateur de croisement

La figure 4.10 montre un exemple d'application de notre opérateur de croisement.

Dans cet exemple, nous avons 3 périodes différentes et une flotte composée de 3 véhicules. Le dépôt est associé à l'identifiant 0. Les demandes de transport sont [0,6], [7,0], [8,4], [9,5], [10,0], [0,3], [0,1] et [11,2].

Le processus principal de notre opérateur peut être divisé en 5 étapes comme suit :

**Étape 1 :** Copier les gènes de Parent1 (et Parent2) dans Enfant1 (et Enfant2) respectivement.

**Étape 2 :** Choisir au hasard une période (qui sera notée  $P_{crossover}$ ) parmi l'ensemble des périodes.

**Étape 3 :** Choisir aléatoirement un gène de la période  $P_{crossover}$  de l'Enfant1 et échanger ce gène avec un autre choisi au hasard de la période  $P_{crossover}$  de l'Enfant2. Ici, nous vérifions que l'échange peut être fait (les contraintes de capacité du véhicule sont respectées). En effet, il est possible qu'en échangeant deux tournées ensemble, nous affectons une de ces tournées à un véhicule dont la capacité ne permet pas de réaliser cette tournée. Dans ce dernier cas, nous choisissons deux autres tournées.

**Étape 4 :** Supprimer les sites en double de chaque enfant qui peuvent être causés par l'échange.

**Étape 5 :** Insérer les sites manquants (non présents dans la solution) pour chaque enfant n'importe où dans ses chromosomes (pas seulement dans  $P_{crossover}$ ) en assurant que chaque insertion respecte toutes les contraintes du problème.

#### 4.4.2.6 Hybridation avec une méthode de recherche locale :

Une fois que les deux enfants sont générés en utilisant notre opérateur de croisement, nous appliquons un algorithme de recherche locale (notée LS) à chaque chromosome modifié. Pour minimiser le temps de calcul, notre procédure LS est appliquée uniquement sur les chromosomes qui ont eu une modification (insertion ou suppression d'un ou plusieurs sites) par rapport aux chromosomes initiaux copiés des parents. L'objectif de LS est de minimiser la distance totale pour un chromosome donné. L'algorithme 12 décrit cette procédure. Le processus se termine après MAX\_LS itérations successives sans amélioration.

#### 4.4.2.7 Test de dominance :

La dernière étape dans notre HGA-GRASP est de tester la dominance de chaque descendant généré. S'il domine au moins un de ses parents, nous le gardons. Sinon, il sera détruit.

L'opérateur de croisement est appliqué MAX\_CO fois et après chaque itération, si le nombre limite de solutions dans la population est dépassé, nous supprimons les plus mauvaises solutions (qui ont les rangs les plus élevés). Cette limite est déterminée en utilisant le paramètre LIMIT\_Sol.

#### 4.4.2.8 Algorithme global :

La figure 4.11 rappelle les étapes de notre méthode "HGA-GRASP".

**Algorithme 12** Procédure LS**Input** : Chromosome  $C$ ,  $MAX\_LS$ ;**Output** : Chromosome  $C$ ;Chromosome  $C\_temp \leftarrow C$ ; $i \leftarrow 0$ ;**while**  $i < MAX\_LS$  **do**

Sélectionnez aléatoirement une demande appariée d'un gène aléatoire  $G$  dans  $C\_temp$ ;

**for** *Tous les gènes dans  $C\_temp \neq G$*  **do**

Essayez d'insérer la demande appariée choisie dans le gène;

**if** *L'insertion est faisable et  $Distance(C\_temp) < Distance(C)$*  **then**

$C \leftarrow C\_temp$ ;

$i \leftarrow 0$ ;

**else**

$i \leftarrow i + 1$ ;

**end**

**end**

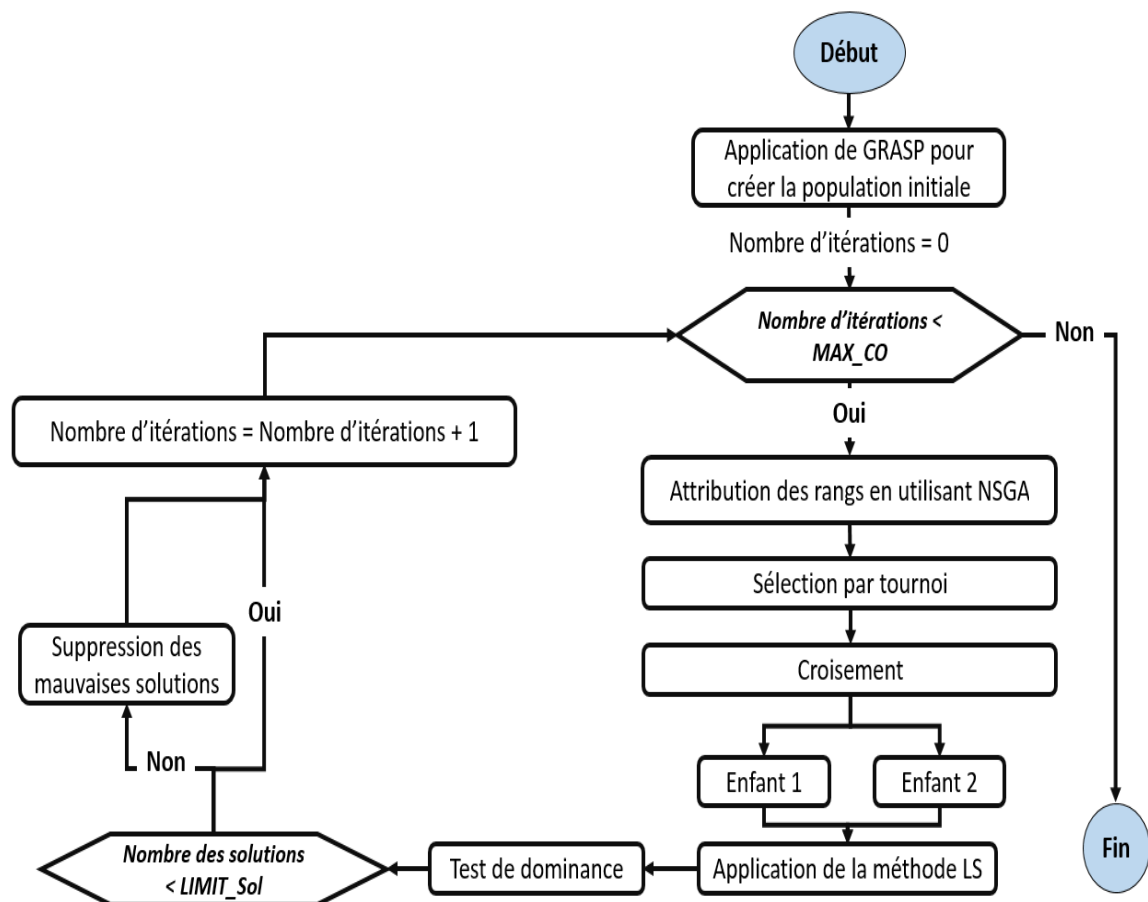
**end**

FIGURE 4.11 – Méthode "HGA-GRASP"

### 4.4.3 Résultats expérimentaux

Dans cette partie, l'efficacité des approches multi-objectifs proposées est étudiée. La métaheuristique proposée est codée en C++ et l'approche lexicographique est résolue en utilisant le solveur commercial CPLEX 12.4. Nos approches sont exécutées avec un processeur Intel Core i7, 2.80 GHz et 16 Go de RAM. Les paramètres d'algorithme sont fixés comme suit :  $MAX\_CO = 1000 + (10 * z)$ ,  $MNIWI = 9$ ,  $K = z/2$ ,  $Max = 20$ ,  $LIMIT\_Sol = 100$ ,  $MAX\_LS2 = 5$  et  $CP = 15$ . Le choix des paramètres a été effectué après plusieurs tests. En effet, l'augmentation de  $MAX\_CO$ ,  $MNIWI$ ,  $Max$ ,  $MAX\_LS2$  et  $CP$  augmentera le temps de résolution. D'autre part, l'augmentation de  $K$  dégradera la qualité de la population initiale en transformant GRASP en une méthode aléatoire. Par contre, si nous diminuons la valeur de  $K$ , le GRASP se rapprochera d'une méthode gloutonne classique. Enfin,  $LIMIT\_SO$  a un impact direct sur la procédure de sélection (si nous augmentons cette valeur, la diversification de notre méthode augmente aussi et si nous la diminuons, l'intensification augmente). Tous les résultats dans cette partie sont obtenus à l'issue de 10 lancements de nos approches pour chaque instance.

Dans un premier temps, nous avons comparé les résultats obtenus par notre approche GRASP seulement et ceux obtenus après l'exécution de la combinaison GRASP-HGA avec les solutions obtenues en utilisant la méthode MILP-Global. Pour cela, nous avons décidé d'utiliser les deux ensembles d'instances déjà générés. Une telle comparaison est possible ici parce que nous avons fixé les valeurs de pénalité à zéro. Le tableau 4.6 montre la distance totale parcourue et le temps de résolution obtenu en utilisant chaque méthode. Un délai de deux heures a été imposé à toutes nos approches (c'est généralement le cas dans la littérature).

L'écart en termes de distance entre la combinaison GRASP-HGA et l'approche GRASP uniquement ainsi que les écarts en termes de distance et de temps de résolution entre GRASP-HGA et la méthode MILP-Global sont représentés respectivement par les trois dernières colonnes. Les écarts sont calculés en utilisant l'équation (3.1). Les résultats intermédiaires obtenus avec l'approche GRASP sont bons pour les petites instances et l'amélioration apportée par le HGA n'est pas remarquable. Contrairement aux grandes instances, où le HGA a notablement amélioré la qualité des solutions obtenues. L'écart entre le GRASP et le GRASP-HGA en termes de distance est en moyenne de 4.65% et peut aller jusqu'à 17.38%. Il est évident que le temps CPU utilisé par notre GRASP-HGA est supérieur à celui utilisé lors de l'application de notre approche GRASP seulement (la différence maximale est d'environ 30 minutes). Les distances obtenues en utilisant la combinaison GRASP-HGA sont très proches de celles fournies par l'approche exacte avec un écart moyen de 0.95% et un écart-type de 1.88%. Le gain en temps de résolution est en moyenne de 75.54%. Ces valeurs sont calculées en fonction des 46 instances résolues en utilisant l'approche MILP-Global. Les solutions avec le "\*" correspondent aux meilleures solutions faisables obtenues par l'approche MILP-Global après deux heures de résolution et celles avec "\*\*\*" correspondent à la borne inférieure si aucune solution réalisable n'a été fournie dans l'intervalle de temps.

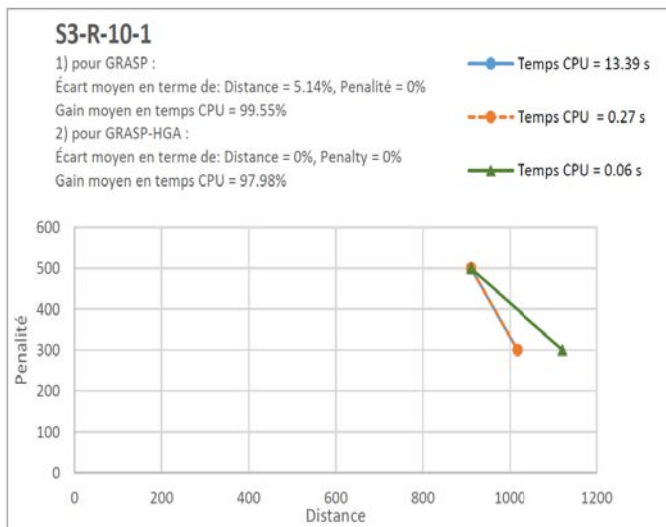
Instance	NP	MILP-Global		GRASP		GRASP-HGA		Écart (GRASP-HGA   GRASP)		Écart (GRASP-HGA   MILP-Global)	
		Distance	Temps CPU (s)	Distance	Temps CPU (s)	Distance	Temps CPU (s)	Distance	Distance	Temps CPU	
S1-R-10-1	10	910.66	3.2	910.66	0.05	910.66	0.21	0	0	-93.44	
S1-R-10-2	10	657.04	5.67	727.79	0.12	657.04	1.45	-9.72	0	-74.43	
S1-R-10-3	10	821.46	6.98	821.46	0.19	821.46	1.73	0	0	-75.21	
S1-R-10-4	10	912.91	6.45	912.91	0.17	912.91	1.67	0	0	-74.11	
S1-R-10-5	10	843.88	6.37	843.88	0.15	843.88	1.62	0	0	-74.57	
S1-R-10-6	10	930.68	6.87	930.68	0.19	930.68	1.73	0	0	-74.82	
S1-R-10-7	10	897.63	6.49	897.63	0.16	897.63	1.59	0	0	-75.5	
S1-R-10-8	10	759.55	7.13	759.55	0.22	759.55	2.12	0	0	-70.27	
S1-R-10-9	10	938.96	7.04	938.96	0.23	938.96	2.09	0	0	-70.31	
S1-R-10-10	10	803.48	7.17	803.48	0.3	803.48	2.2	0	0	-69.32	
S1-R-20-1	10	1814.51	10.73	1818.5	0.33	1814.51	2.97	-0.22	0	-72.32	
S1-R-20-2	10	1706.71	8.95	1855.51	0.33	1706.71	2.68	-8.02	0	-70.06	
S1-R-20-3	10	1829.73	6.35	1851.31	0.29	1829.73	1.67	-1.17	0	-73.7	
S1-R-20-4	10	1493.82	11	1573.07	0.51	1493.82	3.45	-5.04	0	-68.64	
S1-R-20-5	10	1667.67	9.33	1784.01	0.48	1667.67	3.23	-6.52	0	-65.38	
S1-R-20-6	10	1643.01	11.64	1722.48	0.53	1643.01	3.87	-4.61	0	-66.75	
S1-R-20-7	10	1547.68	6.76	1873.34	0.43	1547.68	1.57	-17.38	0	-76.78	
S1-R-20-8	10	1635.76	5.82	1672.39	0.41	1635.76	1.47	-2.19	0	-74.74	
S1-R-20-9	10	1605.32	6.74	1758.48	0.47	1631.23	1.86	-7.24	1.61	-72.4	
S1-R-20-10	10	1524.8	6.1	1615.16	0.46	1524.8	1.78	-5.59	0	-70.82	
S1-R-50-1	10	3242.77*	7200	3872.63	78.63	3432.03	250.15	-11.38	5.84	-96.53	
S1-R-50-2	10	3217.59*	7200	3508.88	392.12	3346.46	2217.23	-4.63	4.01	-69.21	
S1-R-50-3	10	3397.16*	7200	3870.85	13.58	3480.77	78.35	-10.08	2.46	-98.91	
S1-R-50-4	10	3385.38*	7200	3868.62	94.34	3390.62	325.18	-12.36	0.15	-95.48	
S1-R-50-5	10	3129.47*	7200	3489.94	85.77	3204.65	278.12	-8.17	2.4	-96.14	
S1-R-50-6	10	3169.07*	7200	3561.57	38.42	3210.94	125.16	-9.84	1.32	-98.26	
S1-R-50-7	10	3201.02*	7200	3270.11	82.41	3213.55	223.45	-1.73	0.39	-96.9	
S1-R-50-8	10	3169.13*	7200	3537.53	247.81	3241.11	1720.12	-8.38	2.27	-76.11	
S1-R-50-9	10	2996.75	4259.32	3554.29	17.29	3209.37	84.25	-9.7	7.1	-98.02	
S1-R-50-10	10	3286.35*	7200	3811.76	100.41	3351.32	358.22	-12.08	1.98	-95.02	
S2-C-10-1	1	58.72	3.16	58.72	0.03	58.72	0.97	0	0	-69.3	
S2-C-10-2	1	59.42	17.18	59.42	0.25	59.42	2.35	0	0	-86.3	
S2-R-10-1	5	270.93	2.13	270.93	0.03	270.93	0.86	0	0	-59.53	
S2-R-10-2	3	226.99	13.42	226.99	0.52	226.99	4.03	0	0	-70.01	
S2-RC-10-1	2	193.09	2.04	226.44	0.17	193.09	1.02	-14.73	0	-50	
S2-RC-10-2	2	195.15	3.21	199.71	0.19	195.15	1.24	-2.29	0	-61.46	
S2-C-20-1	2	204.18	5.13	234.09	0.24	211.7	1.8	-9.57	3.68	-64.89	
S2-C-20-2	2	212.95	42.12	221.31	1.23	214.17	10.27	-3.22	0.58	-75.62	
S2-R-20-1	2	450	4.81	455.26	0.2	450	1.05	-1.16	0	-78.23	
S2-R-20-2	2	400.3	140.51	423.99	8.94	400.3	70.25	-5.59	0	-50.01	
S2-RC-20-1	2	323.98	3.63	336.35	0.21	323.98	1.25	-3.68	0	-65.45	
S2-RC-20-2	2	327.79	5.12	340.64	0.23	327.79	1.82	-3.77	0	-64.38	
S2-C-50-1	2	435.34	192.13	481.39	11.74	468.31	100.23	-2.72	7.57	-47.83	
S2-C-50-2	2	525.87	305.49	553.38	13.51	535.38	109.23	-3.25	1.81	-64.24	
S2-R-50-1	2	1125.31	1024.17	1175.54	17.82	1125.31	120.21	-4.27	0	-88.26	
S2-R-50-2	2	1122.95*	7200	1170.16	97.24	1126.91	350.22	-3.7	0.35	-95.14	
S2-RC-50-1	2	699.5951**	7200	940.13	105.21	920.25	360.85	-2.11	-	-94.99	
S2-RC-50-2	3	677.2159**	7200	1050.99	120.42	969.25	400.12	-7.78	-	-94.44	

TABLE 4.6 – Résultats expérimentaux sur les ensembles 1 et 2

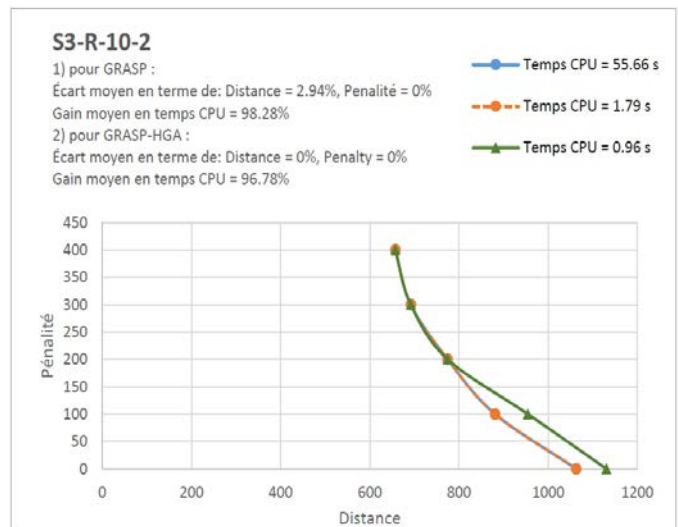
Comme déjà mentionné, toutes les instances appartenant à l'ensemble 1 et 2 ne

prennent pas en compte l'existence d'un coût de pénalité pour la visite d'un site donné dans une période différente de celle demandée. Ainsi, les solutions obtenues par notre métaheuristique qui gère la version bi-objectifs du problème peuvent être comparées avec celles obtenues par l'approche MILP-Global dédiée à la version mono-objectif du problème.

