



HAL
open science

Planification de la recharge et optimisation des tournées dans le cas de flottes captives

Ons Sassi

► **To cite this version:**

Ons Sassi. Planification de la recharge et optimisation des tournées dans le cas de flottes captives. Autre [cs.OH]. Université de Lorraine, 2015. Français. NNT : 2015LORR0303 . tel-01754599

HAL Id: tel-01754599

<https://hal.univ-lorraine.fr/tel-01754599v1>

Submitted on 13 Jan 2022

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

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



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Planification de la recharge et optimisation des tournées dans le cas de flottes captives

THÈSE

présentée et soutenue publiquement le 7 décembre 2015

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

Ons SASSI

Composition du jury

Rapporteurs : Van-Dat CUNG : Professeur, Grenoble INP
Aziz MOUKRIM : Professeur, Université de Technologie de Compiègne

Examineurs : Ammar OULAMARA : Professeur, Université de Lorraine (directeur de thèse)
Nacer BOUDJLIDA : Professeur, Université de Lorraine
Wahiba RAMDANE CHERIF : Maître de conférence, Université de Lorraine
Patrick SIARRY : Professeur, Université Paris-Est, Créteil
Marie-Ange MANIER : Maître de conférence, Université de Technologie
Belfort-Montbéliard

Invité : Alain ROSET : Directeur innovation, Groupe La Poste

Remerciements

Tout d'abord, je tiens à remercier mon directeur de thèse Ammar OULAMARA pour son encadrement, sa disponibilité et ses conseils tout au long de ces trois années de recherche.

Mes remerciements vont également à Mme. Wahiba RAMDANE CHERIF avec qui j'ai pu collaborer tout au long de la période de préparation de thèse. Elle m'a apporté une compréhension plus approfondie des problèmes de tournées de véhicules et des méta-heuristiques.

Je remercie également tous les collaborateurs et partenaires du projet Infini Drive qui m'ont apporté beaucoup de connaissances dans le domaine de la mobilité électrique. Mes remerciements vont plus spécifiquement à Elena Shevchenko et Alain ROSET qui m'ont fait découvrir le monde postal avec tous ses mystères.

Je n'oublie pas de remercier tous ceux dont j'ai pu faire la connaissance au cours de conférences, séminaires et autres occasions, qui m'ont offert des rencontres enrichissantes et aidé d'une manière ou d'une autre.

Enfin, cette thèse de doctorat est évidemment dédiée à ma famille et mes amis pour leur présence, leur amour et leur soutien inconditionnel.

*À mes très chers parents.
Pour tout ce que vous me donnez depuis tant d'années.
À la plus chère des nièces Roukaya.
Pour toute la joie que tu me donnes depuis le jour de ta naissance.*

Sommaire

Chapitre 1

Introduction générale

1.1	Contexte et objectif	1
1.2	Contributions et méthodologie	3
1.3	Structure du mémoire	4

Chapitre 2

Problématique industrielle

2.1	Contexte général	5
2.2	Projet Infini Drive	7
2.3	Écosystème du véhicule électrique	7
2.3.1	Le véhicule électrique	7
2.3.2	Infrastructure de recharge	8
2.3.3	Le réseau électrique	11
2.4	Problème industriel	12
2.4.1	Le métier La Poste	12
2.4.2	Le métier ERDF	14
2.4.3	Problème de gestion opérationnelle d'une flotte de véhicules électriques et thermiques	15
2.5	Conclusion	17

Chapitre 3

Revue de littérature de l'éco-système du véhicule électrique

3.1	Introduction	18
3.2	Notions sur la complexité	19
3.2.1	Classification des problèmes	19
3.2.2	La NP-Complétude	20
3.2.3	La Réduction polynomiale	20

3.3	Algorithmes d'approximation	20
3.4	Introduction à la recherche locale	21
3.4.1	Algorithme de recherche locale	21
3.4.2	Notion de voisinage	21
3.5	Méthodes de résolution combinatoire	22
3.5.1	Méthodes exactes	22
3.5.2	Méthodes approchées	23
3.6	État de l'art de l'écosystème du véhicule électrique	27
3.6.1	Smart Grid et V2G	27
3.6.2	Dimensionnement de l'infrastructure de recharge	28
3.6.3	Optimisation de la recharge	31
3.6.4	Optimisation des tournées de véhicules électriques	33
3.7	Conclusion	35

Chapitre 4

Optimisation conjointe de l'affectation et de la recharge des véhicules électriques

4.1	Introduction	36
4.2	Description du problème et notation	37
4.3	Étude de complexité	38
4.4	Formulation mathématique	39
4.5	Approches de résolution	42
4.5.1	Méthode exacte	42
4.5.2	Heuristique Séquentielle	42
4.5.3	Heuristique globale	51
4.6	Résultats expérimentaux et discussion	52
4.7	Conclusion	58

Chapitre 5

Extensions au problème de base

5.1	Introduction	59
5.2	Dépassement de la puissance souscrite	60
5.3	Vieillessement des batteries	61
5.4	Modèle de véhicule	62
5.5	Modèle d'infrastructure de recharge	63
5.6	Bilan carbone	64
5.7	Optimisation de la flotte de véhicules	65
5.8	Cas d'étude et expérimentations	66

5.8.1	Protocole des expérimentations simulées	66
5.8.2	SC 1 : Scénario de référence	68
5.8.3	SC 2 : Modulation de la puissance de recharge	69
5.8.4	SC 3 : Optimisation de l'affectation et de la recharge des véhicules	70
5.8.5	SC 4 : Vieillessement de la batterie	71
5.8.6	SC 5 : Optimisation du bilan carbone	73
5.8.7	SC 6 : Optimisation des coûts d'exploitation	74
5.8.8	SC 7 : Optimisation de la flotte	76
5.9	Conclusion	77

Chapitre 6

Étude du problème d'ordonnancement d'intervalles sous contraintes d'énergie

6.1	Introduction	78
6.2	Ordonnancement d'intervalles sous contraintes d'énergie	79
6.3	Revue de littérature	80
6.3.1	Problèmes tactiques	81
6.3.2	Problèmes de minimisation des coûts	81
6.3.3	Problèmes de maximisation du profit	82
6.3.4	Problèmes d'ordonnancement d'intervalles en-ligne	83
6.4	Étude du problème	84
6.4.1	Minimal ISEC	84
6.4.2	Maximal ISEC	86
6.4.3	Maximal Weighted ISEC	89
6.5	Conclusion	89

Chapitre 7

Heuristiques pour la résolution du problème de tournées de véhicules électriques

7.1	Introduction	90
7.2	Motivation et description du problème	93
7.3	Formulation mathématique	95
7.4	Résolution par heuristiques	100
7.4.1	Représentation d'une solution au problème EVRP-MF	100
7.4.2	Prise en compte des besoins d'énergie des tournées	101
7.4.3	Heuristique constructive	105
7.4.4	Méthodes de recherche locale	107
7.4.5	Recherche locale à grand voisinage (LNS)	115
7.5	Étude comparative	124

7.5.1	Implémentation et jeux de test	124
7.5.2	Choix des paramètres des algorithmes	125
7.5.3	Résultats numériques	125
7.6	Conclusion	130

Chapitre 8

Approches méta-heuristiques basées sur LNS

8.1	Introduction	134
8.2	ILS avec LNS intégré	135
8.2.1	Principe de ILS	135
8.2.2	ILS avec LNS intégré pour EVRP-MF	137
8.3	Recherche à Voisinage Variable avec LNS intégré	138
8.3.1	Principe de VNS	139
8.3.2	VNS avec LNS intégré pour EVRP-MF	140
8.4	Étude expérimentale	141
8.4.1	Performance de l'algorithme ILS	141
8.4.2	Performance de l'algorithme VNS	145
8.4.3	Étude comparative	147
8.5	Conclusion	150

Chapitre 9

Conclusion générale et perspectives

Bibliographie

Chapitre 1

Introduction générale

Sommaire

1.1 Contexte et objectif	1
1.2 Contributions et méthodologie	3
1.3 Structure du mémoire	4

Ce chapitre d'introduction présente le contexte général et les objectifs de notre étude qui se concentre sur la planification de la recharge et l'optimisation des tournées des véhicules électriques. Il décrit également notre contribution et la structure de ce mémoire.

1.1 Contexte et objectif

La croissance du secteur du transport a connu une réelle explosion au cours de ces dernières années. Cette croissance est principalement attribuée au transport routier qui représente le mode dominant de mobilité avec 90% des trajets effectués par les passagers et 75% des biens transportés¹. En effet, une grande partie des échanges commerciaux et des déplacements des personnes est effectuée par voie routière.

Avec la mondialisation et le développement du e-commerce, la circulation des poids lourds et des véhicules utilitaires ne cesse d'augmenter. Ainsi, qu'il s'agisse du transport de marchandises, de la livraison du courrier et des colis par voie postale ou simplement de la mobilité des personnes, les véhicules utilisés fonctionnent à majorité avec des moteurs à combustion. C'est pourquoi le secteur du transport apparaît clairement comme le secteur d'activité principal en matière d'utilisation de produits pétroliers avec une part actuelle de 50%².

Réellement, le transport des marchandises et la mobilité des personnes sont fortement dépendants de l'utilisation du pétrole, dont l'usage intensif a des impacts néfastes sur l'environnement et sur la dépendance énergétique. Par ailleurs, le transport est une source importante d'émissions de gaz à effet de serre et de polluants. L'automobile représente, de nos jours, 13% des émissions de CO₂ dans le monde et 20% en Europe³. Les grandes agglomérations sont donc saturées par les gaz à effet de serre et les particules fines résultant de la combustion du carburant des voitures thermiques. Les conséquences sont ainsi importantes sur la qualité de vie.

¹Source : IFP Énergies nouvelles, Panorama 2005 : La consommation d'énergie dans le secteur du transport

²Source : IFP Énergies nouvelles, Panorama 2005 : La consommation d'énergie dans le secteur du transport

³Source : <http://www.voiture-electrique-populaire.fr/paradigme-mobilite-individuelle>

Depuis plusieurs années, tout un ensemble de moyens est en train d'être mis en oeuvre afin de réduire les émissions de CO₂ et concilier augmentation de la demande énergétique et limitation des émissions de CO₂. Mais, les voies de réduction des émissions liées au secteur du transport semblent être limitées et longues à mettre en oeuvre. Pour faire face à toutes ces difficultés, de nouveaux services et de nouvelles technologies de mobilité doivent être inventés.

Une première solution pour réduire l'impact environnemental du transport de personnes pourrait consister en une réduction du nombre de véhicules particuliers en circulation. En effet, la voiture particulière génère à elle seule 54.7% des émissions du secteur du transport routier et représente au global 14.4% des rejets de CO₂⁴. Par ailleurs, le taux d'occupation de la voiture reste particulièrement bas pour les déplacements pendulaires (1.1 à 1.2 personnes par voiture), autrement dit, l'immense majorité des conducteurs sont seuls dans leurs véhicules. Il est ainsi important de développer les modes alternatifs permettant de rationaliser l'usage du véhicule individuel. Conscients de cette problématique, les services publics tentent de favoriser et d'encourager le recours au covoiturage et au transport multimodal. La notion de la voiture évolue ainsi, progressivement, pour passer d'une simple propriété à un moyen de transport avec le développement de nouveaux services de mobilité tels que le covoiturage et l'auto-partage.

Favoriser le covoiturage et le combiner avec l'utilisation de transports en commun et des modes de transports individuels propres pourrait ainsi permettre de proposer des alternatives à la voiture individuelle, de mieux répondre aux besoins de la mobilité et de compléter les offres de transport en commun.

Pour faire face aux difficultés économiques, sociétales et environnementales rencontrées par le secteur du transport, une autre solution pourrait concerner le développement de nouvelles technologies de véhicules qui soient plus propres et plus indépendantes de l'usage du pétrole. En effet, il est essentiel de favoriser la pénétration de véhicules à faible consommation et émissions de gaz à effet de serre.

Depuis les années 90, le développement du véhicule électrique devient le défi des constructeurs. Aujourd'hui, la voiture tout électrique commence à percer le marché de l'automobile et son autonomie est en évolution permanente.

Moins polluant et plus silencieux que le véhicule thermique, ce nouveau modèle de véhicules semble avoir sa place pour participer à la révolution de la mobilité urbaine et durable. Le développement du véhicule électrique a été soutenu par les États Généraux de l'Automobile, organisés en janvier 2009, qui ont donné lieu au lancement du plan national "véhicules décarbonés". Par ailleurs, dans la lignée des efforts gouvernementaux pour favoriser l'usage des «véhicules décarbonés», le Groupe La Poste a mené des travaux pour structurer la demande de véhicules électriques en France et amorcer l'émergence et le développement du marché du véhicule électrique.

Malgré les progrès réalisés au niveau de l'écosystème (technologie et services) de la mobilité électrique, les coûts d'achat élevés et l'indisponibilité des infrastructures de recharge adéquates représentent des freins majeurs à l'expansion de l'usage des véhicules électriques par les particuliers et les entreprises.

Pour faire face aux coûts d'achat élevés du véhicule électrique, l'état français a mis en place un bonus écologique pour récompenser les personnes qui acquièrent des véhicules propres et surtout pour amorcer le démarrage de la filière industrielle du véhicule électrique. Ce bonus permet ainsi de réduire l'écart des coûts à l'achat des véhicules électriques et thermiques sans le supprimer

⁴<http://www.notre-planete.info>

complètement.

Concernant la disponibilité des bornes de recharge, la question du développement des infrastructures de recharge en particulier pour les véhicules des entreprises et des collectivités est plus que jamais d'actualité. Le déploiement des véhicules électriques doit alors passer par la conception et l'expérimentation des infrastructures permettant à l'exploitant d'une flotte de véhicules électriques d'assurer une gestion adéquate de sa flotte et d'atteindre durablement ses objectifs.

À l'horizon 2020, ce sont 2 millions de véhicules électriques qui devraient être mis en circulation, selon les objectifs du plan gouvernemental. Un tel développement nécessitera le déploiement d'infrastructures de recharge des véhicules. Le plan prévoit la mise en place de 4 millions de prises dans le secteur privé, à domicile ou en entreprises, et 400.000 prises mises à disposition en voirie ou en parkings publics⁵.

En plus des incitations gouvernementales pour l'achat des véhicules électriques et le déploiement des infrastructures de recharge, un autre volet d'initiatives concerne notamment les appels à projets concernant les "véhicules du futur" du PIA⁶. Ce programme a réservé une part importante des financements à la R&D et l'expérimentation de l'écosystème du véhicule électrique. À cet effet, le projet Infini Drive a été lancé et piloté par La Poste, ERDF et six entreprises et laboratoires de recherche. Il est financé par l'ADEME (Agence de l'Environnement et de la Maîtrise de l'Énergie).

Les axes de recherche de ce projet sont particulièrement multiples et ils reposent principalement sur l'aide à la définition et l'utilisation de solutions technologiques et organisationnelles innovantes. Il s'agit, en particulier, d'optimiser le dimensionnement du parc des véhicules électriques et de l'infrastructure de recharge et d'atteindre les meilleurs équilibres économiques et écologiques. L'atteinte de ces objectifs repose sur l'utilisation des outils d'optimisation, de modélisation et de simulation ainsi que sur la validation de ces outils grâce à des expérimentations et à des scénarios d'usage variés.

Ce travail de thèse rentre dans le cadre du projet Infinidrive dont l'objectif est de développer et de tester des outils d'aide à la décision pour accompagner les industriels dans le déploiement de véhicules électriques dans des flottes captives. Cette thèse considère alors différents problèmes d'optimisation résolus par les techniques de la recherche opérationnelle et de l'optimisation combinatoire. Ces problèmes concernent principalement l'optimisation de l'affectation et de la recharge des véhicules électriques ainsi que l'optimisation des tournées des véhicules électriques dans un contexte industriel.

1.2 Contributions et méthodologie

Dans le cadre de cette thèse, notre objectif est de développer des algorithmes d'optimisation performants pour résoudre deux problèmes principaux : (i) l'optimisation conjointe de l'affectation et de la recharge des véhicules électriques et (ii) l'optimisation des tournées des véhicules électriques.

Le premier problème fait l'objet de la majeure partie de nos travaux. Il est dérivé d'un cas d'usage réel et il est suivi par des longues périodes d'expérimentation. Ce problème consiste à optimiser l'affectation d'un ensemble de tournées à une flotte mixte de véhicules électriques et thermiques et à optimiser simultanément la recharge des véhicules électriques. L'objectif de

⁵Source : COMMISSARIAT GÉNÉRAL AU DÉVELOPPEMENT DURABLE : Les véhicules électriques en perspective : Analyse coûts-avantages et demande potentielle

⁶Programme d'Investissements d'Avenir

l'optimisation est d'assurer la rentabilité des véhicules électriques tout en respectant les contraintes d'ordre opérationnel et technique. Cet objectif varie de l'optimisation des coûts économiques à l'optimisation du bilan carbone et l'optimisation de la flotte et de l'infrastructure de recharge. La conception des tournées n'est pas prise en compte dans le cadre de cette première problématique, ainsi une tournée est modélisée comme étant une tâche ayant certaines caractéristiques et devant être exécutée par un véhicule adéquat. De cette problématique sont dérivés de nouveaux problèmes d'ordonnement d'intervalles sous contraintes d'énergie que nous proposons également d'étudier.

Le second problème consiste à construire des tournées à coût total minimal et à optimiser simultanément la recharge d'une flotte composée de véhicules électriques hétérogènes et de véhicules thermiques identiques. Il s'agit alors de déterminer, pour un ensemble donné de clients et de stations de recharge, par quels véhicules de la flotte les clients seront servis, à quel moment dans la tournée ils le seront, à quel moment les véhicules électriques doivent être chargés et dans quelles stations de recharge ils le seront. Nous considérons pour ce nouveau problème des contraintes réelles qui n'ont jamais été considérées dans la littérature.

1.3 Structure du mémoire

La suite de ce mémoire est organisée comme suit : Le chapitre 2 décrit la problématique industrielle qui a été évoquée dans le cadre du projet Infini Drive et qui consiste à optimiser les usages et la recharge des véhicules électriques dans un contexte industriel.

Le chapitre 3 présente une revue de littérature de plusieurs problématiques liées à l'écosystème du véhicule électrique ainsi que des méthodes de résolution exactes et approchées. Ce chapitre évoque également plusieurs notions utiles telles que la notion de complexité et d'algorithmes d'approximation.

Le chapitre 4 traite le problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques. Nous montrons que ce problème est NP-difficile et nous exposons les méthodes utilisées pour résoudre ce problème. Une méthode exacte basée sur la modélisation linéaire en nombres mixtes et deux heuristiques sont proposées et testées sur des instances réelles et d'autres aléatoires.

Le chapitre 5 traite plusieurs extensions du problème initial en intégrant de nouvelles fonctions objectif et de nouvelles contraintes. Il expose les résultats de plusieurs scénarios d'usage.

Le chapitre 6 est consacré à une étude de notre problème de base de point de vue ordonnancement. Il s'agit alors d'étudier le problème d'ordonnement d'intervalles sous contraintes d'énergie. Nous proposons des études de complexité et des algorithmes d'approximation avec garantie de performance pour plusieurs variantes de ce problème.

Le chapitre 7 introduit un nouveau problème de tournées de véhicules électriques et il expose les heuristiques utilisées pour résoudre ce problème et les résultats obtenus sur des instances généralisées de la littérature.

Le chapitre 8 propose des approches méta-heuristiques basées sur l'algorithme de recherche locale à grand voisinage pour résoudre notre problème de tournées de véhicules électriques. Plusieurs méta-heuristiques hybrides sont ainsi proposées.

Finalement, le chapitre 9 présente une synthèse des travaux que nous avons réalisés et propose quelques perspectives de recherches futures.

Chapitre 2

Problématique industrielle

Sommaire

2.1 Contexte général	5
2.2 Projet Infini Drive	7
2.3 Écosystème du véhicule électrique	7
2.3.1 Le véhicule électrique	7
2.3.2 Infrastructure de recharge	8
2.3.3 Le réseau électrique	11
2.4 Problème industriel	12
2.4.1 Le métier La Poste	12
2.4.2 Le métier ERDF	14
2.4.3 Problème de gestion opérationnelle d'une flotte de véhicules électriques et thermiques	15
2.5 Conclusion	17

Ce chapitre expose le contexte industriel de notre étude. Il définit en détail l'écosystème du véhicule électrique et explique notre problématique industrielle.

2.1 Contexte général

Un changement de l'équation économique accompagne aujourd'hui l'émergence des véhicules électriques. En effet, pour les véhicules thermiques, l'utilisation du véhicule est le principal poste de coût, la possession a un coût réduit. Or, pour les véhicules électriques, c'est la possession qui est un élément principal de coût et l'usage est peu coûteux à court terme mais il représente un enjeu global au niveau de l'établissement. Remplacer un véhicule thermique par un autre électrique n'a alors de sens qu'en y associant une optimisation des usages et une gestion énergétique adaptée. En effet, le développement du véhicule électrique nécessite une gestion intelligente de tout un nouvel écosystème (Figure 2.1) composé des fournisseurs d'infrastructures, fournisseurs d'électricité, constructeurs automobile, etc.

Vu la complexité associée à l'écosystème du véhicule électrique, le développement du marché du véhicule électrique est confronté à de nombreuses difficultés principalement liées au coût et à la gestion de la batterie. Tout d'abord, l'autonomie limitée de la batterie représente un frein au développement du marché du véhicule électrique. Ainsi, la recharge de la batterie constitue une opération clef et présente une des principales barrières du développement économique. De plus,

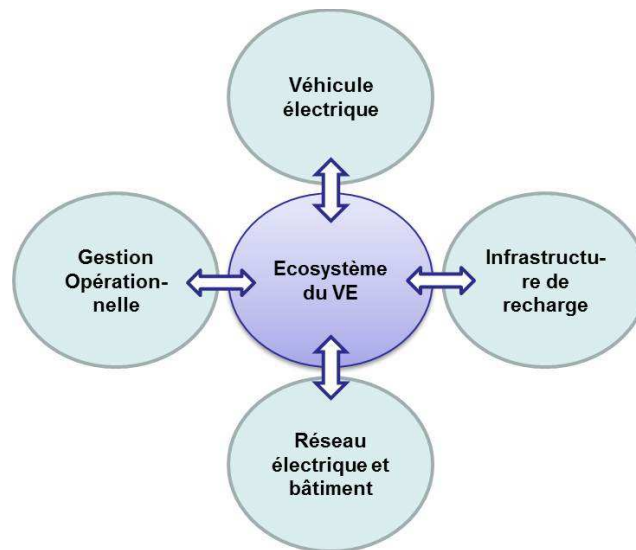


FIG. 2.1: Écosystème du Véhicule Électrique

le temps de recharge peut être assez long en fonction de la technologie de recharge utilisée. Une autre faiblesse est liée au réseau électrique qui est déjà saturé. Une adoption à grande échelle des véhicules électriques peut alors fortement impacter le réseau électrique public qui est déjà confronté à de nombreux défis. Ainsi, le nouvel écosystème des véhicules électriques (véhicules - chargeurs - réseau électrique) conduit à de nouveaux défis d'optimisation visant à développer des modèles efficaces et des outils d'aide à la décision performants pour gérer les flottes de véhicules électriques.

Afin de faire face aux faiblesses liées au véhicule électrique, plusieurs initiatives ont été menées pour soutenir son déploiement. Par exemple, le Comité des Transports et du Tourisme de la Communauté Européenne a voté une directive qui contraint le déploiement de 456.000 stations de recharge pour une utilisation publique en Europe d'ici 2020.

En outre, certains services liés aux infrastructures de recharge tels que la cartographie des services et les services d'itinérance sont déjà proposés par plusieurs start-ups et grandes entreprises. Cependant, pour assurer un déploiement réussi des véhicules électriques dans le court terme, il est nécessaire (1) d'identifier les catégories d'utilisation spécifiques pour lesquelles le véhicule électrique est le plus approprié (par exemple, les transports urbains, la logistique urbaine, etc.) en termes d'autonomie, capacité de la batterie, coûts d'exploitation, etc. et (2) d'assurer une gestion optimale du véhicule électrique dans le cadre de son écosystème (véhicules- chargeurs - réseau électrique - gestion de flotte) en mettant l'accent sur les nouveaux défis d'optimisation visant à développer des modèles et des outils d'aide à la décision efficaces pour gérer l'écosystème des véhicules électriques.

Dans ce cadre, le projet Infini Drive vise à concevoir un dispositif de recharge intelligente pour les flottes captives de véhicules électriques. Il a comme objectif de mettre à disposition des entreprises et des collectivités un système intelligent et communicant entre le véhicule, la borne, le réseau électrique et le système d'information existant, approprié à leur usage et environnement, au-delà des aspects techniques. Plus spécifiquement, le projet Infini Drive se concentre sur le développement d'outils d'aide à la décision performants qui permettent (i) d'établir des plans d'affectation des tournées aux véhicules électriques et thermiques disponibles tout en prenant en compte les besoins

énergétiques des tournées, les distances à parcourir, l'autonomie des véhicules et les contraintes techniques liées au réseau électrique, avec comme objectif d'assurer la rentabilité des véhicules électriques, (ii) de fournir des plans de recharge optimisés tout en cherchant à minimiser les coûts de la recharge.

2.2 Projet Infini Drive

Le projet Infini Drive [1] est un projet lancé en 2012 et financé par l'ADEME dans le cadre du programme Véhicule du Futur des Investissements d'Avenir. Il représente un consortium qui a réuni huit partenaires : le Groupe La Poste, ERDF, Greenovia, G²Mobility, Docapost BPO, LORIA, ARMINES et le laboratoire I3M.

L'objectif de ce projet est de concevoir un système de pilotage intelligent des infrastructures de recharge des véhicules électriques et de permettre un déploiement viable des flottes de véhicules électriques. Le système de pilotage de la recharge vise à assurer un fonctionnement intelligent et cohérent de l'ensemble du système que composent les véhicules, les infrastructures de recharge, le réseau électrique et l'organisation.

De plus, ce projet vise à expérimenter les infrastructures de recharge et fournir des recommandations pour une utilisation massive des véhicules électriques dans le cadre de flottes captives. Les infrastructures déployées devront couvrir l'ensemble des besoins des exploitants de flottes, notamment par rapport à la rentabilité économique, tout en faisant le meilleur usage du réseau et des moyens de production d'électricité. La validation des systèmes de pilotage développés passe par une série d'expérimentations effectuées sur des sites du Groupe La Poste et de ERDF. Les résultats des expérimentations ainsi que des recommandations concernant l'usage du véhicule électrique sont disponibles dans le livre des recommandations [1] rédigé à l'issue du projet.

2.3 Écosystème du véhicule électrique

2.3.1 Le véhicule électrique

Le véhicule électrique est composé d'une batterie pour stocker de l'énergie, d'un moteur électrique à courant continu avec un système de contrôle et d'un chargeur de batterie. La capacité de batterie d'un véhicule électrique est de l'ordre de 22 kWh, lui assurant une autonomie d'environ 150 kilomètres. La recharge de sa batterie dure entre 20 minutes et 10 heures en fonction du type des bornes de recharge utilisées et de la capacité de sa batterie.

Le véhicule électrique présente plusieurs avantages. Il s'agit, tout d'abord, d'un véhicule propre qui n'émet pas de fumées ou de particules lors de la phase de roulage. De plus, il dispose d'un bilan carbone, du puits à la roue, intéressant. Le véhicule électrique ne consomme pas d'énergie pendant les phases de ralenti et sa batterie récupère de l'énergie pendant les phases de décélération. Le véhicule électrique est aussi un véhicule viable avec des dépenses d'entretien réduites. Son moteur est parfaitement silencieux.

Même si le véhicule électrique compte beaucoup d'avantages, certains inconvénients freinent aujourd'hui son bon développement. En effet, il faut compter environ 8 heures pour charger un véhicule électrique en utilisant une prise domestique. Avec une autonomie limitée, des trajets qui dépassent 150 kilomètres sont difficilement envisageables. De plus, le coût d'achat initial du véhicule électrique reste élevé par rapport au coût de son équivalent thermique. Ce coût est principalement lié au coût élevé de la construction des batteries avec un coût d'environ 500 euros/kWh soit une batterie

à environ 10000 euros⁷. Le coût de batterie pénalise alors fortement le véhicule électrique, rendant son coût de possession près de 1.3 fois supérieur à celui des véhicules thermiques équivalents⁸.

2.3.2 Infrastructure de recharge

L'infrastructure de recharge est l'un des facteurs clés de succès du véhicule électrique. La recharge du véhicule électrique nécessite l'installation de bornes de recharge adaptées. Bien que la recharge d'un véhicule électrique semble simple à la majorité des utilisateurs, le système de recharge doit assurer certaines fonctions de mise en sécurité et dialoguer avec le véhicule pendant la connexion et la recharge. Il est également nécessaire d'étudier les types de connectiques à mettre en place côté infrastructure et côté véhicule afin de sécuriser l'opération de recharge.

À l'heure actuelle, plusieurs types de bornes de recharge, plusieurs modes de recharge et plusieurs types de connectiques sont disponibles sur le marché. Les bornes de recharge se distinguent par la puissance du courant délivré et leur tension d'alimentation. Nous identifions trois types de recharge : recharge lente, recharge accélérée et recharge rapide. Les modes de recharge se distinguent par le degré de communication entre le véhicule et l'infrastructure de recharge.

Dans ce qui suit, nous détaillons les différents types de bornes de recharge, les différents modes de recharge et les différents types de prises.

a. Modes de recharge

Les modes de recharge définissent la façon dont le véhicule électrique et l'infrastructure de recharge communiquent. Cette communication est très importante pour la sécurité de la recharge. À ce jour, quatre modes de recharge sont utilisés, dont trois dotés d'un mode de communication avec le réseau électrique.

i- Mode 1 Pour le Mode 1, le véhicule électrique est directement branché au réseau électrique par une prise de courant normalisé pour une intensité standard jusqu'à 16 ampères côté infrastructure. L'intensité de la recharge devra être limitée à 13 ampères, voire 10 ampères pour les prises domestiques pour optimiser leur durée de vie.

Ce mode est caractérisé par une absence de contrôle de recharge. Pour la recharge sur une prise non-dédiée (Modes 1 et 2), la sécurité dépend de l'état de l'installation électrique existante.

ii- Mode 2 Pour le Mode 2, le véhicule électrique est chargé à travers une prise non dédiée avec un dispositif de protection. Ce mode intègre un boîtier de contrôle sur le câble d'alimentation vérifiant l'intégrité du branchement du véhicule. Le mode 2 est recommandé pour une recharge lente occasionnelle et il s'apparente à un fonctionnement de type interrupteur (on/off) au branchement du véhicule. En mode 1, 2, on disposera globalement de 3 kW de puissance et une intensité de courant monophasé de 16 ampères.

iii- Mode 3 Pour le Mode 3, la recharge est effectuée à travers une prise fixe sur circuit dédié. En plus des fonctionnalités du mode 2, il est possible de moduler la recharge en intensité d'un véhicule électrique et d'avoir la possibilité de disposer de fonctions avancées de communication par la présence de fils pilotes. En ce mode, la puissance fournie peut atteindre 43 kW avec une intensité

⁷Source : Les batteries : l'enjeu majeur de la mobilité électrique ?, décembre 2012

⁸Source : COMMISSARIAT GÉNÉRAL AU DÉVELOPPEMENT DURABLE : Les véhicules électriques en perspective : Analyse coûts-avantages et demande potentielle

de courant triphasé de 63 ampères. Ce mode est alors recommandé pour une recharge normale à accélérée quotidienne.

Une infrastructure, composée de points de recharge en mode 3, peut disposer d'un système de pilotage de la recharge pour fixer des règles de recharge plus élaborées qu'un simple pilotage on/off. Par exemple, il est possible d'implémenter des règles sur le nombre de véhicules à charger simultanément ainsi que l'intensité de recharge de chaque véhicule, par exemple, ce mode de pilotage de la recharge est le plus souvent suffisant pour l'usage de véhicules électriques dans une activité répétitive et prédictive.

vi- Mode 4 Pour le Mode 4, la recharge en courant continu est assurée en utilisant un chargeur externe. Ce mode est principalement utilisé pour la recharge rapide. Pour ce mode, en courant continu, on disposera d'une puissance de 50 kW avec une intensité de 125 ampères.

La Figure 2.2 schématise les différents modes de recharge.

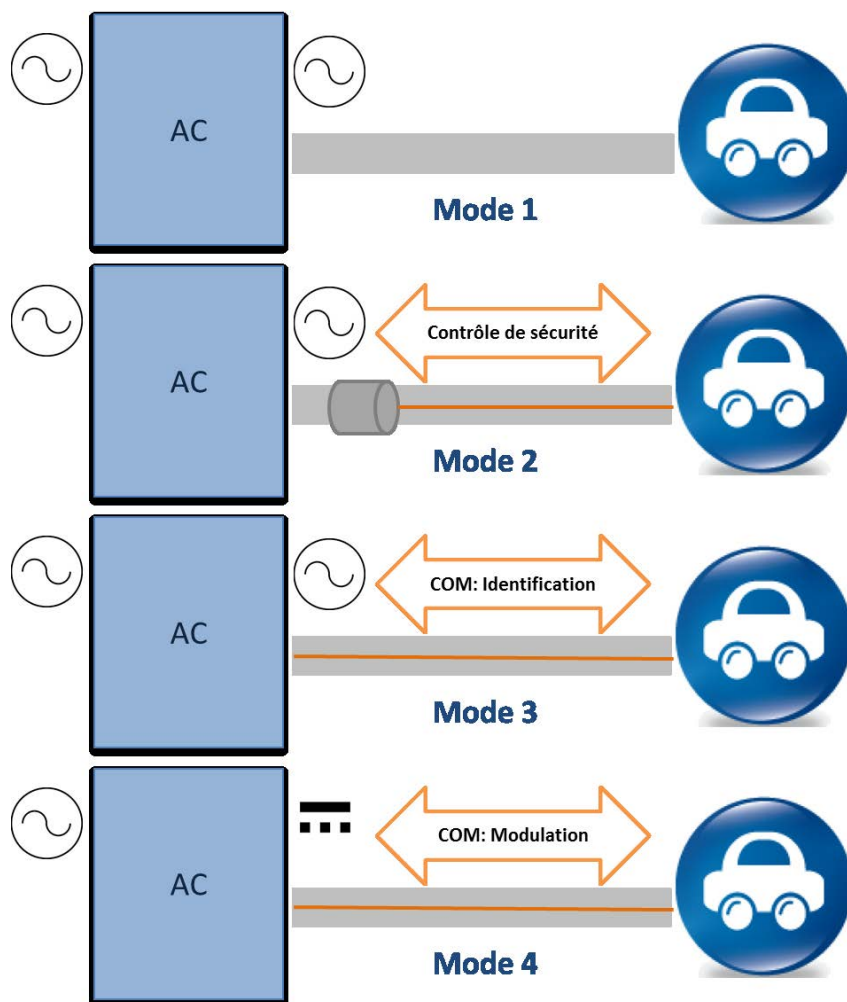


FIG. 2.2: Différents modes de recharge

b. Prises de recharge

Concernant la connectique côté infrastructure, les câbles de recharge peuvent être solidaires de la borne ou nomades. Il existe différents types de prises pour la recharge des véhicules électriques. Sur le marché, trois socles de prises sont actuellement utilisés : type E/F, type 2, type 3.

i- Prise de type E/F Ce type de prise permet la recharge des véhicules électriques, dès leur branchement au réseau, en mode 1 et 2. Ces prises délivrent un ampérage de 8-10 ampères en courant monophasé. Du fait d'une moindre intensité, les temps de recharge associés sont allongés. C'est le type de prise utilisé dans le cadre de recharges occasionnelles. Ces mêmes types de prises peuvent être renforcés pour délivrer un courant monophasé de 16 ampères garantissant la recharge d'une batterie en 6 à 8 heures.

ii- Prise de type 2 et 3 Les prises de type 2 et 3 permettent la recharge des véhicules électriques en mode 3 (permettant des fonctionnalités de communication). Les puissances associées à ces recharges peuvent être plus élevées si nécessaire. Les prises de type 3 sont conformes à la réglementation française actuelle.