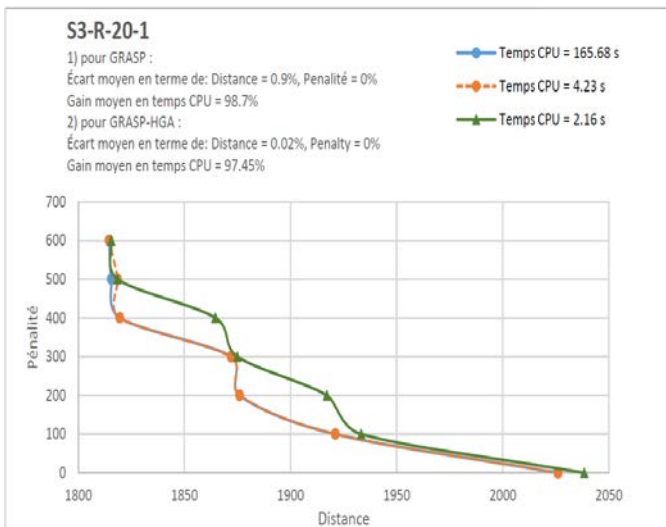
Dans un deuxième temps, nous avons décidé de construire un troisième ensemble composé de 6 instances générées aléatoirement qui prennent en compte la pénalité (égale à 100 si le véhicule visite le site dans une période différente de celle demandée). Ce troisième ensemble va nous servir pour tester nos approches bi-objectifs.



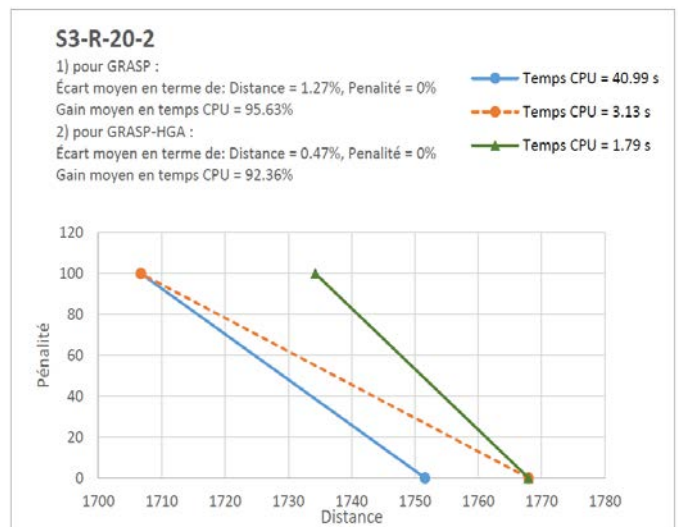
S3-R-10-1



S3-R-10-2



S3-R-20-1



S3-R-20-2

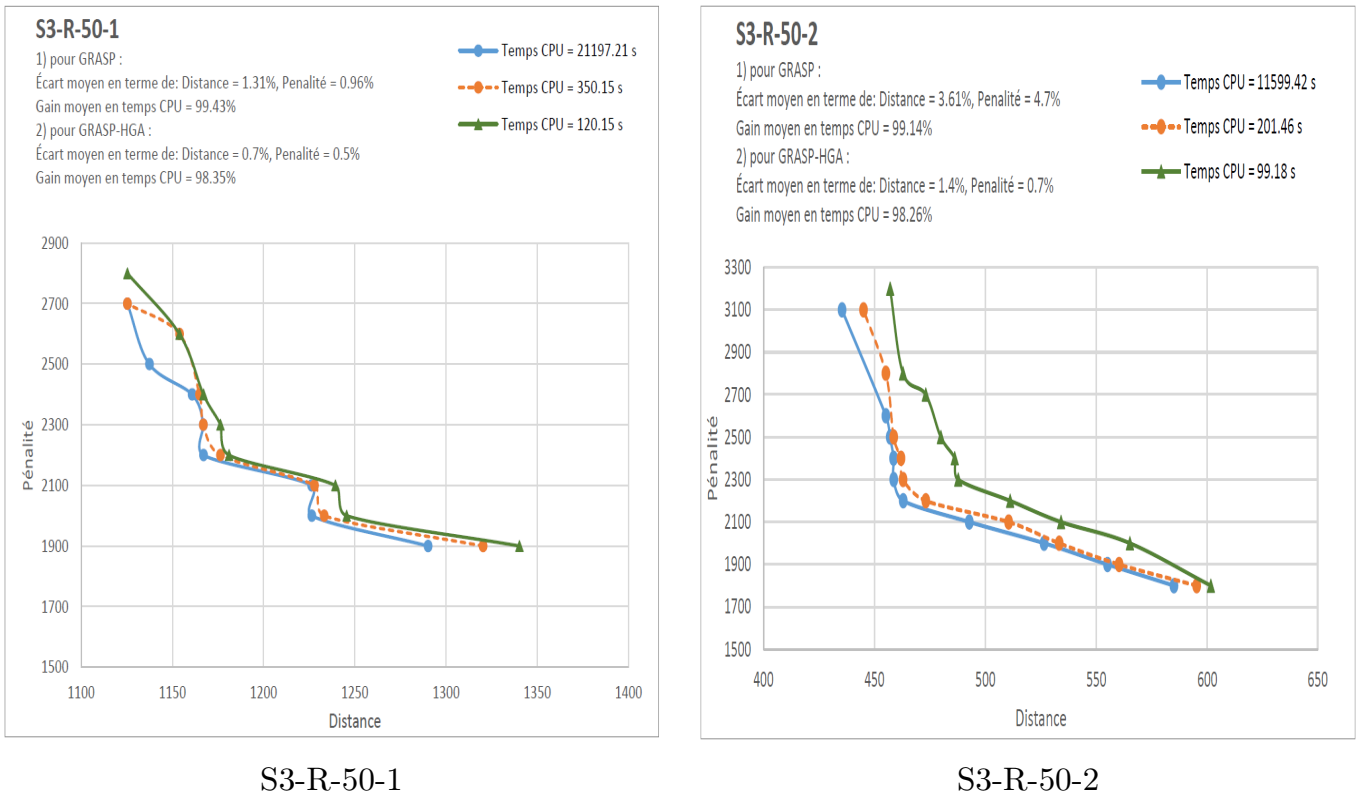


FIGURE 4.13 – Résultats expérimentaux pour l'ensemble 3

Pour chaque instance de l'ensemble 3, la figure 4.13 illustre les solutions obtenues par notre approche métaheuristique représentée par la ligne intermittente orange et le front Pareto-optimal représenté par la ligne bleue. De plus, la ligne verte représente uniquement les résultats fournis par l'approche GRASP. Nous donnons aussi le temps de résolution nécessaire pour obtenir les trois courbes ainsi que le gain de temps en utilisant les approches GRASP et GRASP-HGA par rapport au temps demandé pour extraire le front Pareto-optimal. Les écarts moyens en termes de distance et de pénalité sont également signalés.

L'approche lexicographique nous permet de trouver seulement deux solutions non-dominées :

- $Max_{penalty}$  : cette solution est obtenue en utilisant notre approche lexicographique avec minimisation de la distance puis de la pénalité comme ordre d'importance.
- $Min_{penalty}$  : cette solution est obtenue en minimisant la pénalité puis la distance.

Pour obtenir les autres points du front Pareto, nous appliquons la procédure suivante :

1.  $TPenalite$  égale la pénalité associée avec la solution  $Min_{penalty}$ .

2.  $TPenalite = TPenalite + 100$ . Cette valeur de 100 est due à la valeur de la pénalité si le site est visité dans une période autre que celle désirée. Si  $TPenalite$  est égal à la valeur de pénalité associée avec la solution  $Max_{penalty}$ , nous nous arrêtons là.
3. Nous exécutons notre MILP avec F1 comme fonction objectif après avoir ajouté une contrainte pour imposer que la pénalité totale doit être égale à  $TPenalite$ . Par conséquent, si la solution obtenue n'est dominée par aucune solution existante dans l'ensemble de solutions Pareto, nous ajoutons cette solution à l'ensemble. Puis nous revenons à 2.

En analysant les résultats pour l'ensemble 3, nous remarquons que les solutions obtenues par notre GRASP représentent une bonne population de départ pour notre HGA. Par conséquent, les solutions finales sont très bonnes parce qu'elles sont proches du front Pareto-optimal. De plus, cette bonne performance globale est obtenue dans un petit temps de résolution qui prouve l'efficacité de notre combinaison GRASP-HGA.

Enfin, afin de valider nos procédures pour les instances industrielles, nous avons décidé de générer un quatrième ensemble d'instances composé de six instances générées aléatoirement avec 100, 200, 400, 600, 800 et 1000 nœuds.

Instance	NP	MILP-Global		GRASP		GRASP-HGA		Écart (GRASP-HGA   GRASP)
		Distance	Temps CPU (s)	Distance	Temps CPU (s)	Distance	Temps CPU (s)	Distance
S4-R-100	10	678.27**	7200	1400.14	450.05	1305.75	1100.87	-6.74
S4-R-200	10	1630.62**	7200	7730.04	570.14	7288.84	1465.72	-5.71
S4-R-400	10	Out of memory	3943.75	16740.1	630.87	15874.4	1723.94	-5.17
S4-R-600	10	Out of memory	3874.13	40608.8	781.12	35731.1	2230.45	-12.01
S4-R-800	10	Out of memory	3805.71	61578.3	834.12	57204.7	3120.15	-7.10
S4-R-1000	10	Out of memory	4101.24	107534	1030.45	104064	3942.41	-3.23

TABLE 4.7 – Résultats expérimentaux pour l'ensemble 4

Le tableau 4.7 montre les solutions obtenues en testant nos approches sur les grandes instances de l'ensemble 4. Contrairement à l'approche exacte qui n'a pas donné de solutions réalisables, le tableau montre que nos méthodes approchées ont réussi à fournir des solutions en moins de deux heures. Pour les deux premières instances, l'approche exacte fournit une borne inférieure sans aucune solution faisable et pour les instances restantes de cet ensemble, CPLEX signale une condition de "Out of memory". Pour cet ensemble d'instances, l'écart moyen entre les approches GRASP et GRASP-HGA est de 6.66%.



## 4.5 Analyse de la collaboration entre chargeurs

### 4.5.1 Contexte : Transport Collaboratif en Distribution Urbaine

La collaboration entre chargeurs peut s'effectuer au moins de deux façons. La première est une démarche active entre des chargeurs voisins qui se mettent d'accord sur une période (par exemple un jour) pour une collecte commune. L'idée est ici pour le transporteur de collecter les marchandises chez ses fournisseurs dans une même tournée, d'améliorer le taux de remplissage de ses moyens de transport et ainsi de pouvoir faire bénéficier ses clients d'une remise sur le prix des prestations. Nous avons pu constater que, dans le cadre des différentes méthodes élaborées et testées pour résoudre la variante SPDPTWPD, le fait de minimiser la distance totale parcourue entraînait la collecte ou la livraison de sites voisins et que la maximisation du profit conduisait à un meilleur taux de remplissage des camions (voir sous-section 4.5.2).

La seconde façon d'appréhender cette collaboration peut être qualifiée de passive. En effet, les chargeurs n'ont qu'à accepter le principe de collaboration sur un horizon temporel dont ils fixent la longueur et les modalités. Les horaires de passage pour la collecte et la livraison ne sont plus définis sous la forme d'un créneau horaire précis dans une période donnée (par exemple entre 8h et 10h jeudi prochain), mais comme un ensemble de fenêtres temporelles sur plusieurs périodes (par exemple de 8h à 10h le mardi et le mercredi, et de 10h à 12h le jeudi), avec une date de préférence (par exemple le mercredi). Cet horizon peut être défini en amont de la période initialement prévue ou à la fois en amont et en aval de cette période. Mais parfois, le fait d'anticiper ou de retarder le transport entraîne des conséquences financières (coûts de stockage, pénalités de retard imposées par les clients). Pour compenser ces pertes, un système de pénalités peut être mis en place (soit fixe, soit variable en fonction de l'écart entre la date souhaitée et la date réelle). Le principe de collaboration signifie que tous les partenaires sont censés être gagnants, car un chargeur ne doit pas perdre d'argent pour le bien de tous. Ainsi, il faut s'assurer que même les chargeurs qui subissent des pénalités soient également gagnants, sinon la démarche ne présente aucun intérêt pour eux. Cela induit non seulement que les pénalités soient prises en charge, par exemple déduites du gain global avant répartition de ce gain entre les chargeurs, mais aussi que le prix de la prestation de transport s'en trouve amélioré.

### 4.5.2 Collaboration et tournées mono-période

Dans le but de pouvoir illustrer nos résultats et l'intérêt de l'aspect multi-période, nous avons été amenés à construire un exemple sur trois périodes. Cette instance est construite à partir de la réunion de trois instances mono-périodes (voir annexe A.3). Dans un premier temps, nous allons analyser les résultats pour ces trois exemples.

#### 4.5.2.1 Analyse des résultats pour l'instance Mono-1 :

Si on étudie les différentes demandes de l'exemple Mono-1, il est possible d'envisager des regroupements uniquement basés dans un premier temps sur des critères de localisation. Dans un souci de simplification, nous désignerons les demandes par le couple origine-destination associé. La visualisation des demandes et des tournées de cet exemple est présentée dans les figures suivantes.

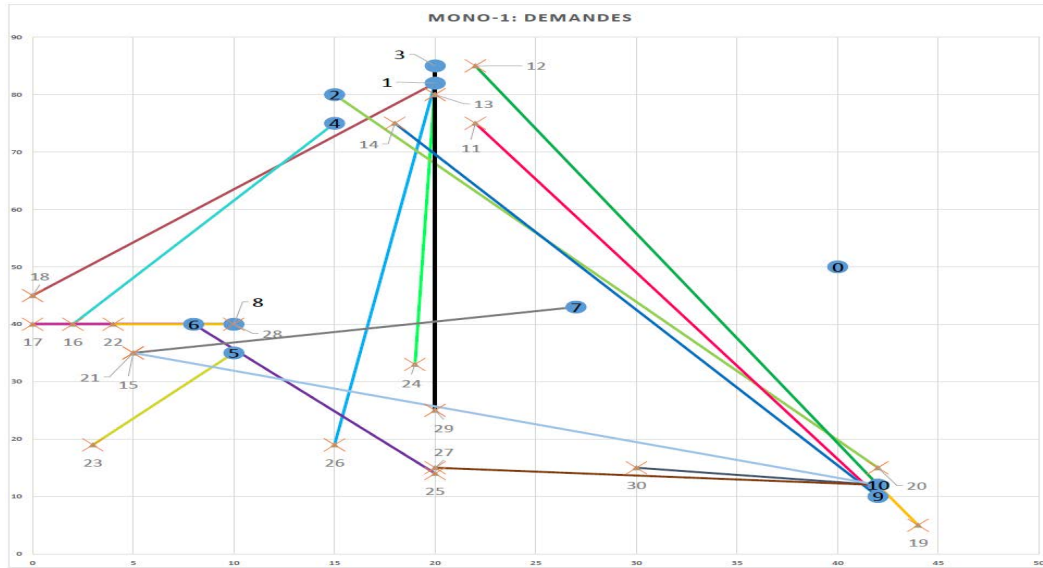


FIGURE 4.14 – Visualisation des demandes pour l'instance Mono-1

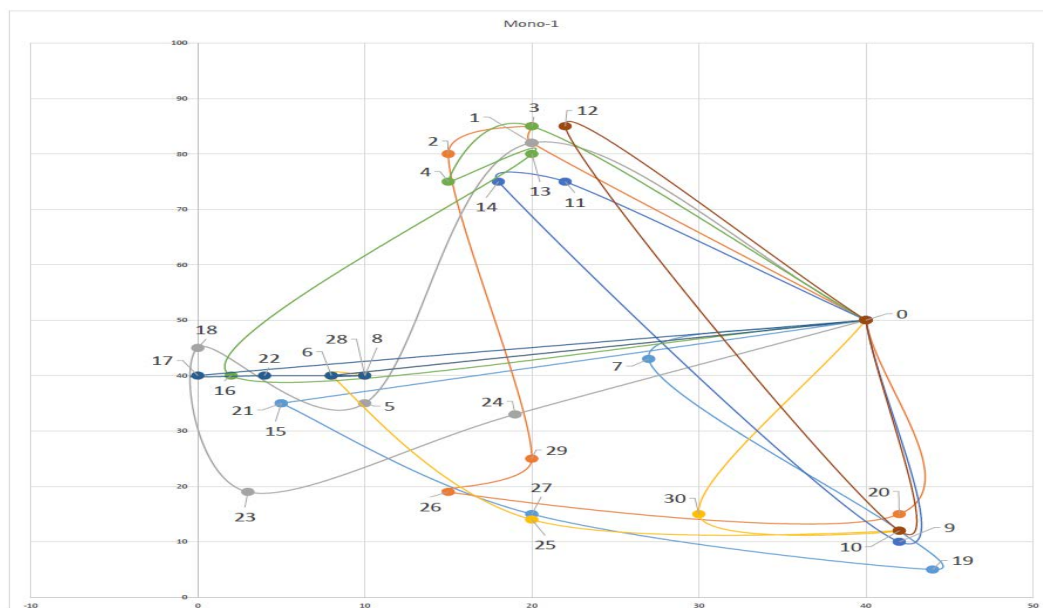


FIGURE 4.15 – Visualisation des tournées pour l'instance Mono-1

En effet, il est possible d'identifier différents types de regroupements possibles. Le premier que nous considérons consiste en des demandes où les fournisseurs sont proches, de même que les clients :

- 9-11, 9-14 et 10-12
- 1-24, 1-26 et 3-29

C'est ce cas qui paraît le plus profitable à la collaboration active des chargeurs. Il est évidemment possible de regrouper les demandes issues d'une même origine :

- $\left. \begin{array}{l} 1-18, 1-24 \text{ et } 1-26 \\ 3-13 \text{ et } 3-29 \end{array} \right\}$  ou proches avec 2-20 et 4-16
- 6-17, 6-25 et 6-28
- $\left. \begin{array}{l} 9-11 \text{ et } 9-14 \\ 10-12, 10-15, 10-19, 10-27 \text{ et } 10-29 \end{array} \right\}$  proches

Il en est de même pour les demandes dont les destinations sont voisines :

- 10-15 et 7-21

Ensuite, des demandes peuvent partager le même axe :

- 6-17, 6-28 et 8-22
- 1-18, 3-13 et 4-16

Pour finir, il reste certaines demandes plus difficiles à regrouper, car elles sont en sens opposé comme la 2-20 par rapport à la 9-11 ou transversales aux autres demandes comme les 5-23 et 6-25. Toutefois, il ne faut pas oublier que d'autres contraintes sont également à prendre en compte. Dans cet exemple, la contrainte de capacité des camions n'interdit pas la plupart des regroupements proposés, sauf pour le site 6 où il est impossible de regrouper toutes les demandes issues de cette origine. Il en est de même pour le regroupement avec la demande 8-22 car la quantité transportée est de 71 pour une capacité camion de 100. Cette forte valeur explique pourquoi dans la tournée réalisant cette demande, la livraison se fait immédiatement après la collecte. Étudions maintenant les tournées proposées par notre algorithme génétique hybride introduit dans la section 3.3 :

- T1 : 0-7-10-19-27-15-21-0
- T2 : 0-1-3-2-29-26-20-0
- T3 : 0-1-5-18-23-24-0
- T4 : 0-6-28-6-25-10-30-0
- T5 : 0-9-14-11-0
- T6 : 0-3-4-13-16-0
- T7 : 0-6-8-22-17-0
- T8 : 0-10-12-0

Le premier regroupement que nous avons proposé, 9-11, 9-14 et 10-12, est découpé sur les tournées T5 et T8. C'est la dernière contrainte, portant sur les fenêtres temporelles, et non encore vérifiée jusqu'à présent qui empêche ce rapprochement.