La Figure 2.3 présente les différents types de connectiques côté infrastructure.



FIG. 2.3: Différents types de connectiques côté infrastructure

Concernant la connectique côté véhicule, deux types de connectiques sont présents sur le marché actuellement : connectique de type 1 et connectique de type 2.

La Figure 2.4 présente les différentes connectiques côté véhicule.

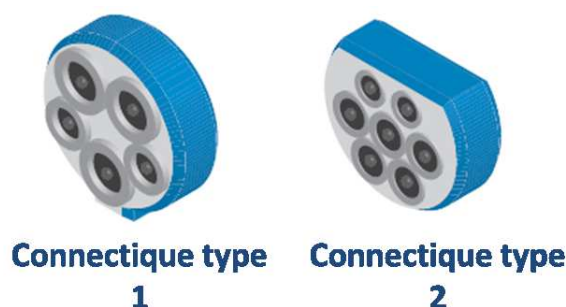


FIG. 2.4: Différents types de connectiques côté véhicules

c. Bornes de recharge

Nous distinguons trois types de bornes de recharge : la recharge normale, la recharge accélérée et la recharge rapide.

i- Recharge normale La recharge normale d'une puissance de 3.7 kVA (3 kW) est accessible à tous les véhicules électriques. Selon les modèles, une recharge de 6 à 8 heures est nécessaire pour recharger entièrement un véhicule électrique. Ce type de recharge s'effectue plutôt la nuit à domicile ou aux parkings des entreprises.

ii- Recharge accélérée Certains véhicules électriques disposent de batteries qui permettent des recharges accélérées ou rapides. La recharge accélérée, d'une puissance de 22 kVA, peut recharger la batterie d'un véhicule compatible en une heure.

iii- Recharge rapide La recharge rapide permet de réduire considérablement le temps de recharge d'un véhicule compatible. En effet, il est possible de charger 80% de sa batterie en moins de 30 minutes. Mais, le coût d'installation de ce type de chargeurs reste élevé par rapport au coût d'installation d'autres types de chargeurs. Ce type de recharge intervient généralement pour des utilisations professionnelles intensives (taxis, flottes de véhicules) ou à l'occasion de longs trajets.

Il existe aujourd'hui trois standards pour la recharge rapide :

- en courant continu (DC) avec une puissance de recharge jusqu'à 170 kW (850 V / 200 A)
- en courant continu (DC) avec une puissance de recharge de 50 kW (500 V / 100 A)
- en courant alternatif (AC) avec une puissance de recharge de 43 kW (400 V / 63 A)

Pour la recharge rapide à 43 kVA en courant alternatif triphasé, une prise «type 2» est nécessaire côté véhicule. Cette prise permet également les recharges normales et accélérées décrites précédemment.

2.3.3 Le réseau électrique

Le réseau électrique est un ensemble d'infrastructures énergétiques permettant d'acheminer l'énergie électrique des centres de production vers les consommateurs d'électricité.

Une installation électrique est caractérisée par sa puissance qui correspond à la quantité d'électricité maximum pouvant être consommée à un moment donné. Si cette puissance est dépassée à cause d'une consommation importante d'électricité, l'installation électrique disjoncte.

a. Contrat électrique

Le contrat électrique est un abonnement qui dépend de la puissance souscrite. Plus la puissance souscrite est élevée, plus le prix de l'abonnement est important et plus l'installation peut soutenir d'électricité sur le réseau à un moment donné. En plus de l'abonnement fixe, la facture d'électricité comprend une part variable qui dépend de la consommation d'électricité. Une facture d'électricité est donc composée d'une partie fixe qui dépend de la puissance souscrite et d'une partie variable qui dépend de la quantité d'électricité consommée et d'une éventuelle pénalité liée à un dépassement de la puissance souscrite.

En France, trois types de contrats électriques existent : le tarif bleu, le tarif jaune et le tarif bleu. Le tarif bleu s'adresse exclusivement aux particuliers et aux entreprises dont les installations sont alimentées en basse tension avec une puissance inférieure à 36 kVA. Pour ce type de contrat, aucun

dépassement de la puissance souscrite n'est autorisé. Le tarif jaune concerne les petites et moyennes entreprises qui ont une puissance de compteur comprise entre 42 et 240 kVA. Le tarif vert concerne les entreprises qui consomment des quantités importantes d'électricité et qui ont une puissance de compteur supérieure à 240 kVA. Pour les tarifs jaune et vert, le dépassement de la puissance souscrite est autorisé mais tout dépassement induit une pénalité horaire.

Pour les sites d'expérimentations La Poste et ERDF, c'est le tarif jaune qui est mis en place.

b. Puissance souscrite

La puissance souscrite est une caractéristique du contrat d'électricité. Elle correspond à la puissance maximale que le consommateur peut appeler sur le réseau électrique. Elle dépend du contrat d'électricité souscrit et du compteur. En cas de dépassement et en fonction de type du contrat souscrit, soit un disjoncteur est programmé pour suspendre l'alimentation électrique soit une pénalité horaire est appliquée.

c. Puissance de raccordement

La puissance de raccordement est la puissance maximale que l'installation de consommation d'électricité peut supporter. C'est une limite physique du réseau. Ainsi, la puissance souscrite doit toujours être inférieure ou égale à la puissance de raccordement. Dans l'hypothèse où un occupant souhaite augmenter sa puissance souscrite au-delà de la puissance de raccordement, des travaux coûteux sont nécessaires afin de modifier l'installation électrique.

2.4 Problème industriel

Les deux groupes utilisateurs du projet Infini Drive, La Poste et ERDF, présentent des contextes organisationnels différents vis-à-vis du véhicule électrique. Dans ce qui suit, nous exposons les particularités du métier La Poste et ERDF avant d'expliquer la problématique industrielle liée à la gestion opérationnelle d'une flotte de véhicules électriques et thermiques.

2.4.1 Le métier La Poste

L'architecture du schéma industriel de La Poste repose sur le principe de concentration puis dispersion du courrier entre les clients expéditeurs et les clients destinataires. Il s'agit, dans un premier temps, de collecter le courrier et ensuite d'assurer son acheminement et son tri dans les plateformes dédiées avant de le distribuer au client final. Les travaux extérieurs postaux se concentrent ainsi autour des opérations de collecte et de remise des envois postaux.

Les véhicules de La Poste réalisent une partie de leur activité le matin à partir de 7h, ils sont souvent remisés le midi pendant une période de 2 à 4 heures, et repartent pour certains pour réaliser une autre partie de l'activité l'après-midi. Ils sont systématiquement remisés le soir sur le site au plus tard à 18h, offrant une large plage horaire de recharge.

L'activité de distribution de courrier aux particuliers dans le cas de La Poste est caractérisée par des arrêts fréquents, en effet, le véhicule peut s'arrêter plus d'une centaine de fois par tournée.

Dans le cadre de son organisation de l'activité et des travaux extérieurs postaux, La Poste est assujettie à des contraintes d'ordre commercial qui structurent sa relation avec ses clients en fonction de l'offre et de la demande. La Poste assure principalement deux types de services : le Service Universel Postal et le service spécifique payant sur demande.

a. Le Service universel postal

Les modalités de la distribution du courrier assurée par La Poste au titre du service universel postal sont définies par l'article L. 1 du code des postes et des communications électroniques.

À ce titre, La Poste est tenue de fournir, sur l'ensemble du territoire, 6 jours sur 7, un service qui répond à des objectifs de qualité portant notamment sur la rapidité et la fiabilité des prestations.

La définition du service universel postal en France concerne des envois de correspondance nationaux et internationaux auxquels il convient d'ajouter, dans les dimensions nationales et internationales, des services d'envois de colis au guichet d'un poids inférieur ou égal à 20 kg, des offres de presse, le service de recommandation, de valeur déclarée, de réexpédition, etc.

Les services de levée (correspondant à la collecte des boîtes aux lettres jaunes) et de distribution relevant du service universel postal sont assurés tous les jours ouvrables selon des plages horaires définies par la réglementation.

Pour répondre aux exigences du service universel postal en termes de qualité du service de distribution offert, La Poste adapte les tournées de facteurs selon la demande et le type du milieu. Une grande partie du volume du courrier destiné au grand public et à certaines entreprises est distribué lors des tournées Quartier-Lettre (QL) appelées également tournées Ménage. Ces tournées correspondent à la distribution des envois de correspondance (lettres de petits formats, objets suivis ou non, etc.).

Les tournées de distribution de colis se distinguent quant à elles des tournées courrier par un parcours lié aux adresses à desservir, elles sont d'une plus grande variabilité (parcours, volume de colis à distribuer) et elles sont exclusivement réalisées à base de voitures. Les envois publicitaires peuvent aussi faire l'objet d'une tournée indépendante.

Pour les grandes communes, les tournées colis sont généralement assurées par la filiale de La Poste Coliposte et les tournées publicité par la filiale Médiapost. En milieu rural, on peut voir fusionner les trois types de tournées cités auparavant au sein d'une même tournée. Ainsi, c'est la tournée QL qui englobe tous les flux logistiques.

b. Les services spécifiques payants

Pour répondre à des besoins spécifiques exprimés par les entreprises et les administrations, des services spécifiques font l'objet de contrats commerciaux individualisés liant La Poste et ses clients. Les clients sous contrat de remise ou de collecte de courrier/colis disposent d'un traitement spécifique de leur courrier et bénéficient d'un autre type de tournées. C'est le service «Courses Spéciales (CS)» qui assure la remise du courrier à des horaires fixes précisés sur le contrat liant La Poste et le client. Ce service consiste en des tournées voiture spécialisées permettant d'effectuer des prestations payantes de remise précoce du courrier.

Afin d'assurer le meilleur service en termes de rapidité et de régularité, des tournées CEDEX (Courrier d'Entreprise à Distribution EXceptionnelle) sont dédiées à la distribution du courrier aux entreprises disposant d'un code CEDEX. On peut ici distinguer deux sous-types de tournées : les entreprises disposant d'un contrat CEDEX se voient distribuer leur courrier à l'heure souhaitée et les entreprises CEDEX sans contrat reçoivent leur courrier entre 9h et 12h.

On distingue également les tournées Clients Importants (CI). Certains clients de La Poste sont considérés comme des « Clients Importants » en raison de la quantité importante de courrier qu'ils reçoivent quotidiennement. Ces clients ne bénéficient pas d'un contrat de remise du courrier payant. Néanmoins, ce type de tournées permet à La Poste de mieux gérer ses flux logistiques.

c. Environnement interne et qualité de service

La Poste gère également des contraintes liées à son environnement interne (contraintes industrielles, techniques, organisationnelles, etc.) qui structurent le modèle économique et social de l'entreprise. Ce qui implique l'émergence d'autres formes de tournées qui visent principalement à améliorer la qualité du service offert par La Poste. Cependant, ces tournées ne constituent pas toujours des tournées complètes de distribution ou de collecte, mais elles agissent sur l'organisation de gestion des véhicules attachés à une entité postale. Parmi ces tournées, nous trouvons les tournées sécables. Cette notion a été introduite par la nouvelle organisation de l'activité postale en Facteur d'Avenir. Ce dispositif offre plus de flexibilité aux organisations pour faire face aux aléas. En effet, en cas d'absence inopinée d'un facteur, l'ensemble de son équipe se partage sa tournée afin de garantir la distribution du courrier le jour même.

Lors des jours faibles (il s'agit, généralement, des lundis et mardis pour les tournées ménage), le volume du courrier est plus faible que pour les autres jours, le titulaire de la tournée sécable est au repos et sa tournée est répartie entre les autres agents de l'équipe. Ce type de tournées permet d'économiser un véhicule, mais ceci augmentera la charge pour les véhicules engagés.

Pour l'activité postale, le poids a une influence sur le dimensionnement des tournées. Ainsi, des coffres-relais sont placés sur les principaux points de passage des villes et des tournées «dépôt-relais» sont réalisées par des véhicules 4 roues afin de charger ces «coffres relais» par le courrier et les colis. Ces tournées se déroulent pendant des plages horaires précises et le respect de ces plages horaires conditionne en grande partie le bon déroulement des tournées de facteurs alimentant leurs sacoches à travers les «coffres relais».

Nous distinguons également les tournées de liaison qui consistent à ramener le courrier mal acheminé à un établissement élémentaire de distribution du courrier où s'effectue l'échange des envois postaux avec les autres entités postales d'une zone donnée.

2.4.2 Le métier ERDF

Une grande partie du métier ERDF est axée sur les Travaux Interventions Clientèles. D'autres métiers tels que les chargés d'affaires travaux et les agents d'astreinte font également partie de l'activité ERDF.

a. Typologie de l'activité Travaux Interventions Clientèle

L'activité TIC (Travaux Interventions Clientèle) consiste à assurer les prestations d'intervention demandées par les fournisseurs d'énergie pour le compte de leurs clients. Ces interventions concernent par exemple la relève d'index de compteurs, le changement de panneau de comptage, etc. Certaines interventions peuvent être à l'initiative d'ERDF (contrôle de conformité, coupure d'électricité pour factures impayées...).

L'activité TIC est totalement programmable, sauf en milieu rural et semi-rural, avec des délais de temps variables :

- Activité classique : rendez-vous à 3 semaines
- Tournée du jour : rendez-vous à 5 jours sur certaines activités
- Tournée prioritaire : rendez-vous à la journée pour certaines interventions

Les activités TIC se divisent en des activités délibérées et des activités programmées. Les activités délibérées correspondent à des prestations de contrôle ou d'entretien (contrôle de compteurs,

remplacement de compteurs, fin de maintien d'alimentation, etc.). Elles peuvent être avec rendez-vous.

Les activités programmées le sont directement par les fournisseurs ou par le distributeur.

La typologie classique des tournées d'une agence urbaine est comme suit :

- 5% de tournées du jour (rendez-vous urgent à 3 ou 5 jours)
- 50% de tournées programmées à 3 semaines sans dépannage
- 20% de tournées programmées à 3 semaines avec dépannage
- 25% de tournées spécifiques

La typologie classique de répartition des tournées en milieu rural est composée de 95% de tournées avec programmation à 3 semaines et dépannage et de 5% de tournées du jour.

Les véhicules d'ERDF sont utilisés toute la journée à partir de 8h et ne sont pas toujours remis le midi sur les sites. Ils sont généralement remis sur le site le soir au plus tard à 19h. Dans une journée, un agent réalise de 15 à 25 prestations.

Concernant la répartition des tournées, environ 50% de l'activité TIC s'effectue en milieu urbain avec retour méridien d'environ 1 heure, dont la longueur est estimée à 60 kilomètres maximum. En milieu rural sont effectuées 25% à 30% des tournées avec une moyenne de 130 kilomètres sans retour méridien. De 15% à 25% des tournées sont effectuées en milieu semi-urbain avec un kilométrage moyen de 110/120 kilomètres, 50% des agents effectuant ces tournées retournent sur le site de travail le midi.

b. Les autres métiers

i- Chargés d'affaires travaux Cette activité assure la maîtrise d'ouvrage de conception et de réalisation des travaux de raccordement ou de mise à niveau du réseau. Ces chargés d'affaires effectuent généralement des déplacements pour rencontrer les équipes travaux. Ils font en moyenne 60 kilomètres par jour inégalement répartis sur la semaine.

ii- Responsables de la préparation du travail des agents d'exploitation Cette activité prépare les chantiers des agents d'exploitation (mode opératoire des chantiers, méthode de travail, etc.). Le kilométrage moyen journalier est d'environ 60 kilomètres comme pour les chargés d'affaires. Ces agents préparent de 4 à 5 chantiers par jour. Le temps de déplacement dépasse 2 heures par jour.

iii- Agents d'astreinte en milieu urbain Basés souvent au coeur des grandes métropoles, les agents d'astreinte effectuent en moyenne 5 sorties de trajet inférieur à 10 kilomètres.

2.4.3 Problème de gestion opérationnelle d'une flotte de véhicules électriques et thermiques

Remplacer des véhicules thermiques par des véhicules électriques est un choix viable si le bilan économique global est favorable à moyen terme et si l'usage du véhicule électrique est adapté aux activités de l'entreprise en termes d'autonomie et de capacité du réseau électrique.

a. Rentabilité économique du véhicule électrique

Pour pouvoir établir le bilan économique d'une flotte de véhicules, il est nécessaire de prendre en compte la totalité des coûts associés à l'achat et l'exploitation de chaque type de véhicule. L'analyse

économique d'un parc automobile est liée à l'investissement sur les véhicules, aux coûts en énergie associés, et également aux autres postes de coûts d'exploitation tels que l'entretien et la sinistralité.

La comparaison des coûts complets des véhicules électriques et thermiques de type utilitaire léger affiche une répartition des coûts avec une part «coûts fixes - coûts variables» inversée :

- Pour le VE, les coûts fixes représentent environ 2/3 du coût total de possession, les coûts variables 1/3. En faisant entrer dans le calcul des coûts fixes une partie des coûts liés à l'infrastructure, la partie fixe est plus conséquente mais traduit bien l'investissement nécessaire au départ. C'est ce que certains appellent le phénomène d'achat de carburant à l'avance.
- Pour le VT, les coûts fixes représentent environ 1/3 du coût total de possession, les coûts variables 2/3.

Dans le cas d'un déploiement massif de véhicules électriques, garantir l'équilibre et l'intérêt économique est primordial. Pour atteindre l'équilibre souhaité, il est important d'assurer une exploitation quotidienne intense des véhicules électriques pour compenser le coût initial d'investissement du parc sans dégrader la qualité du service offert aux clients. Le kilométrage annuel d'un véhicule est alors un paramètre important dans la détermination du point d'équilibre.

La Figure 2.5 présente la notion de point d'équilibre qui correspond au kilométrage à partir duquel le véhicule électrique devient pertinent économiquement.

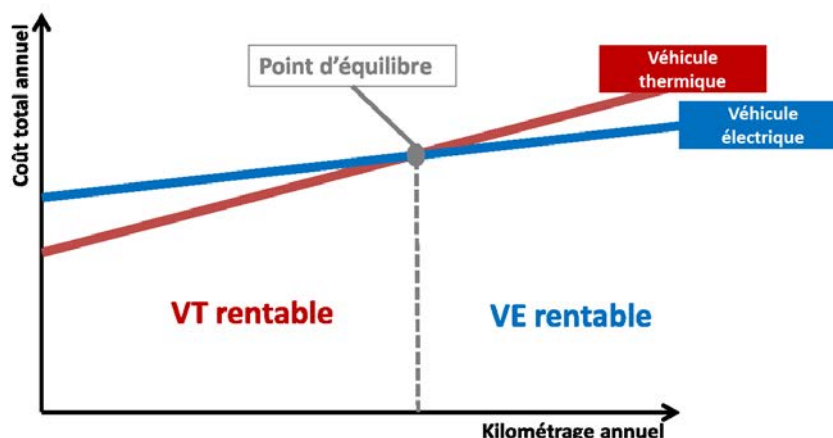


FIG. 2.5: Point d'équilibre du véhicule électrique

b. Contraintes d'autonomie et de recharge

C'est vrai que le véhicule électrique nécessite de rouler fréquemment pour être compétitif par rapport au véhicule thermique mais il est aussi essentiel d'assurer un usage qui s'inscrit dans la limite de l'autonomie réelle des véhicules électriques. En effet, l'usage du véhicule électrique induit des contraintes supplémentaires liées à l'autonomie limitée des batteries.

À la différence du véhicule thermique, le véhicule électrique s'inscrit dans un contexte encore plus large qui concerne l'infrastructure de recharge, le bâtiment et le réseau électrique. La recharge des véhicules électriques peut rajouter une charge supplémentaire sur le réseau électrique. Il est alors nécessaire d'optimiser la recharge des véhicules électriques en fonction de leur consommation et des tournées qui leur seront affectées.

c. Contraintes opérationnelles

Dans le cadre des activités de La Poste et d'ERDF, toutes les tournées sont cadencées et leurs durées sont précises. Ceci exige la prise en compte des heures de début et de fin fixes pour toutes les tournées même pour les tournées de nature flexible. Chaque tournée peut alors être assimilée à une tâche définie par un ensemble de caractéristiques qui serviront comme des données d'entrée pour l'algorithme d'affectation. La modélisation des usages des flottes captives doit permettre de prendre en compte un maximum de paramètres (typologie de tournée, type de véhicule, masse, trafic, conduite, conditions climatiques, topologie, vieillissement de la batterie...) et des contraintes métiers actuelles et à venir (capacité des parkings, nombre de kilomètres à parcourir, réglementation, RH...).

d. Objectif d'une gestion optimisée d'une flotte captive

La rentabilité du projet de déploiement des véhicules électriques doit se traduire par une réorganisation de l'affectation des véhicules sur l'ensemble des tâches à accomplir dans l'établissement pour réduire la part réalisée par les véhicules thermiques.

L'objectif d'une gestion opérationnelle optimisée d'une flotte captive est d'affecter les véhicules aux tournées de façon à assurer un bilan économique et environnemental optimisé et d'optimiser, simultanément, la recharge des véhicules en tenant compte des contraintes tarifaires et des appels de puissance du bâtiment.

Pour une flotte de véhicules électriques dont on connaît l'état de charge courant, établir son planning de charge consiste à déterminer les plages temporelles de connexion des véhicules aux bornes, ainsi que les niveaux d'intensité délivrés par ces bornes. Un planning obéit à des contraintes diverses : rétablir un état suffisant de charge avant utilisation, ne pas dépasser à chaque instant les seuils minimum et maximum de charge interne des batteries, ne pas soutirer trop de puissance électrique du réseau, etc. On peut associer à une politique de recharge un coût, qu'on cherchera à minimiser (euros, carbone émis, puissance maximum appelée...).

2.5 Conclusion

L'émergence des véhicules électriques induit une nouvelle équation économique et un nouvel écosystème plus complexe que celui du véhicule thermique.

Pour assurer un déploiement massif des véhicules électriques au sein des flottes d'entreprises, il est nécessaire de mettre en place des infrastructures de recharge adéquates, fonctionnelles et sécurisées. De plus, il est primordial de garantir l'équilibre et l'intérêt économique. Pour atteindre cet équilibre, il faut utiliser au mieux les véhicules déployés en garantissant un usage suffisant des véhicules électriques pour compenser leur coût à l'achat et les coûts des infrastructures de recharge et, en même temps, un usage qui s'inscrit dans la limite de l'autonomie réelle des véhicules. Il est également important de mettre en oeuvre une recharge intelligente pour optimiser le coût de la recharge et diminuer l'impact environnemental.

Chapitre 3

Revue de littérature de l'éco-système du véhicule électrique

Sommaire

3.1 Introduction	18
3.2 Notions sur la complexité	19
3.2.1 Classification des problèmes	19
3.2.2 La NP-Complétude	20
3.2.3 La Réduction polynomiale	20
3.3 Algorithmes d'approximation	20
3.4 Introduction à la recherche locale	21
3.4.1 Algorithme de recherche locale	21
3.4.2 Notion de voisinage	21
3.5 Méthodes de résolution combinatoire	22
3.5.1 Méthodes exactes	22
3.5.2 Méthodes approchées	23
3.6 État de l'art de l'écosystème du véhicule électrique	27
3.6.1 Smart Grid et V2G	27
3.6.2 Dimensionnement de l'infrastructure de recharge	28
3.6.3 Optimisation de la recharge	31
3.6.4 Optimisation des tournées de véhicules électriques	33
3.7 Conclusion	35

Dans ce chapitre, nous introduisons quelques notions utiles liées à la théorie de la complexité et aux algorithmes d'approximation. Nous exposons également les méthodes de résolution des problèmes d'optimisation les plus connues et nous effectuons une étude bibliographique de l'éco-système du véhicule électrique.

3.1 Introduction

L'émergence de la mobilité électrique et le déploiement du véhicule électrique doivent apporter des solutions aux enjeux liés à la dépendance énergétique vis à vis du pétrole, au réchauffement climatique et à la qualité de l'air. Cependant, le véhicule électrique est un écosystème complexe dont la fiabilité et la rentabilité économique sont conditionnées par un pilotage intelligent et optimisé de

toutes les composantes de cet écosystème. Son émergence dépend alors largement de la disponibilité et du coût des batteries, du développement et de la disponibilité des infrastructures de recharge, de l'intégration de ces infrastructures dans les réseaux électriques, etc.

Pour répondre au développement important du véhicule électrique, plusieurs groupes de recherche travaillent actuellement sur différentes problématiques liées à la mobilité électrique et au concept de "Smart Grid". Depuis 2012, le nombre d'études et de projets portant sur l'écosystème du véhicule électrique a considérablement augmenté. Les problématiques liées au véhicule électrique sont alors plus que jamais d'actualité et attirent de plus en plus l'attention des chercheurs, des industriels et des organisations gouvernementales.

Pour optimiser l'écosystème des véhicules électriques, on fait généralement appel aux techniques de la recherche opérationnelle qui sont orientées vers la recherche de la meilleure façon d'opérer des choix en vue d'aboutir au résultat visé ou au meilleur résultat possible.

Dans ce chapitre, différentes thématiques relatives à l'écosystème du véhicule électrique et plusieurs méthodes d'optimisation sont présentées.

Les travaux de la littérature concernant l'écosystème du véhicule électrique peuvent être structurés en 4 grands thèmes :

- Problème de *Smart Grid* ou *Vehicle to Grid* (V2G)
- Problème de dimensionnement de l'infrastructure de recharge
- Problème d'optimisation de la recharge
- Problème de transport des véhicules électriques

Avant de présenter les travaux de la littérature liés à chacun des thèmes, nous expliquons quelques notions liées à la théorie de complexité et aux algorithmes d'approximation et nous passons en revue les outils d'optimisation et les principales techniques de la recherche opérationnelle.

3.2 Notions sur la complexité

Dans cette section, nous présentons quelques définitions et notions concernant la théorie de la complexité.

La théorie de la complexité s'intéresse seulement aux problèmes de décision. Cette restriction est justifiée par le fait que les autres problèmes, tels que les problèmes d'optimisation, peuvent être facilement transformés en des problèmes de décision équivalents.

Définition. Un problème de décision est un problème dont la réponse est binaire : oui ou non.

3.2.1 Classification des problèmes

Le principe de l'étude de la théorie de la complexité vise à classer les problèmes de décision en fonction de temps d'exécution sur une machine de Turing. Il existe plusieurs classes de complexité dans la littérature dont les plus connues sont les suivantes :

- **La classe P.** Cette classe englobe les problèmes qui peuvent être résolus en un temps polynomial.
- **La classe NP.** Cette classe contient tous les problèmes de décision dont l'ensemble de solutions potentielles a un cardinal au pire exponentiel.

3.2.2 La NP-Complétude

La théorie de la NP-complétude concerne la reconnaissance des problèmes les plus durs de la classe NP. La notion de la difficulté d'un problème qui est introduite dans cette classe signifie que si un problème de cette classe est prouvé être facile alors tous les problèmes de NP le sont. Inversement, si l'un d'eux est prouvé être difficile, alors la classe NP est distincte de la classe P.

3.2.3 La Réduction polynomiale

La réduction polynomiale est la technique de base généralement utilisée pour prouver que des problèmes sont NP-complets.

Soient deux problèmes $P1$ et $P2$, réduire $P1$ en $P2$ signifie transformer une instance quelconque de $P1$ en une instance équivalente de $P2$; ainsi, $P2$ peut servir de sous-routine pour résoudre $P1$. Plus formellement, s'il existe une fonction polynomiale τ , tel que : $\forall x, x \in P1 \Leftrightarrow \tau(x) \in P2$, on dit que le problème $P1$ se réduit au problème $P2$. Le problème $P1$ est alors plus facile à résoudre et si on trouve un algorithme polynomial pour $P2$, on en a un pour $P1$ aussi.

Il est alors possible d'utiliser la réduction polynomiale pour montrer la difficulté de résolution d'un problème en transformant un problème déjà connu comme étant difficile à résoudre vers le problème de départ.

3.3 Algorithmes d'approximation

Dans cette section, nous introduisons le concept d'algorithmes d'approximation et nous présentons quelques définitions qui seront utiles pour la compréhension du reste de ce mémoire.

Contrairement à la résolution d'un problème de la classe P , où la solution est connue et elle peut être obtenue en un temps polynomial, l'approximation d'un problème d'optimisation NP-difficile repose sur l'obtention d'informations sur une solution qu'on ne connaît pas et qu'il est difficile de connaître. Il est donc intéressant d'essayer de borner le mieux possible les erreurs commises par les heuristiques utilisées.

Définition. Un algorithme d'approximation A pour un problème d'optimisation P calcule en un temps polynomial une solution réalisable dont la valeur objectif est proche de celle de la solution optimale.

Idéalement, la solution générée par l'algorithme d'approximation est garantie de différer de la solution optimale uniquement par une constante.

Définition. Un algorithme d'approximation absolue pour un problème d'optimisation P est un algorithme polynomial A pour lequel il existe une constante k telle que $|A(I) - OPT(I)| \leq k$ pour toutes les instances I de P .

Malheureusement, nous devons nous limiter, dans la plus part des cas, à des garanties de performance relatives.

Définition. Soit P un problème d'optimisation et $k \geq 1$. Un algorithme d'approximation de facteur k pour P est un algorithme polynomial A tel que $\frac{1}{k} \times OPT(I) \leq A(I) \leq k \times OPT(I)$ pour toutes les instances I de P . On dit également que A a un rapport de performance (ou une garantie de performance) égal à k .

Dans le cas des algorithmes en-ligne, nous évoquons la notion de rapport de compétitivité.

Définition. Soit A un algorithme en-ligne retournant à chaque étape i une solution valide S_i et soit S_i^* la solution optimale. L'algorithme A a un rapport de compétitivité c (ou est c -compétitif) si on a : $\forall i, c \times |S_i| \geq |S_i^*|$.

L'interprétation du rapport de compétitivité est comme suit. Un algorithme A est c -compétitif si à chaque étape i , il construit une solution valide dont l'objectif est au moins $\frac{1}{c}$ fois celui de la meilleure solution possible.

3.4 Introduction à la recherche locale

3.4.1 Algorithme de recherche locale

Les algorithmes de recherche locale font partie des algorithmes d'amélioration pour lesquels une meilleure solution est générée à chaque itération lors de l'exploration du voisinage de la solution courante.

La recherche locale est fondée sur le principe suivant : l'algorithme part d'une solution initiale réalisable. Cette solution initiale subit, par la suite, un ensemble de modifications locales afin d'améliorer la fonction objectif. L'inconvénient de la méthode de recherche locale est que la recherche peut se trouver bloquée dans un optimum local et ainsi générer une solution finale qui est différente de la solution optimale.

La qualité des solutions générées dépend alors des opérateurs de recherche locale utilisés et des espaces de recherche explorés.

Un espace de recherche représente l'espace des solutions réalisables qui vérifient l'ensemble des contraintes du problème.

3.4.2 Notion de voisinage

La définition d'une structure de voisinage est une étape essentielle dans la mise en place d'un algorithme de recherche locale. Le voisinage d'une solution S est un sous-ensemble de configurations de l'espace de recherche qui est directement atteignable à partir d'une transformation donnée de S . La notion de voisinage permet alors de structurer l'espace de recherche.

Pour illustrer la notion de voisinage, considérons les deux voisinages distincts présentés dans la Figure 3.4.2. Nous prenons l'exemple du problème de voyageur de commerce [2] dont une solution est composée d'une séquence de villes à visiter selon un ordre défini. Selon le voisinage V_1 , deux solutions sont voisines si et seulement si une permutation entre deux composantes de la solution permet de passer d'une solution à une autre. Selon le voisinage V_2 , deux solutions sont voisines si et seulement si une séquence de deux villes est déplacée vers une autre position de la tournée. Depuis une solution S quelconque, l'ensemble $V_1(S)$ est différent de $V_2(S)$.

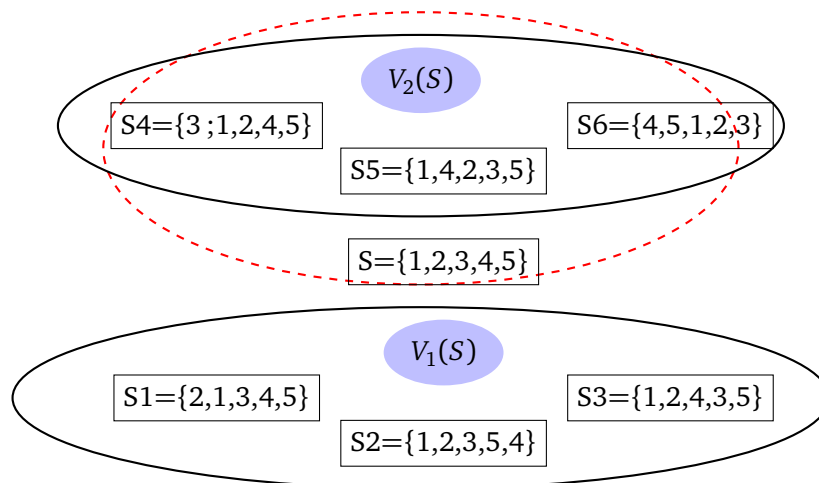


FIG. 3.1: Exemple de voisinage

3.5 Méthodes de résolution combinatoire

Si les méthodes de résolution exactes des problèmes d'optimisation combinatoire ont l'avantage de garantir l'optimalité des solutions obtenues, le temps de calcul nécessaire pour obtenir une solution optimale peut être très élevé. Dans certains cas, il est alors préférable d'accepter des solutions de bonne qualité, sans que l'optimalité ne soit assurée, au profit d'un temps de calcul plus raisonnable.

Dans cette section, nous présentons les principales méthodes de résolution des problèmes d'optimisation. Nous exposerons alors les méthodes exactes et les méthodes approchées les plus utilisées.

3.5.1 Méthodes exactes

Les méthodes exactes sont nombreuses et se caractérisent par le fait qu'elles permettent d'obtenir des solutions dont l'optimalité est garantie. Dans ce qui suit, nous présentons les méthodes exactes les plus connues.