En effet, lorsque le camion aura collecté la marchandise sur les sites voisins 9 et 10, il ne lui sera plus possible d'arriver à temps sur le site 11. Le deuxième regroupement, 1-24, 1-26 et 3-29, se retrouve réparti sur les tournées T2 et T3. Ce regroupement est possible en considérant l'ensemble des contraintes, suivant la tournée 0-3-1-24-29-26-1 de longueur 148. Toutefois, si on retire ces trois demandes aux tournées T2 et T3, il reste alors les demandes suivantes à réaliser : 2-20, 1-18 et 5-23. La tournée la plus courte qui permettrait de réaliser ces demandes serait 0-1-2-18-5-23-20-0 de longueur 187, mais la fenêtre temporelle sur le site 6 n'est pas respectée. Pour respecter cet intervalle, il faut intercaler le site 6 avant le site 3. Mais alors, il faut revenir en 3 pour respecter la contrainte temporelle sur 3, ce qui va dégrader fortement la distance parcourue. C'est pourquoi, si la minimisation de la distance est recherchée, il faut alors couper en deux la tournée, pour obtenir les tournées suivantes 0-1-5-23-18-0 de longueur 169 et 0-2-20-0 de longueur 144. Soit un total de distance pour ces trois tournées de 461 contre 347 pour T1 et T2. Donc ici, l'objectif prime, ce qui n'est pas gênant car au final le prix de la prestation devrait être plus bas à cause de cette distance moins importante. Il reste à comprendre pourquoi la demande 10-30 a été insérée dans la tournée T4 par notre algorithme. Suivant les suppositions de regroupements, cette demande aurait dû être intégrée dans la tournée T1. A cause des intervalles de visite sur les sites, le site 30 aurait été inséré en dernière position dans la tournée T1. Nous aurions obtenu les tournées suivantes 0-7-10-19-27-15-21-30-0 et 0-6-28-6-25-0, mais cette solution est plus longue de 1 par rapport à la solution obtenue. Pour que la collaboration active fonctionne entre les chargeurs, il n'est pas seulement suffisant de définir une fenêtre de temps proche au niveau des origines, mais aussi nécessaire de vérifier que les fenêtres temporelles au niveau des destinations soient également compatibles, sans parler évidemment de la contrainte de capacité du camion.

Un traitement « automatique » et optimisé des regroupements par un algorithme tel que ceux que nous proposons est donc pleinement justifié et nous avons vérifié sur cet exemple que le résultat obtenu est cohérent et en adéquation avec les objectifs de collaboration.

#### 4.5.2.2 Analyse des résultats pour l'instance Mono-2 :

Les figures suivantes représentent comme précédemment la visualisation des demandes et des tournées.

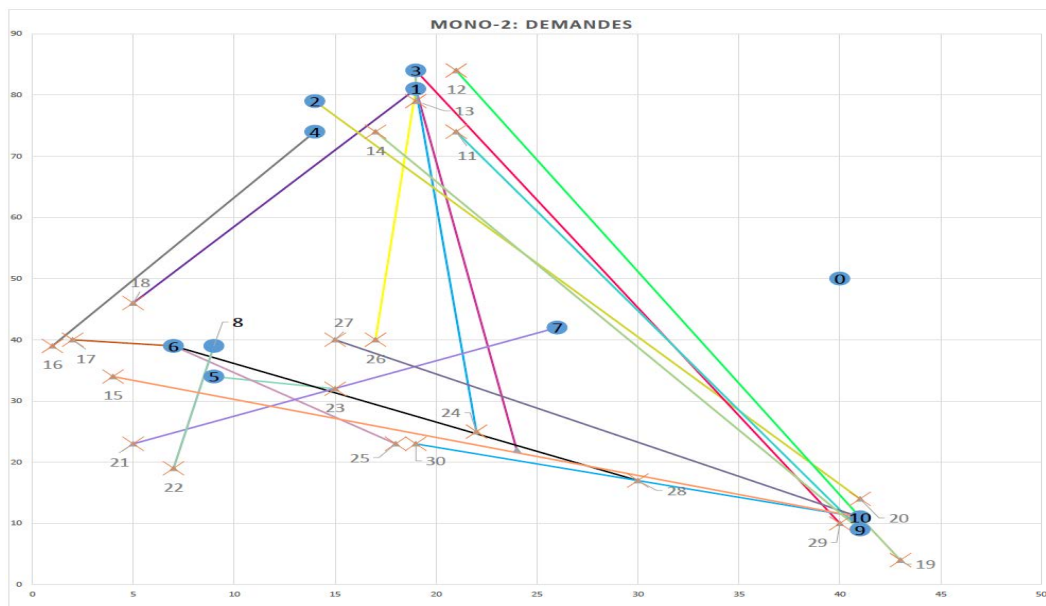


FIGURE 4.16 – Visualisation des demandes pour l'instance Mono-2

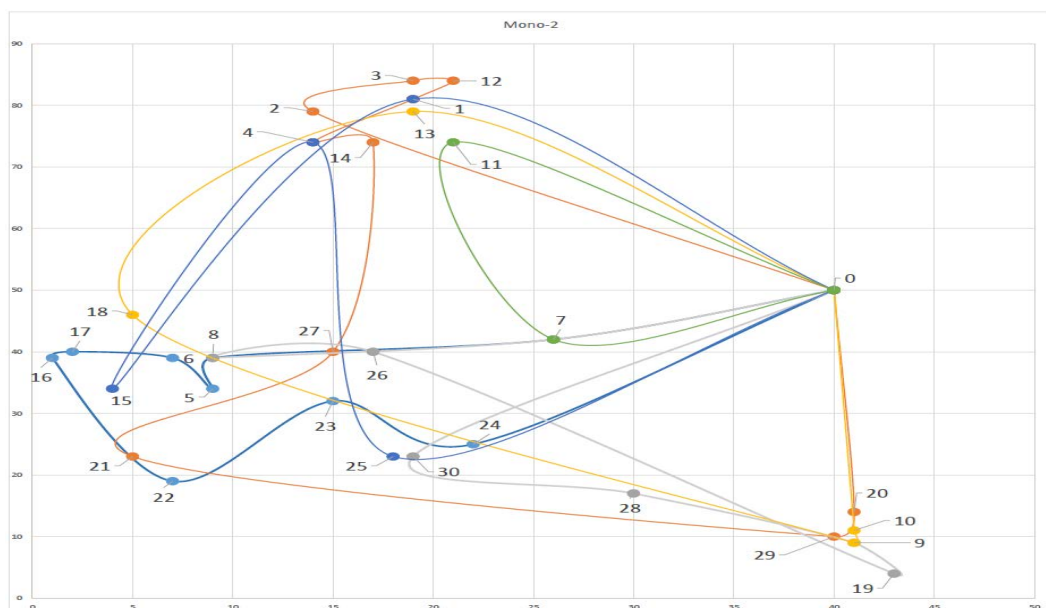


FIGURE 4.17 – Visualisation des tournées pour l'instance Mono-2

Comme pour l'exemple Mono-1, nous allons étudier les différents regroupements. Le premier ensemble rassemble les demandes où les fournisseurs sont proches de même que les clients :

- 1-29 et 3-20
- 1-15, 3-27, 4-21 et 4-25 mais avec des distances un peu plus importantes.

C'est ce cas qui paraît le plus profitable à la collaboration active des chargeurs. Il est évidemment possible de regrouper les demandes issues d'une même origine :

- 1-15 et 1-29
  - 3-27 et 3-20
  - 4-14, 4-21 et 4-25
- } ou proches avec 2-12
- 7-11, 7-19 et 7-22
  - 8-16, 8-23 et 8-26 proches avec 5-24 et 6-17
  - 9-18, 9-28 et 9-30 proches avec 10-13

De même pour les demandes dont les destinations sont voisines avec une distance inférieure à 2 :

- 9-30 et 4-25
- 8-16 et 6-17
- 8-26 et 3-27

Ensuite, des demandes peuvent partager le même axe :

- 6-17, 8-16, 8-23 et 8-26

Pour finir, il reste certaines demandes plus difficiles à regrouper, comme les demandes issues du site 7 qui sont en étoile, de l'intérieur de la ville vers l'extérieur et la demande 10-13 qui est inversé par rapport à 1-29.

Dans cet exemple, considérant les quantités sur certaines demandes supérieures à la moitié de la capacité des camions, certains regroupements sont impossibles. C'est le cas pour les demandes 1-15, 4-21 et 4-25 qui ne peuvent pas se retrouver deux à deux dans la même tournée. Il en est de même pour les paires suivantes 7-11 et 7-19, 9-18 et 9-30 et la dernière 9-18 et 9-28. Étudions maintenant les tournées proposées par notre algorithme :

- T1 : 0-7-8-5-6-17-16-22-23-24-0
- T2 : 0-2-3-12-1-4-14-27-21-29-20-0
- T3 : 0-7-8-26-19-9-28-30-0
- T4 : 0-10-9-18-13-0
- T5 : 0-1-15-4-25-0
- T6 : 0-7-11-0

Le premier regroupement que nous avons proposé, 9-11, 9-14 et 10-12, est découpé sur les tournées T5 et T8. C'est la dernière contrainte, non encore vérifiée jusqu'à présent qui empêche ce rapprochement. En effet, lorsque le camion aura collecté la marchandise sur les sites voisins 9 et 10, il ne lui sera plus possible d'arriver à temps sur le site 11.

#### 4.5.2.3 Analyse des résultats pour l'instance Mono-3 :

Les figures suivantes représentent comme précédemment la visualisation des demandes et des tournées.

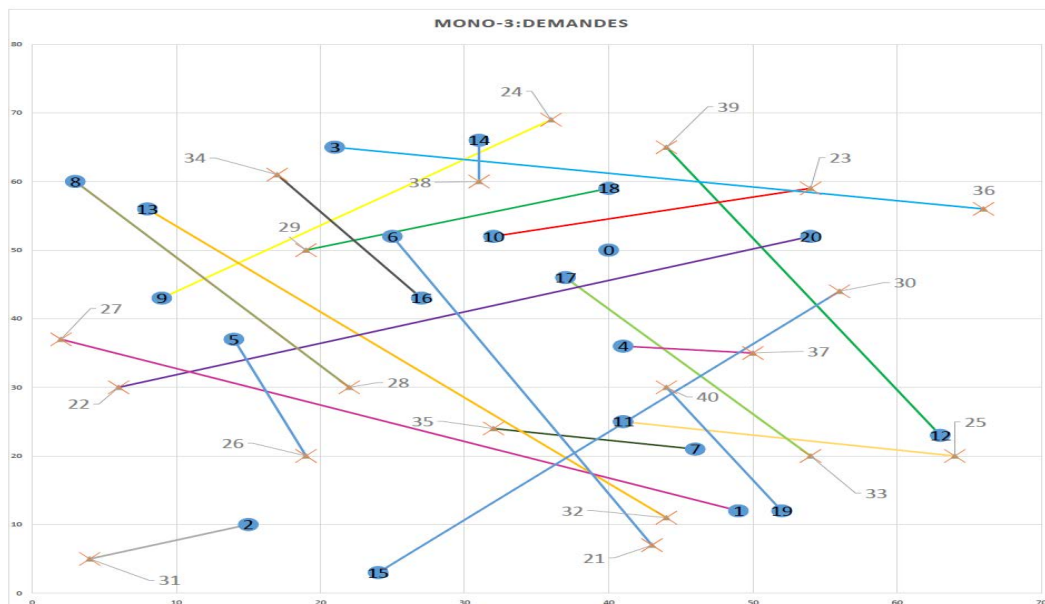


FIGURE 4.18 – Visualisation des demandes pour l'instance Mono-3

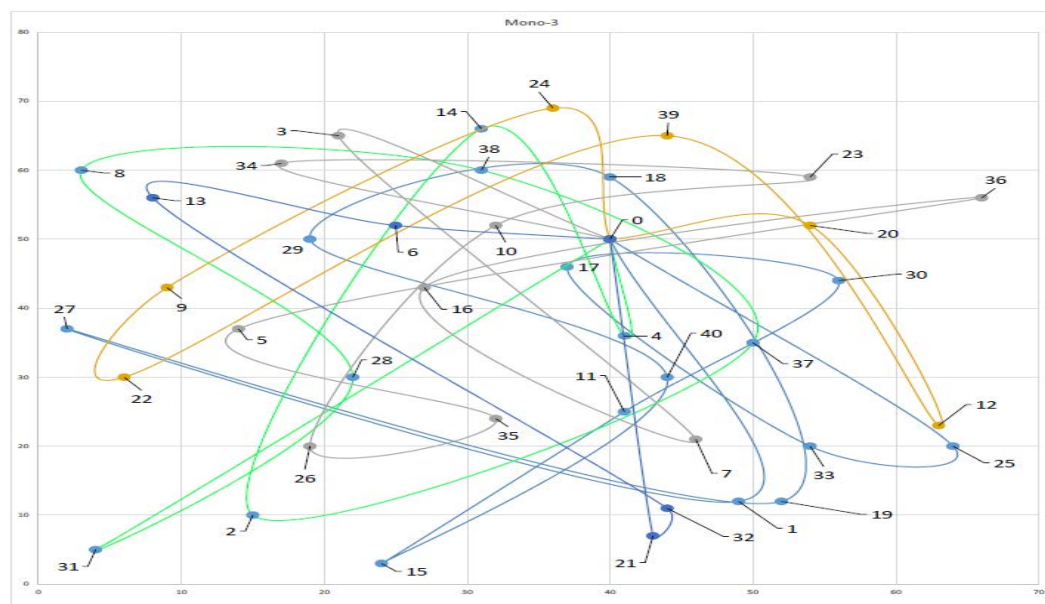


FIGURE 4.19 – Visualisation des tournées pour l'instance Mono-3

L'instance Mono-3 est un peu particulière par rapport aux deux autres instances. En effet, chacun des 20 chargeurs de cette instance est à l'origine d'une seule demande (contre 10 chargeurs dans Mono-1 et Mono-2, voir tableau 4.8). De plus, comme nous pouvons le voir sur la figure 4.18, les sites sont mieux répartis uniformément dans l'espace, et les demandes vont dans tous les sens. Cette configuration rend la constitution des groupes plus difficile. En effet, nous n'avons pas de sites voisins, mais

au cours d'une tournée, le camion peut faire un léger détour pour éventuellement traiter une nouvelle demande. Sans considérer les autres contraintes, nous pourrions ainsi construire la tournée : 0-6-13-8-28-21-32-19-40-0. À partir de cette structure de base, nous pourrions envisager un premier détour pour inclure 5-26 dans la tournée initiale 13-8-5-28-26. Ensuite, de 26, pourquoi ne pas aller réaliser 2-31 ? Ainsi, il est possible d'étirer de plus en plus la tournée. En fait, dans ce cas, nous nous ramenons à ce que nous avons évoqué en tout début de manuscrit, le Lane Covering Problem, avec la problématique de repositionnement de véhicule à vide. Dans notre cas, le véhicule n'est pas forcément vide, mais ces trajets, que nous qualifions d'"inutiles", peuvent avoir un impact non négligeable sur la répartition de la distance totale parcourue sur chaque demande.

Nous observons dans la figure 4.19 que les tournées sont plus longues que pour les autres instances, en moyenne 144 pour Mono-1, 155 pour Mono-2 et finalement 211 pour Mono-3. Elles sont aussi plus enchevêtrées. Les tournées fournies par notre algorithme sont :

- T1 : 0-1-27-19-18-29-40-15-11-30-17-33-25-0
- T2 : 0-4-14-2-37-38-8-28-1-0
- T3 : 0-3-7-16-36-5-35-26-10-23-34-0
- T4 : 0-20-12-39-22-9-24-0
- T5 : 0-6-13-32-21-0

Le tableau 4.8 est la synthèse des résultats obtenus sur les trois périodes. La distance totale de la tournée est répartie entre les différentes demandes associées suivant plusieurs clés de répartition. La première clé est la quantité de marchandises par demande ( $Qté$ ), la seconde est la distance parcourue par un transporteur qui ne réaliserait que cette demande ( $D$ ), et la troisième est le produit de la quantité par la distance précédente ( $Qté * D$ ). Les résultats du tableau représentent le gain en pourcentage entre la distance répartie de la tournée et la distance en trajet direct. Les cases en grisé montrent les cas où la mutualisation d'un camion ne s'avère pas profitable. Dans la majorité des cas, la collaboration semble être intéressante. Pour une demande où l'origine est proche de la destination et la quantité de marchandises associée importante, la clé de répartition suivant la quantité peut être suffisamment importante pour que la distance répartie de la tournée soit supérieure à la distance directe. Donc, plus la tournée incluant cette demande est longue, plus le risque de perte est grand. Toutefois, il est à noter que le gain moyen pour l'ensemble des demandes sur les trois périodes s'élève à 45%.