- La méthode énumérative consiste à effectuer une énumération explicite de toutes les solutions possibles du problème. Le temps de calcul de cette méthode augmente exponentiellement dès que le nombre de variables augmente. Cette méthode s'avère alors peu efficace.
- La programmation dynamique consiste à placer le problème dans une famille de problèmes de même nature mais de difficulté différente puis à trouver une relation de récurrence liant les solutions optimales de ces différents problèmes. La solution optimale d'un problème est alors obtenue en combinant les solutions optimales d'un ensemble de sous-problèmes. Les solutions sont calculées d'une façon ascendante en partant des solutions des sous-problèmes les plus petits pour arriver progressivement aux solutions de l'ensemble.
- La méthode de séparation et d'évaluation (Branch & Bound) consiste à faire une énumération en séparant le problème en sous-problèmes et en évaluant ceux-ci à l'aide d'une relaxation (continue ou lagrangienne principalement) jusqu'à ne plus avoir que des problèmes faciles à résoudre ou dont on sait avec certitude qu'ils ne peuvent pas contenir de solutions optimales. La phase de séparation consiste à diviser le problème en un ensemble de sous-problèmes de telle façon que les ensembles de solutions de tous ces sous-problèmes forment une partition.

La phase d'évaluation des noeuds de l'arbre de recherche permet de trouver l'optimum parmi l'ensemble de solutions réalisables ou, au contraire, de prouver que l'ensemble de solutions ne contient aucune solution intéressante, notamment à travers des comparaisons avec des bornes inférieures et supérieures. Dans ce dernier cas, il n'est plus utile d'effectuer la séparation de l'ensemble de solutions.

- La méthode de génération de colonnes permet de résoudre efficacement des programmes de grande taille. Partant d'un programme linéaire avec un sous-ensemble de colonnes de petite taille (quelques variables), il s'agit d'ajouter les variables qui sont susceptibles d'améliorer le coût de la solution courante. La génération de colonnes répète ce processus d'ajout de variables jusqu'à obtenir une solution satisfaisante à la totalité du problème.

3.5.2 Méthodes approchées

Les méthodes exactes permettent de trouver des solutions optimales pour des problèmes de taille raisonnable. Mais, ces méthodes rencontrent généralement des difficultés avec les instances de taille importante. Pour faire face à ces difficultés, des méthodes approchées peuvent être utilisées. Les méthodes approchées se divisent en des heuristiques spécifiques, conçues pour résoudre un problème particulier et des méta-heuristiques plus performantes et adaptables pour résoudre un grand nombre de problèmes différents. Les heuristiques peuvent être classées en deux catégories :

- Méthodes constructives qui génèrent des solutions à partir d'une solution initiale partielle en essayant d'en ajouter progressivement des éléments jusqu'à ce qu'une solution complète soit obtenue.
- Méthodes d'amélioration qui partent d'une solution initiale complète et l'améliorent d'une façon répétitive en explorant son voisinage afin d'obtenir une solution d'une meilleure qualité. Les algorithmes de recherche locale représentent un vaste espace des méthodes d'amélioration.

La qualité des solutions générées par les heuristiques peut être mesurée analytiquement en montrant que les solutions obtenues sont très proches des solutions optimales, c'est le cas des heuristiques avec garantie de performance. Il est également possible d'évaluer la qualité des solutions expérimentalement en testant les heuristiques développées sur un nombre suffisant d'instances.

Par ailleurs, pour faire face aux difficultés rencontrées par les heuristiques pour obtenir de bonnes solutions, les méta-heuristiques peuvent être adaptées et utilisées pour résoudre différents types de problèmes d'optimisation réputés difficiles. Certaines méta-heuristiques peuvent travailler avec une population de solutions (algorithmes génétiques, colonies de fourmis, etc.) alors que d'autres ne manipulent qu'une solution à la fois (recuit simulé, recherche tabou, etc.).

Le recuit simulé

La méthode du recuit simulé a été introduite comme étant une méthode de recherche locale qui utilise une stratégie pour échapper des minima locaux [3–6]. Le principe du recuit simulé est de parcourir de manière itérative l'espace des solutions. On part avec une solution initialement générée de manière aléatoire dont correspond une énergie initiale et une température initiale généralement élevée.

À chaque itération de l'algorithme, un changement élémentaire est effectué sur la solution. Cette modification fait varier l'énergie du système. Si cette variation est négative (la nouvelle solution améliore la fonction objectif, et permet de diminuer l'énergie du système), elle est acceptée. Si la solution trouvée est moins bonne que la précédente alors elle sera acceptée avec une certaine

probabilité.

L'organigramme de l'algorithme du recuit simulé est donné sur la figure 3.2.

Les inconvénients du recuit simulé sont principalement liés aux réglages de l'algorithme (gestion de la décroissance de la température, etc.) et au temps de calcul qui peut devenir assez long. En revanche, l'algorithme du recuit simulé a l'avantage d'être facile à implémenter.

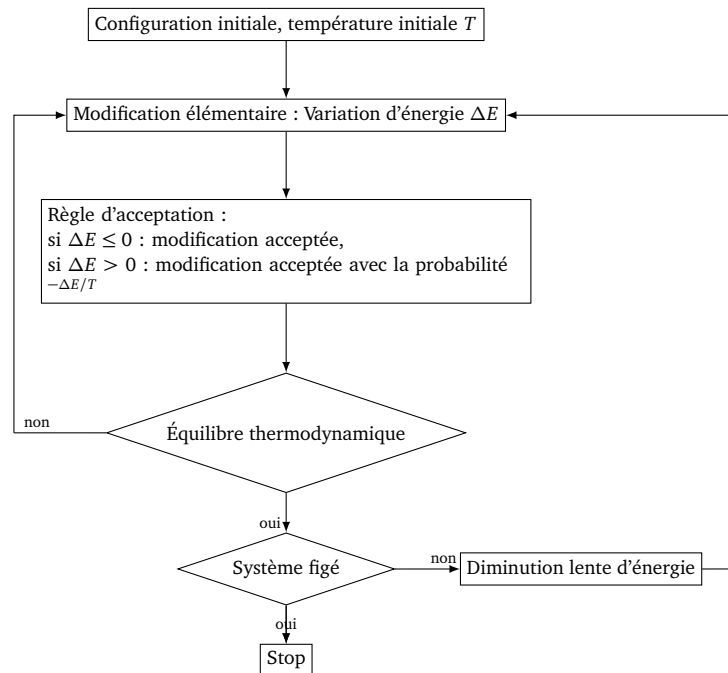


FIG. 3.2: Schéma général de l'algorithme de recuit simulé

Dans la littérature, le recuit simulé a été largement utilisé pour résoudre un grand nombre de problèmes difficiles tels que les problèmes d'ordonnancement [7, 8], les problèmes de tournées de véhicules [9], etc.

La recherche tabou

La méthode de recherche tabou a été proposée depuis environ une trentaine d'années par F. Glover [10]. La recherche tabou est une méthode de recherche locale combinée avec un ensemble de techniques inspirées de la mémoire humaine et permettant d'éviter d'être piégé dans un minimum local ou d'engendrer un cycle [11].

La principale particularité de la recherche tabou est l'utilisation d'une mémoire. En effet, à partir d'une solution donnée, il s'agit d'explorer le voisinage de cette solution et de choisir une nouvelle solution qui minimise la fonction objectif. Il est alors possible d'augmenter la valeur de la fonction objectif si la solution courante de l'algorithme est un minimum local. Afin d'éviter de retomber dans le même minimum local pendant les itérations suivantes, l'idée de la recherche tabou est d'utiliser une sorte de mémoire qui permet d'interdire de revenir sur les dernières positions visitées.

À chaque itération de l'algorithme de recherche tabou, pour passer de la solution courante à la solution suivante, deux étapes sont suivies :

- On construit l'ensemble des solutions réalisables obtenues à partir de la solution courante en appliquant un mouvement élémentaire.

- On évalue la fonction objectif de chaque solution. La solution qui sera retenue est celle avec la meilleure valeur de fonction objectif. Cette nouvelle solution est adoptée même si elle est moins bonne que la solution courante, c'est grâce à cette particularité que la méthode de recherche tabou peut échapper d'un optimum local. Pour éviter de retourner à une solution déjà retenue dans les itérations précédentes, les derniers mouvements effectués sont stockés dans une liste de mouvements interdits appelée "liste tabou".

L'organigramme de l'algorithme de recherche tabou est représenté sur la figure 3.3.

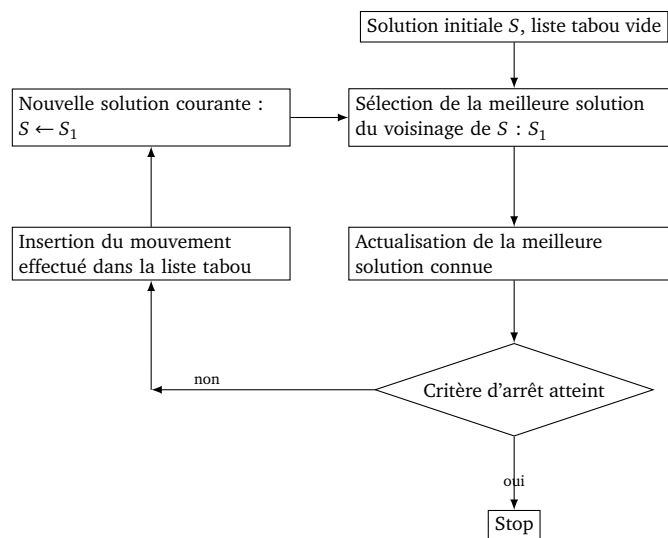


FIG. 3.3: Organigramme de l'algorithme de recherche tabou

Les méthodes de recherche tabou ont permis de fournir des solutions de très bonne qualité et ont prouvé leur efficacité dans la résolution de plusieurs problèmes d'optimisation combinatoire [12, 13].

Les algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique [14, 15]. On part d'une population de solutions initiales arbitrairement choisies (chromosomes). On évalue leur performance (Fitness) relative. Sur la base de ces performances, on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. Quelques individus se reproduisent, d'autres disparaissent et seuls les individus les mieux adaptés sont supposés survivre. On recommence ce cycle jusqu'à ce qu'on trouve une solution satisfaisante. Étant donné que les algorithmes génétiques travaillent avec une population de solutions, ils ont été largement utilisés pour résoudre des problèmes d'optimisation multi-objectif [16].

Les algorithmes de colonies de fourmis

Les algorithmes de colonies de fourmis sont des algorithmes itératifs à population où tous les individus partagent un savoir commun qui leur permet d'orienter leurs futurs choix et d'indiquer aux autres individus des choix à suivre ou à éviter [17].

Le principe de cette méta-heuristique repose sur le comportement particulier des fourmis qui utilisent pour communiquer une substance chimique volatile particulière appelée phéromone grâce

à une glande située dans leur abdomen. En quittant leur nid pour explorer leur environnement à la recherche de la nourriture, les fourmis arrivent à élaborer des chemins qui s'avèrent fréquemment être les plus courts pour aller du nid vers une source de nourriture. Chaque fourmi dépose alors une quantité de phéromones sur ces pistes qui deviendront un moyen de communication avec leurs congénères, les fourmis choisissent ainsi avec une probabilité élevée les chemins contenant les plus fortes concentrations de phéromones à l'aide des récepteurs situés dans leurs antennes.

Les algorithmes de colonies de fourmis ont été utilisés pour résoudre un grand nombre de problèmes, tels que les problèmes de transport [18, 19], les problèmes d'affectation [20], etc.

La recherche locale itérative

La recherche locale itérative, ou encore Iterated Local Search (ILS), repose sur deux phases : une phase de recherche locale et une phase de perturbation. Les étapes principales de ILS ont été proposées par Lourenço et al. [21]. L'algorithme de ILS part d'une solution initiale à laquelle une recherche locale est appliquée pour générer un nouvel optimum local. Ensuite, une perturbation est appliquée sur cet optimum local pour générer un nouveau point de départ. Finalement, une recherche locale est de nouveau appliquée sur la solution perturbée pour générer un nouvel optimum local. Ce nouvel optimum local peut être soit accepté ou rejeté en fonction du critère d'acceptation de l'algorithme. Si la nouvelle solution est acceptée, elle devient le nouveau point de départ de l'algorithme. Ce processus est répété jusqu'à ce qu'un critère d'arrêt soit satisfait.

Dans la littérature, l'algorithme de ILS a été largement utilisé pour résoudre un grand nombre de problèmes d'optimisation [22] tels que les problèmes d'ordonnancement [23], les problèmes d'affectation quadratique [24], le *Team Orienteering Problem* [25], etc.

La recherche à voisinage variable

La recherche à voisinage variable, ou encore Variable Neighborhood Search (VNS), a été introduite par Mladenović et Hansen [26] en 1997. Elle représente une méta-heuristique récente pour la résolution des problèmes d'optimisation. À la place d'utiliser une seule structure de voisinage, l'idée de base de cette méta-heuristique est le changement systématique de la structure de voisinage au sein d'une recherche locale, et ceci dans un ordre prédéfini, afin d'explorer de plus grands espaces de recherche et ainsi échapper des minima locaux.

Depuis son introduction, la recherche à voisinage variable a vu le développement d'un nombre important de variantes de diverses complexités. La plus simple d'entre elles est la descente à voisinage variable (Variable Neighborhood Descent) (voir Hansen et al. [27] pour plus de détails). On trouve également une autre variante appelée recherche à voisinage variable réduite qui est principalement utilisée pour résoudre des instances larges [28].

Dans la littérature, l'algorithme de VNS a été utilisé pour résoudre différents problèmes d'optimisation dans des domaines aussi variés que le data mining, la localisation, l'ordonnancement ou la théorie des graphes [29, 30]. De plus, l'algorithme de VNS a été largement utilisé pour résoudre plusieurs variantes du problème de tournées de véhicules tels que les problèmes de tournées de véhicules avec fenêtres de temps et plusieurs dépôts [31, 32], les problèmes de tournées périodiques [33] et les problèmes de tournées de véhicules très larges [34]. Pour une revue plus complète des applications de VNS, le lecteur est invité à se référer à [27].

3.6 État de l'art de l'écosystème du véhicule électrique

Comme expliqué auparavant, nous présentons, dans cette section, une revue de littérature des problèmes d'optimisation issus de l'écosystème du véhicule électrique. Nous détaillons alors les problèmes de "Smart Grid" et de "Vehicle-to-Grid", les problèmes de dimensionnement de l'infrastructure de recharge, les problèmes d'optimisation de la recharge et les problèmes de transport des véhicules électriques.

3.6.1 Smart Grid et V2G

Le déploiement d'un grand nombre de véhicules électriques conduit à de nouveaux défis mais peut également apporter d'autres opportunités et des solutions à des enjeux énergétiques. En effet, l'intégration des véhicules électriques dans le système de distribution électrique peut, d'une part, ajouter une charge supplémentaire sur le réseau électrique. Mais, les véhicules électriques pourraient, d'autre part, représenter une capacité additionnelle de stockage d'énergie étant donné qu'il est possible d'utiliser l'énergie stockée dans la batterie pour l'injecter sur le réseau en période de forte demande, ou inversement, de charger les véhicules en période de faible demande afin d'absorber le surplus de la production. Il s'agit du concept du «Vehicle-to-Grid» (V2G), littéralement «du véhicule vers le réseau», qui consiste à utiliser les batteries des véhicules électriques comme une capacité de stockage mobile.

Le terme "Vehicle-to-Grid" désigne un système dans lequel les véhicules électriques communiquent avec le réseau électrique pour équilibrer la demande en électricité soit en stockant de l'énergie dans leurs batteries pendant le temps où la demande sur le réseau est faible soit en injectant de l'énergie sur le réseau électrique quand la demande est forte. Ces batteries jouent le rôle d'une source d'énergie qui peut prévenir ou retarder le délestage.

Réellement, le concept de V2G s'intègre dans le contexte général de "Smart Grid". Le terme "Smart Grid" désigne un réseau de distribution d'électricité "intelligent" qui a comme objectif d'optimiser la production, la distribution et la consommation de l'électricité afin d'améliorer l'efficacité énergétique du réseau électrique.

Les études qui se concentrent sur les problèmes de V2G ont comme objectif de définir la connexion entre les batteries des véhicules électriques et le réseau électrique [35], de prouver la validité du concept de V2G [36], d'identifier les nouveaux marchés, d'estimer la viabilité économique de ce modèle [37, 38] et d'optimiser l'énergie dans un contexte de V2G [39, 40].

Les études sur la viabilité économique du modèle de V2G prouvent que le développement des systèmes de V2G pourrait apporter des revenus supplémentaires pour les véhicules propres, assurer plus de stabilité et de fiabilité par rapport au réseau électrique, diminuer les coûts associés au réseau électrique et développer un système rentable de sauvegarde et de stockage de l'énergie [37, 38].

Le problème d'optimisation de l'énergie dans un contexte de V2G consiste à décider si le véhicule électrique doit être chargé ou déchargé pour contribuer à la production énergétique et à assurer l'équilibre entre l'offre et la demande électrique. Un tel équilibre participe au lissage de la courbe de production énergétique et permet ainsi de réduire l'utilisation des centrales thermiques très polluantes en termes d'émission CO₂.

Dans [39], les auteurs proposent une méthode d'optimisation pour gérer la recharge d'un ensemble de véhicules hybrides et électriques dans un contexte de V2G. Chaque véhicule est caractérisé par sa capacité de batterie, il arrive sur le parking à une heure donnée et précise l'heure de départ prévue ainsi que l'état de charge souhaité à son départ. L'objectif est de gérer la recharge et la décharge des véhicules disponibles en déterminant, pour chaque véhicule, les horaires pendant lesquels le véhicule doit injecter de l'énergie sur le réseau électrique et les créneaux pendant lesquels

il doit être chargé. Chaque véhicule doit quitter le parking à l'heure de départ précisée avec l'état de charge souhaité. Afin de générer un planning de recharge et de décharge des batteries tout au long de la journée, une optimisation par essais de particules est appliquée. L'objectif de cette optimisation est de maximiser le profit des propriétaires des véhicules tout en s'assurant que les contraintes liées au réseau électrique et aux propriétaires des véhicules sont satisfaites.

Nicolas et al. [41] décrivent une conception d'un «smart grid» qui permet la gestion d'un pool de batteries interconnectées et qui a comme objectif de minimiser l'impact économique d'un parc de véhicules électriques dont les batteries soutirent et injectent sur le réseau. Pour quantifier les coûts de la gestion d'une flotte électrique composée de 2 millions véhicules, une modélisation mathématique a été proposée et deux cas ont été étudiés : le cas d'un opérateur gestionnaire de la flotte entière de 2 millions de véhicules électriques qui doit d'une part répondre à la demande électrique de ses clients et d'autre part optimiser les périodes de recharge pour réduire ses coûts et le cas d'un producteur d'électricité qui a à sa disposition des centrales de production. Différents profils d'utilisateurs ont été cartographiés selon les moments d'utilisation de leurs véhicules électriques. Pour le premier cas, il s'agit de minimiser les coûts du gestionnaire de flotte. Cet objectif traduit le fait que les prix élevés doivent correspondre à des injections sur le réseau électrique et les prix faibles à la recharge des batteries. Les résultats de la modélisation pour une journée montrent que le pic de la courbe initiale de charge (19h) a été légèrement réduit mais une nouvelle pointe apparaît. Pour le deuxième cas, l'objectif est de minimiser l'écart à la charge moyenne journalière. Les résultats de la modélisation montrent que l'utilisation de cette nouvelle fonction objectif permet de réduire la pointe sans créer d'autres pics.

3.6.2 Dimensionnement de l'infrastructure de recharge

Comme le marché du véhicule électrique est en pleine expansion depuis quelques années, le déploiement des infrastructures de recharge devient une nécessité pour rassurer les usagers du véhicule électrique. De plus, la disponibilité d'une infrastructure de recharge adéquate pourrait dissiper les inquiétudes des particuliers vis à vis de l'autonomie limitée de la batterie et ainsi améliorer le taux d'adoption des véhicules propres. Le dimensionnement et le déploiement des infrastructures de recharge conditionnent alors en grande partie le déploiement des véhicules électriques sur le marché mondial. De point de vue scientifique, plusieurs modèles ont été proposés dans la littérature pour définir les meilleurs emplacements des stations de recharge.

Généralement, positionner des infrastructures afin de répondre aux demandes de clients répartis sur un espace géographique est une problématique retrouvée dans de nombreux problèmes dits de localisation comme le *facility location problem* [42, 43] ou le *set covering problem* [44, 45] qui sont tous des problèmes NP-difficiles. Plusieurs variantes du problème de localisation ainsi que des techniques d'optimisation et des méthodes exactes de résolution ont été exposées dans [44] et [46].

Afin de proposer des modèles de localisation adéquats, il faut commencer par estimer la demande en charge des véhicules électriques. Pour ce faire, des approches d'estimation de la demande en charge se basant sur les flux de trafic traversant les noeuds d'un réseau routier [47, 48] ou se basant sur la matrice des déplacements des utilisateurs des véhicules électriques [49], ont été utilisées.

Concernant la résolution du problème de dimensionnement d'une infrastructure de recharge, plusieurs modèles mathématiques et heuristiques ont été proposés. Touati-Moungla et Jost [50] exposent trois catégories de modèles utilisés pour résoudre les problèmes de localisation des stations-services. Ils distinguent alors les modèles basés sur le "*maximum covering location problem*", les modèles basés sur le "*set covering problem*" et les modèles basés sur le "*maximum covering/shortest path problem*".

Concernant l'objectif de l'optimisation, certaines études s'intéressent à la minimisation des pertes

des utilisateurs sur leur chemin vers les stations de recharge [47, 48], d'autres à la maximisation du nombre de demandes satisfaites [51, 52] ou même à la minimisation du temps total du trajet de chaque véhicule électrique incluant le temps d'attente dans une station de recharge [53], etc.

Par rapport aux caractéristiques des problèmes de dimensionnement des infrastructures de recharge étudiés dans la littérature, nous distinguons (i) les cas où le nombre de stations de recharge est fixé au préalable et il s'agit d'optimiser les localisations de ces stations [53] et (ii) les problèmes dont l'objectif est d'optimiser le nombre, les capacités et les emplacements des stations de recharge à déployer [54].

Afin de concevoir une infrastructure de stations de recharge, Gavranović et al. [55] adoptent une approche basée sur le problème du p -médian où p est une donnée du problème. Cette approche optimise la localisation de p stations de recharge par rapport à un ensemble de clients, de sorte que la somme des distances pondérées entre les clients et les stations de recharge soit minimisée. Le modèle proposé prend également en compte les préférences de localisation exprimées par les managers d'une entreprise sous la forme de scores incorporés dans la fonction objectif. De plus, ce modèle intègre des contraintes sur le nombre maximal de quartiers ainsi que la population maximale que peut servir une station de recharge. La fonction objectif est égale à la somme de la demande de chaque région multipliée par la distance parcourue jusqu'à la station à laquelle elle est affectée, divisée par un score de préférence pour cette localisation.

Li et al [56] utilisent un algorithme génétique pour définir les meilleurs emplacements des stations de recharge. La méthode adoptée est basée sur la théorie de conservation des flux de trafic régional pour prédire le nombre de véhicules électriques à servir dans chaque région. Le modèle proposé consiste à minimiser le coût total relatif au déploiement des stations de recharge. Ce coût est égal à la somme du coût fixe initial d'investissement, du coût variable d'exploitation des stations de recharge et du coût d'usure.

Mehar et al. [54] proposent de résoudre le problème de localisation d'un ensemble de stations de recharge afin de satisfaire la demande des clients. Il est considéré que les stations de recharge ont des capacités limitées et que les clients ont une certaine demande. Le modèle proposé prend également en compte la concentration du trafic dans les zones considérées et la capacité du réseau électrique. Il s'agit de déterminer le nombre et les emplacements des stations de recharge à déployer et d'affecter les clients aux stations déployées. Pour ce faire, deux fonctions objectif doivent être minimisées : le coût de déploiement des stations de recharge et le coût de transport entre les clients et les stations dans lesquelles ils seront servis. Ces deux fonctions sont combinées en une seule fonction objectif en associant un poids à chaque fonction. Pour résoudre ce problème, un algorithme génétique est proposé. Une phase de prétraitement permet de créer des groupements de clients en se basant sur les données concernant le trafic.

Frade et al. [51] utilisent la ville de Lisbonne comme étant un cas d'étude pour l'application d'un modèle de localisation à couverture maximale (*maximal covering location model*) pour répondre à un maximum de demande. L'objectif du problème d'optimisation est de déterminer le nombre, les emplacements et les capacités des stations de recharge à installer. La méthodologie suivie pour résoudre ce problème consiste à estimer la demande et déterminer ensuite les emplacements des stations qui permettent de mieux satisfaire la demande estimée.

L'étude présentée dans [57] propose d'optimiser le dimensionnement et le déploiement d'une infrastructure de recharge pour les véhicules électriques. Cette étude prend en considération de nombreux facteurs tels que la demande en charge des véhicules électriques, des modèles de comportement des utilisateurs, la structure du réseau routier, les coûts de construction et d'exploitation des stations de recharge, les frais engagés par les utilisateurs, etc. Dans le cadre de cette étude, les utilisateurs des véhicules électriques peuvent choisir le chemin le plus court entre les noeuds du réseau routier et les stations de recharge. L'objectif ici est de déterminer le nombre, les

emplacements et les capacités des stations de recharge à déployer tout en cherchant à minimiser les coûts d'investissement et les coûts d'exploitation de l'infrastructure de recharge. Il s'agit également de réduire les coûts engagés par les utilisateurs, à savoir les coûts de recharge et les coûts du trajet effectué pour atteindre les stations de recharge. Après la construction d'un modèle de réseau de transport en utilisant la théorie des graphes, un programme quadratique en nombres mixtes est proposé pour résoudre le problème d'optimisation. Une étude de cas montre que la méthode proposée peut effectivement réduire les coûts d'investissement et d'exploitation des stations de recharge.

Wang [58] expose une modélisation linéaire en nombres entiers du problème de localisation des stations de recharge pour des scooters électriques. L'objectif de cette étude est de déterminer le nombre optimal ainsi que les localisations optimales des stations de recharge tout en cherchant à minimiser les coûts de déploiement des stations et satisfaire la demande des voyageurs partant d'un même point d'origine et arrivant à une même destination. Le modèle proposé prend en compte l'autonomie des véhicules, le profil des conducteurs, les conditions routières, le temps passé à la destination, la distance à parcourir, les contraintes liées au nombre maximal de bornes de recharge à installer par site et la taille de la flotte de véhicules. Le modèle proposé, formulé en utilisant un programme linéaire en nombres entiers, est validé à l'aide d'une étude de cas dans le comté de Penghu. Les résultats obtenus montrent que le modèle proposé est capable d'identifier les paramètres critiques qui ont un impact sur les emplacements des stations de recharge tels que le temps de recharge minimal requis et la taille de la flotte de véhicules. Une analyse de sensibilité suggère que la recharge rapide pourrait réduire considérablement le nombre de stations de recharge requises, ce qui se traduirait par d'importantes économies de coûts.

Ce modèle a été étendu dans [59] où des voyages avec plusieurs points de départ et d'arrivée ont été considérés. Le but des modèles proposés est de déterminer le nombre de stations de changement de batteries et de les placer d'une manière optimale pour couvrir le besoin des utilisateurs de scooters électriques. L'objectif du problème d'optimisation est de minimiser le coût d'installation des stations de recharge de batteries.

Wang et Lin [60] étudient le problème de localisation de stations de recharge, proposant différents types de recharge, afin de charger des véhicules électriques parcourant des plus courts chemins liant des points de départ et des points d'arrivée. Ce problème est une extension du problème de localisation des stations de recharge normale exposé dans [59]. Le but de ce problème est de minimiser le coût de déploiement de ces stations. Le problème est défini par un ensemble de véhicules identiques ayant la même autonomie et un ensemble de stations ayant des capacités infinies. Il est considéré qu'il existe une relation linéaire entre la distance parcourue et la quantité d'énergie consommée.

L'étude proposée par Ip et al. [61] porte sur l'optimisation du déploiement des stations de recharge dans des zones urbaines caractérisées par un trafic dense, des rues étroites et d'autres facteurs complexes tels que la distribution du réseau électrique. Pour résoudre ce problème, un modèle composé de deux étapes est proposé. Ce modèle segmente, dans un premier temps, le trafic routier et la demande associée en un ensemble de groupes de demandes en appliquant un algorithme de regroupement hiérarchique. Ces groupes de demandes représentent les régions dans lesquelles au moins une station de recharge doit être déployée. Le modèle proposé optimise par la suite l'affectation des stations de recharge aux groupes de demandes en utilisant la programmation linéaire. Trois scénarios d'optimisation sont considérés : (1) les stations de recharge ont toutes les mêmes caractéristiques, il s'agit ici de minimiser le coût total d'exploitation en affectant les stations de recharge aux groupes de demandes ; (2) les stations de recharge ont des capacités différentes, dans ce cas, le coût variable de chaque groupe de demande varie d'une station à une autre et (3) des contraintes budgétaires supplémentaires sont considérées et l'objectif est de maximiser la valeur nette totale actualisée de l'investissement.

3.6.3 Optimisation de la recharge

L'autonomie limitée de la batterie du véhicule électrique représente un des obstacles qui empêchent le déploiement à grande échelle du véhicule électrique. Ainsi, optimiser au mieux la recharge des véhicules électriques fait partie des problématiques les plus étudiées dans la littérature. En effet, la recharge non optimisée des véhicules électriques dès qu'ils sont branchés aux bornes de recharge peut avoir un impact considérable sur la performance du réseau électrique [62].

Clement-Nyns et al. [62] proposent d'étudier l'impact de la recharge des véhicules hybrides et électriques sur un réseau résidentiel de distribution. Ces véhicules appartiennent à un ensemble d'utilisateurs qui souhaitent récupérer leurs véhicules entièrement chargés à une heure précise de la journée. Deux scénarios de recharge sont comparés.

Pour le premier scénario, les batteries sont rechargées sans aucune optimisation, i.e., l'opération de recharge commence soit immédiatement lorsque le véhicule est branché soit après un délai d'attente réglable par l'utilisateur.

Pour le deuxième scénario, il s'agit d'optimiser la recharge de l'ensemble des véhicules dans le but de minimiser les pertes de puissance du réseau de distribution lors de la recharge des véhicules. Pour optimiser la recharge, la technique de programmation quadratique a été utilisée. Une deuxième méthode basée sur la programmation dynamique a été également mise en oeuvre, mais cette dernière n'a pas permis d'améliorer le temps de calcul ni la qualité des solutions obtenues par le programme quadratique. Les résultats obtenus, en considérant différents profils de consommation du bâtiment, montrent que l'optimisation de la recharge permet de réduire considérablement les pertes de puissance du réseau de distribution.

Dans ce même contexte, Gan et al. [63] exposent une conception d'un protocole décentralisé pour négocier la planification de la recharge des véhicules électriques. L'objectif global est de combler les creux de demande d'électricité pendant la nuit et d'éviter les pics de consommation au cours de la journée. Pour chaque itération de l'algorithme proposé, les véhicules électriques choisissent leurs profils de recharge pour le jour suivant en se basant sur les prix proposés par le gestionnaire du réseau de distribution d'électricité. La planification de la recharge des véhicules électriques a été présentée sous la forme d'un problème d'optimisation global avec comme objectif la recherche des profils de recharge optimisés. Le scénario considéré modélise une négociation entre le gestionnaire du réseau de distribution d'électricité et un ensemble de véhicules électriques pour établir un plan de recharge durant un intervalle de temps donné. Avant la fin de la période consacrée à la recharge du véhicule, il faut que la batterie soit complètement chargée. Les auteurs proposent un protocole asynchrone et un autre synchrone. À chaque itération des algorithmes développés, chaque véhicule peut choisir son profil de recharge indépendamment des décisions des autres véhicules tout en cherchant à minimiser les coûts compte tenu des prix donnés par le gestionnaire du réseau de distribution d'électricité.

M. Ramezani, et al. [64] utilisent un algorithme évolutionnaire d'optimisation multi-objectifs afin de concevoir un environnement de simulation permettant de générer un ensemble de scénarios en se basant sur certains paramètres qui décrivent une situation d'usage des véhicules électriques et d'optimiser la planification de la recharge des batteries compte tenu de plusieurs contraintes. Les contraintes considérées sont principalement la capacité du réseau de distribution d'électricité et le respect des spécificités des batteries. Concernant la première contrainte, l'hypothèse retenue est que le gestionnaire du réseau de distribution d'électricité fournit, à des intervalles réguliers, la capacité du réseau disponible et le contrôleur de la recharge de la flotte électrique dresse le plan de recharge en se basant sur des prévisions de la capacité du réseau disponible dans le futur. Concernant les batteries, différentes contraintes ont été prises en compte telles que les limites sur l'état de charge, l'intensité maximale du courant tolérée et la possibilité d'effectuer des recharges intermittentes. Plusieurs objectifs ont été considérés pour l'algorithme évolutionnaire. Chaque objectif peut être

pondéré selon les préférences de l'utilisateur et l'algorithme d'optimisation doit fournir la solution qui permet de maximiser la fonction objectif globale. Le premier objectif est de maximiser le profit du fournisseur d'électricité. Le deuxième objectif est de minimiser la dégradation de la batterie. Deux facteurs influençant la durée de vie de la batterie ont été considérés : l'impact d'une recharge discontinue et l'impact de la recharge en elle-même. La dégradation totale de la batterie est définie comme étant la somme des dégradations dues aux deux facteurs cités auparavant. Le dernier objectif est de maximiser la satisfaction des clients. Le client précise le niveau de charge minimal accepté et l'heure de son départ et il peut choisir ses critères de préférence (recharge rapide, le prix le moins élevé, dégradation de batterie minimale). La première étape de l'algorithme d'optimisation est la génération de solutions qui satisfont les contraintes du problème. Une solution décrit un plan de recharge de tous les véhicules électriques et précise le début et la fin de la recharge ainsi que la courbe de recharge utilisée. Ensuite, un algorithme évolutionnaire hybride utilisant une technique de recherche locale permet de générer d'autres planifications pour chaque véhicule jusqu'à ce qu'un optimum soit atteint et aucune autre amélioration ne soit possible. De plus, les mutations et les recombinaisons de plans de recharge permettent de générer de nouvelles solutions qui seront par la suite optimisées localement.

Lee et al. [65] conçoivent un planificateur capable de générer un ordonnancement de tâches consommant de l'énergie électrique (telles que l'utilisation de la machine à laver ou le chargement d'une batterie d'un véhicule électrique) et ceci afin de réduire les pics de la courbe de charge d'électricité. La conception de ce planificateur est basée sur les algorithmes génétiques. Chaque étape de l'algorithme génère une population de solutions et l'évalue selon une fonction objectif afin de sélectionner la meilleure solution et recombinaison l'ancienne population pour construire une nouvelle population de solutions.

Clement et al. [66] proposent un ordonnanceur centralisé qui permet de gérer d'une manière optimale la capacité du réseau afin de recharger les véhicules hybrides et ceci en se basant sur les informations transmises par les utilisateurs des véhicules et qui concernent leur utilisation prévue des véhicules dans le futur. Ces informations sont considérées fiables. Le but de cet ordonnanceur est de coordonner les recharges des véhicules afin de minimiser les pertes d'énergie et d'optimiser l'utilisation du réseau. Les contraintes du problème sont, d'une part, le respect de la puissance maximale du réseau qu'il ne faut pas dépasser. D'autre part, les batteries doivent être complètement rechargées à la fin du cycle. Néanmoins, cette approche ne considère pas le fait que les utilisateurs aient parfois intérêt à fournir des informations erronées. Pour cela, dans [67], les auteurs utilisent un mécanisme d'incitation qui permet de prendre en compte les situations dans lesquelles les utilisateurs mentent sur leurs intentions d'utilisation des véhicules et de générer des mécanismes de paiement qui incitent les utilisateurs à transmettre les informations d'une manière fiable. Ce mécanisme considère des vitesses de recharge hétérogènes et flexibles pour générer des plans de recharge des véhicules.

Le problème d'optimisation du modèle de recharge des véhicules hybrides, défini par une plage horaire et un type de recharge (rapide, lent, etc.), est examiné dans [68]. L'objectif de l'optimisation est de minimiser le coût total de l'électricité et du carburant ainsi que la dégradation de la batterie.

Dans [69], les auteurs présentent des stratégies de contrôle d'énergie fondées sur les principes de l'optimisation quadratique pour recharger des véhicules hybrides afin de réduire les pics de la courbe de charge et raffiner la forme générale de cette courbe. Deux stratégies ont été considérées. La première est une stratégie de contrôle d'énergie qui consiste à suivre les processus de recharge de chaque véhicule indépendamment des autres. La deuxième est une stratégie globale qui contrôle et coordonne les processus de recharge d'un ensemble de véhicules. Chacune de ces stratégies définit un ordonnancement des opérations de recharge en indiquant à chaque fois la plage horaire et le type de charge pour chaque véhicule.

3.6.4 Optimisation des tournées de véhicules électriques

Les problèmes de transport avec des véhicules électriques peuvent être divisés en deux sous-problèmes : (i) le problème de calcul d'itinéraire où l'objectif est de déterminer le meilleur chemin pour un véhicule électrique allant d'une origine à une destination tout en respectant la consommation énergétique, le coût et le temps, et (ii) le problème de conception de tournées où l'objectif est de construire un ensemble de tournées pour servir un ensemble de clients en utilisant des véhicules électriques.

Concernant le problème de construction du meilleur chemin, Artmeier et al. [70] exposent des extensions des algorithmes standards de calcul du plus court chemin pour aborder le problème de recherche du chemin énergétiquement optimal qui pourrait être effectué par un véhicule électrique. Ce problème prend en compte la capacité maximale des batteries des véhicules électriques ainsi que les quantités d'énergie qui peuvent être récupérées par le véhicule électrique lors d'un freinage ou d'une descente d'une pente. Étant donné un point de départ et l'état de charge de la batterie, il s'agit de déterminer un chemin optimal vers une destination préalablement fixée tout en cherchant à maximiser la quantité d'énergie restante dans la batterie à la fin du trajet. L'état de charge du véhicule ne doit jamais être négatif et ne peut pas dépasser la capacité maximale de la batterie. La modélisation du problème est représentée sous la forme d'un graphe orienté dont les noeuds correspondent à des localisations et les arcs correspondent aux segments de routes. À chaque arc du graphe est associée une valeur qui peut être soit positive soit négative. Une valeur positive (respectivement négative) correspond à la quantité d'énergie consommée (respectivement récupérée) par le véhicule quand il parcourt un trajet donné. Les auteurs modélisent le problème étudié comme étant un cas particulier du problème de plus court chemin avec contraintes [71] et proposent un algorithme de résolution ayant une complexité en $O(n^3)$.

Le problème du plus court chemin énergétique est aussi étudié dans [72] et [73]. Dans [73], il est montré que les contraintes liées à la capacité de la batterie peuvent être traduites par des fonctions de coût sur les arcs du graphe considéré en suivant la règle FIFO. Ainsi, l'algorithme de Bellman-Ford peut être utilisé pour résoudre ce problème en $O(nm)$. Une deuxième méthode de résolution consiste à utiliser, dans un premier temps, l'algorithme de Johnson pour calculer une transformation du graphe de départ en un graphe sans poids négatifs. L'algorithme de Dijkstra est alors appliqué, dans un second temps, sur le graphe transformé. L'algorithme développé a une complexité en $O(n \log + m)$ après une phase de prétraitement en $O(nm)$ pour n'importe quel graphe dont les arcs présentent les quantités d'énergie consommées et récupérées par le véhicule électrique.

L'étude présentée dans [72] montre qu'il est possible de calculer les coûts associés aux arcs d'une façon dynamique et il est ainsi possible d'éviter la phase de prétraitement et réduire le temps de calcul de l'algorithme. Un algorithme ayant une complexité en $O(n^2)$ est proposé.

Aucune de ces trois dernières études ne considère la possibilité de charger le véhicule sur certains noeuds du graphe pendant le trajet. Ceci représente la contribution principale présentée dans [74]. Dans cette dernière étude, Sweda et al. proposent de déterminer le chemin ayant le moindre coût pour un véhicule électrique et incluent la possibilité de charger la batterie du véhicule électrique durant le trajet. L'objectif est de déterminer un trajet reliant un point de départ et une destination tel que l'état de charge de la batterie reste toujours entre une valeur minimale et une valeur maximale de l'état de charge. Le problème est modélisé comme un programme dynamique et deux méthodes de résolution sont présentées. La première méthode est une récursion arrière qui trouve la solution optimale quand l'espace d'états est discret, ce qui est le cas sous certaines hypothèses. La deuxième méthode est un algorithme d'approximation basé sur la programmation dynamique.

Dans le cas du problème de construction de tournées, nous évoquons souvent le "*Electric Vehicle Routing Problem*" qui est une extension du problème classique "*Vehicle Routing Problem*" ([75] et

[76]).

Conrad et al. [77] étaient les premiers à évoquer le "*Recharging Vehicle Routing Problem*". Pour ce problème, des véhicules ayant des autonomies limitées peuvent être chargés chez certains clients pendant leurs tournées. Il est considéré que les clients doivent être servis pendant des fenêtres de temps prédéfinies et que le temps de recharge pour chaque véhicule est fixe. Les auteurs présentent une analyse de sensibilité pour comprendre l'impact de l'autonomie du véhicule, du temps de la recharge et de la considération des fenêtres de temps sur les solutions obtenues.

Dans [78], les auteurs introduisent le "*Green Vehicle Routing Problem (GVRP)*" pour lequel les tournées des véhicules et les planning de recharge sont déterminés simultanément. Ce problème est défini par un ensemble de véhicules électriques identiques, un ensemble de clients à servir et un ensemble de stations de recharge. Il est considéré que les véhicules ont une capacité de transport illimitée et que le temps de recharge est constant. Il s'agit de construire un ensemble de tournées, qui commencent et se terminent au dépôt, tel que tous les clients soient servis et toutes les contraintes soient satisfaites. Pour résoudre ce problème, les auteurs proposent un programme linéaire en nombres mixtes (MIP) dont l'objectif est de minimiser la distance parcourue tout en considérant les contraintes sur l'autonomie des véhicules et sur le nombre et la durée des tournées. Les auteurs proposent deux heuristiques constructives pour résoudre ce problème. La première heuristique est une adaptation de la méthode de *Clarke and Wright* et la deuxième est un *density-based clustering algorithm*. Une revue du GVRP est donnée dans [79].

Schneider et al. [80] introduisent le problème de conception de tournées de véhicules électriques avec des fenêtres de temps et des stations de recharge (*Electric Vehicle Routing Problem with Time Windows and Recharging Stations*), noté E-VRPTW. Ils intègrent des contraintes supplémentaires par rapport à ce qui a été introduit dans [78], à savoir des contraintes liées aux demandes des clients et aux capacités limitées des véhicules. De plus, chaque client doit être visité pendant une fenêtre de temps prédéfinie. Pour ce problème, les temps de recharge dépendent de l'état de charge des batteries à leur arrivée aux stations de recharge. À chaque fois qu'un véhicule électrique visite une station de recharge, sa batterie doit être chargée à 100%. L'objectif du problème E-VRPTW est de minimiser le nombre de véhicules utilisés et de minimiser la distance totale parcourue par les véhicules. Pour résoudre ce problème, une méta-heuristique hybride combinant une VNS et une recherche tabou est développée. Des solutions infaisables peuvent être acceptées durant la recherche de la meilleure solution.

Dans [81], les auteurs introduisent le problème de tournées de véhicules électriques avec des fenêtres de temps et une flotte mixte (*Electric Vehicle Routing Problem with Time Windows and Mixed Fleet*) pour optimiser les tournées d'une flotte mixte composée de véhicules électriques et de véhicules thermiques. Pour résoudre ce problème, une recherche locale à grand voisinage adaptatif (*Adaptative Large Neighborhood Search*), renforcée par une recherche locale pour intensification, est proposée.

Un problème similaire est introduit dans [82]. La seule différence ici est de considérer une flotte hétérogène de véhicules qui ont des capacités de transport, des capacités de batteries et des coûts d'acquisition différents. Une recherche locale à grand voisinage adaptatif qui incorpore une recherche locale et une procédure d'étiquetage pour intensification est développée.

Dans [83], les auteurs présentent une variante plus réaliste du problème de tournées de véhicules électriques dans laquelle deux nouvelles contraintes sont considérées : différentes technologies de recharge et la possibilité d'une recharge partielle des véhicules. Des heuristiques constructives et des recherches locales sont proposées et utilisées dans le cadre d'un algorithme de recuit simulé non déterministe.

3.7 Conclusion

Dans ce chapitre, nous avons exposé quelques notions liées à la théorie de complexité et aux algorithmes d'approximation. Nous avons, par la suite, présenté les principales méthodes de résolution exactes et approchées utilisées dans la littérature. Ensuite, nous avons proposé une revue de littérature des principales thématiques liées à l'écosystème du véhicule électrique, à savoir les problèmes de Smart Grid et de V2G, les problèmes de dimensionnement des infrastructures de recharge, les problèmes d'optimisation de la recharge et les problèmes de transport des véhicules électriques.

Chapitre 4

Optimisation conjointe de l'affectation et de la recharge des véhicules électriques

Sommaire

4.1 Introduction	36
4.2 Description du problème et notation	37
4.3 Étude de complexité	38
4.4 Formulation mathématique	39
4.5 Approches de résolution	42
4.5.1 Méthode exacte	42
4.5.2 Heuristique Séquentielle	42
4.5.3 Heuristique globale	51
4.6 Résultats expérimentaux et discussion	52
4.7 Conclusion	58

Ce chapitre décrit notre problème de base qui concerne l'optimisation conjointe de l'affectation et de la recharge des véhicules électriques. Il expose une étude de complexité et les méthodes de résolution développées.

4.1 Introduction

Les véhicules décarbonés constituent un changement de modèle pour l'industrie et une des opportunités de croissance des prochaines années. La clé de réussite de cette innovation dépend principalement de son équation économique et de sa compatibilité avec l'usage souhaité.

Dans le cadre de la mise en place progressive des véhicules électriques, le projet Infini Drive propose l'étude de la faisabilité et des conséquences d'un déploiement d'une flotte de véhicules électriques et la conception et l'expérimentation d'un système de recharge pour les exploitants des flottes captives.

Après avoir présenté une étude bibliographique des principaux problèmes liés à l'écosystème du véhicule électrique, dans ce chapitre, nous nous concentrons sur l'étude et la résolution de notre problème de base qui est l'optimisation conjointe de l'affectation et de la recharge des véhicules électriques, ou encore *Electric Vehicle Scheduling and Charging Problem*, noté EVSCP. Notre objectif est de fournir des outils d'aide à la décision génériques pour une gestion opérationnelle efficace d'une flotte de véhicules électriques et thermiques. Il s'agit alors de prendre en compte les différentes

contraintes d'ordre opérationnel et technique liées à l'usage des véhicules électriques pour répondre aux besoins d'un gestionnaire de flottes captives.

Ce problème n'a pas été étudié dans la littérature. En effet, toutes les études antérieures se concentrent soit sur l'optimisation de l'allocation des véhicules soit sur l'optimisation de la recharge. Aucune autre étude ne propose de résoudre ces deux problèmes simultanément. L'avantage de résoudre le problème d'affectation et de recharge des véhicules électriques d'une façon conjointe est, d'une part, de s'assurer de la faisabilité de la solution d'affectation en termes d'énergie et de capacité du réseau électrique et, d'autre part, de fournir des solutions globales de meilleure qualité (l'optimum global n'est pas toujours égal à la somme des optima locaux).

Dans ce qui suit, nous décrivons, tout d'abord, notre problème en détail et nous étudions sa complexité. Nous proposons par la suite un modèle linéaire en nombres mixtes. Ensuite, nous présentons nos approches de résolution, à savoir une méthode exacte et deux heuristiques. Enfin, nous comparons les performances des différentes méthodes de résolution sur des données réelles et des instances générées aléatoirement.

4.2 Description du problème et notation

Le problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques (EVSCP) est défini par une flotte de véhicules électriques et thermiques, un ensemble de tournées à effectuer et une infrastructure de recharge, l'objectif est d'établir une affectation optimale des tournées aux véhicules et, en même temps, un planning de recharge des véhicules électriques tout en vérifiant un certain nombre de contraintes techniques et opérationnelles.

Nous considérons un ensemble $M_1 = \{1, \dots, m_1\}$ de Véhicules Électriques (VEs) et un ensemble $M_2 = \{1, \dots, m_2\}$ de Véhicules Thermiques (VTs) disponibles pour effectuer un ensemble $N = \{1, \dots, n\}$ de tournées. Le coût fixe du véhicule j est noté CF_j , son coût variable, hors coût de recharge pour les véhicules électriques, est noté CV_j . Chaque véhicule j a une capacité d'emport maximale égale à Q_j . Chaque véhicule peut effectuer au plus une tournée à la fois. Chaque VE j fonctionne avec une batterie caractérisée par une capacité CE_j (kWh) et un état de charge (SoC) à l'instant $t = 0$, noté SoC_j et calculé comme le quotient de la quantité d'énergie disponible dans la batterie du véhicule à l'instant $t = 0$ et la capacité maximale de la batterie CE_j ($0 =$ vide ; $1 =$ pleine). Par ailleurs, à des valeurs élevées ou basses du SoC, la batterie a tendance à se dégrader plus rapidement [68]. Afin d'améliorer sa durée de vie et respecter les conditions de sécurité, SoC_j devrait être dans l'intervalle $[SoC_j^{min}, SoC_j^{max}]$, où SoC_j^{min} et SoC_j^{max} sont les valeurs minimales et maximales acceptables du SoC, respectivement.

Pour charger les VEs, nous considérons qu'au plus m_1 chargeurs sont disponibles. Ces chargeurs peuvent être de K_{max} types différents (recharge lente, recharge accélérée, etc.). Le coût de déploiement d'un chargeur de type k est noté DC_k .

L'horizon d'optimisation $[0, T]$ est divisé en H périodes de temps équidistantes $t = 1, \dots, H$, chacune de durée δ , où t représente l'intervalle de temps $[t - 1, t]$. À chaque période de temps t , chaque chargeur peut fournir une puissance de recharge comprise dans l'intervalle $[p_k^{min}, p_k^{max}]$, où p_k^{min} et p_k^{max} sont les puissances maximales et minimales que peut fournir ce chargeur. Ainsi, si un véhicule électrique j est chargé pendant l'intervalle de temps t avec une puissance p_k^{max} , il récupère une quantité d'énergie égale à $\delta \times p_k^{max}$ (kWh).

Notons CP la puissance physique du bâtiment et SP sa puissance souscrite. La puissance du réseau électrique disponible à chaque intervalle de temps t est notée g_t . Le coût d'électricité à l'instant t est noté c_t .

Un ensemble de n tournées doit être effectué par les véhicules disponibles. Chaque tournée i est caractérisée par une heure de début s_i , une heure de fin f_i , une demande q_i (quantité de biens à livrer pendant cette tournée), un poids w_i (longueur exprimée en kilomètre) ainsi qu'un besoin en énergie E_i .

Une solution faisable à ce problème est composée d'une affectation de toutes les tournées aux véhicules disponibles et d'un planning de recharge des VE's tel que :

- toutes les tournées sont affectées
- aucun véhicule n'est affecté à des tournées qui se recouvrent
- aucun véhicule n'est chargé pendant les horaires des tournées qui lui ont été affectées
- aucun véhicule n'est affecté à une tournée dont la demande dépasse sa capacité maximale d'emport
- la puissance maximale du réseau disponible n'est jamais dépassée
- les puissances maximales et minimales des chargeurs sont respectées pendant la phase de recharge des VE's
- l'état de charge des véhicules ne dépasse jamais le SOC maximal autorisé
- une énergie suffisante est disponible dans la batterie de chaque VE avant le début de chaque tournée

4.3 Étude de complexité

Dans cette partie, nous prouvons que le problème EVSCP est NP-difficile au sens ordinaire. Pour cela, nous utilisons la réduction au problème de partition qui est connu d'être NP-Complet.

Le problème de partition peut être décrit comme suit : Étant donné un ensemble $P = \{a_1, a_2 \dots a_p\}$ de p entiers tel que $\sum_{i=1}^p a_i = 2A$, existe-t-il une partition de P en deux sous-ensembles P_1 et P_2 tel que $\sum_{i \in P_1} a_i = \sum_{i \in P_2} a_i = A$?

Théorème 4.1. *Le problème EVSCP est NP-difficile au sens ordinaire.*

Preuve : Étant donnée une instance aléatoire du problème de partition, nous construisons une instance (\mathcal{S}) du problème EVSCP avec un ensemble de $2p + 2$ tournées. Les instants de début, de fin et les besoins en énergie sont donnés dans le tableau 6.2. Nous supposons que $a_0 = 0$.

TAB. 4.1: Caractéristiques de l'instance (\mathcal{S})

Tournée	Début	Fin	Besoin en énergie
$T_i; i = 1, \dots, 2p$	$A + \sum_0^{i-1} a_k$	$A + \sum_0^i a_k$	a_i
T_{2p+1}	$3A$	$3A + 1$	A
T_{2p+2}	$3A$	$3A + 1$	A

Sans perte de généralité, nous limitons notre analyse au cas de deux véhicules électriques disponibles VE_1 et VE_2 ayant chacun une capacité de batterie de $2A$ (kWh). Nous supposons que deux chargeurs existent. Chaque chargeur peut fournir une puissance de recharge discrète égale à 1 (kW). La capacité du réseau disponible pour la recharge des VE's est limitée à 2 (kW) à chaque intervalle de temps t . À $t = 0$, les batteries des deux VE's sont vides. Une allocation faisable des véhicules aux tournées doit satisfaire les contraintes suivantes :

- Les VE doivent avoir un état de charge suffisant avant le début de chaque tournée.
- La capacité maximale de la batterie ne doit jamais être dépassée pendant la phase de recharge du VE.
- À chaque intervalle de temps t , la puissance totale utilisée pour charger les VE ne doit jamais dépasser la capacité maximale disponible du réseau électrique.

Dans ce qui suit, nous montrons qu'il existe une solution faisable au problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques si, et seulement si, le problème de partition admet une solution.

Tout d'abord, nous supposons que le problème de partition admette une solution et notons P_1 et P_2 les deux sous-ensembles de P . L'ordonnancement souhaité des tournées est obtenu en construisant deux ensembles L_1 et L_2 de tournées tel que les tournées dans L_1 et L_2 correspondent aux ensembles P_1 and P_2 , respectivement ; i.e., $T_i \in L_j$, si $a_i \in P_j$, $i = 1, \dots, 2p$, $j = 1, 2$. Affectons les tournées de L_1 au VE₁ et les tournées de L_2 au VE₂. Les tournées T_{2p+1} et T_{2p+2} sont affectées au VE₁ et VE₂, respectivement. Afin d'effectuer les tournées, les VE sont chargés comme suit : Pendant l'intervalle de temps $[0, A]$, chaque VE est chargé avec une puissance de 1 (kW). À $t = A$, chaque VE récupère alors A (kWh) quantité d'énergie. Pendant l'intervalle de temps $[A, 3A + 1]$, VE₁ effectue toutes les tournées de L_1 pour A unités de temps, et il est chargé quand il n'est pas en tournée (VE₁ est chargé pendant les intervalles de temps $[A, 3A] \cap_{a_i \in P_2} [A + a_{i-1}, A + a_{i-1} + a_i]$), et finalement, il effectue la tournée T_{2p+1} . Le VE₂ effectue les tournées de L_2 et il est chargé pendant l'intervalle $[A, 3A] \cap_{a_i \in P_1} [A + a_{i-1}, A + a_{i-1} + a_i]$ avant d'effectuer la tournée T_{2p+2} . Pendant les différentes phases de recharge, chaque véhicule récupère alors suffisamment d'énergie pour effectuer les tournées qui lui ont été affectées. Le planning de tournées et de recharge obtenu est alors faisable.

Inversement, supposons maintenant qu'il existe une solution au problème d'ordonnancement de tâches sur deux machines. Étant donné que les tournées T_{2p+1} et T_{2p+2} ont les mêmes instants de début et de fin, nous supposons, sans perte de généralité, que T_{2p+1} et T_{2p+2} soient affectées à VE₁ et VE₂, respectivement. Notons R_1 et R_2 les deux ensembles de tournées affectés à VE₁ et VE₂, respectivement. Les sommes $\sum_{a_i \in R_1} a_i = A_1$ et $\sum_{a_i \in R_2} a_i = A_2$ représentent les quantités d'énergie nécessaires pour effectuer les ensembles de tournées R_1 et R_2 . Les véhicules VE₁ et VE₂ ont alors besoin d'une quantité d'énergie totale égale à $A + A_1$ et $A + A_2$, respectivement. Supposons que $A_1 > A$, le véhicule VE₁ a alors besoin d'une quantité d'énergie supérieure à $2A$. De plus, VE₁ peut seulement être chargé pendant l'intervalle de temps $[0, 3A] \cap_{a_i \in R_2} [A + a_{i-1}, A + a_{i-1} + a_i]$ et il peut récupérer au plus une quantité d'énergie totale $E = 3A - A_1 < 2A$ (kWh). Étant donné que $E < 2A$, VE₁ ne peut pas récupérer suffisamment d'énergie pour effectuer les tournées de l'ensemble R_1 en plus de la tournée T_{2p+1} , d'où $A_1 \leq A$. En suivant le même raisonnement, nous obtenons $A_2 \leq A$. Étant donné que $A_1 + A_2 = 2A$, nous concluons que $A_1 = A$ et $A_2 = A$. Par conséquent, R_1 et R_2 représentent une solution au problème de partition.

4.4 Formulation mathématique

Dans cette section, nous proposons un modèle linéaire en nombres mixtes pour notre problème. Avant de présenter le modèle, nous résumons, dans le tableau 4.2, la nomenclature du problème.

Sauf indication contraire, nous considérons, dans ce chapitre, qu'exactement m_1 chargeurs sont disponibles et qu'ils sont tous du même type. Les variables p_k^{\min} et p_k^{\max} seront alors notées p^{\min} et p^{\max} , respectivement.

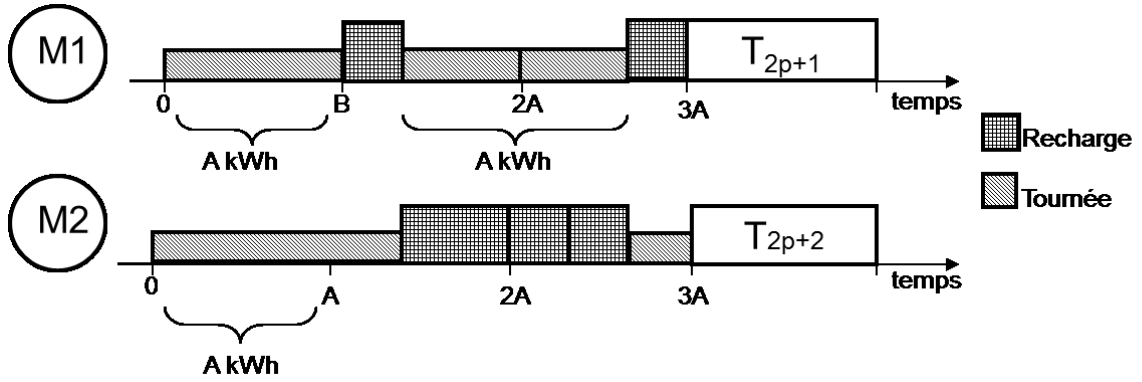


FIG. 4.1: Solution au problème d'ordonnancement sur deux véhicules

Pour notre modèle mathématique, nous introduisons les variables de décision suivantes :

x_{ij} : variable binaire égale à 1 si le véhicule j , $j = 1, \dots, m_1 + m_2$, est affecté à la tournée i , $i = 1, \dots, n$, et 0 sinon.

y_{jt} : variable binaire égale à 1 si VE j , $j = 1, \dots, m_1$, est chargé pendant l'intervalle de temps $t = 1, \dots, H$, et 0 sinon.

p_{jt} : variable réelle qui désigne la puissance de recharge appliquée au véhicule VE j , $j = 1, \dots, m_1$ pendant la période de temps $t = 1, \dots, H$.

Afin de réduire la complexité du modèle proposé et sans perte de généralité, nous supposons que la quantité d'énergie E_i nécessaire pour effectuer la tournée i soit consommée pendant la dernière période de temps f_i de cette tournée ($E_{i,f_i} = E_i$ et $E_{i,l} = 0$, $l = s_i, s_i + 1, \dots, f_i - 1$).

La formulation mathématique (\mathcal{P}) de notre problème est alors comme suit :

$$\text{Lex} \quad \left(\max \sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij} ; \min \sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt} \right) \quad (4.1)$$

$$\sum_{j=1}^{m_1+m_2} x_{ij} = 1, \quad \forall i \quad (4.2)$$

$$x_{ij} + \sum_{i' \in V(i)} x_{i'j} \leq 1, \quad \forall i, \forall j \quad (4.3)$$

$$\sum_{i=1}^{i=n} q_i \times x_{ij} \leq Q_j, \quad \forall j \quad (4.4)$$

$$\sum_{t=s_i}^{f_i} y_{jt} + (f_i - s_i + 1) \times x_{ij} \leq (f_i - s_i + 1), \quad \forall i, \forall j \in M_1 \quad (4.5)$$

$$\sum_{j=1}^{m_1} p_{jt} \leq g_t, \quad \forall t \quad (4.6)$$

$$p^{\min} \times y_{jt} \leq p_{jt}, \quad \forall t, \forall j \in M_1 \quad (4.7)$$

TAB. 4.2: Définition des paramètres de EVSCP

$[0, T]$	horizon d'optimisation
$t = 1, \dots, H$	discrétisation de l'intervalle de temps T
$M_1 = \{1, \dots, m_1\}$	ensemble de m_1 véhicules électriques
$M_2 = \{1, \dots, m_2\}$	ensemble de m_2 véhicules thermiques
CF_j	coût fixe du véhicule j
CV_j	coût variable du véhicule j
Q_j	capacité d'emport maximale du véhicule j
CE_j	capacité de batterie du véhicule électrique j
SoC_j^0, SoC_j^t	état de charge du véhicule électrique j à l'instant $t = 0$, et à l'instant $t > 0$, respectivement
$SoC_j^{\min}, SoC_j^{\max}$	valeurs minimales et maximales acceptables du SoC
K_{max}	nombre maximal de types de bornes de recharge disponibles
DC_k	coût de déploiement d'un chargeur de type k
p_k^{\min}, p_k^{\max}	puissance minimale et maximale que peut fournir un chargeur
CP	puissance physique du bâtiment
SP	puissance souscrite du bâtiment
g_t	puissance du réseau disponible pour la recharge des VEs pendant l'intervalle de temps t
c_t	coût de l'énergie pendant l'intervalle de temps t
n	nombre de tournées
s_i, f_i	heure de début et heure de fin de la tournée i
q_i	quantité de biens à livrer pendant la tournée i
w_i	longueur de la tournée i (km)
E_i	besoin en énergie de la tournée i (kWh)

$$p_{jt} \leq p^{\max} \times y_{jt}, \quad \forall t, \forall j \in M_1 \quad (4.8)$$

$$SoC_j^0 + \frac{\sum_{t \leq s_i-1} \delta \times p_{jt} - \sum_{l=1/f_i \leq s_i-1} E_{l,f_i} \times x_{lj}}{CE_j} \leq SoC_j^{\max}, \quad \forall i, \forall j \in M_1 \quad (4.9)$$

$$SoC_j^0 + \frac{\sum_{t \leq f_i} \delta \times p_{jt} - \sum_{l=1/f_i \leq f_i} E_{l,f_i} \times x_{lj}}{CE_j} \geq SoC_j^{\min}, \quad \forall i, \forall j \in M_1 \quad (4.10)$$

Les contraintes (4.2) assurent que chaque tournée est affectée à exactement un véhicule. Les contraintes (4.3) garantissent qu'aucun véhicule n'est affecté à des tournées qui se recouvrent. Les contraintes (4.4) assurent qu'aucun véhicule n'est affecté à une tournée dont le volume de biens à livrer dépasse la capacité d'emport maximale du véhicule. Les contraintes (4.5) interdisent la recharge du VE pendant la tournée. Les contraintes (4.6) assurent qu'à chaque intervalle de temps t , la puissance totale utilisée pour charger les VEs ne dépasse pas la capacité maximale disponible du réseau électrique. Les contraintes (4.7) et (4.8) assurent que la puissance de recharge des VEs est comprise entre p^{\min} et p^{\max} . Les contraintes (4.9) et (4.10) garantissent que l'état de charge de chaque VE est dans l'intervalle $[SoC^{\min}, SoC^{\max}]$ pendant tout l'horizon de planification $[0, T]$.

Notre objectif est d'optimiser deux fonctions lexicographiques [84, 85]; c'est à dire, les fonctions objectif sont classées dans l'ordre de leur importance et les problèmes d'optimisation sont résolus un à la fois. Plus précisément, si z^* est la valeur optimale de la fonction objectif principale, alors l'optimisation de la fonction objectif secondaire est soumise à ce que la fonction objectif principale soit délimitée par z^* .

Pour notre problème, le premier objectif consiste à maximiser la distance totale parcourue par les VEs ($\sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij}$). Le deuxième objectif est de minimiser le coût total de recharge des VEs

$$(\sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt}).$$

Notre choix d'opter pour l'optimisation lexicographique s'explique par le fait que, pour les applications réelles d'optimisation des usages des véhicules électriques, une grande part du gain est obtenue grâce à l'optimisation de l'affectation des tournées. En effet, ce gain est environ 1000 euros/jour pour une flotte de 10 VEs, toutefois, le gain obtenu grâce à l'optimisation de la recharge est d'environ 30 euros/jour.

Nous pouvons également penser à optimiser le coût total d'exploitation de la flotte plutôt que maximiser la distance parcourue par les VEs et minimiser les coûts de recharge. Cependant, étant donné que l'introduction des véhicules électriques dans les flottes d'entreprises est récente, les données disponibles ne sont pas suffisantes pour évaluer correctement les coûts opérationnels des VEs. En outre, les véhicules électriques sont loués et doivent être vendus après 4 ans d'exploitation avec un objectif de 64000 kilomètres après 4 ans. L'objectif des gestionnaires de flotte est alors d'atteindre le total de 64000 km en maximisant l'utilisation des véhicules électriques.

4.5 Approches de résolution

Dans cette partie, nous exposons les méthodes de résolution développées. Nous commençons par décrire notre algorithme de résolution exacte avant d'expliquer nos deux heuristiques.

4.5.1 Méthode exacte

Notre méthode de résolution exacte consiste à résoudre le modèle mathématique proposé en deux étapes en utilisant CPLEX. Dans la première étape, l'objectif est de maximiser la distance totale parcourue par les VEs ($\sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij}$). La solution optimale générée par cette étape joue le rôle d'une solution initiale dans la deuxième partie. Une nouvelle contrainte qui assure que le nombre de kilomètres parcourus par les VEs est plus grand que la valeur de la fonction objectif trouvée dans la première étape, est ajoutée au modèle linéaire dans la deuxième étape. Ensuite, le nouveau modèle est résolu avec la deuxième fonction objectif ($\sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt}$). Ces deux étapes sont décrites dans l'Algorithme 1.

Algorithme 1 Méthode exacte

- 1: **Donnés** : Instance du problème EVSCP
 - 2: **Sortie** : Une solution au problème EVSCP
 - 3: **Étape 1** : Résoudre le MIP avec comme objectif $K = \sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij}$
 - 4: Soit K^* la valeur de la fonction objectif de la solution optimale S^*
 - 5: **Étape 2** : Ajouter la contrainte $\sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij} \geq K^*$ au MIP
 - 6: Résoudre le nouveau MIP avec la fonction objectif $\sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt}$ en partant de la solution S^*
-

4.5.2 Heuristique Séquentielle

Dans cette partie, nous décrivons une heuristique séquentielle - SH pour résoudre le problème EVSCP. Cette heuristique est principalement basée sur l'idée suivante : Pour chaque VE, un ensemble de tournées est sélectionné et affecté à ce VE. Ensuite, un planning de recharge est établi pour ce VE tout en respectant toutes les contraintes du problème. Cette heuristique entrelace deux

étapes : l'étape de sélection des tournées et l'étape de planification de la recharge. Pour les véhicules thermiques, il s'agit d'appliquer seulement l'algorithme de sélection des tournées sur les tournées qui n'ont pas pu être affectées à des véhicules électriques.

L'algorithme 3 donne une vue d'ensemble de notre heuristique séquentielle.

Algorithme 2 Vue d'ensemble de l'heuristique séquentielle

- 1: **Données** : Données du problème EVSCP
 - 2: **Sortie** : Une solution au problème EVSCP
 - 3: **pour** chaque VE **faire**
 - 4: Appliquer *l'algorithme de sélection des tournées* pour générer un ensemble de tournées réalisables par ce véhicule
 - 5: Étant donné l'ensemble des tournées sélectionnées, appliquer *l'algorithme d'optimisation de la recharge* pour minimiser le coût de recharge de ce véhicule
 - 6: **fin pour**
 - 7: **pour** chaque VT **faire**
 - 8: Appliquer *l'algorithme de sélection des tournées*
 - 9: **fin pour**
-

a. Algorithme de sélection des tournées

Pour affecter un ensemble de tournées à un VE, nous avons choisi d'utiliser le problème de clique à poids maximum (MWCP) [86, 87].

Le problème de clique à poids maximum est défini comme suit : Soit $\mathcal{G} = (\mathcal{V}; \mathcal{E})$ un graphe non orienté et pondéré, où \mathcal{V} est l'ensemble des sommets et \mathcal{E} est l'ensemble des arêtes. À chaque arête i est associé un poids w_i . Deux sommets distincts sont dits adjacents s'ils sont connectés par une arête. Étant donné un ensemble de sommets \mathcal{S} , le poids de \mathcal{S} sera noté $W(\mathcal{S}) = \sum_{i \in \mathcal{S}} w_i$.