Pour les chargeurs, la collaboration reste globalement intéressante, mais avec une forte disparité dans les résultats. En moyenne, le gain est de 33% avec un écart type de 28%. Les résultats présentés dans le tableau 4.8 dépendent de la répartition utilisée. En effet, la tournée réalisant toutes les demandes du chargeur RC6 a pour longueur 117. Or nous considérons dans notre répartition les distances pour chacune des demandes de ce chargeur et cette somme est égale à 253. Ainsi pour être rentable,



la collaboration doit avoir un gain supérieur à 112%  $(=(253-117)/117)$ . Cela n'est pas le cas pour ce chargeur. L'intérêt d'une collaboration varie suivant la règle de répartition de ce gain entre l'ensemble des chargeurs.

Demandes	Période 1			Période 2			Période 3		
	Qté	D	Qté*D	Qté	D	Qté*D	Qté	D	Qté*D
d1	0.69	0.48	0.67	0.91	0.69	0.91	0.07	0.17	0.28
d2	-0.16	0.27	-0.03	0.64	0.69	0.58	0.66	0.42	0.65
d3	0.64	0.62	0.67	0.98	0.69	0.98	0.95	0.42	0.95
d4	0.17	0.45	0.19	0.68	0.58	0.64	0.82	0.41	0.83
d5	0.04	0.48	0.11	0.47	0.72	0.37	0.59	0.42	0.46
d6	0.33	0.47	0.2	0.09	0.72	0.29	0.37	0.38	0.50
d7	0.24	0.59	0.23	0.17	0.14	0.15	0.19	0.42	0.01
d8	0.58	0.27	0.48	0.81	0.72	0.81	0.47	0.41	0.36
d9	0.75	0.49	0.76	-0.12	0.69	-0.04	0.01	0.17	0.30
d10	0.55	0.62	0.66	0.74	0.4	0.72	0.18	0.13	0.29
d11	0	0	0	0.01	0.4	0.04	0.48	0.38	0.45
d12	0.9	0.47	0.9	0	0	0	0.08	0.13	0.00
d13	0.65	0.59	0.65	0.54	0.58	0.56	0.31	0.17	0.05
d14	0.81	0.62	0.81	0.56	0.58	0.5	0	0.42	0.07
d15	0.23	0.49	0.22	0.96	0.72	0.94	0.40	0.38	0.55
d16	0.63	0.48	0.59	0.92	0.72	0.89	0.30	0.41	0.17
d17	0.5	0.62	0.41	0.1	0.14	0.12	-0.35	0.42	0.30
d18	0.9	0.59	0.9	0.51	0.58	0.64	0.19	0.41	0.39
d19	0.12	0.47	0.31	0.96	0.69	0.96	0.30	0.38	0.15
d20	0.7	0.45	0.68	0.9	0.72	0.93	0.20	0.41	0.36
<b>moyenne</b>	<b>0.46</b>	<b>0.48</b>	<b>0.47</b>	<b>0.54</b>	<b>0.56</b>	<b>0.55</b>	<b>0.31</b>	<b>0.34</b>	<b>0.35</b>
Chargeurs	Qté	D	Qté*D	Qté	D	Qté*D	Qté	D	Qté*D
RC1	0.15	-0.19	0.11	0.35	0.11	0.32	0.07	0.17	0.28
RC2	0.24	0.59	0.23	0.09	0.72	0.29	0.66	0.42	0.65
RC3	0.18	0.13	0.27	0.79	0.54	0.76	0.95	0.42	0.95
RC4	0.58	0.27	0.48	0.27	0.37	0.24	0.82	0.41	0.83
RC5	0.04	0.48	0.11	0.98	0.69	0.98	0.59	0.42	0.46
RC6	-0.28	-0.14	-0.3	0.91	0.69	0.91	0.37	0.38	0.50
RC7	0.55	0.62	0.66	0.38	0.4	0.33	0.19	0.42	0.01
RC8	0.17	0.45	0.19	-0.31	0.15	-0.15	0.47	0.41	0.36
RC9	0	0	0	0.13	0.32	0.13	0.01	0.17	0.30
RC10	0.06	-0.09	0.03	0.74	0.4	0.72	0.18	0.13	0.29
RC11							0.48	0.38	0.45
RC12							0.08	0.13	0.00
RC13							0.31	0.17	0.05
RC14							0.00	0.42	0.07
RC15							0.40	0.38	0.55
RC16							0.30	0.41	0.17
RC17							-0.35	0.42	0.30
RC18							0.19	0.41	0.39
RC19							0.30	0.38	0.15
RC20							0.20	0.41	0.36
<b>moyenne</b>	<b>0.17</b>	<b>0.21</b>	<b>0.18</b>	<b>0.43</b>	<b>0.44</b>	<b>0.45</b>	<b>0.31</b>	<b>0.34</b>	<b>0.35</b>

TABLE 4.8 – Synthèse de solutions sur les trois périodes

### 4.5.3 Collaboration et tournées multi-périodes

Dans cette partie, nous allons considérer le problème de PDPTWPD dans sa globalité. Les demandes ne sont plus tenues de se terminer dans une période donnée mais peuvent être réalisées dans n'importe laquelle des trois périodes avec ou sans pénalité. L'objectif est de confirmer les bons résultats obtenus dans nos travaux précédents [Manier *et al.* 2016]. Comme nous avons pu le voir dans le paragraphe précédent, cela ne paraît pas aussi simple.

Avant de présenter le tableau 4.11 avec les résultats globaux par chargeur dans le cas sans pénalité, nous pouvons remarquer, dans le tableau 4.9, que sur les 60 demandes de l'exemple, 10 d'entre elles ne présentent pas de gain sur au moins une règle de répartition. Comme précédemment, l'effet de la quantité en est la cause principale. La forte disparité des quantités transportées lors d'une tournée fait que certaines demandes se voient fortement pénalisées alors que les autres, au contraire, sont favorisées en terme de répartition de la distance parcourue. De plus, comme les demandes peuvent être réalisées durant n'importe quelle période, la répartition des demandes par période dans notre solution est surprenante, comme le montre le tableau 4.9. En effet durant la première période, 6 demandes de Mono-1, 7 demandes de Mono-2 et 11 demandes de Mono-3 ont été effectuées. Ainsi, seulement deux demandes de Mono-3 ont été livrées en période 3.

	<b>Mono-1</b>	<b>Mono-2</b>	<b>Mono-3</b>
<b>P1</b>	6	7	11
<b>P2</b>	6	10	7
<b>P3</b>	8	3	2

TABLE 4.9 – Distribution des demandes de Mono-i par période j : cas sans pénalités

Globalement, le gain sur les demandes est un peu plus faible, car la moyenne n'est que de 42% (tableau 4.10). Par contre, il est possible de remarquer que cette évolution n'est pas la même suivant les périodes : faible baisse pour la première, perte un peu plus accentuée pour la deuxième et une amélioration pour la troisième période. Cette dernière hausse peut s'expliquer par le fait que les demandes associées se retrouvent regroupées avec des demandes des autres périodes.

À partir du tableau 4.11, la conclusion est la même par rapport aux chargeurs : la baisse est globale, alors que pour les périodes 1 et 3 nous pouvons remarquer une amélioration. Toutefois, les inconvénients de la collaboration, tout regroupement de demandes n'étant pas forcément profitable, se trouvent exacerbés du fait que cela s'applique sur trois périodes et non plus une seule.

Demandes	Période 1			Période 2			Période 3		
	Qté	D	Qté*D	Qté	D	Qté*D	Qté	D	Qté*D
d1	0.56	0.48	0.59	0.88	0.40	0.90	0.31	0.52	0.59
d2	0.06	0.58	0.31	0.73	0.52	0.78	-0.50	0.22	-0.16
d3	0.29	0.29	0.14	0.98	0.45	0.98	0.93	0.36	0.94
d4	0.03	0.34	0.17	0.40	0.45	0.29	0.69	0.35	0.72
d5	0.23	0.55	0.18	0.36	0.41	0.15	0.57	0.48	0.49
d6	0.07	0.36	0.17	-0.46	0.36	-0.39	0.68	0.40	0.76
d7	0.49	0.52	0.34	0.39	0.58	0.29	0.24	0.34	0.15
d8	0.62	0.44	0.64	0.52	0.35	0.31	0.70	0.34	0.70
d9	0.71	0.62	0.71	-0.06	0.45	0.10	0.35	0.48	0.56
d10	0.57	0.41	0.67	0.69	0.62	0.67	0.78	0.24	0.82
d11	0.39	0.22	0.16	0.16	0.36	0.07	0.83	0.41	0.84
d12	0.61	0.22	0.69	-0.18	0.41	0.06	0.47	0.29	0.43
d13	0.55	0.48	0.49	0.47	0.51	0.30	0.54	0.48	0.41
d14	0.88	0.51	0.82	-0.17	0.40	-0.25	-0.26	0.24	-0.27
d15	0.07	0.62	0.05	0.92	0.62	0.91	0.61	0.29	0.63
d16	0.73	0.58	0.70	0.91	0.58	0.88	0.47	0.24	0.37
d17	0.20	0.48	0.22	0.27	0.44	0.25	-0.27	0.51	0.21
d18	0.90	0.36	0.85	0.35	0.48	0.52	-0.39	0.35	-0.02
d19	-0.01	0.24	0.10	0.97	0.55	0.97	0.62	0.48	0.65
d20	0.52	0.55	0.57	0.80	0.62	0.90	-0.32	0.29	-0.06
moyenne	<b>0.42</b>	<b>0.44</b>	<b>0.43</b>	<b>0.45</b>	<b>0.48</b>	<b>0.44</b>	<b>0.35</b>	<b>0.37</b>	<b>0.44</b>

TABLE 4.10 – Synthèse par demande pour l'instance multi-périodes sans pénalités

Chargeurs	Période 1			Période 2			Période 3		
	Qté	D	Qté*D	Qté	D	Qté*D	Qté	D	Qté*D
RC1	0.03	-0.20	-0.02	0.47	0.35	0.40	-0.27	0.51	0.21
RC2	0.49	0.52	0.34	-0.46	0.36	-0.39	0.62	0.48	0.65
RC3	0.32	0.10	0.42	0.57	0.14	0.40	0.47	0.24	0.37
RC4	0.62	0.44	0.64	0.27	0.33	0.18	0.68	0.40	0.76
RC5	0.23	0.55	0.18	0.98	0.45	0.98	-0.39	0.35	-0.02
RC6	-0.11	0.15	0.01	0.88	0.40	0.90	0.78	0.24	0.82
RC7	0.57	0.41	0.67	-0.14	0.27	-0.07	0.69	0.35	0.72
RC8	0.03	0.34	0.17	-0.37	-0.23	-0.11	0.61	0.29	0.63
RC9	-0.20	0.27	-0.22	0.06	0.21	-0.10	0.35	0.48	0.56
RC10	-0.04	-0.27	-0.21	0.69	0.62	0.67	0.70	0.34	0.70
RC11							0.93	0.36	0.94
RC12							0.31	0.52	0.59
RC13							0.47	0.29	0.43
RC14							0.83	0.41	0.84
RC15							-0.26	0.24	-0.27
RC16							-0.32	0.29	-0.06
RC17							0.57	0.48	0.49
RC18							-0.50	0.22	-0.16
RC19							0.24	0.34	0.15
RC20							0.54	0.48	0.41
moyenne	<b>0.20</b>	<b>0.23</b>	<b>0.19</b>	<b>0.30</b>	<b>0.29</b>	<b>0.29</b>	<b>0.35</b>	<b>0.37</b>	<b>0.44</b>

TABLE 4.11 – Synthèse par chargeur pour l'instance multi-périodes sans pénalités

Si nous revenons au tableau 4.9, nous remarquons également que les demandes des chargeurs sont effectuées essentiellement sur les deux premières périodes, où un plus grand nombre de demandes est livré que dans le cas mono-période. La période 3 est de ce fait moins chargée. Les tournées sur la première période sont un peu plus chargées, donc un peu plus longues. De même pour la période 3, le nombre de véhicules a été minimisé car le nombre de demandes à traiter est relativement faible.

Nous avons ensuite introduit les pénalités pour analyser quels en sont les impacts sur la collaboration. Les demandes de l'instance Mono- $X$ , avec  $X$  vaut de 1 à 3, peuvent être réalisées sur la période  $X$  sans aucune pénalité. Par contre, si une demande est réalisée dans une autre période  $Y$ , une pénalité est mise en place. Cette dernière est égale à  $200 * |X - Y|$ .

Les résultats présentés dans le tableau 4.13 semblent s'améliorer notamment sur les deux dernières périodes. Mais la répartition des demandes au cours de ces trois périodes a fortement changé pour les deux premières périodes, comme le montre le tableau 4.12.

	<b>Mono-1</b>	<b>Mono-2</b>	<b>Mono-3</b>
<b>P1</b>	4	9	10
<b>P2</b>	9	5	8
<b>P3</b>	7	6	2

TABLE 4.12 – Distribution des demandes de Mono- $i$  par période  $j$  : cas avec pénalités

Il est difficile d'appréhender le rôle des pénalités, car nous aurions pu penser que cela aurait forcé les demandes à être traitées dans leur période initiale. Mais, le seul changement que nous observons consiste dans le fait que les tournées solutions sont différentes (elles ne font pas intervenir les mêmes demandes), et que la distance totale parcourue par l'ensemble des tournées a diminué de l'ordre de 5%.

Demandes	Période 1			Période 2			Période 3		
	Qté	D	Qté*D	Qté	D	Qté*D	Qté	D	Qté*D
d1	0.83	0.66	0.82	0.91	0.51	0.91	-0.01	0.39	0.04
d2	0.32	0.66	0.48	0.74	0.40	0.73	0.06	0.26	0.09
d3	0.42	0.51	0.55	0.98	0.31	0.99	0.95	0.50	0.93
d4	-0.19	0.40	-0.02	0.64	0.59	0.68	0.75	0.38	0.81
d5	0.58	0.53	0.47	0.59	0.44	0.42	0.53	0.40	0.42
d6	-0.42	0.38	-0.54	0.00	0.00	0.00	0.77	0.51	0.78
d7	0.62	0.59	0.52	0.55	0.66	0.47	-0.14	0.31	0.00
d8	0.53	0.50	0.45	0.85	0.66	0.85	0.48	0.50	0.50
d9	0.00	0.00	0.00	-0.09	0.51	-0.04	0.47	0.44	0.66
d10	0.48	0.50	0.58	0.68	0.51	0.59	0.83	0.50	0.83
d11	0.76	0.41	0.64	0.23	0.31	0.12	0.72	0.51	0.76
d12	0.79	0.38	0.81	0.13	0.50	-0.08	0.55	0.36	0.67
d13	0.55	0.74	0.51	0.49	0.59	0.60	-0.01	0.36	-0.08
d14	0.77	0.39	0.67	0.04	0.41	0.01	0.15	0.44	0.30
d15	0.34	0.24	0.28	0.92	0.74	0.90	0.67	0.38	0.76
d16	0.47	0.74	0.51	0.83	0.74	0.80	-0.17	0.26	-0.20
d17	0.21	0.51	0.18	0.14	0.24	0.20	-0.34	0.50	0.11
d18	0.87	0.74	0.86	0.56	0.53	0.64	0.16	0.41	0.53
d19	0.43	0.53	0.52	0.93	0.26	0.93	0.62	0.36	0.60
d20	0.43	0.51	0.38	0.79	0.74	0.88	0.18	0.39	0.29
moyenne	<b>0.44</b>	<b>0.50</b>	<b>0.43</b>	<b>0.55</b>	<b>0.48</b>	<b>0.53</b>	<b>0.36</b>	<b>0.41</b>	<b>0.44</b>

TABLE 4.13 – Synthèse par demande pour l'instance multi-périodes avec pénalités

Chargeurs	Période 1			Période 2			Période 3		
	Qté	D	Qté*D	Qté	D	Qté*D	Qté	D	Qté*D
RC1	0.03	0.30	0.02	0.59	0.52	0.52	-0.34	0.50	0.11
RC2	0.62	0.59	0.52	0.00	0.00	0.00	0.62	0.36	0.60
RC3	0.45	0.53	0.54	0.73	0.51	0.71	-0.17	0.26	-0.20
RC4	0.53	0.50	0.45	0.32	0.28	0.29	0.77	0.51	0.78
RC5	0.58	0.53	0.47	0.98	0.31	0.99	0.16	0.41	0.53
RC6	-0.27	0.25	-0.33	0.91	0.51	0.91	0.83	0.50	0.83
RC7	0.48	0.50	0.58	0.08	0.26	-0.01	0.75	0.38	0.81
RC8	-0.19	0.40	-0.02	-0.28	-0.41	-0.18	0.67	0.38	0.76
RC9	-0.62	-0.71	-0.68	0.21	0.28	0.22	0.47	0.44	0.66
RC10	0.22	-0.08	0.15	0.68	0.51	0.59	0.48	0.50	0.50
RC11							0.95	0.50	0.93
RC12							-0.01	0.39	0.04
RC13							0.55	0.36	0.67
RC14							0.72	0.51	0.76
RC15							0.15	0.44	0.30
RC16							0.18	0.39	0.29
RC17							0.53	0.40	0.42
RC18							0.06	0.26	0.09
RC19							-0.14	0.31	0.00
RC20							-0.01	0.36	-0.08
moyenne	<b>0.18</b>	<b>0.28</b>	<b>0.17</b>	<b>0.42</b>	<b>0.28</b>	<b>0.40</b>	<b>0.36</b>	<b>0.41</b>	<b>0.44</b>

TABLE 4.14 – Synthèse par chargeur pour l'instance multi-périodes avec pénalités

Au niveau des gains par chargeur, il fallait s'assurer que ceux-ci permettent de couvrir les pénalités. Dans le tableau 4.14, l'évolution du gain moyen, entre les versions sans et avec pénalités varie de 29 à 33%. Mais cette évolution permettra-t-elle de prendre en charge les pénalités ? C'est une question qui reste en suspens, ne serait-ce que du fait du choix de la répartition des coûts d'une tournée.

#### 4.5.4 Bilan

Pour augmenter les avantages de la collaboration, il est important de comprendre que tout regroupement n'est pas forcément intéressant. Nous avons donné des pistes de règles qui permettraient de construire comme d'éviter certains regroupements. La collaboration entre deux demandes étant proches par les deux extrémités semble plus appropriée que celle entre des demandes qui ne sont voisines que par une extrémité. La règle de répartition de la distance de la tournée sur les différents chargeurs a une influence non négligeable sur les résultats. Nous pouvons supposer qu'en milieu urbain, on transportera essentiellement des colis d'un poids relativement semblable, donc nous pourrions exclure ce paramètre de la règle de répartition. Mais si nous transportons des marchandises industrielles, cette hypothèse n'est plus vérifiée. Même si le concept de pénalités est intéressant sur le papier, il reste néanmoins à le préciser. La méthode de calcul, par exemple basée sur des coûts de stockage, ne doit pas interdire de livrer une demande en dehors de sa période initiale par une valeur trop importante. Mais elle doit aussi limiter la combinatoire de l'affectation de la demande à une période par une valeur pas trop faible. N'oublions pas non plus que les résultats qui sont présentés ont été obtenus avec une métaheuristique. Donc, la solution est une solution approchée dont nous ne pouvons pas connaître l'écart avec la solution optimale. De plus, nous avons observé que cette méthode a tendance à charger les premières périodes. Mais, cette surcharge entraînant des tournées plus longues peut avoir un effet néfaste sur la collaboration. Toutefois, les essais ayant confirmé l'efficacité de notre métaheuristique sur de nombreux exemples, cette méthode est la seule permettant d'obtenir une solution en un temps raisonnable, même si la taille de l'instance est importante.

## 4.6 Conclusion

La satisfaction de toutes les demandes de transport est indispensable sur un horizon temporel donné, d'où la nécessité de planifier les tournées sur un ensemble de périodes. Le but de ce chapitre était alors de fournir des méthodes de résolution pour une nouvelle variante du PDP qui est le Multi-periods Pickup and Delivery Problem with Time Windows and Paired Demands (Mu-PDPTWPD).

Après en avoir proposé une modélisation sous forme d'un programme linéaire mixte en nombres entiers, nous avons montré qu'une résolution optimale de ce modèle global, et parfois même l'obtention d'une solution faisable, est très difficile de par la complexité de cette variante. En effet, à la complexité d'un problème

---

de PDPTWPD classique, vient s'ajouter l'affectation de chaque demande à une période de l'horizon considéré. Pour y remédier, nous avons d'abord élaboré et testé des méthodes reposant sur une stratégie abordant le Mu-PDPTWPD comme une séquence de SPDPTWPDs (variante décrite et résolue aux chapitres 2 et 3). Elles fournissent donc des solutions pour lesquelles l'optimalité n'est pas garantie (en tout cas pas atteinte pour la plupart des instances traitées). Nous avons ensuite développé une méthode métaheuristique, considérant le problème dans sa globalité et qui nous permet de plus de prendre en compte un nouvel objectif lié aux pénalités d'avance ou de retard de livraison. Cette nouvelle stratégie se matérialise sous forme d'une combinaison entre un algorithme génétique hybridé avec une méthode de recherche locale et un GRASP. Les résultats fort encourageants de nos tests nous ont poussés à confronter cette nouvelle méthode avec l'intérêt d'une collaboration entre chargeurs. Nous en avons déduit que la collaboration est intéressante mais pas sous n'importe quelle condition. Pour ne pas allonger inutilement les tournées, il est préférable de ne pas vouloir toujours regrouper les demandes. De plus, l'estimation des gains dépend fortement de la règle de répartition utilisée. Mais compte tenu de l'hétérogénéité des flux urbains, en termes de quantités et de distance à parcourir, ce choix ne s'avère pas si simple.





# Conclusion et perspectives

Cette thèse est le fruit de trois années de travail de recherche. Pendant cette période, nous nous sommes concentrés sur la résolution de problèmes de planification de tournées en nous focalisant sur deux nouvelles variantes du problème de collecte et de livraison. Ces dernières peuvent traduire la collaboration entre les chargeurs dont les demandes de transport peuvent être regroupées et confiées à un seul transporteur qui essaie, à son tour, de mutualiser ses véhicules en transportant simultanément plusieurs demandes de transport.

Cette collaboration, sous les deux formes que nous avons nommées active et passive, était le point de départ de nos travaux de recherche dans le cadre du projet ANR-TCDU (voir figure 5.1). La collaboration "active", qui repose sur le fait que les chargeurs prennent l'initiative et décident de collaborer, est traitée dans les chapitres 2 et 3. L'interprétation de ce type de collaboration nous a conduit à la résolution de la variante SPDPTWPD dans laquelle les chargeurs regroupent leurs demandes sur une seule période. Cette variante prend en considération l'incapacité du transporteur à honorer toutes les demandes. Le but est donc de choisir celles qui améliorent le gain (en minimisant le coût) d'une éventuelle collaboration. La collaboration "passive" est indirectement initiée par les chargeurs qui se mettent d'accord simplement sur l'idée de collaborer sur un horizon temporel divisé en plusieurs périodes. Chaque chargeur choisit sa période de préférence et il peut accepter une visite anticipée ou ultérieure contre une pénalité. Le transporteur optimise alors ses tournées par période de manière à minimiser le coût total sur l'ensemble des périodes. La variante Mu-PDPTWPD traitée au chapitre 4, correspond à ce type de collaboration. Dans ce même chapitre, nous avons notamment résolu le Mu-PDPTWPD en le considérant comme une séquence de SPDPTWPDs. Ce traitement englobe à la fois la collaboration "active" à chaque période et "passive" à cause de l'existence d'un ensemble de périodes.

<b>Collaboration « active »</b>	<b>Collaboration « passive »</b>	
<b>SPDPTWPD</b>	<b>Mu-PDPTWPD</b>	
<b>Mono-période</b>	<b>Séquentiel</b>	<b>Global</b>
	<b>Multi-périodes</b>	

FIGURE 5.1 – Relation entre type de collaboration et variante traitée

Si on s'extrait du contexte du projet ANR-TCDU, et que l'on se place plutôt

d'un point de vue académique, alors nos principales contributions sont les suivantes, schématisées par la figure 5.2 :

- formalisation, modélisation et résolution de deux nouvelles variantes de problèmes de tournées.
- génération de nouveaux jeux de tests, utiles pour ces variantes.
- étude d'un point de vue mono-objectif, comme bi-objectifs.
- pour chaque cas (variante, nombre d'objectifs), élaboration, tests et comparaisons de plusieurs algorithmes basés sur des méthodes exactes et approchées.

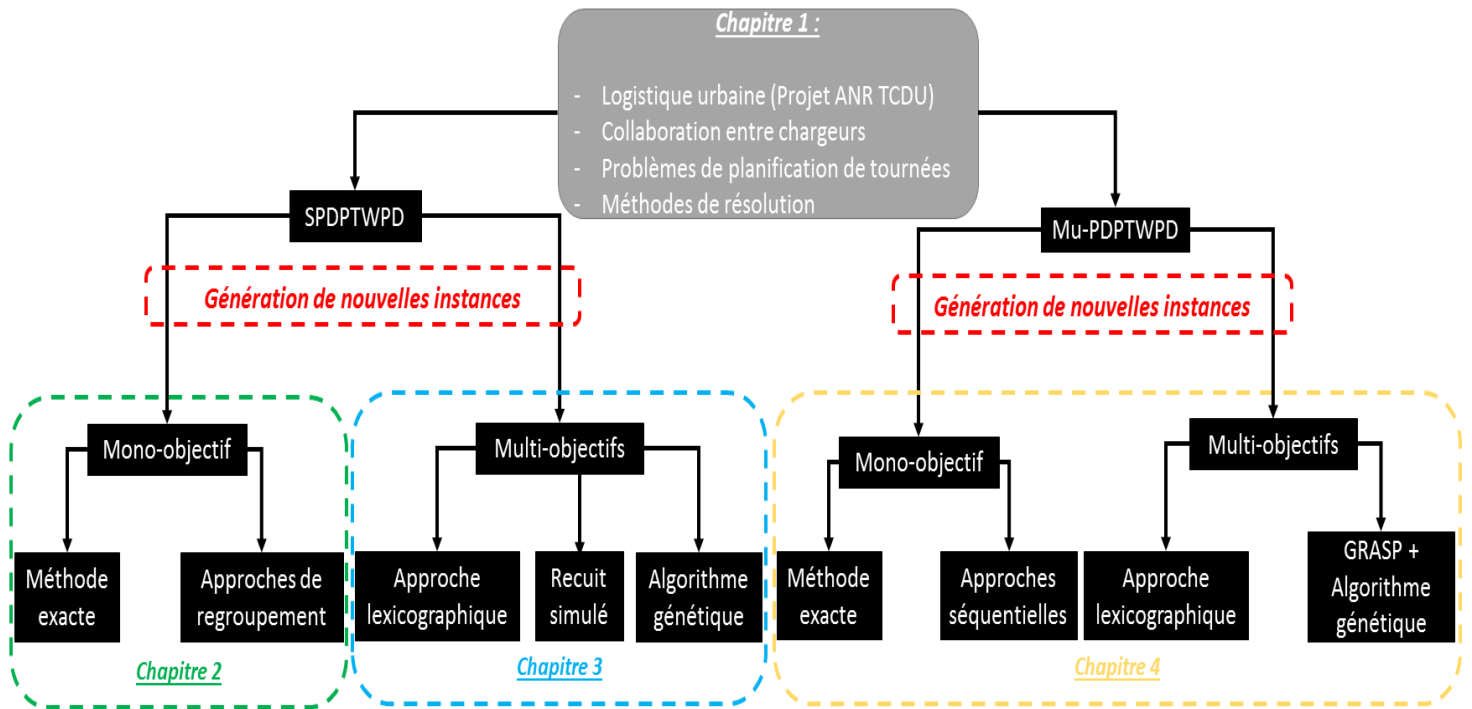


FIGURE 5.2 – Principales contributions

Les modèles et les méthodes développés devraient être prochainement intégrés dans la plateforme web d'aide à la décision qui sera délivrée à l'issue du projet ANR TCDU (date prévue : mars 2019). À l'aide de cette plateforme, les chargeurs pourront déposer leurs demandes de transport en précisant toutes les informations indispensables afin de satisfaire ces demandes (quantité à transporter, point de collecte, point de livraison, etc). Ensuite, une de nos méthodes sera utilisée pour regrouper ces demandes en plusieurs lots en optimisant un ou plusieurs objectifs donnés dans le but de garantir aux chargeurs le meilleur prix. Les résultats fournis par nos méthodes constitueront les données d'entrée pour les différents algorithmes développés par les autres partenaires du projet. À la fin de la démarche d'optimisation, le système d'aide à la décision fournira une solution finale qui consiste à affecter chaque demande

de transport déposée par un chargeur à un transporteur spécifique. Pour finir, la plateforme aidera les transporteurs à optimiser leurs tournées finales prenant en considération l'ensemble des demandes à satisfaire.

Les travaux décrits dans ces différents chapitres ont été publiés dans deux revues internationales, sept conférences internationales et trois conférences et colloques nationaux. Ils ont également été présentés sous la forme d'une communication lors d'une journée scientifique du GDR Macs. De plus, ils ont donné lieu à deux soumissions récentes, à une revue internationale et à une revue nationale (voir la liste globale des publications à la suite de cette partie).

En termes de perspectives et en plus de l'implémentation /utilisation de nos algorithmes dans le prototype du projet ANR TCDU, plusieurs travaux sont envisagés dans le futur. En premier lieu, des améliorations des approches proposées peuvent être faites. Bien que les modèles mathématiques proposés aient réussi à résoudre les variantes considérées, des extensions de ces modèles permettraient de prendre en compte d'autres aspects rencontrés dans les applications réelles : par exemple, l'existence de plusieurs dépôts, la possibilité de diviser les opérations de collecte et de livraison d'une même demande, la possibilité de livrer une demande avec deux ou plusieurs véhicules en passant par un ou plusieurs points de transfert de types parkings ou points relais, etc. En outre, l'ajout d'objectifs supplémentaires serait intéressant, comme la minimisation des émissions de CO<sub>2</sub>.

La performance des méthodes exactes proposées est également un domaine d'amélioration, en particulier au niveau du temps de résolution. Pour répondre à ce défi, plusieurs solutions peuvent être envisagées, par exemple un travail sur la modélisation mathématique elle-même afin de réduire le nombre des variables et des contraintes et donc faciliter la résolution du problème (une première tentative dans ce sens a été introduite en sous-section 2.5). De même, l'utilisation d'autres méthodes exactes représente une bonne piste à explorer, tout en restant conscients des limitations de telles méthodes pour résoudre les grandes instances en un temps de calcul raisonnable.

Les méthodes approchées que nous avons développées peuvent aussi faire l'objet d'améliorations. Par exemple, les différents paramètres des métaheuristiques peuvent être étudiés plus profondément en effectuant des analyses statistiques supplémentaires et ainsi atteindre leurs meilleures valeurs. Ce problème de réglage peut être résolu en réalisant un plan d'expérience ou de manière plus complexe en construisant un module supplémentaire afin d'extraire automatiquement les valeurs optimales des paramètres pour chaque instance.

L'utilisation des algorithmes génétiques hybrides et du recuit simulé en particulier est également une question à aborder. En effet, nous avons utilisé dans ce travail un algorithme génétique comme une métaheuristique basée sur une population de solutions afin de résoudre les variantes considérées. D'autres méthodes basées sur

la population telles que les algorithmes de colonies de fourmis et l'optimisation par essais particuliers (PSO) peuvent également être testées et comparées aux algorithmes génétiques proposés. Une récente étude, dans laquelle j'ai collaboré avec un autre doctorant de l'équipe, a fait l'objet d'une soumission à une revue internationale. Dans ce travail, nous avons utilisé la PSO pour résoudre le SPDPTWPD en ajoutant la possibilité qu'une demande puisse passer par un ou plusieurs points de transfert. En ce qui concerne les métaheuristiques à base de solution unique, d'autres méthodes peuvent être appliquées et comparées avec notre recuit simulé, par exemple : ALNS (Adaptive Large Neighborhood Search), etc.

La quantification des gains espérés par la collaboration entre les chargeurs a ouvert de nouvelles perspectives, non seulement dans le choix de la méthode de répartition des coûts, mais aussi dans le fait de considérer ces gains comme un objectif à maximiser. En effet, dans nos diverses approches, nous avons considéré la distance comme un objectif représentatif des coûts/gains associés aux demandes. Néanmoins en pratique, le coût effectif n'est pas simplement proportionnel à la distance parcourue, d'autres facteurs sont à prendre en compte tels que la quantité transportée (ou le poids, ou le volume...). De plus, la détermination du gain par demande réalisée n'est pas obligatoirement la meilleure approche pour évaluer l'intérêt d'une collaboration. Le gain par chargeur est à considérer, même s'il est difficile à calculer si les demandes sont réparties sur plusieurs tournées donc potentiellement entre plusieurs transporteurs. Néanmoins, intégrer la maximisation de ces gains (de leur moyenne par exemple ou de leur minimum) comme un des objectifs dans nos algorithmes permettrait sans doute d'évaluer encore plus précisément, et en tout cas plus directement, les avantages d'une collaboration entre chargeurs. Par ailleurs, il peut s'avérer intéressant, voire nécessaire, de définir des principes pour une collaboration fructueuse, c'est-à-dire un ensemble de règles qui indiqueraient les regroupements de demandes à envisager ou à éviter, en amont des algorithmes de résolution ou à intégrer à ces derniers.

Finalement, la robustesse de solutions est un point qui ne cesse pas d'attirer l'attention de la communauté scientifique. Des travaux post-doctorat seront consacrés au traitement des variantes précédemment étudiées dans un contexte robuste où le but est d'avoir une solution qui reste bonne pour différents scénarios possibles. Chaque scénario peut représenter une incertitude donnée, par exemple, nous pourrions gérer les incertitudes au niveau du temps de déplacement entre deux sites.

# Publications

## Revue internationale :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "A lexicographic approach for the bi-objective selective pickup and delivery problem with time windows and paired demands", *Annals of Operations Research*, pp.1-19, doi :10.1007/s10479-017-2500-9.
2. Al Chami, Z., Manier, H., Manier, M.-A., & Chebib, E.. "An advanced GRASP-HGA combination to solve a multi-period Pickup and Delivery Problem", *Expert Systems with Applications*, Volume 105, 2018, Pages 262-272, doi : 10.1016/j.eswa.2018.03.040.
3. Al Chami, Z., Manier, H., & Manier, M.-A.. "Multi-Period Pickup and Delivery Problem with Time Windows and Paired Demands", *Computers & Industrial Engineering*. (soumis)
4. Peng, Z., Al Chami, Z., Manier, H., & Manier, M.-A.. "Multi-Objectives Particle Swarm Optimization for selective pickup and delivery problem with transfers", *Swarm and Evolutionary Computation*. (soumis)

## Revue nationale :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "A new compact two-indexes formulation for a Pickup and Delivery Problem", *Revue Recherche Opérationnelle de OpenScience*, numéro spécial "Jeunes Chercheurs du GDR MACS au congrès mondial de l'IFAC WC 2017 - JD MACS 2017". (soumis)

## Conférences internationales :

1. Al Chami, Z., Manier, H., Manier, M.-A., & Khalil, M.. "Two cluster-based approaches for the Pick-up and Delivery Problem with Time Windows", *5th annual workshop of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog2016)*, Nantes, France (june 6-8, 2016).
2. Al Chami, Z., Manier, H., & Manier, M.-A.. "Exact Model for a Selective Pick up and Delivery Problem with Time Windows", *session poster à 8th IFAC/IEEE/IFORS/IFIP Conference on Manufacturing Modelling Management and Control (MIM 2016)*, Troyes, France (june 28-30, 2016).
3. Manier, H., Manier, M.-A., & Al Chami, Z.. "Shippers' collaboration in city logistics", *8th IFAC/IEEE/IFORS/IFIP Conference on Manufacturing Modelling Management and Control (MIM 2016)*, IFAC-PapersOnLine, Volume 49, Issue 12, pp. 1880-1885, Troyes, France (june 28-30, 2016).

4. Al Chami, Z., Manier, H., & Manier, M.-A.. "New model for a variant of Pick up and Delivery Problem", *IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2016)*, pp.1708-1713, Budapest, Hungary (October 9-12, 2016).
5. Al Chami, Z., Manier, H., Manier, M.-A., & Fitouri, C.. "A hybrid genetic algorithm to solve a multi-objective Pickup and Delivery Problem", *20th World Congress of the International Federation of Automatic Control (IFAC WC'17)*, IFAC-PapersOnLine, Volume 50, Issue 1, pp. 14656-14661, Toulouse, France (9-14 July 2017).
6. Al Chami, Z., Manier, H., & Manier, M.-A.. "A new compact two-index formulation for a Pickup and Delivery Problem", *20th World Congress of the International Federation of Automatic Control (IFAC WC'17)*, Toulouse, France (9-14 July 2017).
7. Al Chami, Z., El Flity, H., Manier, H., & Manier M.-A.. "A New Metaheuristic to Solve a Selective Pickup and Delivery Problem", *4th International Conference on Logistics Operations Management (GOL'18)*, Le Havre, France (April 10-12, 2018).

#### Conférences nationales :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "Méthode exacte pour la résolution du problème de collecte et de livraison sélectif avec fenêtres temporelles", *Dix-septième Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'16)*, Compiègne, France (10-12 février 2016).
2. Al Chami, Z., Manier, H., & Manier, M.-A.. "Un algorithme génétique amélioré pour un problème de ramassage et de livraison", *Colloque FUTUR-MOB'17 : Préparer la transition vers la mobilité autonome*, Montbéliard, France (5-7 septembre 2017).
3. Al Chami, Z., Manier, H., & Manier, M.-A.. "Différentes approches pour la résolution d'un problème multipériodes de collecte et de livraison", *Dix-neuvième Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'18)*, Lorient, France (21-23 février 2018).