Une clique du graphe \mathcal{G} est un sous-ensemble de \mathcal{V} dans lequel tous les sommets sont deux à deux adjacents. Une clique S est dite maximale si son cardinal est le plus grand possible (c'est-à-dire qu'elle possède le plus grand nombre de sommets). Une clique à poids maximum \mathcal{S} est une clique qui n'est incluse dans aucune autre clique ayant un poids supérieur à $W(\mathcal{S})$.

Le problème de sélection de tournées est représenté à travers un graphe pondéré indirect $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, où chaque sommet $i \in \mathcal{V}$ représente une tournée caractérisée par un poids (la longueur de la tournée en kilomètre). Nous supposons que les sommets soient triés dans l'ordre croissant de leurs heures de début s_i . Deux sommets i et j sont connectés par une arête si les tournées i et j sont disjointes. Le problème de sélection de tournées est alors équivalent au MWCP sur le graphe \mathcal{G} avec des contraintes supplémentaires.

Le MWCP est NP-difficile [88]. Balas and Yu [89] présentent des cas particuliers qui peuvent être résolus en temps polynomial sur des classes de graphes particuliers. Le graphe d'intervalle est l'un des graphes spécifiques pour lesquels le MWCP devient un problème polynomial. Par contre, dans notre cas, même si le graphe \mathcal{G} est un graphe d'intervalle, le MWCP sur le graphe \mathcal{G} doit s'assurer qu'une solution faisable pour la recharge des VEs existe.

Un exemple de construction de graphe est donné dans la Figure 4.2 où 4 tournées A, B, C, et D, dont les caractéristiques sont données dans le tableau 7.3, sont présentées. Le graphe correspondant

à cette instance est composé de 4 sommets et de 4 arêtes. Le nombre de kilomètres de chaque tournée représente le poids associé au sommet correspondant.

TAB. 4.3: Caractéristiques des tournées

Tournée	Début	Fin	Kilométrage
A	5	9	10
B	10	15	9
C	5	8	7
D	12	15	5

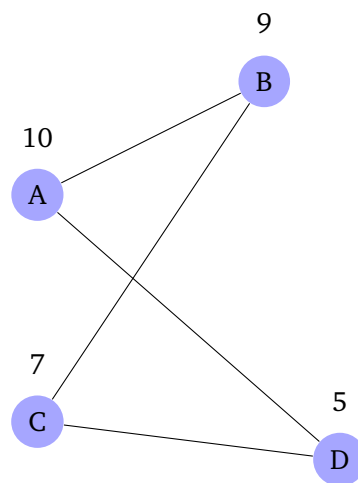


FIG. 4.2: Exemple de construction de graphe

Pour résoudre le problème de sélection de tournées, nous proposons une heuristique qui détermine la clique de poids maximum dans le graphe \mathcal{G} et s'assure qu'un planning faisable de recharge existe.

i- Fonctionnement de l'algorithme

Notre algorithme est composé de deux étapes. Dans la première étape, une clique \mathcal{C} est construite en utilisant un algorithme glouton. Dans la deuxième étape, le poids de \mathcal{C} est amélioré en retirant ou en ajoutant un ou plusieurs sommets. Les deux étapes sont répétées jusqu'à ce que toutes les conditions soient remplies. La clique ayant le meilleur poids est mise à jour à chaque itération. Les détails des deux étapes de l'algorithme sont comme suit :

- **Étape initiale.** Initialement, la clique \mathcal{C} est vide. L'algorithme sélectionne aléatoirement un sommet i et l'ajoute à \mathcal{C} . Pour élargir cette clique, une liste de sommets candidats $L = \{v \in \mathcal{V} \mid v \text{ est connecté à tous les noeuds de } \mathcal{C} \text{ et } v \notin \mathcal{C}\}$ est créé. L'algorithme choisit ensuite un sommet v^* de la liste L et l'ajoute à \mathcal{C} , où v^* est le sommet ayant la plus grande somme des poids de ses sommets adjacents, i.e., $W(v^*) = \max_{v \in L} \sum_{k \in V(v)} w_k$, où $V(v)$ est l'ensemble des sommets adjacents à v , qui satisfait la *condition d'admission* liée à la faisabilité d'un planning de recharge. Cette procédure est répétée jusqu'à ce que L devienne vide ou tous les noeuds de L ne vérifient pas la *condition d'admission*.

- **Étape d'amélioration.** Quand il n'y a plus de sommets qui pourraient améliorer le poids de la clique \mathcal{C} , l'algorithme procède à l'élimination des sommets de la clique. Ainsi, pour chaque $i \in \mathcal{C}$, l'algorithme détermine la liste des noeuds $N_i = \{v \notin \mathcal{C} \mid v \text{ est connecté à tous les sommets de } \mathcal{C} \setminus \{i\}\}$, et exclut de \mathcal{C} le sommet i tel que $W(N_i) \geq W(N_j), \forall j \in \mathcal{C}$, où $W(N_i) = \sum_{k \in N_i} w_k$. Une nouvelle liste L de sommets candidats est déterminée et le meilleur sommet de L est ajouté à \mathcal{C} comme décrit dans l'étape initiale. Le processus d'ajout et/ou suppression de sommets est répété jusqu'à ce qu'un temps limite de calcul soit atteint. Lorsque l'étape d'amélioration est arrêtée, la meilleure clique est mise à jour, la clique courante est supprimée et l'algorithme redémarre avec la première étape jusqu'à ce que la condition d'arrêt soit atteinte. Enfin, l'algorithme génère la meilleure clique trouvée parmi toutes les cliques construites et supprime les noeuds de la meilleure clique de \mathcal{G} .
- **Condition d'arrêt.** L'algorithme s'arrête si un temps limite de traitement est atteint ou si le poids de la clique trouvée est égale au poids total du graphe.

b- Condition d'admission

La condition d'admission assure que les tournées correspondantes aux noeuds de la clique courante peuvent être effectuées par un véhicule électrique, c'est-à-dire, il existe un planning de recharge faisable qui permet au véhicule électrique d'avoir suffisamment d'énergie pour effectuer l'ensemble des tournées de la clique.

L'existence d'un planning de recharge faisable est garantie par la métrique me , où me représente la quantité d'énergie minimale que doit être injectée dans la batterie du véhicule électrique avant qu'il n'effectue la première tournée de la clique, i.e., $me = \sum_{k=1}^{|\mathcal{C}|} (E_k - Y_k)$, où Y_k est la quantité d'énergie que peut être chargée entre deux tournées consécutives $k-1$ et k . En d'autres termes, $Y_k = \delta \times \sum_{t=f_{k-1}+1}^{s_k-1} \min\{p^{\max}, p'_t\}$, où p'_t est la puissance résiduelle du réseau électrique pendant l'intervalle de temps t . Nous supposons que $Y_1 = 0$. Un sommet v est ajouté à la clique \mathcal{C} , si, en suivant l'ordre chronologique des tournées, pour chaque sommet $j \in \mathcal{C}$, $R_j = \sum_{k=1}^j (E_k - Y_k)$ est inférieur à la capacité de batterie du véhicule électrique et $R_j \leq \delta \times \sum_{t=0}^{s_1-1} \min\{p^{\max}, p'_t\}$.

La Figure 4.3 représente un exemple de calcul de la métrique me . Soit la clique 1 composée des tournées A et B avec un besoin en énergie de 5 kWh. La métrique me correspondant à cette clique est égale à 11. Soit la clique 2 composée des tournées C et D avec un besoin en énergie de 4 kWh. La quantité d'énergie qui peut être récupérée entre l'instant de fin de la tournée C et l'instant de début de la tournée D est au plus égale à 3 kWh. La métrique me de cette clique est égale à 5.

La Figure 4.4 représente un exemple illustratif de l'algorithme de construction de tournées. Un graphe est au début construit, une première clique composée des tournées 2, 3 et 5 avec un poids total de 60 est ensuite définie. Cette clique est affectée au VE 1. Finalement, une deuxième clique composée des tournées 1 et 4 est déterminée, elle est affectée au VE 2.

b. Algorithme d'optimisation de la recharge

Étant donnée la meilleure clique \mathcal{C} générée par l'algorithme de sélection de tournées, l'objectif ici est de fournir un planning de recharge optimal qui minimise le coût total de recharge et satisfait les contraintes du problème décrites auparavant.

Si on reprend l'exemple schématisé sur la Figure 4.4, une solution au problème d'optimisation des véhicules électriques engagés est donnée dans la Figure 4.5.

Dans ce qui suit, nous proposons deux algorithmes d'optimisation de la recharge. Le premier algorithme est basé sur une formulation du problème de flot à coût minimum (PFCM) et fournit

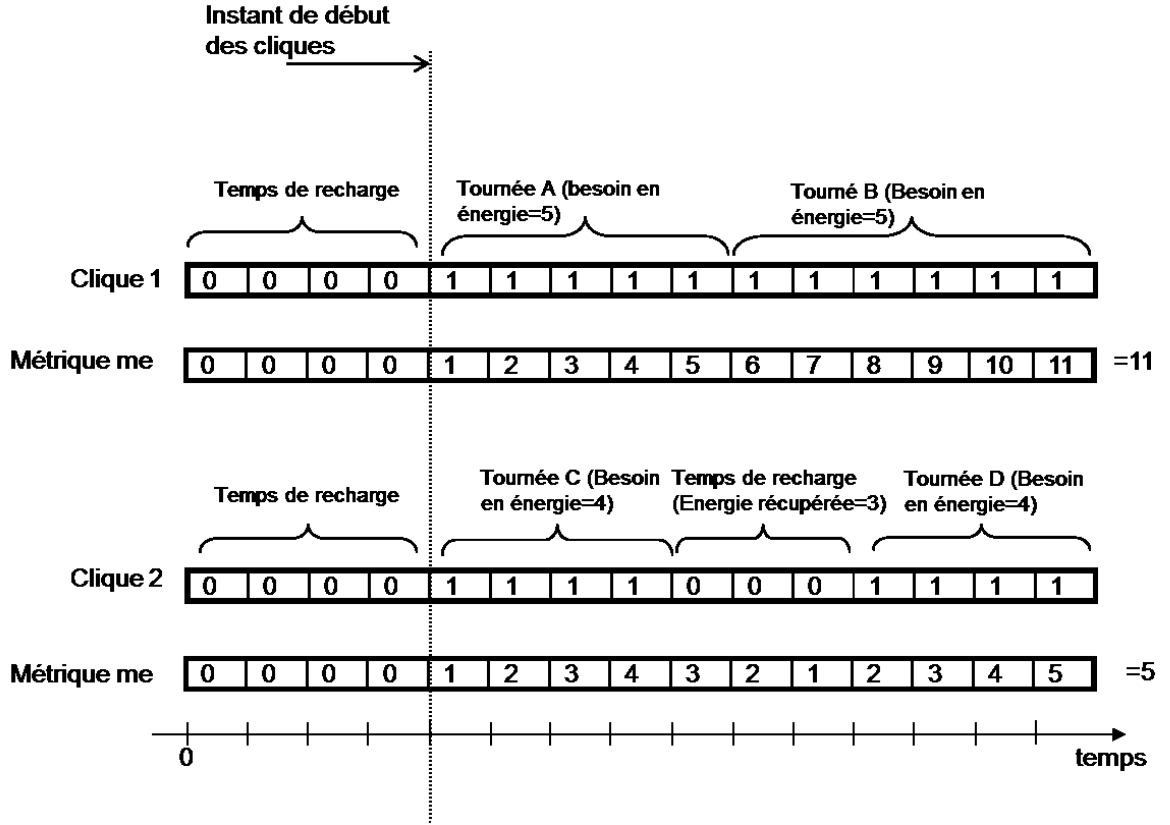


FIG. 4.3: Calcul de la valeur de la métrique.

une solution optimale en $O(((H + |\mathcal{C}|) \log(H + |\mathcal{C}|))^2)$. Le deuxième algorithme donne une solution optimale en $O((H + |\mathcal{C}|)^2)$.

i- Algorithme de recharge basé sur le PFCM

Dans cette section, nous montrons que le problème de recharge peut être réduit au problème de flot à coût minimum [90] dans un réseau construit comme suit. Pour un ensemble donné V de sommets de la clique \mathcal{C} , un réseau $\mathcal{G} = (\mathcal{W}, \mathcal{A})$ est défini comme suit :

- L'ensemble des sommets \mathcal{W} est composé de (i) une source s , (ii) les sommets t_l représentant les périodes de temps $[l-1, l]$, à part les périodes de temps pendant lesquels le véhicule électrique n'est pas disponible pour la recharge, i.e., $l \in \{1, \dots, H\} \setminus \cup_{v \in \{1, \dots, |V|\}} \{s_v + 1, \dots, f_v\}$, (iii) les sommets v_k , $k = 1, \dots, |V|$, qui représentent les tournées indexées par ordre croissant de leurs heures de début s_k , (iv) un puits p .
- L'ensemble \mathcal{A} d'arêtes est composé des : (i) arêtes (s, t_l) ayant une capacité maximale $a_l = \min\{p^{max}, p'_l\} \times \delta$, où p'_l est la capacité résiduelle du réseau électrique pendant l'intervalle de temps l , et un coût c_l correspondant au coût de l'électricité pendant la période de temps l , (ii) arêtes (t_l, v_k) si $l < s_k + 1$ et $l > f_{k-1}$ avec une capacité $+\infty$ et un coût nul, (iii) arêtes (v_{k-1}, v_k) , $k = 2, \dots, |V|$ avec une capacité égale à $CE_j - E_{k-1}$ et un coût nul, (iv) arêtes (v_k, p) , $k = 1, \dots, |V|$ avec une borne supérieure et une borne inférieure de la capacité égale à E_k et un coût nul.

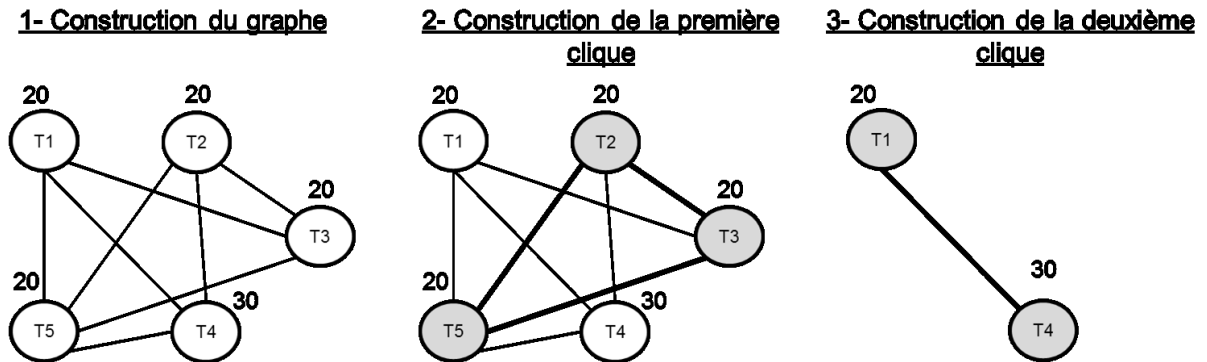


FIG. 4.4: Schéma illustratif du déroulement de l'algorithme de sélection des tournées

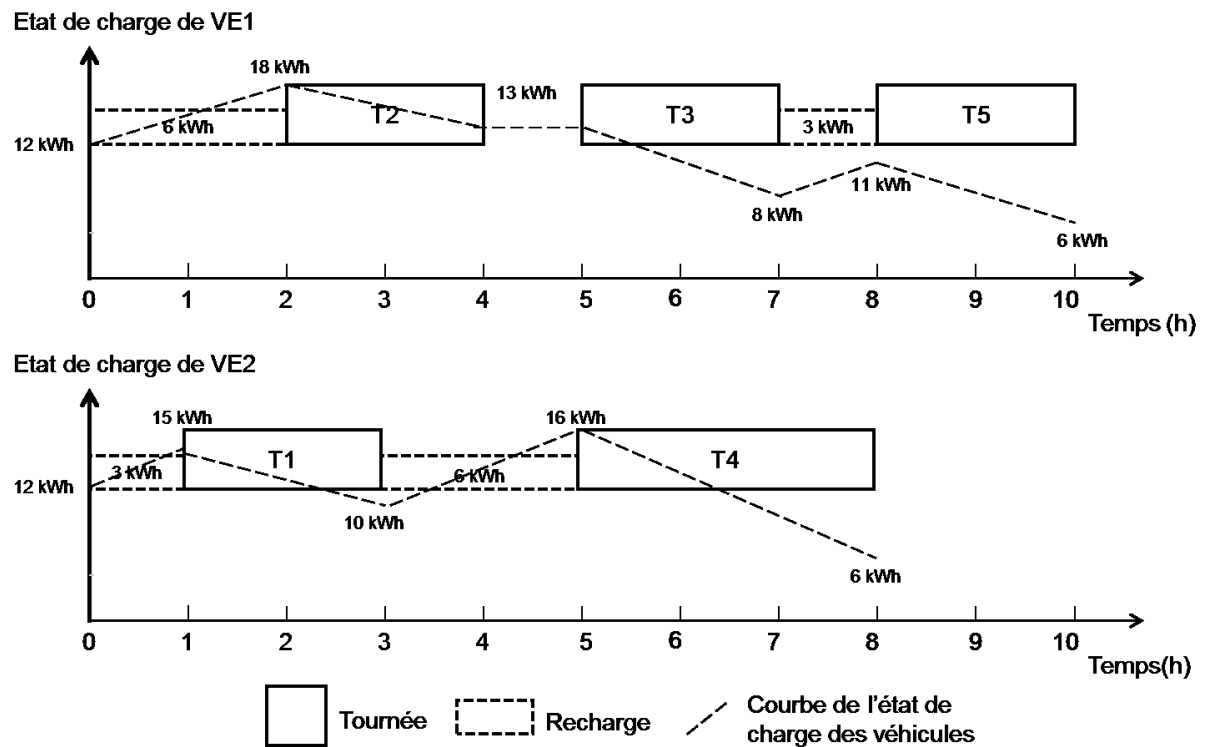


FIG. 4.5: Solution au problème d'optimisation de la recharge

Soit $f(i, j)$ le flot parcourant l'arête $(i, j) \in \mathcal{A}$ dans une solution optimale du PFCM avec un coût total $\sum_{l \in \{1, \dots, H\} \setminus \cup_{v \in V} \{s_v + 1, \dots, f_v\}} c_l \times f(s, t_l)$. Nous définissons un planning faisable pour le problème de recharge de la façon suivante : à chaque intervalle de temps l , où $l \in \{1, \dots, H\} \setminus \cup_{v \in V} \{s_v + 1, \dots, f_v\}$, appliquons sur le véhicule électrique j une puissance de recharge $p_{j,l} = \frac{f(s, t_l)}{\delta}$. En utilisant cette procédure, toute solution faisable au PFCM pourrait être transformée en une solution faisable S au problème de recharge avec un coût $C(S) = \sum_{l \in \{1, \dots, H\} \setminus \cup_{v \in V} \{s_v + 1, \dots, f_v\}} c_l \times f(s, t_l)$. Ainsi, le coût total de la recharge est minimisé quand le coût du flot est minimisé. Un exemple de construction de graphe est schématisé dans la Figure 4.5.2 où deux tournées v_1 et v_2 sont affectées à un VE j . La tournée

v_1 doit être exécutée pendant l'intervalle de temps t_4 et consomme une quantité d'énergie égale à $E1$ kWh. La tournée v_2 est réalisée pendant l'intervalle de temps t_8 et est caractérisée par une consommation énergétique égale à $E2$ kWh. Il s'agit de faire circuler la quantité $E1 + E2$ de la source s jusqu'au sommet puits p .

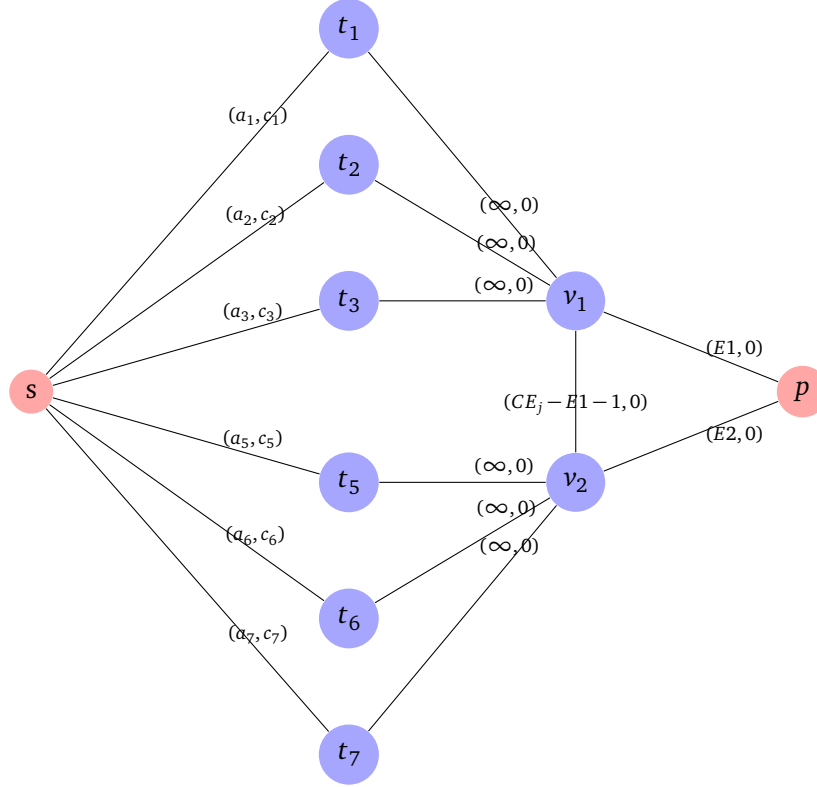


FIG. 4.6: Représentation du problème de recharge sous la forme d'un problème de flot à coût minimum

Lemme 4.2. Une solution optimale au problème de recharge peut être calculée en $O(((H + |V|) \log(H + |V|))^2)$.

Preuve : Le PFCM peut être résolu en $O(((H + |V|) \log(H + |V|))^2)$ [91]. La solution optimale au problème de flot à coût minimum se transforme en une solution optimale au problème de recharge en $O(H)$.

Dans le paragraphe suivant, un algorithme glouton de recharge, fonctionnant en $O(H \times |V|)$, est proposé.

ii- Algorithme glouton de recharge

Soit $C = (c_1, \dots, c_H)$ et $G = (g_1, \dots, g_H)$ deux vecteurs de H éléments, où c_t et g_t sont le coût d'électricité et la puissance disponible du réseau électrique pendant l'intervalle de temps $[t - 1, t]$, respectivement. Soit $E = (e_1, \dots, e_H)$ le vecteur d'énergie, où e_t , $t = 1, \dots, H$, représente soit la quantité d'énergie chargée pendant l'intervalle de temps $[t - 1, t]$ soit la quantité d'énergie consommée si le véhicule électrique est en tournée pendant l'intervalle de temps $[t - 1, t]$. e_t est initialisé à $\frac{-E_k}{f_k - s_k + 1}$ si $t \in [s_k, f_k]$, $\forall k \in V$, et à 0 sinon.

L'idée de notre algorithme glouton de recharge est d'injecter dans chaque véhicule électrique la quantité maximale d'énergie pendant les intervalles de temps les moins chers. À chaque intervalle de temps $[t - 1, t]$, e_t est égale à la quantité maximale d'énergie que peut être chargée pendant l'intervalle de temps $[t - 1, t]$ si le véhicule électrique n'est pas en tournée. Pour calculer la valeur de e_t , la puissance résiduelle de la batterie du véhicule électrique et la capacité résiduelle du réseau électrique pendant l'intervalle de temps $[t - 1, t]$ doivent être prises en compte. Plus précisément, l'algorithme commence par trier le vecteur C dans l'ordre croissant de c_t . Pour chaque valeur c_t tel que $e_t = 0$ (l'intervalle de temps $[t - 1, t]$ n'est pas encore examiné et le véhicule électrique j est disponible pour la recharge), soit V_t l'ensemble de tournées tel que $V_t \subseteq V$ et $v_k \in V_t$ si, et seulement si, $s_k > t$. Pour chaque $v_k \in V_t$ triées dans l'ordre de leurs indices, (i) calculer l'énergie $E_k^{t,B}$ qui peut être injectée dans le véhicule sans dépasser la capacité de la batterie ($E_k^{t,B} = SoC^{max} \times CE_j - \sum_{l=1}^{s_k} e_l$), (ii) calculer l'énergie $E_k^{t,V}$ nécessaire pour effectuer la tournée v_k ($E_k^{t,V} = Soc_j^{min} \times CE_j - \sum_{l=1}^{f_k} e_l$). Soit $E_k^t = \min\{E_k^{t,B}, E_k^{t,V}\}$. Après avoir parcouru les tournées de V_t , l'énergie E^t qui doit être chargée pendant l'intervalle de temps $[t - 1, t]$ est $E^t = \min\{p^{max} \times \delta, g_t \times \delta, \min_{k \in V_t} \{E_k^t\}\}$. Finalement, les vecteurs E et G sont mis à jour ($e_t := E^t$ et $g_t := g_t - \frac{E^t}{\delta}$).

L'algorithme 3 décrit plus en détail notre approche gloutonne pour l'optimisation de la recharge des véhicules électriques.

La Figure 4.7 schématise les trois premières itérations de notre algorithme de recharge. À chaque nouvelle itération, les valeurs mises à jour sont présentées en rouge. Supposons que le véhicule électrique soit affecté aux tournées C et D. La première tournée nécessite 3 kWh et la deuxième 2 kWh. La batterie du véhicule électrique est initialement vide. Considérons que les bornes de recharge peuvent fournir une puissance de recharge inférieure ou égale à 3 kW. L'horizon de planification est divisé en périodes de temps de 15 minutes. Pendant chaque intervalle de temps, le véhicule électrique peut récupérer une quantité d'énergie qui est au plus égale à 0.75 kWh. Remarquons qu'à la deuxième itération, la puissance du réseau disponible est égale à 2 kW. Le véhicule électrique peut alors au plus récupérer 0.5 kWh.

Théorème 4.3. *L'algorithme 3 génère une solution optimale au problème d'optimisation de recharge en $O(H \times |V|)$.*

Preuve : Soit S la solution de recharge générée par l'algorithme de recharge glouton et soit S^* la solution optimale générée par l'algorithme de recharge basé sur le PFCM. Notons $C(S)$ et $C(S^*)$ les coûts des solutions S et S^* , respectivement. Soit t_1 la première période de temps tel que $p_{j,t_1}(S) \neq p_{j,t_1}(S^*)$ où c_{t_1} est le coût d'électricité le moins cher. Par construction, l'algorithme de recharge glouton applique, à chaque période de temps, la puissance maximale disponible pour charger le VE j . Nous avons alors $p_{j,t_1}(S) > p_{j,t_1}(S^*)$. Soit t_2 un autre intervalle de temps tel que $p_{j,t_2}(S) < p_{j,t_2}(S^*)$. Soit S' une nouvelle solution dérivée de S^* où $p_{j,t_i}(S^*) = p_{j,t_i}(S') \forall i \neq 1, 2$, $p_{j,t_1}(S') = p_{j,t_1}(S^*) + \epsilon$ et $p_{j,t_2}(S') = p_{j,t_2}(S^*) - \epsilon$, où $0 < \epsilon \leq p_{j,t_1}(S) - p_{j,t_1}(S^*)$. Il est clair que S' est une solution faisable au problème de recharge. Par ailleurs $C(S') = C(S^*) - c_{t_2} + c_{t_1} < C(S^*)$ étant donné que $c_{t_2} > c_{t_1}$. Ceci contredit le fait que S^* soit une solution optimale. Il en résulte que $\forall i$, $p_{j,t_i}(S) = p_{j,t_i}(S^*)$. En conséquence, S est une solution optimale au problème de recharge.

La complexité de l'algorithme de recharge glouton est dominée par les deux boucles 5 et 7 de l'algorithme 3. L'algorithme glouton génère une solution optimale en $O(H \times |V|)$.

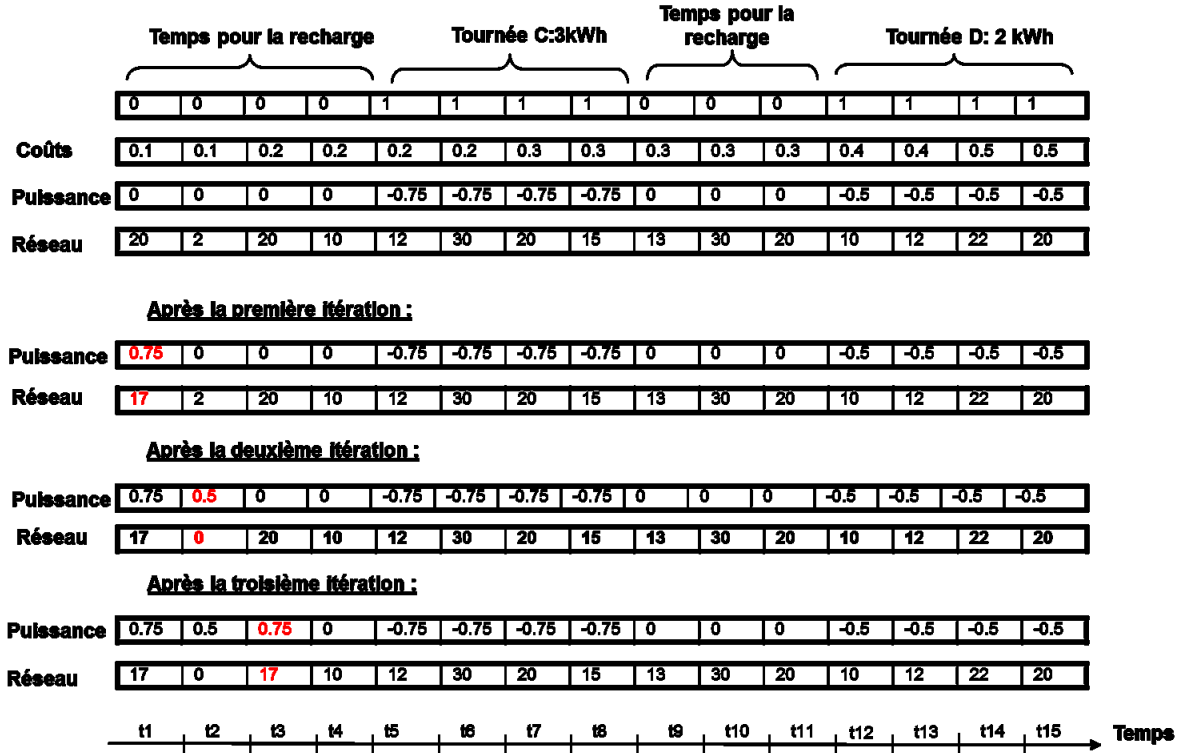


FIG. 4.7: Fonctionnement de l'algorithme glouton de recharge.

Algorithme 3 Algorithme glouton de recharge

- 1: **Entrées** : Un ensemble de tournées affectées au VE j
 - 2: **Sorties** : Planning de recharge pour le VE j
 - 3: Soit $C = (c_1, \dots, c_H)$, $G = (g_1, \dots, g_H)$ and $E = (e_1, \dots, e_H)$ trois vecteurs de H éléments, où c_t , $t = 1, \dots, H$, est le coût d'électricité pendant l'intervalle de temps $[t - 1, t]$, g_t , $t = 1, \dots, H$, est la puissance résiduelle de réseau électrique pendant l'intervalle de temps $[t - 1, t]$ et e_t , $t = 1, \dots, H$, est soit la quantité d'énergie chargée pendant l'intervalle de temps $[t - 1, t]$ ou la quantité d'énergie consommée pendant $[t - 1, t]$ si le véhicule électrique est en tournée. Le paramètre e_t est initialisé à $\frac{-E_k}{f_k - s_k + 1}$ si $t \in [s_k, f_k]$, $\forall k \in V$ et à 0 sinon.
 - 4: Trier le vecteur C dans l'ordre croissant des coûts c_t et soit $C = (c_{\pi(1)}, \dots, c_{\pi(H)})$ le vecteur obtenu
 - 5: **pour** chaque intervalle de temps $[\pi(t) - 1, \pi(t)]$ tel que $e_{\pi(t)} = 0$ **faire**
 - 6: Soit $V_{\pi(t)}$ un ensemble de tournées tel que $V_{\pi(t)} = \{v_k \in V | s_k > \pi(t)\}$ est trié dans l'ordre croissant des indices des tournées.
 - 7: **pour** chaque v_k in $V_{\pi(t)}$ **faire**
 - 8: Calculer $E_k^{\pi(t)} = \min\{SoC_j^{max} \times CE_j - \sum_{l=1}^{s_k} e_l ; SoC_j^{min} \times CE_j - \sum_{l=1}^{f_k} e_l\}$
 - 9: **fin pour**
 - 10: Calculer $E^{\pi(t)} = \min\{P^{max} \times \delta, g_{\pi(t)} \times \delta ; \min_{k \in V_{\pi(t)}} \{E_k^{\pi(t)}\}\}$
 - 11: Mettre à jour $e_{\pi(t)} = E^{\pi(t)}$ et $g_{\pi(t)} = g_{\pi(t)} - \frac{E^{\pi(t)}}{\delta}$
 - 12: **fin pour**
 - 13: Le planning de recharge est donné par le vecteur des puissances $P = (\frac{\max\{0, e_1\}}{\delta}, \dots, \frac{\max\{0, e_H\}}{\delta})$ qui doivent être appliquées au VE j .
-

4.5.3 Heuristique globale

Dans cette section, nous proposons une Heuristique Globale - GH pour résoudre le problème EVSCP. L'algorithme GH procède en deux étapes. La première étape a comme objectif d'affecter les tournées aux véhicules tout en s'assurant de l'existence d'un planning de recharge faisable. Ensuite, étant donnée une affectation faisable des tournées aux véhicules, la deuxième étape détermine un planning de recharge optimal pour tous les véhicules électriques en utilisant le PFCM. Les deux étapes sont détaillées dans le paragraphe suivant.

Dans la première étape, l'algorithme de sélection de tournées est utilisé pour affecter les tournées aux véhicules et une nouvelle *Procédure d'Estimation* de la puissance disponible sur le réseau, est introduite. En effet, dans l'algorithme séquentiel, le planning de recharge est déterminé après l'étape de sélection de tournées, ceci permet de mettre à jour les valeurs de la capacité disponible du réseau à chaque intervalle de temps t . Par contre, dans l'algorithme GH, le planning de recharge de tous les véhicules électriques est déterminé dans la deuxième étape.

Afin de s'assurer de l'existence d'un planning de recharge pour tous les véhicules électriques dans la seconde étape de l'algorithme GH, nous avons besoin d'estimer la puissance du réseau électrique disponible après l'affectation d'un ensemble de tournées à chaque véhicule. Ainsi, pour chaque véhicule, après l'algorithme de sélection de tournées, une nouvelle *Procédure d'estimation* de la puissance du réseau électrique disponible est introduite. En effet, la capacité du réseau électrique est mise à jour en considérant que le véhicule électrique est chargé le plus tard possible. Plus précisément, l'algorithme de recharge glouton est appliqué pour simuler la recharge du véhicule électrique et mettre à jour la puissance du réseau disponible à chaque instant t . Afin de s'assurer que le véhicule est chargé au plus tard, dans l'algorithme 3, au lieu de trier le vecteur de coût C (étape 3 de l'algorithme 3) dans l'ordre croissant de c_i , le vecteur C est trié dans l'ordre décroissant des indices de temps t .

Étant données les tournées affectées aux véhicules, la deuxième étape de l'algorithme GH détermine un planning de recharge optimal pour tous les véhicules. Plus précisément, étant donnés les ensembles L_1, \dots, L_{m_1} , de tournées affectées aux VE_1, \dots, VE_{m_1} , respectivement, le problème d'optimisation de la recharge est modélisé sous la forme d'un problème de flot à coût minimum dans un réseau $\mathcal{G} = (\mathcal{W}, \mathcal{A})$ et il est défini comme suit :

- L'ensemble des sommets \mathcal{W} est composé : (i) d'une source s , (ii) des sommets t_l représentant les périodes de temps $[l-1, l]$, (iii) des sommets $v_k^j \in L_j$, $j = 1, \dots, m_1$ représentant les tournées indexées dans le sens croissant de leurs heures de début, et (iv) un puits p .
- L'ensemble \mathcal{A} des arêtes orientées et ayant des capacités limitées, est composé : (i) des arcs (s, t_l) avec des capacités limitées $a_l = \min\{p^{max}, g_l\} \times \delta$ et un coût c_l correspondant au coût de l'énergie pendant la période de temps l , (ii) des arcs (t_l, v_k^j) si $t_l < s_k^j + 1$ et $t_l > f_{k-1}^j$ avec une capacité $+\infty$, et un coût nul, (iii) des arcs (v_{k-1}^j, v_k^j) , $k = 2, \dots, |L_j|$ avec une capacité maximale égale à $CE_j - E_{k-1}$ et un coût nul, (iv) des arcs (v_k^j, p) , $k = 1, \dots, |L_j|$, $j = 1, \dots, m_1$, avec une borne supérieure et une borne inférieure de la capacité égale à E_k^l , et un coût nul.

Soit $f(i, j)$ le flot traversant l'arc $(i, j) \in \mathcal{A}$ dans une solution optimale du problème de flot avec un coût total $\sum_{l=1}^H c_l \times f(s, t_l)$. Nous définissons un planning de recharge optimale de cette façon : à chaque intervalle de temps l , où $l \in \{1, \dots, H\}$, appliquer sur le véhicule électrique j une puissance de recharge $p_{j,l} = \frac{f(t_l, v_k^j)}{\delta}$, où $(t_l, v_k^j) \in \mathcal{A}$. Il est facile de montrer que le planning de recharge obtenu est optimal. La preuve est identique à celle du théorème 4.3. En utilisant cette procédure, une

solution optimale du problème PFCM peut être transformée en une solution optimale S au problème de recharge. Ainsi, quand le coût du flot est minimisé, le coût total de recharge est minimisé. Une solution optimale au problème de recharge peut être trouvée en $O(((H + |L|)\log(H + |L|))^2)$, où $L = \cup_{j=1}^{m_1} L_j$.

Algorithme 4 Algorithme GH

- 1: **Entrées** : Instance du problème EVSCP
 - 2: **Sorties** : Une solution réalisable
 - 3: **pour** chaque VE **faire**
 - 4: Appliquer *l'algorithme de sélection des tournées* décrit dans la Section 4.5.2
 - 5: Mettre à jour la capacité du réseau électrique disponible en appliquant la *procédure d'estimation* ; i.e., trier le vecteur de coût C dans l'ordre décroissant des indices de temps t et appliquer l'algorithme 3 comme décrit auparavant
 - 6: **fin pour**
 - 7: Appliquer l'algorithme de recharge pour déterminer un planning de recharge optimal pour tous les véhicules électriques
 - 8: **pour** chaque VT **faire**
 - 9: Appliquer *l'algorithme de sélection des tournées*
 - 10: **fin pour**
-

4.6 Résultats expérimentaux et discussion

Pour évaluer la qualité des solutions obtenues par la méthode exacte et les deux heuristiques, nous avons conduit des expérimentations intensives sur des instances réelles et d'autres aléatoirement générées.

Les instances réelles ont été fournies par le Groupe La Poste. Toutes les expérimentations sont effectuées sur un processus Intel Xeon E5620 2.4GHz avec une mémoire RAM de 8GB. Les algorithmes développés sont codés en C++. Le solveur ILOG CPLEX 12.5 est utilisé pour résoudre le modèle linéaire en nombres mixtes.

Les résultats générés par les différents algorithmes sont comparés en termes de la qualité des solutions générées et du temps de calcul. La procédure d'optimisation est basée sur une période de 24h. Les coûts d'électricité sont basés sur les tarifs appliqués par EDF. Pour ces expérimentations, au plus m_1 chargeurs pouvant effectuer une recharge normale et fournir une puissance de recharge variant entre 0.0(kW) et 3.7(kW), peuvent être utilisés.

La première instance de données est composée de 18 véhicules thermiques et 8 véhicules électriques. Tous les véhicules électriques considérés ont une capacité de batterie de 22(kWh). 44 tournées doivent être affectées aux véhicules disponibles. La deuxième instance de données est composée de 14 véhicules thermiques, 8 véhicules électriques ayant une capacité de batterie de 22(kWh) et 4 véhicules électriques avec des batteries de 23.5(kWh). 46 tournées doivent être affectées aux véhicules disponibles. Pour ces deux instances réelles, CPLEX et les deux heuristiques ont réussi à générer des solutions optimales en peu de temps de calcul (moins de 3 secondes).

Pour des raisons de confidentialité, nous ne sommes pas autorisés à publier les détails des

instances et les résultats des expérimentations liées aux instances réelles. En revanche, nous résumons nos principaux résultats dans le tableau 4.4. La colonne 'Solution réelle' représente la solution mise en place sur les sites d'expérimentation et la colonne *Approche exacte* représente les solutions générées par notre méthode exacte. Les indicateurs suivants sont relevés : (i) le nombre de véhicules électriques et thermiques utilisés dans la solution, (ii) la distance parcourue par les VEs et les VTs (iii) la quantité d'énergie consommée par les VEs, (iv) le coût de la recharge et (v) la puissance maximale atteinte lors de la recharge des VEs.

Du tableau 4.4, nous pouvons faire les observations suivantes :

- Tous les VEs sont utilisés et moins de VTs sont utilisés dans la solution optimale par comparaison à la solution réelle.
- La distance totale parcourue par les VEs est améliorée de 71.3% pour l'instance 1 et de 21.8% pour l'instance 2, par comparaison à la solution existante. La quantité d'énergie nécessaire augmente de 60.2% pour l'instance 1 et de 23.5% pour l'instance 2, ce qui est cohérent avec le fait que les VEs soient beaucoup plus utilisés par rapport à la solution réelle.
- Le coût de l'énergie consommée par les VEs dans la solution optimale est très proche du coût de l'énergie de la solution réelle pour l'instance 1 et il est encore mieux pour l'instance 2. Ceci peut être expliqué par le fait que, pour la solution réelle, les VEs sont rechargés à leur retour des tournées et aucune optimisation des coûts n'est mise en place. Par contre, pour la solution optimale, les VEs sont chargés lorsque le coût de l'électricité est bas et la courbe de recharge est lissée pendant toute l'opération de recharge.

TAB. 4.4: Résultats numériques sur des instances réelles

	Instance 1		Instance 2	
	Solution réelle	Approche exacte	Solution réelle	Approche exacte
Nombre de VEs	8	8	12	12
Nombre de VTs	18	17	14	12
Km faits par les VEs	380	651	562	685
Km faits par les VTs	1020	749	516	393
Énergie consommée par les VEs	94.36	152.07	72.88	90.04
Coût de recharge	4.28	4.90	5.72	4.93
Pic de consommation (kVA)	24	23.5	36	27

La deuxième partie des expérimentations réalisées concerne la validation de nos approches de résolution sur des instances générées aléatoirement.

Ces instances aléatoires sont générées de la façon suivante : Le nombre total des véhicules $nv = m_1 + m_2$ prend ses valeurs de l'ensemble $\{40, 80, 120, 160, 200\}$. Le nombre de véhicules électriques m_1 est un ratio de nv ($m_1 = ev \times nv$). ev prend ses valeurs de l'ensemble $\{0.25, 0.5, 0.75, 1.0\}$. Un paramètre a est utilisé pour distinguer la capacité de batterie des véhicules électriques, à savoir $a = 1$ implique que tous les véhicules électriques ont une capacité de batterie de 22(kWh) et $a = 2$ signifie que 50% des véhicules électriques ont une capacité de batterie de 22(kWh) et le reste a une capacité de batterie de 16(kWh). Au plus m_1 bornes de recharge, pouvant fournir une puissance de recharge comprise entre 0(kW) et 3.7(kW), peuvent être utilisées pour charger les véhicules électriques. Le paramètre tt est lié au nombre total des tournées. Si $tt = 1$, alors $n = nv \times c$, où $c \in [1.2, 1.3]$. Si $tt = 2$, alors $n = nv \times c$, où $c \in [1.5, 1.6]$. Toutes les tournées commencent au plus tôt à 6h et

se terminent au plus tard à 20h et leur besoin en énergie varie entre 0.15(kWh) et 0.35(kWh) par kilomètre.

50% de ces tournées sont longues ; c'est-à-dire, elles durent entre 5 et 7 heures, elles commencent entre 6h et 13h et leurs longueurs se situent dans l'intervalle $[50(km), 80(km)]$. Le reste des tournées sont dites moyennes ; c'est-à-dire, elles durent entre 2 et 4 heures et leurs longueurs sont dans l'intervalle $[20(km), 45(km)]$. Une fraction de $\frac{1}{6}$ des tournées commencent entre 6h et 8h, $\frac{1}{3}$ commencent entre 8h et 10h, $\frac{1}{6}$ se terminent entre 18h et 20h et $\frac{1}{3}$ se terminent entre 16h et 18h. Concernant la capacité du réseau électrique disponible pour la recharge des véhicules électriques, nous considérons que si $t \in [00am, 6am]$, $g_t = \frac{2}{3} \times ev \times nv \times p^{\max}$, si $t \in [6am, 6pm]$, $g_t = \frac{1}{5} \times ev \times nv \times p^{\max}$ et si $t \in [6pm, 00am]$, $g_t = \frac{1}{2} \times ev \times nv \times p^{\max}$, où $p^{\max} = 3.7(kW)$.

Pour chaque valeur de nv , 16 classes d'instances sont alors construites et pour chaque classe, 10 instances sont générées aléatoirement. Chaque classe est notée $nv_w_ev_x_a_y_tt_z$, où $w = nv$, x prend les valeurs 1, 2, 3, 4 si ev est égal à : 0.25, 0.50, 0.75, 1, respectivement, $y = a = \{1, 2\}$ et $z = tt = \{1, 2\}$.

Les résultats numériques générés par CPLEX, l'algorithme *SH* et l'algorithme *GH* sont résumés dans le tableau 4.5. Un temps limite de 3600 secondes est fixé pour chaque instance. Notons que dans le tableau 4.5, nous comparons uniquement les valeurs de la première fonction objectif $(\sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij})$ étant donné que les coûts de la recharge dépendent de la solution d'affectation.

Dans le tableau 4.5, nous précisons les paramètres suivants : (i) *Success* : est le pourcentage des instances résolues parmi toutes les instances testées de la même classe. Une instance est dite "résolue" si une solution pour cette instance a été générée dans la limite du temps maximum de résolution, même si cette solution n'est pas optimale, (ii) $Gap_{UB}(M)$: est l'écart entre la meilleure solution S_M générée par une méthode $M \in \{EM, SH, GH\}$ (*EM*, *SH*, et *GH* sont CPLEX, algorithme *SH* et algorithme *GH*, respectivement) par rapport à la borne supérieure (UB) générée par CPLEX, i.e., $Gap_{UB}(S_M) = \frac{UB - S_M}{S_M}$, (iii) $Gap_{max}(M)$: est l'écart entre la meilleure solution générée par les méthodes *EM*, *SH* et *GH* et la solution générée par la méthode $M \in \{EM, SH, GH\}$, c'est-à-dire, $Gap_{max}(M) = \frac{\max(S_{EM}, S_{SH}, S_{GH}) - S_M}{S_M}$, et, (iv) *Time* : est le temps moyen d'exécution nécessaire pour générer une solution d'affectation et de recharge au problème EVSCP. Le symbole '*' signifie que le temps limite d'exécution a été atteint sans qu'une solution faisable ne soit générée.

TAB. 4.5: Résultats de performance de CPLEX, SH et GH sur des instances aléatoires.

Instance	Méthode exacte			Algorithme SH			Algorithme GH					
	Success	Gap _{UB} (EM)	Gap _{max} (EM)	Time	Success	Gap _{UB} (SH)	Gap _{max} (SH)	Time	Success	Gap _{UB} (GH)	Gap _{max} (GH)	Time
nv_40_ce_1_a_1_tt_1	90.00%	5.27%	0.00%	3600.00	100.00%	6.72%	1.36%	0.07	100.00%	5.94%	0.62%	0.29
nv_40_ce_1_a_1_tt_2	90.00%	10.62%	1.02%	3602.00	100.00%	10.88%	1.29%	0.09	100.00%	9.46%	0.00%	0.26
nv_40_ce_1_a_2_tt_1	90.00%	10.29%	2.95%	3601.98	100.00%	7.36%	0.29%	0.07	100.00%	7.14%	0.00%	0.23
nv_40_ce_1_a_2_tt_2	100.00%	14.85%	3.68%	3608.23	100.00%	11.38%	0.61%	0.09	100.00%	10.71%	0.00%	0.27
nv_40_ce_2_a_1_tt_1	90.00%	8.54%	0.00%	3602.18	100.00%	13.70%	4.76%	0.09	100.00%	9.84%	1.21%	6.39
nv_40_ce_2_a_1_tt_2	100.00%	11.96%	0.00%	3601.00	100.00%	16.57%	4.17%	0.12	100.00%	12.61%	0.63%	6.57
nv_40_ce_2_a_2_tt_1	90.00%	13.39%	3.32%	3601.00	100.00%	12.75%	2.76%	0.09	100.00%	9.73%	0.00%	5.86
nv_40_ce_2_a_2_tt_2	100.00%	17.97%	4.37%	3601.00	100.00%	16.43%	3.13%	0.14	100.00%	12.89%	0.00%	7.36
nv_40_ce_3_a_1_tt_1	90.00%	0.50%	0.00%	3601.00	100.00%	0.50%	0.00%	0.11	100.00%	0.50%	0.00%	0.59
nv_40_ce_3_a_1_tt_2	100.00%	7.19%	0.00%	3602.00	100.00%	8.46%	1.18%	0.16	100.00%	8.23%	0.96%	0.93
nv_40_ce_3_a_2_tt_1	90.00%	1.40%	0.90%	3606.47	100.00%	0.46%	0.46%	0.12	100.00%	0.00%	0.00%	0.61
nv_40_ce_3_a_2_tt_2	100.00%	12.57%	4.58%	3601.00	100.00%	8.12%	0.49%	0.17	100.00%	7.59%	0.00%	1.26
nv_40_ce_4_a_1_tt_1	90.00%	0.00%	0.00%	13.47	100.00%	0.46%	0.46%	0.12	100.00%	0.00%	0.00%	0.60
nv_40_ce_4_a_1_tt_2	100.00%	0.00%	0.00%	29.96	100.00%	0.00%	0.00%	0.22	100.00%	0.00%	0.00%	1.19
nv_40_ce_4_a_2_tt_1	90.00%	0.00%	0.00%	19.14	100.00%	0.00%	0.00%	0.13	100.00%	0.00%	0.00%	0.60
nv_40_ce_4_a_2_tt_2	100.00%	0.00%	0.00%	52.64	100.00%	0.00%	0.00%	0.19	100.00%	0.00%	0.00%	1.32
nv_80_ce_1_a_1_tt_1	100.00%	18.65%	1.80%	3601.00	100.00%	19.75%	2.77%	0.33	100.00%	16.53%	0.00%	14.19
nv_80_ce_1_a_1_tt_2	100.00%	26.14%	5.57%	3601.00	100.00%	23.19%	3.15%	0.51	100.00%	19.43%	0.00%	12.37
nv_80_ce_1_a_2_tt_1	100.00%	24.43%	7.00%	3601.00	100.00%	20.05%	3.25%	0.29	100.00%	16.27%	0.00%	10.84
nv_80_ce_1_a_2_tt_2	100.00%	32.55%	10.40%	3601.00	100.00%	22.83%	2.57%	0.41	100.00%	19.75%	0.00%	12.33
nv_80_ce_2_a_1_tt_1	100.00%	20.47%	5.57%	3601.20	100.00%	14.12%	0.31%	0.39	100.00%	13.77%	0.00%	3.44
nv_80_ce_2_a_1_tt_2	100.00%	28.40%	11.23%	3601.48	100.00%	14.58%	0.14%	0.55	100.00%	14.42%	0.00%	4.18
nv_80_ce_2_a_2_tt_1	100.00%	35.45%	18.26%	3601.85	100.00%	14.30%	0.47%	0.35	100.00%	13.76%	0.00%	3.15
nv_80_ce_2_a_2_tt_2	100.00%	52.92%	32.61%	3602.36	100.00%	14.76%	0.32%	0.49	100.00%	14.40%	0.00%	3.92
nv_80_ce_3_a_1_tt_1	100.00%	1.18%	0.00%	818.78	100.00%	1.46%	0.27%	0.39	100.00%	1.35%	0.17%	4.88
nv_80_ce_3_a_1_tt_2	100.00%	15.49%	4.89%	3601.98	100.00%	10.05%	0.49%	0.63	100.00%	9.51%	0.00%	5.98
nv_80_ce_3_a_2_tt_1	100.00%	7.00%	5.47%	3606.53	100.00%	1.47%	0.09%	0.41	100.00%	1.38%	0.00%	4.20
nv_80_ce_3_a_2_tt_2	60.00%	25.87%	15.54%	3602.01	100.00%	8.61%	0.00%	0.60	100.00%	8.72%	0.10%	5.24

Instance	Méthode exacte				Algorithme SH				Algorithme GH			
	Success	Gap _{UB} (EM)	Gap _{max} (EM)	Time	Success	Gap _{UB} (SH)	Gap _{max} (SH)	Time	Success	Gap _{UB} (GH)	Gap _{max} (GH)	Time
nv_80_ce_4_a_1_tt_1	100.00%	0.00%	0.00%	117.31	90.00%	0.12%	0.12%	0.47	100.00%	0.00%	0.00%	4.93
nv_80_ce_4_a_1_tt_2	90.00%	0.00%	0.00%	798.39	90.00%	0.12%	0.12%	0.56	100.00%	0.00%	0.00%	6.34
nv_80_ce_4_a_2_tt_1	100.00%	0.00%	0.00%	217.83	100.00%	0.00%	0.00%	0.42	100.00%	0.00%	0.00%	5.01
nv_80_ce_4_a_2_tt_2	30.00%	0.00%	0.00%	1062.29	90.00%	0.00%	0.00%	0.55	100.00%	0.00%	0.00%	4.29
nv_120_ce_1_a_1_tt_1	100.00%	24.62%	6.93%	3601.74	100.00%	17.05%	0.47%	0.64	100.00%	16.51%	0.00%	3.54
nv_120_ce_1_a_1_tt_2	100.00%	48.84%	22.46%	3601.21	100.00%	20.88%	0.00%	1.02	100.00%	20.88%	0.00%	3.93
nv_120_ce_1_a_2_tt_1	100.00%	37.80%	17.91%	3601.01	100.00%	17.01%	0.39%	0.61	100.00%	16.56%	0.00%	3.14
nv_120_ce_1_a_2_tt_2	100.00%	73.91%	40.38%	3601.76	100.00%	21.77%	0.41%	1.08	100.00%	21.27%	0.00%	4.15
nv_120_ce_2_a_1_tt_1	100.00%	103.17%	63.42%	3602.63	100.00%	15.66%	0.83%	0.76	100.00%	14.72%	0.00%	8.35
nv_120_ce_2_a_1_tt_2	50.00%	111.66%	78.48%	3602.42	100.00%	19.50%	1.46%	1.33	100.00%	17.78%	0.00%	18.65
nv_120_ce_2_a_2_tt_1	100.00%	115.33%	80.87%	3603.01	100.00%	15.41%	0.73%	0.83	100.00%	14.57%	0.00%	8.66
nv_120_ce_2_a_2_tt_2	20.00%	95.76%	66.32%	3602.42	100.00%	18.57%	1.20%	1.23	100.00%	17.16%	0.00%	17.01
nv_120_ce_3_a_1_tt_1	100.00%	31.21%	29.70%	3603.29	100.00%	1.48%	0.71%	0.85	100.00%	0.77%	0.00%	15.15
nv_120_ce_3_a_1_tt_2	0.00%	*	*	*	100.00%	10.45%	1.88%	1.48	100.00%	8.41%	0.00%	15.86
nv_120_ce_3_a_2_tt_1	90.00%	29.44%	28.09%	3603.97	100.00%	1.19%	0.58%	0.89	100.00%	0.61%	0.00%	15.10
nv_120_ce_3_a_2_tt_2	0.00%	*	*	*	100.00%	10.86%	2.26%	1.72	100.00%	8.41%	0.00%	13.23
nv_120_ce_4_a_1_tt_1	90.00%	0.00%	0.00%	504.42	100.00%	0.00%	0.00%	0.79	100.00%	0.00%	0.00%	12.05
nv_120_ce_4_a_1_tt_2	0.00%	*	*	*	100.00%	2.10%	2.10%	1.61	100.00%	0.00%	0.00%	14.22
nv_120_ce_4_a_2_tt_1	30.00%	0.00%	0.00%	730.19	0.00%	0.33%	0.33%	0.80	100.00%	0.00%	0.00%	8.13
nv_120_ce_4_a_2_tt_2	0.00%	*	*	*	0.00%	*	*	*	100.00%	0.13%	0.00%	8.58
nv_160_ce_1_a_1_tt_1	100.00%	374.04%	146.35%	3603.88	100.00%	20.50%	0.56%	1.37	100.00%	19.84%	0.00%	5.87
nv_160_ce_1_a_1_tt_2	50.00%	223.20%	155.19%	3603.36	100.00%	25.20%	0.12%	1.99	100.00%	25.06%	0.00%	8.22
nv_160_ce_1_a_2_tt_1	90.00%	882.71%	450.83%	3602.62	100.00%	19.36%	0.00%	1.35	100.00%	19.36%	0.00%	1.65
nv_160_ce_1_a_2_tt_2	50.00%	150.93%	100.99%	3604.58	100.00%	23.86%	0.24%	2.35	100.00%	23.56%	0.00%	6.68
nv_160_ce_2_a_1_tt_1	10.00%	294.76%	240.23%	3606.24	100.00%	16.82%	0.68%	1.41	100.00%	16.03%	0.00%	63.09
nv_160_ce_2_a_1_tt_2	0.00%	*	*	*	100.00%	19.70%	0.89%	2.49	100.00%	18.90%	0.00%	132.13
nv_160_ce_2_a_2_tt_1	40.00%	223.84%	181.29%	3603.72	100.00%	13.97%	0.67%	1.51	100.00%	13.22%	0.00%	40.83
nv_160_ce_2_a_2_tt_2	0.00%	*	*	*	100.00%	19.31%	0.83%	2.45	100.00%	18.33%	0.00%	113.56
nv_160_ce_3_a_1_tt_1	0.00%	*	*	*	100.00%	6.16%	4.39%	1.57	100.00%	1.69%	0.00%	168.93
nv_160_ce_3_a_1_tt_2	0.00%	*	*	*	100.00%	19.14%	6.68%	2.84	100.00%	11.69%	0.00%	152.09
nv_160_ce_3_a_2_tt_1	0.00%	*	*	*	100.00%	6.27%	4.33%	1.57	100.00%	1.86%	0.00%	159.00

Instance	Méthode exacte				Algorithme SH				Algorithme GH			
	Success	Gap _{UB} (EM)	Gap _{max} (EM)	Time	Success	Gap _{UB} (SH)	Gap _{max} (SH)	Time	Success	Gap _{UB} (GH)	Gap _{max} (GH)	Time
nv_160_ce_3_a_2_tt_2	0.00%	*	*	*	100.00%	19.20%	6.80%	2.75	100.00%	11.61%	0.00%	129.82
nv_160_ce_4_a_1_tt_1	0.00%	*	*	*	90.00%	0.06%	0.95%	1.60	100.00%	0.00%	0.00%	26.24
nv_160_ce_4_a_1_tt_2	0.00%	*	*	*	30.00%	1.00%	0.94%	2.59	100.00%	0.05%	0.00%	137.98
nv_160_ce_4_a_2_tt_1	0.00%	*	*	*	100.00%	0.00%	0.00%	1.51	100.00%	0.00%	0.00%	29.09
nv_160_ce_4_a_2_tt_2	0.00%	*	*	*	20.00%	1.54%	1.54%	2.69	100.00%	0.00%	0.00%	121.15
nv_200_ce_1_a_1_tt_1	60.00%	390.39%	390.39%	3607.04	100.00%	21.75%	0.43%	2.11	100.00%	21.23%	0.00%	16.83
nv_200_ce_1_a_1_tt_2	0.00%	*	*	*	100.00%	22.28%	0.32%	4.32	100.00%	22.06%	0.00%	15.26
nv_200_ce_1_a_2_tt_1	20.00%	558.04%	447.35%	3600.00	100.00%	20.92%	0.80%	2.16	100.00%	20.22%	0.00%	15.11
nv_200_ce_1_a_2_tt_2	0.00%	*	*	*	100.00%	22.56%	0.80%	4.79	100.00%	21.58%	0.00%	16.73
nv_200_ce_2_a_1_tt_1	0.00%	*	*	*	100.00%	22.42%	4.01%	3.06	100.00%	17.70%	0.00%	184.33
nv_200_ce_2_a_1_tt_2	0.00%	*	*	*	100.00%	26.08%	3.79%	6.40	100.00%	21.48%	0.00%	109.74
nv_200_ce_2_a_2_tt_1	0.00%	*	*	*	100.00%	22.39%	4.06%	3.03	100.00%	17.62%	0.00%	185.07
nv_200_ce_2_a_2_tt_2	0.00%	*	*	*	100.00%	29.07%	4.07%	6.17	100.00%	24.03%	0.00%	127.91
nv_200_ce_3_a_1_tt_1	0.00%	*	*	*	100.00%	13.01%	6.66%	2.96	100.00%	5.95%	0.00%	164.35
nv_200_ce_3_a_1_tt_2	0.00%	*	*	*	30.00%	27.48%	7.13%	6.79	100.00%	18.99%	0.00%	173.80
nv_200_ce_3_a_2_tt_1	0.00%	*	*	*	100.00%	12.59%	6.55%	2.88	100.00%	5.67%	0.00%	122.31
nv_200_ce_3_a_2_tt_2	0.00%	*	*	*	30.00%	28.17%	7.24%	7.09	100.00%	19.52%	0.00%	188.96
nv_200_ce_4_a_1_tt_1	0.00%	*	*	*	100.00%	0.00%	0.00%	2.60	100.00%	0.00%	0.00%	183.20
nv_200_ce_4_a_1_tt_2	0.00%	*	*	*	70.00%	0.29%	0.29%	5.22	100.00%	0.00%	0.00%	109.60
nv_200_ce_4_a_2_tt_1	0.00%	*	*	*	100.00%	0.00%	0.00%	13.27	100.00%	0.00%	0.00%	110.67
nv_200_ce_4_a_2_tt_2	0.00%	*	*	*	70.00%	0.25%	0.25%	6.05	100.00%	0.00%	0.00%	156.08

Du tableau 4.5, nous pouvons tirer les conclusions suivantes :

- La méthode exacte réussit à résoudre des petites et moyennes instances (flottes avec 40 et 80 véhicules), mais elle échoue à résoudre des instances de grande taille (flottes de 120, 160 et 200 véhicules). Pour des larges instances avec un petit nombre de VEs ($ce = 0.25$), l'algorithme de résolution exacte réussit à trouver des solutions faisables. De plus, quand la flotte de VEs est hétérogène ($a = 2$), la méthode exacte a une performance limitée, en effet, elle ne réussit pas à trouver des solutions de bonne qualité comme il est le cas pour les classes d'instances avec une flotte homogène de VEs.
- L'heuristique *GH* réussit à résoudre toutes les instances, alors que l'heuristique *SH* échoue à résoudre 44 instances parmi les 800 instances. Ceci concerne particulièrement les grandes instances (par exemple, les instances des classes *nv_200_ce_3_a_2_tt_1* et *nv_200_ce_4_a_1_tt_2*). Ceci peut s'expliquer par le fait que l'algorithme *SH* définit un planning de recharge optimal pour chaque VE et ceci d'une façon séquentielle sans anticiper l'existence d'une solution de recharge faisable pour les derniers VEs traités par l'algorithme.
- Concernant la performance des heuristiques proposées, le tableau 4.5 montre que les deux heuristiques ont une très bonne performance. L'algorithme *GH* résout à l'optimalité 180 instances parmi les 800 instances testées, et a la meilleure performance avec Gap_{max} inférieur à 1% par comparaison aux meilleures solutions trouvées par la méthode exacte et l'heuristique *SH* et un écart Gap_{UB} d'au plus 20% par comparaison aux bornes supérieures générées par CPLEX.
L'heuristique *SH* réussit à trouver les solutions optimales pour 89 instances et génère des bonnes solutions avec un gap Gap_{max} d'au plus 7% et un Gap_{UB} d'au plus 30% par comparaison aux bornes supérieures générées par CPLEX pour les grandes instances.
- Le temps d'exécution des heuristiques *SH* and *GH* sont relativement courts. Environ 190 secondes sont nécessaires pour résoudre des instances de grande taille avec jusqu'à 200 véhicules en utilisant l'heuristique *GH*. Un temps de calcul plus court, avec au plus 13 secondes, est nécessaire pour trouver des bonnes solutions pour des grandes instances en utilisant l'heuristique *SH*.

4.7 Conclusion

Dans ce chapitre, nous avons considéré le problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques. Nous avons prouvé que ce nouveau problème est NP-difficile au sens ordinaire.

Pour résoudre ce problème, nous avons proposé un programme linéaire en nombres mixtes. Une méthode exacte, qui consiste à résoudre le programme linéaire en deux étapes en utilisant CPLEX, a été proposée. Nous avons également proposé deux heuristiques performantes. La première est une heuristique séquentielle qui utilise deux sous-algorithmes pour affecter un ensemble de tournées à chaque VE disponible en utilisant le problème de clique à poids maximum et pour optimiser la recharge en utilisant le problème de flot à coût minimum.

La deuxième heuristique génère une solution au problème d'affectation des tournées aux véhicules disponibles tout en s'assurant de la faisabilité de la recharge. Ensuite, elle optimise le planning de recharge.

Les résultats d'expérimentation sur des instances aléatoires montrent que les deux heuristiques génèrent des solutions de bonne qualité et ceci en un temps de calcul raisonnable.

Chapitre 5

Extensions au problème de base

Sommaire

5.1	Introduction	59
5.2	Dépassement de la puissance souscrite	60
5.3	Vieillessement des batteries	61
5.4	Modèle de véhicule	62
5.5	Modèle d'infrastructure de recharge	63
5.6	Bilan carbone	64
5.7	Optimisation de la flotte de véhicules	65
5.8	Cas d'étude et expérimentations	66
5.8.1	Protocole des expérimentations simulées	66
5.8.2	SC 1 : Scénario de référence	68
5.8.3	SC 2 : Modulation de la puissance de recharge	69
5.8.4	SC 3 : Optimisation de l'affectation et de la recharge des véhicules	70
5.8.5	SC 4 : Vieillessement de la batterie	71
5.8.6	SC 5 : Optimisation du bilan carbone	73
5.8.7	SC 6 : Optimisation des coûts d'exploitation	74
5.8.8	SC 7 : Optimisation de la flotte	76
5.9	Conclusion	77

Ce chapitre présente plusieurs extensions à notre problème de base. Différents scénarios d'usage sont proposés et testés sur des instances réelles.

5.1 Introduction

Pour valider les outils d'optimisation développés, plusieurs scénarios d'expérimentation ont été menés afin de tester différentes situations et vérifier le comportement et les réactions du véhicule électrique vis à vis de différentes conditions d'usage.

Dans ce chapitre, nous proposons plusieurs extensions au problème de base présenté dans le chapitre précédent. L'objectif de ces extensions est de garantir un bilan économique et écologique positif pour le véhicule électrique tout en prenant en compte des contraintes d'usage réelles et des contraintes techniques supplémentaires liées aux batteries des VEs. Il s'agit également d'étudier l'impact du déploiement des VEs sur l'organisation actuelle et sur le réseau électrique. Pour valider l'intérêt de ces extensions, plusieurs scénarios ont été testés. Ces scénarios sont structurés en deux

classes : (i) la classe des scénarios opérationnels (modulation de la recharge et optimisation de la gestion opérationnelle d'une flotte de véhicules) et (ii) la classe des scénarios tactiques (optimisation de la durée de vie des batteries des véhicules électriques, réduction des émissions de CO₂ de la flotte de véhicules, prise en compte de différents modèles de VE, optimisation de l'infrastructure de recharge, optimisation des coûts d'exploitation et optimisation de la flotte).

Pour chacun des scénarios considérés, un modèle mathématique est présenté et les résultats d'expérimentation sont détaillés.

5.2 Dépassement de la puissance souscrite

En fonction des caractéristiques du contrat d'électricité souscrit, il peut être possible de dépasser la puissance souscrite du bâtiment. Dans le cas où le dépassement de la puissance souscrite est autorisé, une pénalité horaire est appliquée à l'issue de chaque dépassement.

Le contrat d'électricité souscrit par les entreprises autorise souvent le dépassement de la puissance souscrite, c'est également le cas pour les sites d'expérimentation de La Poste et d'ERDF.

Dans cette section, nous considérons le cas où il est possible de dépasser la puissance souscrite du bâtiment sans dépasser sa puissance de raccordement. Nous souhaitons alors vérifier si le fait d'autoriser le dépassement de la puissance souscrite permettrait de mieux utiliser les véhicules électriques et de réduire les coûts d'exploitation de la flotte de véhicules.