#### Journée scientifique :

1. Al Chami, Z., Manier, H., & Manier, M.-A.. "Modélisation des problèmes de tournées riches en logistique urbaine", *Journées STP du GDR MACS session du Groupe de Travail Bermudes*, Nancy, (9-10 novembre 2017).

# Annexes

## A.1 Création d'instances pour le SPDPTWPD

Dans cette section, nous présentons la façon avec laquelle nous avons généré les nouvelles instances du SPDPTWPD. Comme nous l'avons déjà évoqué, ces jeux d'instances ont été générés afin de tester les méthodes développées aux chapitres 2 et 3. Les nouvelles instances sont obtenues à partir des instances de référence élaborées par [Li & Lim 2003] basées sur les instances du VRPTW créées par [Solomon 1987]. Deux modifications majeures ont été faites : l'ajout d'un profit par site (fixe pour les instances du groupe 1 et variable pour celles du groupe 2) et la possibilité d'avoir trois types de flux.

1	200								
0	40	50	0	0	1236	0	0	0	0
1	45	70	-20	0	1125	90	0	0	10
2	42	68	-20	727	782	90	0	0	15
3	38	70	-10	534	605	90	13	0	4
4	35	66	-10	357	410	90	14	0	5
5	35	69	-20	448	505	90	15	0	10
6	25	85	-30	0	1107	90	16	0	30
7	22	85	-40	567	620	90	0	0	10
8	20	80	-20	384	429	90	18	0	2
9	15	80	-20	278	345	90	19	0	10
10	30	52	-10	0	1135	90	20	0	5
11	38	68	20	255	324	90	0	0	10
12	40	69	20	621	702	90	0	0	4
13	42	65	10	0	1130	90	0	3	10
14	42	66	10	0	1129	90	0	4	5
15	40	66	20	0	1130	90	0	5	10
16	22	75	30	30	92	90	0	6	15
17	20	85	40	475	528	90	0	0	10
18	15	75	20	179	254	90	0	8	30
19	18	75	20	99	148	90	0	9	10
20	28	55	10	732	777	90	0	10	12

FIGURE A.1 – Exemple d'une instance du SPDPTWPD

La figure A.1 montre un exemple d'une instance avec 20 nœuds et 1 véhicule. La première ligne donne les informations liées à la flotte de véhicules (leur nombre et leurs capacités). En ce qui concerne les autres lignes, elles donnent les informations liées avec les nœuds ainsi que le dépôt identifié par 0. La première colonne donne alors l'identifiant de chaque site. Les coordonnées sont illustrées par la deuxième et la troisième colonne. La quantité de marchandises est donnée par la quatrième colonne. Les dates d'ouverture et de fermeture sont représentées respectivement par la cinquième et la sixième colonne. Pour les fournisseurs, l'identifiant du fournisseur (fourni par la septième colonne) est égal à 0, tandis que l'identifiant du client correspondant est donné par la huitième colonne. Cet identifiant est parfois égal à 0 comme par exemple pour le site 11. Cela veut dire que ce site est appairé avec le dépôt. Également, pour les clients, l'identifiant client dans la figure A.1 est égal à 0, et l'identifiant fournisseur est soit égal à 0 si appairage avec le dépôt (par exemple le

site 1), soit non nul si le client est appairé avec un autre site (par exemple le site 5). La dernière colonne montre le profit associé à chaque site. L'instance montrée, à titre d'exemple dans cette section, appartient au deuxième groupe parce que le profit n'est pas fixe. Ces instances sont accessibles via ce lien : [https://drive.google.com/drive/folders/1ic9RwY1qqNhAA\\_69-hnuqZxKXj4eZGRm?usp=sharing](https://drive.google.com/drive/folders/1ic9RwY1qqNhAA_69-hnuqZxKXj4eZGRm?usp=sharing).

## A.2 Création d'instances pour le Mu-PDPTWPD

D'autres nouvelles instances ont été générées dans le cadre de cette thèse afin de valider les méthodes développées au chapitre 4 pour résoudre le Mu-PDPTWPD. Ces instances sont réparties sur deux ensembles : pour le premier, les instances sont générées de façon aléatoire et nous pourrions avoir des demandes qui peuvent être honorées sur une ou plusieurs périodes parmi celles existantes. En ce qui concerne le deuxième ensemble, nous utilisons les instances du SPDPTWPD sur lesquelles nous ajoutons diverses périodes tout en considérant que chaque demande peut être honorée sur toutes ces périodes.

À titre d'exemple, l'instance illustrée dans la figure A.2 appartient à l'ensemble 1. Les quatre premières lignes donnent respectivement le nombre de dépôts, le nombre de fournisseurs et celui de clients ainsi que le nombre de véhicules. La vitesse de chaque véhicule est donnée par la cinquième ligne. Les trois lignes suivantes peuvent être négligées. En effet, elles représentent des variables spécifiques utilisées pour calculer le coût du transport pour une distance donnée. Mais dans nos approches, nous avons considéré la distance comme fonction objectif et donc nous ne les avons pas prises en considération. La neuvième ligne montre la capacité maximale pour chaque véhicule. Le temps de service par site est donné par la ligne 10. Les lignes restantes donnent respectivement la matrice de distances entre tous les nœuds, les fournisseurs de chaque client, les clients de chaque fournisseur, les fenêtres temporelles par site pour l'ensemble de périodes, les pénalités par site à payer lors d'une visite anticipée ou retardée et la quantité demandée par chaque site. Toutes ces instances sont accessibles via ce lien : <https://drive.google.com/drive/folders/1c7cZAvEDtNDj1oVK1Af0HcWPnMj2EkjF>.



```

w=1;
f=5;
c=5;
v=1;
Vitesse=[2];
CK=[0.476];
CC=[19.04];
CJ=[155.69];
Q=[200];
S=[0,10,10,10,10,10,10,10,10,10];
dd=[
[0,92.9731,47.8853,67.082,36.0555,98.0867,57.3847,44.7772,53.8516,46.5725,56.4358],
[92.9731,0,66.0984,84.0952,118.727,9.21954,45.7056,66.6108,42.0476,125.98.4937],
[47.8853,66.0984,0,26.4008,83.672,67.0671,20.3961,3.16228,26.9258,93.7763,33.1059],
[67.082,84.0952,26.4008,0,102.84,82.5409,41.7732,28.7924,50.9902,113.565,21.4709],
[36.0555,118.727,83.672,102.84,0,125.431,90.6035,80.5295,84.8999,11,90.1388],
[98.0867,9.21954,67.0671,82.5409,125.431,0,47.0106,68,45.3542,132.136,98.4784],
[57.3847,45.7056,20.3961,41.7732,90.6035,47.0106,0,21.0238,10.4403,99.5691,53.1413],
[44.7772,66.6108,3.16228,28.7924,80.5295,68,21.0238,0,26.4008,90.6201,33.6155],
[53.8516,42.0476,26.9258,50.9902,84.8999,45.3542,10.4403,26.4008,0,93.2363,59.9416],
[46.5725,125,93.7763,113.565,11,132.136,99.5691,90.6201,93.2363,0,101.124],
[56.4358,98.4937,33.1059,21.4709,90.1388,98.4784,53.1413,33.6155,59.9416,101.124,0]];
Fournisseur=[1,3,1,6,1];
Client=[1,8,1,1,10];
fenetre=[
[<0,440>,<0,440>,<0,440>,<0,440>,<0,440>,<0,440>,<0,440>,<0,440>,<0,440>,<0,440>],
[<0,0>,<0,0>,<0,0>,<0,0>,<64,103>,<64,103>,<0,0>,<0,0>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<95,134>,<95,134>,<0,0>,<0,0>,<0,0>],
[<61,112>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<65,123>,<65,123>,<65,123>,<0,0>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<80,119>,<80,119>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<65,102>,<65,102>,<0,0>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<308,361>,<308,361>,<0,0>,<0,0>,<0,0>],
[<0,0>,<63,109>,<63,109>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<246,298>,<246,298>,<0,0>,<0,0>],
[<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<0,0>,<65,97>,<0,0>,<0,0>]];
penalite=[
[1000,1000,1000,1000,0,0,1000,1000,1000,1000],
[1000,1000,1000,1000,1000,0,0,1000,1000,1000],
[0,1000,1000,1000,1000,1000,1000,1000,1000,1000],
[1000,1000,1000,1000,0,0,0,1000,1000,1000],
[1000,1000,1000,1000,1000,1000,0,0,1000,1000],
[1000,1000,1000,1000,1000,1000,0,1000,1000,1000],
[1000,0,0,1000,1000,1000,1000,1000,1000,1000],
[1000,1000,1000,1000,1000,1000,0,0,1000,1000],
[1000,1000,1000,1000,1000,1000,1000,1000,0,1000,1000]];
q=[0,77,61,73,76,72,-9,-61,-61,-72,-45];

```

FIGURE A.2 – Exemple d’une instance du Mu-PDPTWPD

### A.3 Création des instances utilisées dans la section 4.5

Les instances utilisées en section 4.5 afin de valider l’avantage de collaborer entre les chargeurs ont été générées en deux temps. Premièrement, trois instances mono-périodes ont été créées aléatoirement avec 20 demandes par instance. Nous avons nommé ces instances Mono-x avec x vaut de 1 à 3. Par exemple, l’instance Mono-1 est illustrée dans la figure A.3. Les deux premières lignes montrent respectivement le nombre de périodes (égal à 1 dans le cas mono-période) et la durée maximale. Ensuite, les lignes suivantes donnent le nombre de dépôts, fournisseurs, clients, demandes et véhicules. La vitesse moyenne et la capacité maximale de chaque véhicule sont fournies dans les lignes 8 et 9. La dixième ligne n’est pas utilisée par nos approches. En effet, elle associe chaque véhicule à un dépôt mais comme nous traitons des variantes mono-dépôt, nous pouvons négliger cette ligne. Pour la matrice de distances

et les fenêtres temporelles, nous commençons par le dépôt puis les fournisseurs pour terminer avec les clients. Dans la dernière partie de l'instance, nous avons la liste de demandes à satisfaire. Pour chaque demande, nous montrons respectivement l'identifiant du site fournisseur et celui du client, la quantité à transporter, le temps de chargement et celui de déchargement ainsi que le profit.

```

Number_of_periods=1;
Duration_of_periods=2000;
Number_of_depots=1;
Number_of_suppliers=10;
Number_of_customers=20;
Number_of_demands=20;
Number_of_vehicles=20;
Average_speed=[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1];
Maximal_capacity=[100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100];
Associated_depot=[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1];
Distances_matrix=[
[0,37.7359,...,36.4006],
.
.
.
[36.4006,...,0]];
Time_window=[
[<0,440>],
[<37,67>],
.
.
.
[<80,380>]];
Demands=[
[1,24,20,10,12,12],
.
.
.
[6,17,28,7,11,38]];

```

FIGURE A.3 – Exemple d'une instance Mono-x

Site	Période souhaitée
<b>Dépôt</b>	
<b>Fournisseurs Mono-1</b>	<b>1</b>
<b>Fournisseurs Mono-2</b>	<b>2</b>
<b>Fournisseurs Mono-3</b>	<b>3</b>
<b>Clients Mono-1</b>	<b>1</b>
<b>Clients Mono-2</b>	<b>2</b>
<b>Clients Mono-3</b>	<b>3</b>

FIGURE A.4 – Création des instances multi-périodes

Deuxièmement, deux instances multi-périodes ont été générées en concaténant les trois instances mono-période (voir figure A.4). La différence entre les deux instances multi-périodes est la présence des pénalités dans une d'elles. Toutes ces instances sont accessibles via ce lien : [https://drive.google.com/open?id=1KaVAwk4G396wQeQGokcxgc00SLbv\\_u2](https://drive.google.com/open?id=1KaVAwk4G396wQeQGokcxgc00SLbv_u2)



# Bibliographie

- [Abdoun *et al.* 2016] Otman Abdoun, Chakir Tajani, Jaafar Abouchabaka et Haimoudi El Khatir. *Improved Genetic Algorithm to Solve Asymmetric Traveling Salesman Problem*. *Int. J. Open Problems Compt. Math*, vol. 9, no. 4, pages 42–55, 2016. (Cité en page 17.)
- [Adewumi & Adeleke 2016] Aderemi Oluyinka Adewumi et Olawale Joshua Adeleke. *A survey of recent advances in vehicle routing problems*. *International Journal of System Assurance Engineering and Management*, pages 1–18, 2016. (Cité en page 19.)
- [Aghezzaf & El Fahim 2015] Brahim Aghezzaf et Hassan El Fahim. *Solving the capacitated team orienteering problem with time windows through variable neighborhood search*. *International Review on Computers and Software (IRECOS)*, vol. 10, no. 11, pages 1134–1142, 2015. (Cité en page 23.)
- [Ai & Kachitvichyanukul 2009] The Jin Ai et Voratas Kachitvichyanukul. *Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem*. *Computers & Industrial Engineering*, vol. 56, no. 1, pages 380–387, 2009. (Cité en page 20.)
- [Andreatta & Lulli 2008] Giovanni Andreatta et Guglielmo Lulli. *A multi-period TSP with stochastic regular and urgent demands*. *European Journal of Operational Research*, vol. 185, no. 1, pages 122–132, 2008. (Cité en page 18.)
- [Avcı & Topaloglu 2016] Mustafa Avcı et Seyda Topaloglu. *A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery*. *Expert Systems with Applications*, vol. 53, pages 160–171, 2016. (Cité en page 25.)
- [Belhaiza *et al.* 2014] Slim Belhaiza, Pierre Hansen et Gilbert Laporte. *A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows*. *Computers & Operations Research*, vol. 52, pages 269–281, 2014. (Cité en pages 20 et 102.)
- [Bellman 1952] Richard Bellman. *On the theory of dynamic programming*. *Proceedings of the National Academy of Sciences*, vol. 38, no. 8, pages 716–719, 1952. (Cité en page 28.)
- [Belloso *et al.* 2015] Javier Belloso, Angel A Juan, Javier Faulin et Adrián Serrano. *Using multi-start biased randomization of heuristics to solve the vehicle routing problem with clustered backhauls*. In *Proc. ICAOR*, page 16, 2015. (Cité en page 24.)
- [Belloso *et al.* 2017] Javier Belloso, Angel A Juan, Enoc Martinez et Javier Faulin. *A biased-randomized metaheuristic for the vehicle routing problem with clustered and mixed backhauls*. *Networks*, vol. 69, no. 3, pages 241–255, 2017. (Cité en page 24.)

- [Ben-Said *et al.* 2016] Asma Ben-Said, Racha El-Hajj et Aziz Moukrim. *An adaptive heuristic for the Capacitated Team Orienteering Problem*. IFAC-PapersOnLine, vol. 49, no. 12, pages 1662–1666, 2016. (Cité en page 23.)
- [Benavent *et al.* 2015] Enrique Benavent, Mercedes Landete, Enrique Mota et Gregorio Tirado. *The multiple vehicle pickup and delivery problem with LIFO constraints*. European Journal of Operational Research, vol. 243, no. 3, pages 752–762, 2015. (Cité en pages 29 et 33.)
- [Benjelloun *et al.* 2010] Abderrahim Benjelloun, Teodor Gabriel Crainic et Yvon Bigras. *Towards a taxonomy of City Logistics projects*. Procedia-Social and Behavioral Sciences, vol. 2, no. 3, pages 6217–6228, 2010. (Cité en page 12.)
- [Bettinelli *et al.* 2014] Andrea Bettinelli, Alberto Ceselli et Giovanni 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, pages 171–197, 2014. (Cité en pages 28 et 33.)
- [Biggs *et al.* 1976] Norman Biggs, E Keith Lloyd et Robin J Wilson. *Graph theory*, 1736-1936. Oxford University Press, 1976. (Cité en page 16.)
- [Bortfeldt *et al.* 2015] Andreas Bortfeldt, Thomas Hahn, Dirk Männel et Lars Mönch. *Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints*. European Journal of Operational Research, vol. 243, no. 1, pages 82–96, 2015. (Cité en page 24.)
- [Butt & Cavalier 1994] Steven E Butt et Tom M Cavalier. *A heuristic for the multiple tour maximum collection problem*. Computers & Operations Research, vol. 21, no. 1, pages 101–111, 1994. (Cité en page 21.)
- [Carrabs *et al.* 2007] Francesco Carrabs, Raffaele Cerulli et Jean-François Cordeau. *An additive branch-and-bound algorithm for the pickup and delivery traveling salesman problem with LIFO or FIFO loading*. INFOR : Information Systems and Operational Research, vol. 45, no. 4, pages 223–238, 2007. (Cité en pages 28, 32 et 33.)
- [Chen *et al.* 2014] Qingfeng Chen, Kunpeng Li et Zhixue Liu. *Model and algorithm for an unpaired pickup and delivery vehicle routing problem with split loads*. Transportation Research Part E : Logistics and Transportation Review, vol. 69, pages 218–235, 2014. (Cité en page 25.)
- [Chen *et al.* 2017] Yushi Chen, Shunli Ma, Xi Chen et Pedram Ghamisi. *Hyper-spectral data clustering based on density analysis ensemble*. Remote Sensing Letters, vol. 8, no. 2, pages 194–203, 2017. (Cité en page 53.)
- [Coene *et al.* 2010] Sofie Coene, A Arnout et Frits CR Spieksma. *On a periodic vehicle routing problem*. Journal of the Operational Research Society, vol. 61, no. 12, pages 1719–1728, 2010. (Cité en page 52.)
- [Dai & Chen 2012] Bo Dai et Haoxun Chen. *Mathematical model and solution approach for carriers' collaborative transportation planning in less than truckload transportation*. International Journal of Advanced Operations Management, vol. 4, no. 1-2, pages 62–84, 2012. (Cité en page 15.)