Soit XP la pénalité appliquée à chaque dépassement de la puissance souscrite pendant un intervalle de temps t . Nous introduisons les nouvelles variables de décision suivantes :

- DQ_t : variables réelles positives désignant la quantité d'énergie en dépassement de la puissance souscrite à chaque intervalle de temps t , $t = 1, \dots, H$
- DF_t : variables binaires égales à 1 si $DQ_t > 0$ et 0 sinon, $t = 1, \dots, H$

Nous obtenons la formulation mathématique suivante :

$$\text{Lex} \quad \left(\max \sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij} ; \min \sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt} + \sum_{t=1}^H XP \times DQ_t \right) \quad (5.1)$$

S.c. :

(4.2)-(4.5)

(4.7)-(4.10)

$$\sum_{j=1}^{m_1} p_{jt} \leq g_t + (CP - SP), \quad \forall t = 1, \dots, H \quad (5.2)$$

$$\sum_{j=1}^{m_1} p_{jt} \leq g_t + DQ_t, \quad \forall t = 1, \dots, H \quad (5.3)$$

$$DQ_t \leq (CP - SP) \times DF_t, \quad \forall t = 1, \dots, H \quad (5.4)$$

Pour ce nouveau modèle linéaire, les contraintes (4.6) du chapitre précédent ont été remplacées par les contraintes (5.2), (5.3) et (5.4). Notre fonction objectif a été également modifiée pour prendre en compte les coûts supplémentaires liés au dépassement de la puissance souscrite.

5.3 Vieillessement des batteries

Comme les véhicules électriques sont constamment rechargés et déchargés, ils perdent lentement de leurs capacités de batteries, ce qui résulte en une autonomie réduite des véhicules.

Dans cette section, nous proposons d'optimiser l'usage et la recharge des véhicules électriques tout en essayant de prolonger la durée de vie des batteries. Pour cela, nous prenons en compte les recommandations proposées dans la littérature et les suggestions des constructeurs de véhicules.

Les modèles des batteries considérées, dans le cadre de cette étude, sont les modèles Zebra et Lithium-ion similaires aux batteries des véhicules Citroen Berlingo électrique et Kango ZE utilisés par le Groupe La Poste et ERDF.

L'optimisation de la durée de vie des batteries dépend, entre autres, du protocole de recharge de ces batteries. Bashash et al. [68] montrent que la batterie se dégrade plus rapidement quand son état de charge atteint des niveaux très bas ou des niveaux très élevés surtout si elle est chargée en utilisant des puissances élevées. De plus, une recharge discontinue pourrait réduire la durée de vie des batteries. Des interruptions dans le processus de recharge doivent alors être évitées ou limitées afin de protéger les batteries des VEs [92]. Par ailleurs, la capacité des batteries diminue quand le nombre de cycles recharge-décharge augmente et ceci dépend également des puissances de recharge utilisées.

Appelons "phase de recharge" un ensemble d'intervalles de temps consécutifs durant lesquels un véhicule électrique est chargé. Par exemple, si un véhicule est chargé de 11 :00 à 12 :00 et ensuite de 14 :00 à 17 :00, nous considérons dans ce cas deux phases de recharge. En d'autres mots, le nombre de phases de recharge est égal au nombre de fois la recharge d'un véhicule électrique est arrêtée et reprise durant l'horizon d'optimisation.

Pour éviter une dégradation rapide de la batterie à cause d'une recharge discontinue, deux autres contraintes de recharge sont considérées :

- La durée d'une phase de recharge doit être supérieure à une durée minimale donnée L .
- Le nombre de phases de recharge pour chaque véhicule électrique est limité.

Soit l le nombre d'intervalles de temps nécessaires pour couvrir la durée L et soit t_0 un intervalle de temps virtuel pour lequel $y_{jt_0} = 0, \forall j \in M_1$.

Nous obtenons la formulation mathématique suivante :

$$\text{Lex} \quad \left(\max \sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij} ; \min \sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt} \right) \quad (5.5)$$

S.c. :
(4.2)-(4.10)

$$\sum_{l=t}^{l=t+l} y_{jkl} \geq l \times (y_{jkt} - y_{jkt-1}), \quad \forall t = 1 \dots H, \forall j = 1 \dots m_1, \forall k \quad (5.6)$$

Les contraintes (5.6) sont ajoutées aux contraintes (4.2)-(4.10) quand une durée minimale pour chaque phase de recharge est exigée.

En intégrant la contrainte qui exige que le nombre de phases de recharge pour chaque VE doit être inférieur à Ph_{max} , nous utilisons des nouvelles variables positives f_{jt} définies par les contraintes suivantes :

$$f_{jt} \geq y_{jkt} - y_{jkt-1}, \quad \forall t = 1 \dots H, \forall j = 1 \dots m_1 \quad (5.7)$$

$$\sum_{t=1}^H f_{jt} \leq Ph_{max}, \quad \forall j = 1 \dots m_1 \quad (5.8)$$

Nous obtenons la formulation mathématique suivante :

$$\text{Lex} \quad \left(\max \sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij} ; \min \sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt} \right) \quad (5.9)$$

S.c. :

(4.2)-(4.10)

(5.7)-(5.8)

5.4 Modèle de véhicule

Les expérimentations effectuées sur deux modèles de VE montrent que ces véhicules ne peuvent pas être chargés avec des valeurs continues de puissance mais plutôt avec des puissances discrètes. En effet, chaque valeur de la puissance de recharge (ou intensité de recharge) doit appartenir à un ensemble prédéfini de valeurs possibles X . Ces valeurs dépendent de la technologie de la batterie de chaque VE. Par exemple, pour une puissance de recharge maximale de 3kVA (ce qui est équivalent à une intensité maximale de 16A), trois valeurs de puissances peuvent être utilisées pour un véhicule Kango ZE : 1.8 kW (8A), 2.7 kW (12A) et 3.3 kW (16A). Si une puissance de recharge de 2.2 kW est appliquée sur le VE pendant une période de temps, le véhicule récupère une quantité totale d'énergie égale à $1.8 \times \delta$ (kWh) plutôt que $2.2 \times \delta$ (kWh).

Dans cette section, nous prenons en compte les paliers de puissance qu'un VE peut accepter.

Soit n_{jtv} une variable binaire égale à 1 si le VE j est chargé pendant l'intervalle de temps t avec une intensité a_v et 0 sinon. Notre formulation mathématique est comme suit :

$$\sum_v n_{jtv} \leq 1, \quad \forall t = 1 \dots H, \forall j \in M_1 \quad (5.10)$$

$$\sum_v n_{jtv} = y_{jkt}, \quad \forall t = 1 \dots H, \forall j \in M_1, \forall k \quad (5.11)$$

D'autre part, la puissance de recharge maximale ne dépend pas seulement du type du chargeur mais aussi du modèle de batterie. Les expérimentations sur différents modèles de batteries montrent que les véhicules Citroen Berlingo électrique ne peuvent pas être chargés avec une puissance supérieure à 10 A. Par contre, les batteries des véhicules Kango ZE peuvent être chargées à 16 A. Dans ce cas, les contraintes (4.8) doivent être remplacées par les contraintes suivantes :

$$p_{jt} \leq \min(p_k^{\max}, p^{(\max,j)} \times y_{jt}), \quad \forall t, \forall j \in M_1 \quad (5.12)$$

où $p^{(\max,j)}$ est la capacité de recharge maximale acceptée par le VE j .

De plus, la durée minimale recommandée d'une phase de recharge L peut également dépendre de la technologie de batterie du VE. Dans ce cas, les contraintes (5.6) doivent être remplacées par les contraintes suivantes :

$$\sum_{k=t}^{k=t+l_j} y_{jk} \geq l_j \times (y_{jt} - y_{jt-1}), \quad \forall t = 1 \dots H, \forall j = 1 \dots m_1 \quad (5.13)$$

où l_j est le nombre d'intervalles de temps nécessaires pour couvrir la durée de recharge minimale recommandée pour le VE j .

5.5 Modèle d'infrastructure de recharge

Dans le cadre du métier La Poste et ERDF, tous les VEs sont chargés après les tournées sur les parkings des établissements. Nous considérons dans cette section deux types de recharge : la recharge normale qui peut délivrer une puissance maximale de 3 kW et la recharge accélérée délivrant une puissance maximale de 22 kW.

Étant donné un ensemble de tournées et une flotte de véhicules électriques, nous proposons, dans cette section, d'optimiser l'infrastructure de recharge.

Notre objectif est de déterminer le nombre optimal de bornes de recharge normale et de bornes de recharge accélérée permettant un usage maximal des véhicules électriques.

Pour cela, comme dans le chapitre précédent, nous utilisons un modèle linéaire en variables mixtes en deux étapes en utilisant CPLEX. Dans la première étape, l'objectif est de maximiser la distance totale parcourue par les VEs. Dans la deuxième étape, l'objectif est de minimiser le coût de la recharge et le coût de déploiement de l'infrastructure de recharge.

Nous introduisons les variables de décision suivantes :

x_{ij} : variable binaire égale à 1 si le véhicule j , $j = 1, \dots, m_1 + m_2$, est affecté à la tournée i , $i = 1, \dots, n$, et 0 sinon.

y_{jtk} : variable binaire égale à 1 si VE j , $j = 1, \dots, m_1$, est chargé pendant l'intervalle de temps $t = 1, \dots, H$ en utilisant une borne de recharge de type $k = 1 \dots, K_{max}$, et 0 sinon.

p_{jtk} : variable réelle qui désigne la puissance de recharge appliquée au véhicule VE j , $j = 1, \dots, m_1$ pendant la période de temps $t = 1, \dots, H$ en utilisant une borne de recharge de type $k = 1, \dots, K_{max}$.

b_k : variables réelles positives représentant le nombre de bornes de recharge de type k à déployer, $\forall k = 1, \dots, K_{max}$.

Nous obtenons la formulation mathématique suivante :

$$\text{Lex} \quad \left(\max \sum_{i=1}^n \sum_{j=1}^{m_1} w_i \times x_{ij} ; \min \sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt} + \sum_{k=1}^K DC_k \times b_k \right) \quad (5.15)$$

$$\sum_{j=1}^{m_1+m_2} x_{ij} = 1, \quad \forall i \quad (5.16)$$

$$x_{ij} + \sum_{i' \in V(i)} x_{i'j} \leq 1, \quad \forall i, \forall j \quad (5.17)$$

$$\sum_{t=s_i}^{f_i} y_{jtk} + (f_i - s_i + 1) \times x_{ij} \leq (f_i - s_i + 1), \quad \forall i, \forall j \in M_1, \forall k \quad (5.18)$$

$$\sum_{j=1}^{m_1} p_{jtk} \leq g_t, \quad \forall t, \forall k \quad (5.19)$$

$$p_k^{\min} \times y_{jtk} \leq p_{jtk}, \quad \forall t, \forall j \in M_1, \forall k \quad (5.20)$$

$$p_{jtk} \leq p_k^{\max} \times y_{jtk}, \quad \forall t, \forall j \in M_1, \forall k \quad (5.21)$$

$$SoC_j^0 + \frac{\sum_{t \leq s_i-1} \delta \times p_{jtk} - \sum_{l=1/f_l \leq s_i-1} E_{l,f_l} \times x_{lj}}{CE_j} \leq SoC_j^{\max}, \quad \forall i, \forall j \in M_1, \forall k \quad (5.22)$$

$$SoC_j^0 + \frac{\sum_{t \leq f_i} \delta \times p_{jtk} - \sum_{l=1/f_l \leq f_i} E_{l,f_l} \times x_{lj}}{CE_j} \geq SoC_j^{\min}, \quad \forall i, \forall j \in M_1, \forall k \quad (5.23)$$

$$\sum_{j=1}^{m_1} y_{jtk} \leq b_k, \quad \forall t, \forall k \quad (5.24)$$

$$\sum_{k=1}^{K_{\max}} b_k \leq m_1, \quad (5.25)$$

Les contraintes (5.16)-(5.23) sont similaires aux contraintes (4.2)-(4.10) mais elles prennent en compte différents types de bornes de recharge. Les contraintes (5.24) assurent qu'au plus b_k chargeurs de type k sont utilisés à chaque intervalle de temps t . Les contraintes (5.25) garantissent qu'au plus m_1 chargeurs sont utilisés.

5.6 Bilan carbone

En France, le secteur des transports est le principal émetteur de gaz à effet de serre. Si la phase de construction du véhicule contribue à la part des émissions CO₂ du secteur de transport, c'est la phase d'usage du véhicule qui est le principal contributeur à la dégradation de la qualité de l'air par les émissions de CO₂ et les particules fines.

Un véhicule électrique roulant en France émet, en moyenne, 12 g de CO₂ par kilomètre parcouru, contre une moyenne de 117 g de CO₂ au kilomètre pour le parc français actuel ⁹. Pour les gestionnaires de flotte, le bilan environnemental peut être un motif de choix du véhicule électrique. Dans cette section, nous proposons alors d'étudier le bilan carbone d'une flotte de véhicules électriques et thermiques et de minimiser les émissions en polluants de cette flotte.

Pour pouvoir comparer le bilan carbone d'un VE à celui d'un VT, nous adoptons une analyse "du puits à la roue" sur la phase d'usage pour évaluer le bilan écologique des VEs et des VTs et pour minimiser les émissions totales en CO₂ d'une flotte mixte de véhicules. Cette analyse consiste à mesurer les émissions en CO₂ d'un véhicule dès l'extraction des combustibles jusqu'à leur utilisation.

Même si le VE n'émet pas de gaz à effet de serre lors de la phase de roulage, la production de l'électricité utilisée pour recharger les batteries des VEs peut être une source importante d'émissions CO₂. L'incidence écologique des VEs dépend alors de la façon dont est produite l'électricité utilisée pour charger leur batterie. En effet, si la recharge des VEs s'effectue pendant des horaires de pic de consommation, la production d'électricité peut s'effectuer à partir des centrales thermiques qui génèrent des grandes quantités de polluants. Les horaires de recharge ont alors un impact significatif

⁹Rapport au nom de la commission des affaires économiques de Frédérique Massat, 15 avril 2014.

sur les quantités de gaz à effet de serre émises. Pour cette raison, le bilan environnemental lié à la phase d'usage du véhicule électrique dépend fortement des sources de production d'électricité et du mix électrique de chaque pays. Par exemple, le mix électrique français, lié au nucléaire et peu carboné, est favorable à l'usage du véhicule électrique.

Concernant les véhicules thermiques, la combustion du carburant pendant l'usage du véhicule est une source principale des émissions CO₂.

La quantité de CO₂ émise pendant la phase d'usage des VEs est calculée en multipliant la quantité d'énergie utilisée pour charger les VEs par le facteur d'émission (quantité de CO₂ émise par unité d'énergie produite exprimée en g_{éq}CO₂/kWh).

Le choix du type de facteur d'émission dépend du système à analyser. Il existe plusieurs façons de comptabiliser les émissions de gaz à effet de serre engendrées par l'utilisation de l'électricité pour la recharge des véhicules électriques. Les facteurs d'émissions, selon les périmètres considérés, peuvent varier de 30 à 930 g_{éq}CO₂/kWh.

Pour comparer les émissions entre le véhicule électrique et le véhicule thermique, un facteur d'émissions moyen annuel a été retenu tenant compte des émissions de GES du puits à la roue, et des imports et exports d'électricité. Pour comparer les différentes politiques de recharge journalières et hebdomadaires entre elles, un facteur d'émissions horaire sur la base des données RTE a été retenu.

La quantité de CO₂ émise pendant la phase d'usage d'un VT est calculée en multipliant le nombre de kilomètres parcourus par le facteur d'émission de gaz à effet de serre (émissions de CO₂ en gramme par litre) et par la consommation du VT en gasoil. Tous les VTs fonctionnant avec le même type de gasoil, ont le même facteur d'émission de gaz à effet de serre.

Nous obtenons le modèle mathématique suivant :

$$\text{Min} \sum_{j=1}^{m_1} \sum_{t=1}^H CO_t^{\text{VE}} \times \left(\frac{\delta}{3600}\right) \times p_{jt} + \sum_{i=1}^n \sum_{j=m_1+1}^{m_2} \text{Cons}_j^{\text{VT}} \times w_i \times CO_t^{\text{VT}} \quad (5.26)$$

S.c. :

(4.2)-(4.10)

L'objectif (5.26) consiste à minimiser la quantité de CO₂ émise pendant la phase de recharge des VEs et lors de l'usage des VTs et il est soumis aux contraintes (4.2)-(4.10).

5.7 Optimisation de la flotte de véhicules

Étant donné un ensemble de tournées et une infrastructure de recharge, nous proposons, dans cette section, d'optimiser la flotte de véhicules en minimisant son coût total de possession (TCO) qui définit le coût total d'un véhicule dès son achat jusqu'à sa revente passant par la phase d'exploitation et de maintenance.

L'objectif est de déterminer le nombre optimal de VEs et de VTs nécessaires pour effectuer toutes les tournées. Le TCO est composé d'un coût fixe et d'un coût variable. Le coût fixe représente les coûts indépendants des usages et du kilométrage réalisé mais pouvant être payés sur une durée plus ou moins longue. Le TCO d'un VE englobe le coût d'achat ou de location du véhicule, de batterie, d'infrastructure de recharge (un point de charge par véhicule électrique), du contrat d'énergie et de fiscalité.

Le coût variable représente les coûts liés à l'utilisation du véhicule et qui sont proportionnels au kilométrage réalisé. Il englobe le coût lié à la consommation énergétique du véhicule (diesel ou électricité) et d'autres coûts liés à l'entretien/maintenance du véhicule, assurance, sinistralité, etc.

Pour calculer le coût d'exploitation par kilomètre d'un VE, le coût d'électricité (euros/km) et la consommation du véhicule (quantité d'énergie nécessaire pour effectuer 100 km) doivent être connus.

Le coût de la consommation énergétique pour la phase d'exploitation d'un véhicule doit être calculé en multipliant la consommation énergétique par kilomètre par le coût par unité d'énergie. Par exemple, si le coût d'électricité est 0.11 euros le kWh et le véhicule consomme 11kWh pour parcourir 100 km, le coût d'un kilomètre est de 0.01 euros.

Pour calculer le coût d'exploitation par kilomètre d'un VT, le coût de gasoil et la quantité de carburant nécessaire pour parcourir 100 kilomètres doivent être connus.

Pour optimiser la flotte de véhicules, nous utilisons les variables de décision suivantes :

UVE_j : variable binaire qui est égale à 1 si le VE $j = 1, \dots, m_1$ est utilisé pour effectuer au moins une tournée et 0 sinon.

UVT_j : variable binaire qui est égale à 1 si le VT $j = m_1 + 1, \dots, m_2$ est utilisé pour effectuer au moins une tournée et 0 sinon.

Notre formulation mathématique est comme suit :

$$\begin{aligned} \text{Min } & \sum_{j=1}^{m_1} CF_j^{\text{VE}} \times UVE_j + \sum_{j=m_1+1}^{m_2} CF_j^{\text{VT}} \times UVT_j + \sum_{i=1}^n \sum_{j=1}^{m_1} CV_j^{\text{VE}} \frac{w_i}{100} \times \\ & x_{ij} + \sum_{i=1}^n \sum_{j=m_1+1}^{m_2} CV_j^{\text{VT}} \frac{w_i}{100} \times x_{ij} + \sum_{j=1}^{m_1} \sum_{t=1}^H \delta \times c_t \times p_{jt} \end{aligned} \quad (5.27)$$

S.c. :

(4.2)-(4.10)

$$\sum_{i=1}^{i=n} x_{ij} \leq n \times (UVE_j), \forall j = 1, \dots, m_1 \quad \sum_{i=1}^{i=n} x_{ij} \leq n \times (UVT_j), \forall j = m_1 + 1, \dots, m_2 \quad (5.28)$$

5.8 Cas d'étude et expérimentations

5.8.1 Protocole des expérimentations simulées

L'objectif de ces expérimentations, menées sur des instances du Groupe La Poste, est de comprendre l'influence de l'introduction des véhicules électriques sur le métier de distribution du courrier et de définir les modifications à apporter sur l'organisation actuelle du métier postal pour tirer un maximum d'avantages de l'utilisation des véhicules électriques. Il s'agit également de mesurer la variation dans les coûts d'exploitation et les émissions de gaz à effet de serre liés à la flotte de véhicules. Pour atteindre ces objectifs, un ensemble de scénarios a été mis en place. La section suivante donne la liste des scénarios simulés et donne une description rapide de chaque scénario.

a. Description des scénarios simulés

Les scénarios simulés sont structurés en deux classes : (i) la classe des scénarios opérationnels (scénario de référence, scénario de modulation de la recharge, scénario optimisation de l'affectation et de la recharge, scénario tournées sensibles) et (ii) la classe des scénarios tactiques (scénario vieillissement de la batterie, scénario optimisation carbone, scénario optimisation des coûts d'exploitation).

Scénario de référence : Ce scénario de référence reprend l'organisation actuelle de l'activité postale de chacun des deux sites d'expérimentation. La recharge des véhicules électriques se fait

la nuit à partir du début de la période des heures creuses. Ce scénario joue le rôle d'un point de référence par rapport auquel les autres scénarios seront comparés.

Scénario modulation de la recharge : Pour ce scénario, aucune optimisation de l'affectation des tournées aux véhicules n'est introduite. En se basant sur l'organisation actuelle, il s'agit d'optimiser la recharge des véhicules électriques.

Scénario optimisation de l'affectation et la recharge : Pour ce scénario, il s'agit d'optimiser simultanément l'affectation des tournées et la recharge des véhicules électriques. L'objectif de l'optimisation est de maximiser l'usage des véhicules électriques et de minimiser leur coût total de recharge.

Scénario vieillissement de la batterie : Pour ce scénario, nous étudions l'impact de la prise en compte des facteurs de vieillissement de la batterie sur l'organisation postale à travers la restriction de l'amplitude de l'intervalle de recharge. Nous étudions également les conséquences de cette restriction sur le coût d'exploitation des véhicules électriques.

Scénario optimisation carbone : Dans ce scénario, nous optimisons à la fois l'affectation des tournées aux véhicules et le planning de recharge des véhicules électriques. L'objectif est de minimiser les quantités de CO₂ émises lors de l'exploitation des véhicules thermiques et lors de la recharge des véhicules électriques.

Scénario optimisation des coûts d'exploitation : Dans ce scénario, nous optimisons l'affectation des tournées aux véhicules et le planning de recharge des véhicules électriques. L'objectif est d'optimiser les coûts induits par l'exploitation des véhicules thermiques et électriques.

Scénario optimisation de la flotte de véhicules : Dans ce scénario, nous optimisons la flotte de véhicules à déployer avec comme objectif de minimiser le coût total de possession de la flotte.

b. Description des instances

Les expérimentations ont été effectuées sur les instances fournies par deux sites d'expérimentation du Groupe La Poste. Il s'agit des mêmes instances présentées dans le chapitre précédent.

c. Liste des indicateurs mesurés

Une liste des indicateurs à mesurer est élaborée pour comparer les différents scénarios entre eux. Cette liste est présentée dans le tableau ci-dessous.

Pour des raisons de confidentialité, nous avons préféré ne pas présenter les résultats liés aux coûts d'exploitation de la flotte. Nous donnons, toutefois, quelques remarques liées à la variation de ces coûts en fonction des scénarios considérés.

TAB. 5.1: Liste des indicateurs mesurés

Paramètre	Description
VE	Nombre de véhicules électriques utilisés
VT	Nombre de véhicules thermique utilisés
KM VE	Distance totale parcourue par les véhicules électriques
KM VT	Distance totale parcourue par les véhicules thermiques
Énergie consommée VE	Énergie totale consommée par les véhicules électriques (KWh)
CO ₂ VE	Quantité de CO ₂ émise par les VEs exprimée (g eq. CO ₂) (quantité dûe à la phase de recharge de ces véhicules)
Pic de consommation	Puissance maximale appelée lors de la recharge des VEs (KVA). Cette puissance ne tient pas compte du dépassement dû à la consommation du bâtiment

5.8.2 SC 1 : Scénario de référence

a. Description du scénario

Pour ce scénario de référence, aucune optimisation n'est effectuée. L'objectif de ce scénario est de construire un cas de référence par rapport auquel d'autres scénarios peuvent être comparés. L'engagement des véhicules est fourni par l'organisation actuelle sur les sites d'expérimentation. Tous les véhicules électriques sont chargés sans aucune optimisation et en suivant les règles suivantes :

- Recharge en parallèle de l'ensemble des véhicules à 100%
- Recharge de chaque véhicule avec l'intensité maximale de 16A
- Recharge à partir du début de la période "heures-creuses" le soir
- Une seule phase de recharge pour chaque véhicule électrique est autorisée

b. Résultats

Le tableau 5.2 présente les résultats obtenus pour le scénario de référence.

TAB. 5.2: Résultats du scénario de base

Indicateurs	Instance 1	Instance 2
VE	8	12
VT	18	14
KM VE	380	562
KM VT	1020	516
Énergie consommée	94.36	72.88
Coût de recharge	4.28	5.72
CO ₂ VT	287.77	145.58
CO ₂ VE	7.93	10.23
Énergie en dépassement	0	54.6
Coût dépassement	0	33.67

c. Analyse des résultats

Pour l'instance 1, au total 1400 kilomètres sont effectués par la flotte de véhicules chaque jour. Une part de 380 kilomètres est réalisée par les VEs, soit un pourcentage d'environ 27%. Pour l'instance 2, 562 kilomètres (52%) sur un total de 1078 kilomètres sont réalisés par les VEs. En moyenne, chaque VE effectue 47.5 et 46.83 kilomètres par jour pour les instances 1 et 2, respectivement. Ces moyennes restent inférieures au seuil d'utilisation souhaitée des VEs.

5.8.3 SC 2 : Modulation de la puissance de recharge

a. Description du scénario

Pour ce scénario, une optimisation du planning de recharge est introduite. Vu l'utilisation actuelle des VEs au sein des sites d'expérimentation, l'objectif est de minimiser les coûts de recharge en se basant sur une modulation de la puissance de recharge. L'intensité de recharge doit appartenir à l'ensemble des valeurs possibles suivant : $X = \{8 \text{ A}, 12 \text{ A}, 16 \text{ A}\}$. L'objectif de ce scénario est d'évaluer l'amélioration au niveau du coût total de recharge des VEs, obtenue grâce à la modulation de la puissance de recharge. Les règles à suivre pour optimiser la recharge sont :

- Recharge en parallèle de l'ensemble des véhicules à 100%
- Les véhicules peuvent être chargés par des puissances de 16A, 12A ou 8A
- Recharge à partir du début de la période 'heures creuses' le soir
- Une seule phase de recharge pour chaque véhicule électrique est autorisée

b. Variantes du scénario

Deux variantes de ce scénario sont simulées :

- **SC 2.1. "Avec recharge méridienne"** : La recharge méridienne consiste à recharger les véhicules qui rentrent sur le site d'expérimentation aux alentours de midi avant de repartir pour d'autres prestations l'après-midi. Cela peut améliorer la rentabilité et diminuer les coûts d'exploitation de la totalité de la flotte. Pour cette variante, nous considérons que la recharge méridienne est autorisée.
- **SC 2.2. "Sans recharge méridienne"** : Pour cette variante, nous considérons que la recharge méridienne est interdite.

c. Résultats

Le tableau 5.3 résume les résultats obtenus pour le scénario de modulation de la puissance de recharge.

d. Analyse des résultats

À partir des résultats de ce scénario, nous pouvons tirer les informations suivantes :

- L'optimisation du coût de la recharge permet de diminuer ce coût de 1% pour l'instance 1 et de 3% pour l'instance 2 par rapport au scénario de référence.
- L'interdiction de la recharge méridienne pour l'instance 1 ne permet pas d'avoir une solution de recharge par rapport à l'engagement prévu des VE. En effet, un VE est affecté à deux tournées dont le besoin en énergie est de 18.85 kWh, ce qui correspond à 90% de la capacité de la batterie d'un VE, avec un seuil retour de 20%, ces deux tournées ne sont donc pas réalisables.

TAB. 5.3: Résultats du scénario de modulation de la puissance de recharge

Indicateur	Instance 1		Instance 2	
	SC 2.1	SC 2.2	SC 2.1	SC 2.2
VE	8	Pas de solution	12	12
VT	18		14	14
KM VE	380		562	562
KM VT	1020		516	516
Énergie consommée	94.36		72.88	72.88
Coût de recharge	4.23		5.7	5.7
CO ₂ VT	287.7		145.58	145.58
CO ₂ VE (Kg. eq. CO2)	7.85		10.42	10.42
Énergie en dépassement	0		0	0
Coût dépassement	0		0	0

- Pour l’instance 2, le besoin en énergie des tournées réalisables par les VE est faible, ainsi les VE, chargés à 100% la nuit, n’ont pas besoin d’être chargés à nouveau à la méridienne.
- Une diminution minimale de la quantité de CO₂ émise (-1% pour l’instance 1 et -1% pour l’instance 2) est observée.

5.8.4 SC 3 : Optimisation de l’affectation et de la recharge des véhicules

a. Description du scénario

L’objectif de ce scénario est de maximiser la distance totale parcourue par les VEs et de minimiser les coûts de la recharge tout en vérifiant les contraintes (4.2)-(4.10). Au plus m_1 bornes de recharge normale ($X=\{8 A, 12 A, 16 A\}$) peuvent être utilisées. La valeur du SoC minimal accepté est fixée à 20% et la valeur du SoC maximal accepté est fixée à 95%. Initialement, l’état de charge des batterie est égal à 0 (SoC=0). L’optimisation de la recharge consiste à autoriser la modulation de la puissance de recharge des véhicules et de ne charger que la quantité d’énergie nécessaire pour effectuer les tournées. L’objectif est de minimiser le coût total de la recharge des véhicules.

b. Variantes du scénario

Pour ce scénario, nous distinguons les trois variantes suivantes :

- **SC 3.1. Sans recharge méridienne** : Pour cette variante, la recharge méridienne est interdite.
- **SC 3.2 : Avec recharge méridienne** : Pour cette variante, la recharge méridienne est autorisée.
- **SC 3.3 : Avec recharge méridienne et borne de recharge accélérée** : Pour cette variante, la recharge méridienne est autorisée. De plus, une borne de recharge accélérée est disponible. Cette borne ne peut être utilisée qu’entre 12h et 14h.

c. Résultats

Le tableau 5.4 résume les résultats du scénario de l’optimisation de l’affectation et de la recharge des véhicules

TAB. 5.4: Résultats du scénario de l'optimisation de l'affectation et de la recharge des véhicules

Indicateur	Instance 1			Instance 2		
	SC 3.1	SC 3.2	SC 3.3	SC 3.1	SC 3.2	SC 3.3
VE	8	8	8	12	12	12
VT	19	17	17	12	12	12
KM VE	614.88	649.74	667.59	685	685	685
KM VT	802.19	767.32	749.47	393	393	393
Énergie consommée	133.78	152.07	162.85	90.04	90.04	90.04
Coût de recharge	4.13	4.84	5.27	2.73	2.73	2.73
CO ₂ VT	226.32	216.48	211.54	110.87	121.59	110.87
CO ₂ VE	7.63	8.87	9.61	4.98	5	4.98
Énergie en dépassement	0	0	0.01	0	0	0
Coût dépassement	0	0	0	0	0	0

d. Analyse des résultats

Pour l'instance 1, nous remarquons que :

- l'optimisation globale (affectation et recharge) permet d'augmenter le nombre de kilomètres effectués par les VE de 92%, l'autorisation de la recharge méridienne permet d'augmenter ce nombre de 97% et l'utilisation de la borne accélérée n'apporte pas de gain supplémentaire ;
- la quantité de CO₂ émise par les véhicules thermiques diminue de 35% et celle émise par les VEs diminue de 4% ;
- le coût d'exploitation de la flotte diminue de 33%.

Pour l'instance 2, nous pouvons tirer les remarques suivantes :

- L'optimisation globale (affectation et recharge) permet d'augmenter le nombre de kilomètres effectués par les VEs de 15%. De plus, l'autorisation de la recharge méridienne ne permet pas d'améliorer la valeur de la fonction objectif, ceci s'explique par la faible consommation d'énergie des tournées affectées aux VEs. L'utilisation de la borne de recharge accélérée permet d'augmenter le nombre de kilomètres de 21%. Le faible pourcentage d'augmentation du nombre de kilomètres pour cette instance est dû au fait que le nombre de tournées réalisables par les VE soit réduit.
- La quantité de CO₂ émise par les véhicules thermiques diminue de 16% et celle émise par les véhicules électriques diminue de 55%.
- Le coût d'exploitation de la flotte diminue de 9%.

5.8.5 SC 4 : Vieillessement de la batterie

a. Description du scénario

Dans ce scénario, nous optimisons l'affectation des véhicules aux tournées ainsi que le planning de recharge des véhicules électriques tout en évitant la dégradation de la capacité de la batterie. Dans la littérature scientifique, le vieillissement de la batterie s'accélère si une décharge profonde ou une recharge à 100% de la batterie est effectuée systématiquement. Dans ce scénario, nous testons plusieurs variantes avec des intervalles de seuil min et de seuil max différents. Nous varions alors l'intervalle de SoC [SoC_j^{\min} , SoC_j^{\max}] et nous comparons le nombre de kilomètres parcourus par les

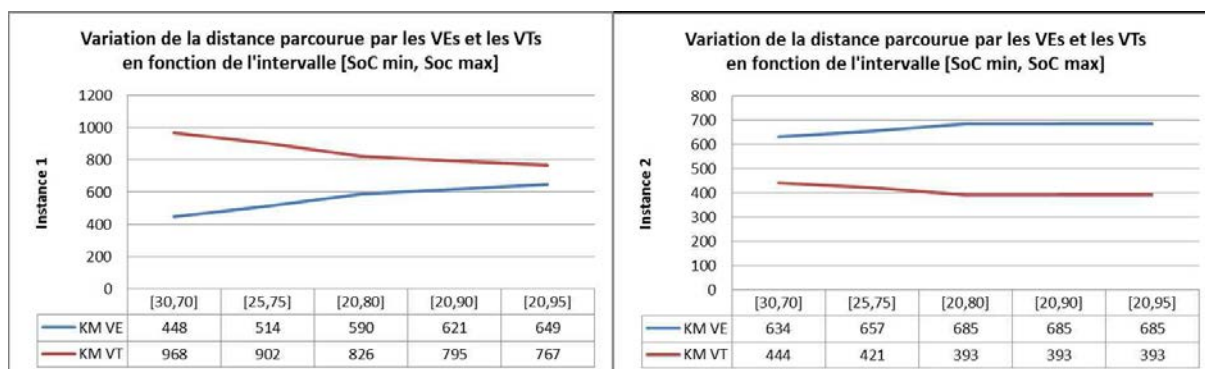


FIG. 5.1: Variation de la distance parcourue par les VEs et les VTs en fonction de l'intervalle $[SoC_j^{\min}, SoC_j^{\max}]$.

VEs. Cinq intervalles sont étudiés : [30 , 70] (SC 4.1), [25 , 75] (SC 4.2), [20 , 80] (SC 4.3), [20 , 90] (SC 4.4), [20 , 95] (SC 4.5).

b. Résultats

Les courbes de la figure 5.1 illustrent la variation de la distance parcourue par les VEs et les VTs en fonction de l'intervalle $[SoC_j^{\min}, SoC_j^{\max}]$ lié aux instances 1 et 2. Nous remarquons ainsi que la distance parcourue par les VEs augmente en fonction de l'amplitude de l'intervalle du SoC.

c. Analyse des résultats

Pour l'instance 1, nous pouvons remarquer les faits suivants :

- Les résultats des scénarios SC.4.1-SC.4.5, montrent une augmentation du nombre de kilomètres faits par les VEs de 10% à chaque augmentation de l'amplitude de l'intervalle de recharge de 10%, sauf pour le SC.4.4 où l'amplitude augmente de 10% alors que le nombre de kilomètres n'augmente que de 4%.
- La variation de la quantité totale de CO₂ émise par la flotte est assez faible, une différence de seulement 7% est constatée entre SC.4.1. et SC.4.5. Nous constatons la même chose par

rapport au coût d'exploitation de la flotte, où la variation est de seulement 10% entre le SC.4.1 et le SC.4.5.