- [Dantzig & Ramser 1959] George B Dantzig et John H Ramser. *The truck dispatching problem*. Management science, vol. 6, no. 1, pages 80–91, 1959. (Cité en page 18.)
- [Dantzig & Wolfe 1960] George B Dantzig et Philip Wolfe. *Decomposition principle for linear programs*. Operations research, vol. 8, no. 1, pages 101–111, 1960. (Cité en page 28.)
- [Dayarian *et al.* 2015] Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau et Walter Rei. *A branch-and-price approach for a multi-period vehicle routing problem*. Computers & Operations Research, vol. 55, pages 167–184, 2015. (Cité en pages 20 et 102.)
- [Dayarian *et al.* 2016] Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau et Walter Rei. *An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem*. Transportation Research Part E : Logistics and Transportation Review, vol. 95, pages 95–123, 2016. (Cité en pages 20 et 102.)
- [Deng *et al.* 2015] Yong Deng, Yang Liu et Deyun Zhou. *An improved genetic algorithm with initial population strategy for symmetric TSP*. Mathematical Problems in Engineering, vol. 2015, 2015. (Cité en page 17.)
- [Dias & Tsuzuki 2010] Guilherme Marques Dias et Marcos de Sales Guerra Tsuzuki. *System for Shippers Collaboration*. IFAC Proceedings Volumes, vol. 43, no. 4, pages 240–245, 2010. (Cité en page 14.)
- [Durand *et al.* 2010] Bruno Durand, Jesus Gonzales-Féliu et Frédéric Henriot. *La logistique urbaine, facteur clé de développement du B to C*. Logistique & Management, vol. 18, no. 2, pages 7–19, 2010. (Cité en page 4.)
- [Ehrgott 2005] Matthias Ehrgott. *Multicriteria Optimization. vol. 491 of Lecture Notes in Economics and Mathematical Systems*, 2005. (Cité en page 65.)
- [El-Hajj 2015] Racha El-Hajj. *Vehicle routing problems with profits, exact and heuristic approaches*. PhD thesis, Compiègne, 2015. (Cité en pages 21 et 47.)
- [Ergun *et al.* 2007] Özlem Ergun, Gültekin Kuyzu et Martin Savelsbergh. *Shipper collaboration*. Computers & Operations Research, vol. 34, no. 6, pages 1551–1560, 2007. (Cité en page 14.)
- [Errico *et al.* 2016] Fausto Errico, Teodor Gabriel Crainic, Federico Malucelli et Maddalena Nonato. *A Benders decomposition approach for the symmetric TSP with generalized latency arising in the design of semiflexible transit systems*. Transportation Science, vol. 51, no. 2, pages 706–722, 2016. (Cité en page 17.)
- [Escobar *et al.* 2014] John Willmer Escobar, Rodrigo Linfati, Paolo Toth et Maria G Baldoquin. *A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem*. Journal of Heuristics, vol. 20, no. 5, pages 483–509, 2014. (Cité en page 20.)

- [Favaretto *et al.* 2007] Daniela Favaretto, Elena Moretti et Paola Pellegrini. *Ant colony system for a VRP with multiple time windows and multiple visits*. Journal of Interdisciplinary Mathematics, vol. 10, no. 2, pages 263–284, 2007. (Cit  en pages 20 et 102.)
- [Feo & Resende 1995] Thomas A Feo et Mauricio GC Resende. *Greedy randomized adaptive search procedures*. Journal of global optimization, vol. 6, no. 2, pages 109–133, 1995. (Cit  en page 117.)
- [Filippi & Stevanato 2013] Carlo Filippi et Elisa Stevanato. *Approximation schemes for bi-objective combinatorial optimization and their application to the TSP with profits*. Computers & Operations Research, vol. 40, no. 10, pages 2418–2428, 2013. (Cit  en page 17.)
- [Furtado *et al.* 2017] Maria Gabriela S Furtado, Pedro Munari et Reinaldo Morabito. *Pickup and delivery problem with time windows : a new compact two-index formulation*. Operations Research Letters, vol. 45, no. 4, pages 334–341, 2017. (Cit  en page 33.)
- [Gaudioso & Paletta 1992] Manlio Gaudioso et Giuseppe Paletta. *A heuristic for the periodic vehicle routing problem*. Transportation Science, vol. 26, no. 2, pages 86–92, 1992. (Cit  en pages 20 et 102.)
- [Ghoseiri & Ghannadpour 2010] Keivan Ghoseiri et Seyed Farid Ghannadpour. *Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm*. Applied Soft Computing, vol. 10, no. 4, pages 1096–1107, 2010. (Cit  en page 20.)
- [Glover 1986] Fred Glover. *Future paths for integer programming and links to artificial intelligence*. Computers & operations research, vol. 13, no. 5, pages 533–549, 1986. (Cit  en page 30.)
- [Goksal *et al.* 2013] Fatma Pinar Goksal, Ismail Karaoglan et Fulya Altiparmak. *A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery*. Computers & Industrial Engineering, vol. 65, no. 1, pages 39–53, 2013. (Cit  en page 25.)
- [Goldberg *et al.* 1987] David E Goldberg, Jon Richardson *et al.* *Genetic algorithms with sharing for multimodal function optimization*. In Genetic algorithms and their applications : Proceedings of the Second International Conference on Genetic Algorithms, pages 41–49. Hillsdale, NJ : Lawrence Erlbaum, 1987. (Cit  en page 81.)
- [Gonzalez-Feliu & Salanova 2012] Jesus Gonzalez-Feliu et Josep-Maria Salanova. *Defining and evaluating collaborative urban freight transportation systems*. Procedia-Social and Behavioral Sciences, vol. 39, pages 172–183, 2012. (Cit  en page 12.)
- [Gonzalez-Feliu 2013] Jesus Gonzalez-Feliu. *Vehicle routing in multi-echelon distribution systems with cross-docking : A systematic lexical-metanarrative analysis*. Computer and Information Science, vol. 6, no. 3, page 28, 2013. (Cit  en page 4.)



- [Gunawan *et al.* 2015] Aldy Gunawan, Hoong Chuin Lau et Kun Lu. *An iterated local search algorithm for solving the orienteering problem with time windows*. In European Conference on Evolutionary Computation in Combinatorial Optimization, pages 61–73. Springer, 2015. (Cité en page 23.)
- [Gunawan *et al.* 2016] Aldy Gunawan, Hoong Chuin Lau et Pieter Vansteenwegen. *Orienteering problem : A survey of recent variants, solution approaches and applications*. European Journal of Operational Research, vol. 255, no. 2, pages 315–332, 2016. (Cité en page 22.)
- [Haimès 1971] Yacov Y Haimès. *On a bicriterion formulation of the problems of integrated system identification and system optimization*. IEEE transactions on systems, man, and cybernetics, vol. 1, no. 3, pages 296–297, 1971. (Cité en page 65.)
- [He *et al.* 2009] Ruhan He, Weibin Xu, Ji Xia Sun et Bingqiao Zu. *Balanced k-means algorithm for partitioning areas in large-scale vehicle routing problem*. In Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on, volume 3, pages 87–90. IEEE, 2009. (Cité en page 52.)
- [Holland 1973] John H Holland. *Genetic algorithms and the optimal allocation of trials*. SIAM Journal on Computing, vol. 2, no. 2, pages 88–105, 1973. (Cité en pages 30 et 77.)
- [Ismail *et al.* 2011] Sahbi Ben Ismail, François Legras et Gilles Coppin. *Synthèse du problème de routage de véhicules*. PhD thesis, Dépt. Logique des Usages, Sciences Sociales et de l’Information (Institut Mines-Télécom-Télécom Bretagne-UEB) ; Laboratoire en sciences et technologies de l’information, de la communication et de la connaissance (UMR CNRS 6285-Télécom Bretagne-Université de Bretagne Occidentale-Université de Bretagne Sud), 2011. (Cité en page 18.)
- [Jain 2010] Anil K Jain. *Data clustering : 50 years beyond K-means*. Pattern recognition letters, vol. 31, no. 8, pages 651–666, 2010. (Cité en page 54.)
- [Jiang *et al.* 2014] Jun Jiang, Kien Ming Ng, Kim Leng Poh et Kwong Meng Teo. *Vehicle routing problem with a heterogeneous fleet and time windows*. Expert Systems with Applications, vol. 41, no. 8, pages 3748–3760, 2014. (Cité en page 20.)
- [Johnson 1989] Ellis L Johnson. *Modeling and strong linear programs for mixed integer programming*. In Algorithms and model formulations in mathematical programming, pages 1–43. Springer, 1989. (Cité en page 28.)
- [Jozefowicz 2013] Nicolas Jozefowicz. *Optimisation combinatoire multi-objectif : des méthodes aux problèmes, de la Terre à (presque) la Lune*. PhD thesis, Institut National Polytechnique de Toulouse (INP Toulouse), 2013. (Cité en page 65.)
- [Kara *et al.* 2013] Imdat Kara, Ozge Nimet Koc, Fulya Altıparmak et Berna Deniz. *New integer linear programming formulation for the traveling salesman*

- problem with time windows : minimizing tour duration with waiting times.* Optimization, vol. 62, no. 10, pages 1309–1319, 2013. (Cité en page 18.)
- [Karabulut & Tasgetiren 2014] Korhan Karabulut et M Fatih Tasgetiren. *A variable iterated greedy algorithm for the traveling salesman problem with time windows.* Information Sciences, vol. 279, pages 383–395, 2014. (Cité en page 18.)
- [Karbowska-Chilinska & Zabielski 2014] Joanna Karbowska-Chilinska et Pawel Zabielski. *Genetic algorithm solving the orienteering problem with time windows.* In Advances in Systems Science, pages 609–619. Springer, 2014. (Cité en page 23.)
- [Kennedy & Eberhart 1995] J. Kennedy et R. Eberhart. *Particle swarm optimization.* In Neural Networks, 1995. Proceedings., IEEE International Conference on, volume 4, pages 1942–1948 vol.4, Nov 1995. (Cité en page 31.)
- [Kirkpatrick *et al.* 1983] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchiet *al.* *Optimization by simulated annealing.* science, vol. 220, no. 4598, pages 671–680, 1983. (Cité en page 29.)
- [Kobeaga *et al.* 2018] Gorka Kobeaga, María Merino et Jose A Lozano. *An efficient evolutionary algorithm for the orienteering problem.* Computers & Operations Research, vol. 90, pages 42–59, 2018. (Cité en page 23.)
- [Kuo & Zulvia 2017] RJ Kuo et Ferani E Zulvia. *Hybrid genetic ant colony optimization algorithm for capacitated vehicle routing problem with fuzzy demand - A case study on garbage collection system.* In Industrial Engineering and Applications (ICIEA), 2017 4th International Conference on, pages 244–248. IEEE, 2017. (Cité en page 20.)
- [Lal 2015] La logistique urbaine : connaitre et agir. CEREMA, 2015. (Cité en page 6.)
- [Land & Doig 1960] Ailsa H Land et Alison G Doig. *An automatic method of solving discrete programming problems.* Econometrica : Journal of the Econometric Society, pages 497–520, 1960. (Cité en page 27.)
- [Lap 2018] 2018. (Cité en page 6.)
- [Li & Lim 2003] Haibing Li et Andrew Lim. *A metaheuristic for the pickup and delivery problem with time windows.* International Journal on Artificial Intelligence Tools, vol. 12, no. 02, pages 173–186, 2003. (Cité en pages ix, 30, 32, 33, 48, 50, 56, 68, 69, 70, 72, 112 et 151.)
- [Li *et al.* 2016] Yuan Li, Haoxun Chen et Christian Prins. *Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests.* European Journal of Operational Research, vol. 252, no. 1, pages 27–38, 2016. (Cité en page 15.)
- [Liao & Ting 2010] Xin-Lan Liao et Chuan-Kang Ting. *An evolutionary approach for the selective pickup and delivery problem.* In Evolutionary Computation (CEC), 2010 IEEE Congress on, pages 1–8. IEEE, 2010. (Cité en pages 31, 32 et 33.)

- [Liao & Ting 2012] Xin-Lan Liao et Chuan-Kang Ting. *Evolutionary algorithms using adaptive mutation for the selective pickup and delivery problem*. In Evolutionary Computation (CEC), 2012 IEEE Congress on, pages 1–8. IEEE, 2012. (Cité en pages 31, 32 et 33.)
- [Liao & Ting 2013] Xin-Lan Liao et Chuan-Kang Ting. *Solving the biobjective selective pickup and delivery problem with memetic algorithm*. In Computational Intelligence In Production And Logistics Systems (CIPLS), 2013 IEEE Workshop on, pages 107–114. IEEE, 2013. (Cité en pages 31, 32 et 33.)
- [Liao et al. 2017] Ching-Jong Liao, Cheng-Hsiung Lee et Wan-Yin Chen. *A Hybrid Tabu Search Algorithm for the Variable Periodic Vehicle Routing Problem*. Arabian Journal for Science and Engineering, vol. 42, no. 2, pages 513–535, 2017. (Cité en pages 20 et 102.)
- [Lin & Vincent 2015] Shih-Wei Lin et F Yu Vincent. *A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows*. Applied Soft Computing, vol. 37, pages 632–642, 2015. (Cité en pages 23 et 102.)
- [Lin & Vincent 2017] Shih-Wei Lin et F Yu Vincent. *Solving the team orienteering problem with time windows and mandatory visits by multi-start simulated annealing*. Computers & Industrial Engineering, 2017. (Cité en page 23.)
- [Liu et al. 2014a] Ran Liu, Zhibin Jiang et Na Geng. *A hybrid genetic algorithm for the multi-depot open vehicle routing problem*. OR spectrum, vol. 36, no. 2, pages 401–421, 2014. (Cité en page 20.)
- [Liu et al. 2014b] Ran Liu, Xiaolan Xie et Thierry Garaix. *Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics*. Omega, vol. 47, pages 17–32, 2014. (Cité en page 102.)
- [Log 2014] *innovation in supply chain : mutualisation et collaboration transversale*, oct 2014. (Cité en page 13.)
- [Lu & Dessouky 2004] Quan Lu et Maged Dessouky. *An exact algorithm for the multiple vehicle pickup and delivery problem*. Transportation Science, vol. 38, no. 4, pages 503–514, 2004. (Cité en pages 32 et 33.)
- [Lu & Dessouky 2006] Quan Lu et Maged M Dessouky. *A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows*. European Journal of Operational Research, vol. 175, no. 2, pages 672–687, 2006. (Cité en pages 32 et 33.)
- [Lysgaard 2010] Jens Lysgaard. *The pyramidal capacitated vehicle routing problem*. European Journal of Operational Research, vol. 205, no. 1, pages 59–64, 2010. (Cité en page 20.)
- [Mancini 2016] Simona Mancini. *A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet : formulation and adaptive large neighborhood search based matheuristic*. Transportation Research Part C : Emerging Technologies, vol. 70, pages 100–112, 2016. (Cité en pages 20 et 102.)

- [Manier *et al.* 2016] H. Manier, M.-A. Manier et Z. Al Chami. *Shippers' collaboration in city logistics*. IFAC-PapersOnLine, vol. 49, no. 12, pages 1880 – 1885, 2016. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016. (Cité en page 138.)
- [Masmoudi *et al.* 2017] Mohamed Amine Masmoudi, Kris Braekers, Malek Masmoudi et Abdelaziz Dammak. *A hybrid genetic algorithm for the heterogeneous dial-a-ride problem*. Computers & Operations Research, vol. 81, pages 1–13, 2017. (Cité en page 25.)
- [Matai *et al.* 2010] Rajesh Matai, Surya Singh et Murari Lal Mittal. *Traveling salesman problem : an overview of applications, formulations, and solution approaches*. In Traveling salesman problem, theory and applications. InTech, 2010. (Cité en page 16.)
- [Melkonian 2007] Vardges Melkonian. *LP-based solution methods for the asymmetric TSP*. Information processing letters, vol. 101, no. 6, pages 233–238, 2007. (Cité en page 17.)
- [Mesquita *et al.* 2013] Marta Mesquita, Alberto Murta, Ana Paias et Laura Wise. *TSP with multiple time-windows and selective cities*. In International Conference on Computational Logistics, pages 158–172. Springer, 2013. (Cité en pages 18 et 102.)
- [Mitra 2005] Subrata Mitra. *An algorithm for the generalized vehicle routing problem with backhauling*. Asia-Pacific Journal of Operational Research, vol. 22, no. 02, pages 153–169, 2005. (Cité en page 25.)
- [Molenbruch *et al.* 2017] Yves Molenbruch, Kris Braekers, An Caris et Greet Vanden Berghe. *Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation*. Computers & Operations Research, vol. 77, pages 58–71, 2017. (Cité en page 25.)
- [Moscato *et al.* 1989] Pablo Moscato *et al.* *On evolution, search, optimization, genetic algorithms and martial arts : Towards memetic algorithms*. Caltech concurrent computation program, C3P Report, vol. 826, page 1989, 1989. (Cité en page 31.)
- [Nagata & Kobayashi 2010] Yuichi Nagata et Shigenobu Kobayashi. *A memetic algorithm for the pickup and delivery problem with time windows using selective route exchange crossover*. Parallel Problem Solving from Nature, PPSN XI, pages 536–545, 2010. (Cité en pages 31, 32 et 33.)
- [Nagy *et al.* 2013] Gábor Nagy, Niaz A Wassan, M Grazia Speranza et Claudia Archetti. *The vehicle routing problem with divisible deliveries and pickups*. Transportation Science, vol. 49, no. 2, pages 271–294, 2013. (Cité en page 25.)
- [Nanry & Barnes 2000] William P Nanry et J Wesley Barnes. *Solving the pickup and delivery problem with time windows using reactive tabu search*. Transportation Research Part B : Methodological, vol. 34, no. 2, pages 107–121, 2000. (Cité en pages 30, 32 et 33.)

- [Nowak *et al.* 2008] Maciek Nowak, Özlem Ergun et Chelsea C White III. *Pickup and delivery with split loads*. *Transportation Science*, vol. 42, no. 1, pages 32–43, 2008. (Cité en pages 29, 32 et 33.)
- [Padberg & Rinaldi 1991] Manfred Padberg et Giovanni Rinaldi. *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*. *SIAM review*, vol. 33, no. 1, pages 60–100, 1991. (Cité en page 28.)
- [Paletta & Triki 2004] Giuseppe Paletta et Chefi Triki. *Solving the asymmetric traveling salesman problem with periodic constraints*. *Networks*, vol. 44, no. 1, pages 31–37, 2004. (Cité en page 18.)
- [Pan *et al.* 2013] Shenle Pan, Eric Ballot et Frédéric Fontane. *The reduction of greenhouse gas emissions from freight transport by pooling supply chains*. *International Journal of Production Economics*, vol. 143, no. 1, pages 86–94, 2013. (Cité en page 14.)
- [Pankratz 2005] Giseller Pankratz. *A grouping genetic algorithm for the pickup and delivery problem with time windows*. *Or Spectrum*, vol. 27, no. 1, pages 21–41, 2005. (Cité en pages 31, 32 et 33.)
- [Papalitsas *et al.* 2015] Christos Papalitsas, Konstantinos Giannakis, Theodore Andronikos, Dimitrios Theotokis et Angelo Sifaleras. *Initialization methods for the TSP with time windows using variable neighborhood search*. In *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on*, pages 1–6. IEEE, 2015. (Cité en page 18.)
- [Park *et al.* 2017] Junhyuk Park, Jongsung Lee, SeHwan Ahn, Jungryul Bae et Hyun-chul Tae. *Exact Algorithm for the Capacitated Team Orienteering Problem with Time Windows*. *Mathematical Problems in Engineering*, vol. 2017, 2017. (Cité en page 23.)
- [Parragh *et al.* 2008a] Sophie N Parragh, Karl F Doerner et Richard F Hartl. *A survey on pickup and delivery models part ii : Transportation between pickup and delivery locations*. *Journal für Betriebswirtschaft*, vol. 58, no. 2, pages 81–117, 2008. (Cité en page 18.)
- [Parragh *et al.* 2008b] Sophie N. Parragh, Karl F. Doerner et Richard F. Hartl. *A survey on pickup and delivery problems. Part I : Transportation between customers and depot*. *Journal für Betriebswirtschaft*, vol. 58, no. 1, pages 21–51, 2008. (Cité en page 23.)
- [Parragh *et al.* 2008c] Sophie N. Parragh, Karl F. Doerner et Richard F. Hartl. *A survey on pickup and delivery problems. Part II : Transportation between pickup and delivery locations*. *Journal für Betriebswirtschaft*, vol. 58, no. 2, pages 81–117, 2008. (Cité en page 23.)
- [Patier & Routhier 2009] Danièle Patier et Jean-Louis Routhier. *Introduction au dossier thématique : La logistique urbaine, acquis et perspectives*. *Les Cahiers scientifiques du transport*, no. 55, pages 5–10, 2009. (Cité en page 4.)
- [Paulsen *et al.* 2015] Niklas Paulsen, Florian Diedrich et Klaus Jansen. *Heuristic Approaches to Minimize Tour Duration for the TSP with Multiple Time*

- Windows*. In OASIS-OpenAccess Series in Informatics, volume 48. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015. (Cité en pages 18 et 102.)
- [Pedro *et al.* 2013] Odivaney Pedro, Rodney Saldanha et Ricardo Camargo. *A tabu search approach for the prize collecting traveling salesman problem*. Electronic Notes in Discrete Mathematics, vol. 41, pages 261–268, 2013. (Cité en page 17.)
- [PIE 2013] 2013. (Cité en page 5.)
- [Pinto *et al.* 2015] Telmo Pinto, Claudio Alves, José Valério De Carvalho et Ana Moura. *An insertion heuristic for the capacitated vehicle routing problem with loading constraints and mixed linehauls and backhauls*. FME Transactions, vol. 43, no. 4, pages 311–318, 2015. (Cité en page 24.)
- [Pisinger & Ropke 2007] David Pisinger et Stefan Ropke. *A general heuristic for vehicle routing problems*. Computers & operations research, vol. 34, no. 8, pages 2403–2435, 2007. (Cité en page 20.)
- [Polat 2017] Olcay Polat. *A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups*. Computers & Operations Research, vol. 85, pages 71–86, 2017. (Cité en page 25.)
- [Pre 2006] 2006. (Cité en pages ix, 6 et 7.)
- [Qin *et al.* 2009] Hu Qin, Andrew Lim et Dongsheng Xu. *The selective traveling salesman problem with regular working time windows*. Studies in Computational Intelligence, vol. 214, pages 291–296, 2009. (Cité en page 17.)
- [Qiu-ai & Zhen-ping 2012] HUANG Qiu-ai et LI Zhen-ping. *Mathematical Model and Algorithm for VRP with Multiple Time Windows*. Logistics Technology, vol. 13, page 064, 2012. (Cité en pages 20 et 102.)
- [Qureshi *et al.* 2009] AG Qureshi, E Taniguchi et Tadashi Yamada. *An exact solution approach for vehicle routing and scheduling problems with soft time windows*. Transportation Research Part E : Logistics and Transportation Review, vol. 45, no. 6, pages 960–977, 2009. (Cité en page 20.)
- [Rahmani *et al.* 2016] Nastaran Rahmani, Boris Detienne, Ruslan Sadykov et François Vanderbeck. *A Column Generation Based Heuristic for the Dial-A-Ride Problem*. In International Conference on Information Systems, Logistics and Supply Chain (ILS), 2016. (Cité en page 25.)
- [Rezki & Aghezzaf 2017] Hasnaa Rezki et Brahim Aghezzaf. *The bi-objective orienteering problem with budget constraint : GRASP\_ILS*. In Logistics and Supply Chain Management (LOGISTIQUA), 2017 International Colloquium on, pages 25–30. IEEE, 2017. (Cité en page 23.)
- [Ribeiro & Laporte 2012] Glaydston Mattos Ribeiro et Gilbert Laporte. *An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem*. Computers & operations research, vol. 39, no. 3, pages 728–735, 2012. (Cité en page 20.)

- [Righini & Salani 2009] Giovanni Righini et Matteo Salani. *Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming*. Computers & Operations Research, vol. 36, no. 4, pages 1191–1203, 2009. (Cité en page 29.)
- [Ropke & Cordeau 2009] Stefan Ropke et Jean-François Cordeau. *Branch and cut and price for the pickup and delivery problem with time windows*. Transportation Science, vol. 43, no. 3, pages 267–286, 2009. (Cité en pages 32 et 33.)
- [Ropke & Pisinger 2006a] Stefan Ropke et David Pisinger. *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows*. Transportation science, vol. 40, no. 4, pages 455–472, 2006. (Cité en pages 29, 32 et 33.)
- [Ropke & Pisinger 2006b] Stefan Ropke et David Pisinger. *A unified heuristic for a large class of vehicle routing problems with backhauls*. European Journal of Operational Research, vol. 171, no. 3, pages 750–775, 2006. (Cité en page 24.)
- [Ropke *et al.* 2007] Stefan Ropke, Jean-François Cordeau et Gilbert Laporte. *Models and branch-and-cut algorithms for pickup and delivery problems with time windows*. Networks, vol. 49, no. 4, pages 258–272, 2007. (Cité en pages 28, 32 et 33.)
- [Routhier *et al.* 2002] Jean-Louis Routhier, Jean-Guy DUFOUR et Danièle PATIER. Du transport de marchandises en ville à la logistique urbaine. Ed. Techniques Ingénieur, 2002. (Cité en page 4.)
- [Salmon 2011] K Salmon. *Pratiques de logistique collaborative : quelles opportunités pour les PME/ETI*. Pôle interministériel de prospective et, 2011. (Cité en page 12.)
- [Sareni & Krahenbuhl 1998] Bruno Sareni et Laurent Krahenbuhl. *Fitness sharing and niching methods revisited*. IEEE transactions on Evolutionary Computation, vol. 2, no. 3, pages 97–106, 1998. (Cité en page 81.)
- [Savelsbergh & Sol 1995] Martin WP Savelsbergh et Marc Sol. *The general pickup and delivery problem*. Transportation science, vol. 29, no. 1, pages 17–29, 1995. (Cité en page 23.)
- [Schilde *et al.* 2009] Michael Schilde, Karl F Doerner, Richard F Hartl et Guenter Kiechle. *Metaheuristics for the bi-objective orienteering problem*. Swarm Intelligence, vol. 3, no. 3, pages 179–201, 2009. (Cité en page 23.)
- [Schrijver 1998] Alexander Schrijver. Theory of linear and integer programming. John Wiley & Sons, 1998. (Cité en page 28.)
- [Shi *et al.* 2009] Xiaoyan Shi, Fanggeng Zhao et Yancheng Gong. *Genetic algorithm for the one-commodity pickup-and-delivery vehicle routing problem*. In Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on, volume 1, pages 175–179. IEEE, 2009. (Cité en page 25.)

- [Solomon 1987] Marius M Solomon. *Algorithms for the vehicle routing and scheduling problems with time window constraints*. Operations research, vol. 35, no. 2, pages 254–265, 1987. (Cit  en pages 48, 68 et 151.)
- [S rensen 2003] Kenneth S rensen. *A framework for robust and flexible optimisation using metaheuristics*. Quarterly Journal of the Belgian, French and Italian Operations Research Societies, vol. 1, no. 4, pages 341–345, 2003. (Cit  en page 18.)
- [Souffriau *et al.* 2013] Wouter Souffriau, Pieter Vansteenwegen, Greet Vanden Bergh  et Dirk Van Oudheusden. *The multiconstraint team orienteering problem with multiple time windows*. Transportation Science, vol. 47, no. 1, pages 53–63, 2013. (Cit  en pages 23 et 102.)
- [Srinivas & Deb 1994] Nidamarthi Srinivas et Kalyanmoy Deb. *Multiojective optimization using nondominated sorting in genetic algorithms*. Evolutionary computation, vol. 2, no. 3, pages 221–248, 1994. (Cit  en page 79.)
- [Ting & Liao 2013] Chuan-Kang Ting et Xin-Lan Liao. *The selective pickup and delivery problem : formulation and a memetic algorithm*. International Journal of Production Economics, vol. 141, no. 1, pages 199–211, 2013. (Cit  en pages 31, 32 et 33.)
- [Toth & Vigo 2002] Paolo Toth et Daniele Vigo. *The vehicle routing problem, Society for industrial and applied mathematics*. SIAM Monographs on Discrete Mathematics and Applications, 2002. (Cit  en page 23.)
- [Tricoire *et al.* 2010] Fabien Tricoire, Martin Romauch, Karl F Doerner et Richard F Hartl. *Heuristics for the multi-period orienteering problem with multiple time windows*. Computers & Operations Research, vol. 37, no. 2, pages 351–367, 2010. (Cit  en pages 23 et 102.)
- [Valle *et al.* 2009] Cristiano Arbex Valle, Alexandre Salles da Cunha, Geraldo Robson Mateus et Leonardo C Martinez. *Exact algorithms for a selective vehicle routing problem where the longest route is minimized*. Electronic Notes in Discrete Mathematics, vol. 35, pages 133–138, 2009. (Cit  en page 20.)
- [Van Laarhoven & Aarts 1987] Peter JM Van Laarhoven et Emile HL Aarts. *Simulated annealing*. In Simulated annealing : Theory and applications, pages 7–15. Springer, 1987. (Cit  en page 30.)
- [Vansteenwegen *et al.* 2011] Pieter Vansteenwegen, Wouter Souffriau et Dirk Van Oudheusden. *The orienteering problem : A survey*. European Journal of Operational Research, vol. 209, no. 1, pages 1–10, 2011. (Cit  en page 21.)
- [Venkateshan & Mathur 2011] Prahalad Venkateshan et Kamlesh Mathur. *An efficient column-generation-based algorithm for solving a pickup-and-delivery problem*. Computers & Operations Research, vol. 38, no. 12, pages 1647–1655, 2011. (Cit  en pages 28, 32 et 33.)
- [Verbeeck *et al.* 2017] C dric Verbeeck, Pieter Vansteenwegen et El-Houssaine Aghezaf. *The time-dependent orienteering problem with time windows : a fast ant*



- colony system*. Annals of Operations Research, pages 1–25, 2017. (Cité en page 23.)
- [Vincent *et al.* 2017] F Yu Vincent, Parida Jewpanya, Ching-Jung Ting et AAN Perwira Redi. *Two-level particle swarm optimization for the multi-modal team orienteering problem with time windows*. Applied Soft Computing, vol. 61, pages 1022–1040, 2017. (Cité en page 23.)
- [Wang & Kopfer 2014] Xin Wang et Herbert Kopfer. *Collaborative transportation planning of less-than-truckload freight*. OR spectrum, vol. 36, no. 2, pages 357–380, 2014. (Cité en page 15.)
- [Wang & Lu 2009] Chung-Ho Wang et Jiu-Zhang Lu. *A hybrid genetic algorithm that optimizes capacitated vehicle routing problems*. Expert Systems with Applications, vol. 36, no. 2, pages 2921–2936, 2009. (Cité en page 20.)
- [Wang *et al.* 2015] Chao Wang, Dong Mu, Fu Zhao et John W Sutherland. *A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows*. Computers & Industrial Engineering, vol. 83, pages 111–122, 2015. (Cité en pages vii, 25, 49 et 50.)
- [Wei *et al.* 2017] Lijun Wei, Zhenzhen Zhang, Defu Zhang et Stephen CH Leung. *A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints*. European Journal of Operational Research, 2017. (Cité en page 20.)
- [Wright 1991] Alden H Wright. *Genetic algorithms for real parameter optimization*. In Foundations of genetic algorithms, volume 1, pages 205–218. Elsevier, 1991. (Cité en page 77.)
- [Xiao *et al.* 2012] Yiyong Xiao, Qiuhong Zhao, Ikou Kaku et Yuchun Xu. *Development of a fuel consumption optimization model for the capacitated vehicle routing problem*. Computers & Operations Research, vol. 39, no. 7, pages 1419–1431, 2012. (Cité en page 20.)
- [Xu *et al.* 2017] Dongyang Xu, Kunpeng Li, Xuxia Zou et Ling Liu. *An unpaired pickup and delivery vehicle routing problem with multi-visit*. Transportation Research Part E : Logistics and Transportation Review, vol. 103, pages 218–247, 2017. (Cité en page 25.)
- [Yao *et al.* 2013] Baozhen Yao, Ping Hu, Mingheng Zhang et Shuang Wang. *Artificial bee colony algorithm with scanning strategy for the periodic vehicle routing problem*. Simulation, vol. 89, no. 6, pages 762–770, 2013. (Cité en pages 20 et 102.)
- [Yearwood 2010] John Yearwood. Technologies for supporting reasoning communities and collaborative decision making : Cooperative approaches : Cooperative approaches. IGI Global, 2010. (Cité en page 12.)
- [Zou *et al.* 2013] Shuilong Zou, Jin Li et Xueqian Li. *A Hybrid Particle Swarm Optimization Algorithm for Multi-Objective Pickup and Delivery Problem with Time Windows*. JCP, vol. 8, no. 10, pages 2583–2589, 2013. (Cité en pages 31, 32 et 33.)



## Résumé :

De nos jours, le transport joue un rôle clé dans la vie des pays modernes, en particulier pour les flux de marchandises. La logistique des flux entre régions, pays et continents a bénéficié d'innovations technologiques et organisationnelles assurant efficacité et efficience. Il n'en a pas été de même à l'échelle urbaine, plus particulièrement dans les centres-villes : la gestion des flux dans un environnement caractérisé par une forte densité démographique n'a pas encore véritablement trouvé son modèle d'organisation. Aujourd'hui, la logistique urbaine ou encore la gestion "du dernier kilomètre" constitue donc un enjeu de premier plan, tant socio politique et environnemental qu'économique. La logistique urbaine est caractérisée par la présence de plusieurs acteurs (chargeurs ou propriétaires de marchandises, clients, transporteurs, autorités publiques, ...) ayant chacun des priorités différentes (réduction de la pollution, amélioration de la qualité de service, minimisation de la distance totale parcourue, ...). Pour relever ces défis, un des leviers possibles consiste à optimiser les tournées de distribution et/ou collecte de marchandises, dans le contexte et sous les contraintes de la ville.

Le but de ce travail de thèse réside alors dans la planification de la distribution des marchandises dans un réseau logistique, abordée sous un angle de collaboration entre les chargeurs. Cette collaboration consiste à regrouper les demandes de divers chargeurs pour optimiser le taux de chargement des camions et obtenir de meilleurs prix de transport. Ici, la gestion du « dernier kilomètre » s'apparente à ce que l'on identifie dans la littérature comme le Pickup and Delivery Problem (PDP). Dans le cadre de cette thèse, nous nous intéressons à des variantes de ce problème plus adaptées au contexte urbain. Après avoir réalisé un état de l'art sur les problèmes d'optimisation combinatoire autour du transport et les méthodes utilisées pour leur résolution, nous étudions deux nouvelles variantes du problème de collecte et de livraison : le Selective PDP with Time Windows and Paired Demands et le Multi-periods PDP with Time Windows and Paired Demands. La première permet aux transporteurs de livrer le maximum de clients dans une journée par exemple ; avec la seconde, et en cas d'impossibilité de livraison dans cette période, on détermine la meilleure date de livraison en minimisant la distance parcourue. Chacune d'elles fait l'objet d'une description formelle, d'une modélisation mathématique sous forme de programme linéaire, puis d'une résolution par des méthodes exacte, heuristiques et métaheuristiques, dans des cas mono-objectif et multi-objectifs. La performance de chaque approche a été évaluée par un nombre substantiel de tests sur des instances de différentes tailles issues de la littérature et/ou que nous avons générées. Les avantages et les inconvénients de chaque approche sont analysés, notamment dans le cadre de la collaboration entre chargeurs.

**Mots-clés :** Transport, Optimisation, Problèmes de tournées

## Abstract:

Nowadays, transportation plays a key role in our modern countries' life, in particular for the goods flows. The logistics of flows between regions, countries and continents have benefited from technological and organizational innovations ensuring efficiency and effectiveness. It has not been the same at the urban scale, especially in city centers: the management of flows in a high population density environment has not yet found its organizational model. Today, urban logistics or "last mile" management is therefore a major issue, both socio-political and environmental as well as economic. Urban logistics is characterized by several actors (shippers or owners of goods, customers, carriers, public authorities, ...) each with different priorities (reduction of pollution, improvement of service quality, minimization of total distance traveled, ...). To overcome these challenges, one possible lever is to optimize the distribution and/or collection of goods in the context and under the constraints of the city.

The goal of this PhD work is then to plan the distribution of goods in a logistics network, approached from a collaboration angle between shippers. This collaboration consists in grouping the demands of several shippers to optimize the loading rate of the trucks and to obtain better transport prices. Here, managing the "last mile" is similar to what is known in the literature as the Pickup and Delivery Problem (PDP). In this thesis, we are interested in variants of this problem more adapted to the urban context. After having realized a state of the art on the combinatorial optimization problems around the transport and the methods used for their resolution, we study two new variants of the problem of collection and delivery: the Selective PDP with Windows and Paired Demands and the Multi-period PDP with Windows and Paired Demands. The first allows carriers to deliver the maximum number of customers in a day for example; with the second, and in case of impossibility of delivery in this period, we determine the best delivery date by minimizing the distance traveled. Each of them is the subject of a formal description, of a mathematical modeling in the form of a linear program, then of a resolution by exact methods, heuristics and metaheuristics, in single-objective and multi-objective cases. The performance of each approach was evaluated by a substantial number of tests on instances of different sizes from the literature and / or that we generated. The advantages and drawbacks of each approach are analyzed, in particular in the context of collaboration between shippers.

**Keywords:** Transport, Optimization, Routing problems

The logo for SPIM (École doctorale SPIM) features the letters 'S', 'P', 'I', and 'M' in a stylized, white, sans-serif font. The 'S' is the largest and most prominent, with the other letters stacked to its right. A blue horizontal bar is positioned to the left of the 'S'.

■ École doctorale SPIM - Université de Technologie Belfort-Montbéliard

F - 90010 Belfort Cedex ■ tél. +33 (0)3 84 58 31 39

■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