- L'intervalle de recharge [20,80] (SC.4.3) semble être un bon compris.

Pour l'instance 2, nous constatons que la variation du nombre de kilomètres effectués par les VEs est de 10% entre le SC.4.1 et le SC.4.5, cela est dû à la faible consommation d'énergie des tournées effectuées par les VEs. Nous constatons la même chose pour la quantité totale de CO₂ émise par la flotte et le coût total d'exploitation de la flotte. Ainsi, pour cette instance, il est tout à fait possible de charger les batteries dans l'intervalle [30,70].

5.8.6 SC 5 : Optimisation du bilan carbone

a. Description du scénario

Dans ce scénario, nous optimisons l'affectation des véhicules aux tournées ainsi que le planning de recharge des véhicules électriques. L'objectif est de minimiser les émissions de CO₂ engendrées par l'utilisation des véhicules thermiques dans la phase de roulage et les émissions du CO₂ durant la phase de recharge.

Pour les expérimentations, nous avons utilisé le mix électrique français pour calculer les émissions en CO₂ des VEs : nucléaire : 74.5%, hydro-électrique : 16.3%, thermique : 9.1%, source d'énergie renouvelable : 0.1%. Toutes les données qui concernent les quantités de CO₂ émises lors de la production d'électricité en France, par kWh produit, sont fournies par le Réseau de Transport d'Électricité RTE.

En utilisant les données collectées toutes les 15 minutes sur une période de 830 jours (du 01/01/2011 au 09/03/2013), cinq profils d'émissions CO₂ ont été identifiés : des émissions très élevées en hiver (novembre, février et mars essentiellement) avec une petite variation au cours de journée, des émissions élevées, des émissions moyennes, des émissions faibles et des émissions très faibles qui correspondent clairement à la période de l'été et des weekends avec une variation assez élevée au cours de la journée.

Ce scénario est composé de quatre variantes décrites dans le tableau 5.5

TAB. 5.5: Variantes du scénario d'optimisation du bilan carbone

Variantes	Hypothèses
SC 5.1	L'estimation des consommations des tournées et des émissions CO ₂ est basée sur des données enregistrées en été, recharge méridienne autorisée, recharge normale
SC 5.2	L'estimation des consommations des tournées et des émissions CO ₂ est basée sur des données enregistrées en été, recharge méridienne autorisée, recharge normale et recharge accélérée
SC 5.3	L'estimation des consommations des tournées et des émissions CO ₂ est basée sur des données enregistrées en hiver, recharge méridienne autorisée, recharge normale
SC 5.4	L'estimation des consommations des tournées et des émissions CO ₂ est basée sur des données enregistrées en hiver, recharge méridienne autorisée, recharge normale et recharge accélérée

b. Résultats

Les tableaux 5.6 et 5.7 présentent les résultats du scénario d'optimisation du bilan carbone.

TAB. 5.6: Résultats de l'optimisation du bilan carbone pour l'instance 1

Indicateur	SC 5.1	SC 5.2	SC 5.3	SC 5.4
VE	8	8	8	8
VT	18	18	19	18
KM VE	649.74	667.59	542.11	551.12
KM VT	767.32	749.47	874.96	865.95
Énergie consommée	152.07	162.85	159.24	162.48
Coût de recharge	5.1	5.36	11.22	11.44
CO ₂ VT	216.48	211.45	246.8	244.31
CO ₂ VE	8.57	9.02	12.21	12.44
Énergie en dépassement	138.52	115.34	129.02	105.55
Coût dépassement	80.82	90.92	84.18	90.92

TAB. 5.7: Résultats de l'optimisation du bilan carbone pour l'instance 2

Indicateur	SC 5.1	SC 5.2	SC 5.3	SC 5.4
VE	12	12	12	12
VT	12	12	12	12
KM VE	685	685	685	685
KM VT	393	393	393	393
Énergie consommée	90.04	90.04	140.03	140.03
Coût de recharge	2.73	2.73	9.86	9.22
CO ₂ VT	110.87	110.87	110.87	110.87
CO ₂ VE	5.03	4.54	10.89	10.17
Énergie en dépassement	0	0	84.95	76.14
Coût dépassement	0	0	53.88	53.88

c. Analyse des résultats

Les résultats des tableaux 5.6 et 5.7 montrent une forte corrélation entre la maximisation de l'usage des VEs et la minimisation des émissions CO₂. La quantité de CO₂ émise par les VTs n'a pas changé par comparaison au scénario 3 étant donné que cette quantité est directement liée au nombre de kilomètres parcourus par les VTs. En revanche, nous remarquons une légère diminution d'environ 5% dans la quantité de CO₂ émise lors de la phase de recharge des VEs.

Nous remarquons également une forte augmentation dans le coût de recharge (5%) et surtout le coût de dépassement qui passe de 0 € à plus que 80 € pour l'instance 1.

Concernant l'impact des conditions climatiques sur les résultats obtenus, nous remarquons :

- une diminution d'environ 20% dans le nombre de kilomètres effectués par les VEs en hiver ainsi qu'une augmentation dans le coût de recharge et la quantité de CO₂ émise pour l'instance 1 ;
- une augmentation dans l'énergie consommée, le coût de recharge et le coût de dépassement pour l'instance 2 même si la distance parcourue par les VEs n'a pas changé.

5.8.7 SC 6 : Optimisation des coûts d'exploitation

a. Description du scénario

Pour ce scénario, nous optimisons l'affectation des véhicules aux tournées ainsi que le planning de recharge des véhicules électriques. L'objectif est de minimiser les coûts d'exploitation des véhicules

thermiques et électriques. Pour ce scénario, nous varions les paramètres suivants :

- Été/Hiver : la courbe RTE prise en compte est calculée en se basant sur la moyenne des deux dernières années pendant la période estivale ou hivernale.
- Recharge méridienne : la recharge méridienne peut être interdite ou autorisée.
- Prix du gasoil : le prix du gasoil est fixé soit à 1.5 euros/litre soit à 2.5 euros/litre.

En variant les paramètres ci-dessus, nous obtenons les variantes suivantes :

TAB. 5.8: Variantes du scénario d'optimisation des coûts d'exploitation de la flotte

Variante	Hypothèses
SC 6.1	Été, recharge méridienne interdite, gasoil à 1.5 euros/L, recharge normale
SC 6.2	Été, recharge méridienne interdite, gasoil à 2.5 euros/L, recharge normale
SC 6.3	Été, recharge méridienne autorisée, gasoil à 1.5 euros/L, recharge normale et accélérée
SC 6.4	Hiver, recharge méridienne autorisée, gasoil à 1.5 euros/L, recharge normale et accélérée

b. Résultats

Les tableaux 5.9 et 5.10 présentent les résultats des scénarios d'optimisation des coûts d'exploitation et d'optimisation de la flotte de véhicules.

TAB. 5.9: Résultats du scénario d'optimisation des coûts d'exploitation pour l'instance 1.

Indicateur	SC 6.1	SC 6.2	SC 6.3	SC 6.4
VE	8	8	8	8
VT	19	19	19	19
KM VE	610.25	614.88	665.62	555.95
KM VT	806.82	802.19	751.45	861.12
Énergie consommée	138.84	133.78	167.78	168.25
Coût de recharge	5.39	4.81	6.59	13.83
CO ₂ VT	227.62	226.32	212	242.94
CO ₂ VE	9.04	8.06	11.22	14.48
Énergie en dépassement	73.72	91.31	41.39	66.57
Coût dépassement	67.35	84.18	94.29	87.55

c. Analyse des résultats

Pour l'instance 1, les résultats des deux premiers scénarios montrent qu'en variant le coût du gasoil, le nombre de kilomètres faits par les VEs augmente légèrement. De plus, en se basant sur les résultats des scénarios 3 et 4, nous constatons qu'en hiver la consommation énergétique des VEs augmente considérablement, ce qui se traduit par une forte baisse dans la distance totale parcourue par les VEs. L'instance 2 semble moins sensible aux conditions climatiques et à la variation du coût du gasoil.

TAB. 5.10: résultats du scénario d'optimisation des coûts d'exploitation pour l'instance 2

Indicateur	SC 6.1	SC 6.2	SC 6.3	SC 6.4
VE	12	12	12	12
VT	12	12	12	12
KM VE	685	685	685	685
KM VT	393	393	393	393
Énergie consommée	90.04	90.04	90.04	140.03
Coût de recharge	5.14	5.14	4.49	15.51
CO ₂ VT	110.87	110.87	110.87	110.87
CO ₂ VE	8.43	8.43	7.21	15.67
Énergie en dépassement	4.6	4.6	0	0
Coût dépassement	26.94	26.94	0	0

5.8.8 SC 7 : Optimisation de la flotte

a. Description du scénario

Pour ce scénario, l'objectif est de dimensionner le parc de véhicules. Étant donné un ensemble de tournées et une infrastructure de recharge, il s'agit de déterminer le nombre de VE et de VT à déployer afin de minimiser le coût total de possession du parc de véhicules. En plus des paramètres du scénario 6, nous varions également, pour ce scénario, le coût d'achat du véhicule électrique. Nous étudions alors les variantes présentées dans le tableau 5.11.

TAB. 5.11: Variantes du scénario d'optimisation de la flotte

Variante	Hypothèses
SC 7.1	Été, recharge méridienne autorisée, gasoil à 1.5 euros/L, recharge normale
SC 7.2	Été, recharge méridienne autorisée, gasoil à 1.5 euros/L, recharge normale, VE à -40%
SC 7.3	Été, recharge méridienne autorisée, gasoil à 2.5 euros/L, recharge normale, VE à -40%
SC 7.4	Été, recharge méridienne autorisée, gasoil à 1.5 euros/L, recharge normale et accélérée
SC 7.5	Hiver, recharge méridienne autorisée, gasoil à 1.5 euros/L, recharge normale et accélérée

b. Résultats

Les tableaux 5.12 et 5.13 présentent les résultats obtenus pour les différentes variantes du scénario d'optimisation de la flotte.

c. Analyse des résultats

D'après les résultats obtenus, nous pouvons faire les remarques suivantes :

- Pour les scénarios 6.1, 6.2, 6.4 et 6.5, seulement 3 VEs sont utilisés pour l'instance 1. En fixant le coût du gasoil à 2.5 €/L et en diminuant de 40% le coût d'achat des VEs, le nombre de VEs utilisés atteint 7.

TAB. 5.12: Résultats du scénario d'optimisation de la flotte pour l'instance 1.

Indicateur	SC 6.1	SC 6.2	SC 6.3	SC 6.4	SC 6.5
VE	3	3	7	3	3
VT	19	19	15	19	19
KM VE	260.68	260.68	549.60	265.36	229.44
KM VT	1156.39	1156.39	867.47	1151.70	1187.63
Énergie consommée	54.26	54.26	120.96	66.57	66.64
Coût de recharge	1.67	1.67	3.74	2.18	4.75
CO ₂ VT	326.25	326.25	244.73	324.93	335.06
CO ₂ VE	3.08	3.08	6.88	3.97	5.29
Énergie en dépassement	0	0	0	0	0
Coût dépassement	0	0	0	0	0

TAB. 5.13: Résultats du scénario d'optimisation de la flotte pour l'instance 2.

Indicateur	SC 6.1	SC 6.2	SC 6.3	SC 6.4	SC 6.5
VE	10	11	10	10	10
VT	14	13	14	14	14
KM VE	602.0	634.0	602.0	602	602
KM VT	476.0	444	476.00	476.0	476.0
Énergie consommée	79.01	83.72	121.22	79.01	121.22
Coût de recharge	2.40	2.55	8	2.40	7.99
CO ₂ VT	134.29	125.26	134.29	134.29	134.29
CO ₂ VE	4.35	4.68	8.99	4.35	8.99
Énergie en dépassement	0	0	0	0	0
Coût dépassement	0	0	0	0	0

- Pour l'instance 2, le nombre optimal des VEs varie entre 10 et 11 en fonction des scénarios, ce qui n'est pas très loin du nombre actuel de VEs.

5.9 Conclusion

Dans ce chapitre, nous avons proposé plusieurs extensions à notre problème de référence en variant les fonctions objectif et les contraintes. Nous avons également présenté différents scénarios d'usage et comparé les résultats obtenus sur deux instances réelles du Groupe La Poste. Malgré les améliorations apportées grâce à notre algorithme d'optimisation par rapport aux solutions actuelles, une nouvelle réorganisation des tournées permettrait une meilleure exploitation des véhicules électriques.

Chapitre 6

Étude du problème d'ordonnancement d'intervalles sous contraintes d'énergie

Sommaire

6.1 Introduction	78
6.2 Ordonnancement d'intervalles sous contraintes d'énergie	79
6.3 Revue de littérature	80
6.3.1 Problèmes tactiques	81
6.3.2 Problèmes de minimisation des coûts	81
6.3.3 Problèmes de maximisation du profit	82
6.3.4 Problèmes d'ordonnancement d'intervalles en-ligne	83
6.4 Étude du problème	84
6.4.1 Minimal ISEC	84
6.4.2 Maximal ISEC	86
6.4.3 Maximal Weighted ISEC	89
6.5 Conclusion	89

Dans ce chapitre, nous étudions notre problème d'optimisation conjointe de l'affectation et de la recharge des VEs du point de vue de la théorie d'ordonnancement. Nous introduisons alors un nouveau problème d'ordonnancement d'intervalles sous contraintes d'énergie.

6.1 Introduction

La théorie de l'ordonnancement de tâches représente une branche fondamentale de la recherche opérationnelle. Elle trouve son intérêt dans plusieurs domaines tels que la gestion de projets, la production, l'administration et l'informatique. La résolution des problèmes d'ordonnancement contribue ainsi à aider les entreprises à améliorer et à rentabiliser aux mieux leurs activités administratives et industrielles et leurs systèmes de production et de maintenance.

Dans le domaine de la recherche opérationnelle, les problèmes d'ordonnancement sont formulés en termes de machines et de jobs. Les machines représentent les ressources et les jobs représentent les tâches qui doivent être exécutées en utilisant les ressources disponibles. Dans un problème traditionnel d'ordonnancement, des tâches caractérisées par une durée d'exécution, une date de début au plus tôt et une date de fin au plus tard, doivent être ordonnancées sur un nombre défini de machines. Il y a alors une liberté dans la détermination des dates de début d'exécution des tâches. Le

planificateur utilise cette liberté pour construire un calendrier de tâches qui satisfait les contraintes temporelles (délais, contraintes de précédence, etc.) et les contraintes portant sur la consommation et la disponibilité des ressources requises. L'objectif de ce planificateur est d'optimiser un critère donné. Par exemple, en minimisant le temps d'exécution total des tâches, l'objectif est de mesurer le temps minimal nécessaire pour que les machines exécutent toutes les tâches indiquées.

Les problèmes d'ordonnancement varient selon la nature des opérations (tâches, intervalles, etc.), les caractéristiques des ressources (limitées, renouvelables, etc.), les contraintes portant sur les critères à optimiser (nombre de tâches à exécuter, makespan, etc.), etc. Si le problème d'ordonnancement prend également en compte des contraintes portant sur l'exploitation des ressources, il est dit sous contraintes. Les contraintes sont, généralement, des relations entre les caractéristiques temporelles des opérations avec le temps (la précédence des opérations) ou des relations temporelles décrivant les disponibilités des ressources. La complexité des problèmes d'ordonnancement augmente avec celle de l'exploitation des ressources.

Dans ce chapitre, nous focalisons notre étude sur les problèmes d'ordonnancement d'intervalles, également connus sous le nom de problèmes d'ordonnancement de tâches fixes ou encore de k-track assignment problems. Ces problèmes sont définis par des dates de début et de fin de tâches qui sont fixées à l'avance. Une tâche est alors assimilée à un intervalle $[s_j; f_j]$ durant lequel elle doit être exécutée sans interruption.

La recherche sur les différents problèmes d'ordonnancement est abondamment présente dans la littérature de la recherche opérationnelle. Dans la section 6.2, nous décrivons notre problème d'intervalles sous contraintes d'énergie. La section 6.3 présente une étude bibliographique du problème d'ordonnancement d'intervalles. Dans la section 6.4, nous nous intéressons à l'étude de complexité et d'approximabilité de notre problème. La section 6.5 conclut ce chapitre.

6.2 Ordonnancement d'intervalles sous contraintes d'énergie

Dans ce chapitre, nous étudions un nouveau problème d'ordonnancement. En effet, le problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques peut être vu comme étant un problème d'ordonnancement d'intervalles sous contraintes supplémentaires d'énergie. Nous proposons alors d'étudier la complexité de plusieurs variantes de ce problème et de développer des algorithmes d'approximation pour les résoudre.

Notre problème d'ordonnancement d'intervalles sous contraintes d'énergie est défini par un ensemble de n tâches indépendantes. Chaque tâche i est définie par des dates de début et de fin fixes et elle nécessite une quantité d'énergie e_i pour son exécution. Toutes les tâches doivent être exécutées sur m machines indépendantes fonctionnant avec des batteries. Chaque batterie est définie par une capacité maximale égale à B . L'énergie stockée dans la batterie est utilisée par la machine lors de l'exécution des tâches. Si une machine n'a pas suffisamment d'énergie pour exécuter une tâche, sa batterie pourrait être chargée en utilisant des chargeurs. Par contre, l'énergie totale que peuvent récupérer toutes les machines au même moment ne peut pas dépasser R à chaque intervalle de temps. Chaque machine ne peut exécuter qu'une seule tâche à la fois et ne peut pas exécuter des tâches pendant sa phase de recharge. Chaque tâche peut être affectée à n'importe quelle machine pourvu que toutes les contraintes soient satisfaites.

L'objectif ici est de déterminer un ordonnancement des tâches ainsi qu'un ordonnancement des tâches de recharge sur les machines disponibles afin de satisfaire toutes les contraintes.

À part les contraintes classiques liées au problème d'ordonnancement d'intervalles sur m machines parallèles, nous considérons les contraintes supplémentaires suivantes :

1. Contraintes de batterie : Ces contraintes concernent la capacité de batterie de chaque machine

à stocker de l'énergie. Nous distinguons le cas où toutes les batteries des machines ont des capacités illimitées ($B = +\infty$) et le cas où chaque machine a une capacité de batterie limitée ($B \in \mathbb{N}$).

2. Contraintes de ressources : Ces contraintes concernent la quantité d'énergie disponible à chaque instant.
3. Contraintes de chargeur : Chaque chargeur peut fournir à chaque machine une quantité limitée d'énergie. Nous distinguons le cas où chaque chargeur ne peut fournir qu'une quantité limitée d'énergie et le cas où chaque chargeur peut fournir une quantité illimitée d'énergie pourvu que cette quantité ne dépasse pas R .

Dans ce qui suit, nous donnons un exemple illustratif d'une variante de notre problème où toutes les contraintes d'énergie doivent être vérifiées et l'objectif est de déterminer le nombre minimal de machines nécessaires pour exécuter toutes les tâches.

Exemple. Nous considérons un problème d'ordonnement de trois intervalles. Les dates de début, les dates de fin et les ressources nécessaires de ces intervalles sont données dans le tableau 6.1. Ces intervalles doivent être exécutés en utilisant des machines parallèles dont la capacité de batteries est égale à 6 unités. Nous considérons que chaque chargeur peut fournir 3 unités de ressources et, à chaque période de temps t , $R = 6$ unités. Pour cette instance, le nombre minimal de machines

TAB. 6.1: Instance du problème d'ordonnement d'intervalles sous contraintes d'énergie

Intervalle	Date de début	Date de fin	Ressources requises
j_1	1	3	3
j_2	2	4	6
j_3	4	7	6

nécessaires pour exécuter trois intervalles est égal à 3. Une solution réalisable est présentée dans la figure 6.1. La tâche J_3 ne peut être exécutée ni sur la machine 1 ni sur la machine 2 bien que ces deux machines soient disponibles à la date de début de J_3 . Ceci est dû au fait qu'aucune machine ne dispose de suffisamment d'énergie pour exécuter J_3 et pendant l'intervalle de temps $[3,4]$, la machine 2 peut au plus récupérer 3 unités d'énergie supplémentaires étant donné que chaque chargeur peut au plus fournir 3 unités d'énergie.

6.3 Revue de littérature

Dans un problème d'ordonnement d'intervalles, chaque tâche a une date de début et de fin imposée, on assimile donc une tâche à un intervalle. Il existe pour ce type de problèmes une littérature importante [93, 94].

Nous distinguons les variantes suivantes du problème d'ordonnement d'intervalles :

- Problèmes tactiques
- Problèmes de minimisation des coûts
- Problèmes de maximisation du profit
- Problèmes d'ordonnement d'intervalles en-ligne

Les trois premières variantes concernent la version hors-ligne du problème d'ordonnement d'intervalles et la dernière variante concerne sa version en-ligne.

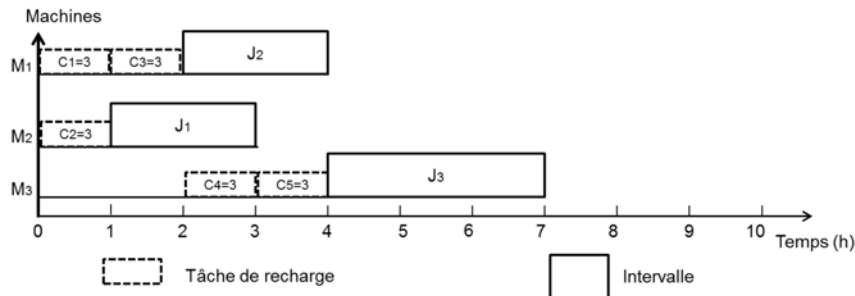


FIG. 6.1: Solution au problème d'ordonnancement d'intervalles sous contraintes d'énergie sur trois machines parallèles

6.3.1 Problèmes tactiques

Dans un problème d'ordonnancement hors-ligne, toutes les caractéristiques des tâches sont connues à l'avance. Dantzig et Fulkerson [95] et Gertsbakh et Stern [96] étudient le problème d'ordonnancement d'intervalles pour lequel n intervalles et m machines sont donnés et la question est de savoir s'il existe un ordonnancement possible de tous les intervalles sur les machines disponibles. Ce problème de faisabilité est équivalent au problème d'ordonnancement d'intervalles basique [93], connu sous le nom du problème tactique, pour lequel l'objectif est de déterminer le nombre minimal de machines nécessaires pour exécuter toutes les tâches. Ce problème peut être résolu à l'optimalité en $O(n \log n)$ [97]. Le problème d'ordonnancement d'intervalles au niveau tactique est également considéré dans [98]. Ce problème consiste à déterminer le nombre minimal de machines parallèles et hétérogènes tel qu'il existe un ordonnancement faisable d'un ensemble donné de tâches sur ces machines. Chaque tâche appartient à une classe spécifique de tâches. Les auteurs étudient la complexité de ce problème et présentent des algorithmes exacts et approchés pour le résoudre.

6.3.2 Problèmes de minimisation des coûts

Dans cette section, nous considérons le cas où les tâches sont données et l'objectif est de réduire les coûts totaux induits par la réalisation de toutes les tâches sur des machines non identiques (notez que si les machines étaient identiques, nous revenons au problème de base d'ordonnancement d'intervalles). Les machines peuvent différer en fonction de plusieurs critères, par exemple, chaque machine peut n'être disponible que pendant certains intervalles de temps. À titre d'exemple, considérons le cas où à chaque machine j , $1 \leq j \leq m$ est associé un intervalle de disponibilité et l'utilisation d'une machine j induit un coût donné. L'objectif ici est de minimiser les coûts totaux d'exécution de toutes les tâches.

Bhatia et al. [99] considèrent le problème où m types différents de machines sont disponibles. Une machine de type j correspond à un intervalle de disponibilité donné, $1 \leq j \leq m$. Ainsi, pour chaque

intervalle de disponibilité, un nombre illimité de machines existe. Un coût est associé à l'utilisation de chaque type de machines. Deux versions NP-difficiles du problème sont proposées. La première version concerne la minimisation du coût total d'exécution de toutes les tâches sur les machines disponibles. Dans le cas où chaque tâche peut être exécutée sur n'importe quel type de machines, Bhatia et al. proposent un algorithme d'approximation avec un rapport d'approximation de 3. La deuxième version du problème consiste à choisir un sous-ensemble de tâches à exécuter sur un ensemble défini de machines tel que le profit total des tâches exécutées soit maximisé. Plusieurs algorithmes d'approximation avec garantie de performance sont proposés pour résoudre ce problème.

Dans le cas où l'objectif est de minimiser le nombre de machines utilisées, un algorithme d'approximation ayant un rapport d'approximation de 2 est proposé.

Une autre façon d'avoir des machines qui ne sont pas identiques est de considérer un ordre linéaire pour les machines disponibles. De plus, pour chaque tâche i , nous précisons une machine "maximale" $m(i)$, $1 \leq m(i) \leq m$ sur laquelle cette tâche peut être exécutée. Ainsi, les machines $1, \dots, m(i)$ peuvent exécuter la tâche i alors que les machines $m(i) + 1, \dots, m$ ne peuvent pas l'exécuter, $i = 1, \dots, n$. Si un coût est associé à chaque machine, l'objectif de ce problème est de minimiser les coûts d'exécution de toutes les tâches.

Dans le cas où il existe plusieurs types de machines et un ordre linéaire pour ces types, Bhatia et al. [99] proposent un algorithme d'approximation ayant un rapport d'approximation égal à 2. Dans le cas de deux types de machines, Dondeti and Emmons [100] et Huang et Lloyd [101] montrent que le problème étudié peut être résolu en un temps polynomial. À partir de trois classes de machines, le problème d'ordonnement d'intervalles devient NP-difficile. Quelques résultats d'approximation concernant ce problème sont donnés dans [101].

Faneyte et al. [102] considèrent le cas où il existe une distance arbitraire entre chaque paire de tâches et le coût d'une machine dépend de la distance parcourue par la machine pour exécuter ses tâches. L'objectif ici est de minimiser le coût total d'exécution de toutes les tâches. Les auteurs prouvent que le problème qui ne prend en compte que deux types de machines est NP-difficile et proposent un algorithme Branch-and-Price pour le résoudre.

Les machines disponibles peuvent n'être autorisées à exécuter que certaines tâches. Par conséquent, les machines diffèrent par rapport à l'ensemble des tâches qu'elles peuvent exécuter. Pour ce problème, Jansen [103] propose un algorithme d'approximation dont la complexité est en $O(\log n)$.

6.3.3 Problèmes de maximisation du profit

Une variante importante du problème d'ordonnement d'intervalles est le cas où le nombre de machines disponibles est une donnée du problème et l'objectif est de maximiser le nombre (ou le poids total) des tâches à exécuter. Ce problème peut être résolu en utilisant une formulation du problème de flot à coût minimal [104, 105]. Dans le cas où chaque tâche a un poids unitaire, un algorithme glouton peut être utilisé pour déterminer le nombre maximal de tâches à exécuter [106, 107].

Arkin et al. [108] considèrent le problème d'ordonnement de n intervalles sur des machines indépendantes dans le cas où un poids est associé à chaque intervalle. L'objectif de ce problème est de maximiser le poids total des intervalles exécutés. Un algorithme en $O(n^2 \log n)$ est proposé. Si chaque tâche peut seulement être exécutée sur un sous-ensemble aléatoire de machines, ce problème devient NP-difficile. Pour un nombre fixe de machines, un algorithme dont la complexité est en $O(n^{k+1})$ est présenté.

Dans le cas où des intervalles de disponibilité sont associés à chaque machine, Brucker et

Nordmann [109] proposent un algorithme en $O(n^{m-1})$ qui maximise le nombre de tâches exécutées.

Plus récemment, Ng et al. [110] étudient le problème d'ordonnancement d'intervalles dans lequel chaque machine peut exécuter un sous-ensemble spécifique de tâches. Chaque exécution d'une tâche sur une machine génère un gain. L'objectif est de trouver un ordonnancement d'un ensemble de tâches sur les machines disponibles tel que le gain total soit maximisé. Ce problème est prouvé d'être NP-difficile au sens fort. Un algorithme exacte, basé sur une réduction au problème de clique à poids maximum, et plusieurs heuristiques sont proposés pour résoudre ce problème.

Krumke et al. [111] proposent d'étudier le problème d'ordonnancement d'intervalles sur des machines ayant des vitesses différentes. L'objectif de ce problème est de maximiser le nombre d'intervalles exécutés. La version hors-ligne de ce problème est prouvée d'être NP-difficile. Pour la version en-ligne, une borne inférieure de $\frac{5}{3}$ du facteur de compétitivité de n'importe quel algorithme déterministe en-ligne est prouvée. De plus, deux règles d'acceptation d'intervalles sont proposées et il est montré que n'importe quel algorithme en-ligne, qui utilise ces deux règles, est 2-compétitif. Dans [112], un index de priorité et une catégorie sont associés à chaque tâche. L'objectif est de déterminer un ordonnancement de tâches sur les machines tel que la priorité totale des tâches exécutées soit maximale. Chaque machine ne peut exécuter que les tâches appartenant à un sous-ensemble prédéfini de classes de tâches. Dans le cas de machines identiques et d'une seule classe de tâches, un algorithme exacte est proposé. Il s'agit d'un algorithme de flot à coût minimum. Dans le cas général (différentes classes de machines et différentes classes de tâches), une formulation mathématique et deux heuristiques sont présentées.

Un autre type de problèmes d'ordonnancement consiste à prendre en compte les contraintes de ressources [113]. Ce type de problèmes est souvent appelé "Resource Allocation Problem" (RAP). Ce problème est défini par un ensemble d'intervalles où chaque intervalle est défini par un besoin de ressources et une valeur (par exemple : profit, priorité, etc.). L'objectif du problème est de déterminer un ensemble d'intervalles à exécuter sur une machine qui dispose d'un nombre limité de ressources tel que la valeur totale des tâches exécutées soit maximale. Une généralisation de ce problème consiste à considérer des ressources qui varient en fonction du temps [114]. Darmann et al. [115] prouvent que le problème RAP reste NP-difficile même si les profits associés aux tâches sont identiques.

Pour plus de détails concernant les modèles existants, les études de complexité et les algorithmes de résolution du problème d'ordonnancement d'intervalle et plusieurs de ses variantes, nous pouvons nous référer à [116] et [93].

6.3.4 Problèmes d'ordonnancement d'intervalles en-ligne

Dans la version en ligne du problème d'ordonnancement d'intervalles, un ensemble de tâches est présenté à un algorithme d'ordonnancement dans l'ordre des heures de début de ces tâches. Pour chaque tâche, l'algorithme d'optimisation doit décider soit de l'affecter à une machine disponible soit de la rejeter sans savoir pendant combien de temps cette tâche doit être exécutée. L'algorithme d'ordonnancement en-ligne n'a alors pas accès à toutes les données, contrairement aux algorithmes hors-ligne. Des applications, des études de complexité et des algorithmes de résolution du problème général d'ordonnancement en-ligne sont disponibles dans [106, 117, 118, 118].

Dans [117], Lipton et Tomkins introduisent le problème d'ordonnancement d'intervalles dans sa version en-ligne. L'objectif ici est de maximiser la durée totale des intervalles exécutés tout en s'assurant qu'aucune paire de tâches exécutées ne se recouvre. Les auteurs montrent qu'aucun algorithme ne peut avoir un ratio de compétitivité meilleur que $O(\log \Delta)$, où Δ est le ratio entre le plus long et le plus court intervalle. Ils proposent un algorithme $O((\log \Delta)^{1+\epsilon})$ -compétitif (i.e., la

solution calculée n'est que $(\log \Delta)^{1+\epsilon}$ fois pire que l'optimal hors-ligne). Un algorithme 2-compétitif est proposé dans le cas où les intervalles ont une des deux durées possibles : soit 1 soit une durée très supérieure à 1. Ce dernier résultat a été généralisé par Goldman et al. [119] au cas où un délai est associé à chaque tâche, i.e., un délai δ_i d'un intervalle i signifie que si un intervalle est accepté, son exécution doit alors commencer entre s_i et $s_i + \delta_i$. Woeginger [120] expose le cas où une seule machine est disponible et un poids est associé à chaque intervalle. Pour ce problème, il est possible d'interrompre l'exécution d'un intervalle qui a été accepté pour accepter un nouvel intervalle. Le poids de l'intervalle interrompu est alors perdu. L'objectif ici est de maximiser le poids total des intervalles acceptés et qui n'ont pas été interrompus. Pour une classe spécifique d'instances pour laquelle les poids des intervalles et leurs durées sont liés par une fonction convexe croissante, Woeginger propose un algorithme d'approximation ayant une garantie de performance égale à 4.

Dans le cas où un poids unitaire est associé à chaque intervalle et m machines sont disponibles, Faigle et al. [106] considèrent qu'un intervalle est perdu s'il n'est pas exécuté d'une façon continue dès l'arrivée des données le concernant. L'objectif de ce problème est de minimiser le nombre d'intervalles perdus. Les auteurs remarquent qu'un algorithme glouton est un algorithme en-ligne qui génère toujours une solution optimale. Cet algorithme a été étendu, dans [121], au cas où des fenêtres de temps sont associées à chaque machine.

6.4 Étude du problème

Dans le cadre de cette étude, nous nous concentrons sur deux cas : (1) le cas où les machines peuvent être chargées pendant l'horizon d'optimisation quand elles ne sont pas en train d'exécuter des tâches, ce cas est appelé *Recharge Autorisée* et (2) le cas où toutes les machines sont entièrement chargées à l'instant $t = 0$ et elles ne peuvent pas être chargées pendant l'horizon de planification, ce cas est appelé *Recharge Interdite*. Le cas (2) est rencontré quand $R = 0$. Pour les deux cas, nous nous intéressons à une étude de complexité et une étude de faisabilité.

Dans le cas où les tâches ne peuvent pas être toutes exécutées sur un nombre donné de machines, nous considérons deux critères d'optimisation, (i) la maximisation du nombre de tâches exécutées et quand des poids sont associés aux tâches, (ii) l'exécution d'un ensemble de tâches avec une somme totale de poids maximale. Ainsi, pour les cas (1) et (2), nous adressons trois problèmes d'optimisation :

- i- Quel est le nombre minimal de machines nécessaires pour exécuter toutes les tâches? Ce problème sera noté Minimal ISEC.
- ii- Si le nombre de machines disponibles est fixé, quel est le nombre maximal de tâches qui peuvent être exécutées? Ce problème sera noté Maximal ISEC.
- iii- Si un poids est associé à chaque tâche, quel est le poids total maximal des tâches qui peuvent être exécutées en utilisant m machines? Ce problème sera noté Maximal Weighted ISEC.

6.4.1 Minimal ISEC

L'objectif de ce problème est de déterminer le nombre minimal de machines nécessaires pour exécuter un ensemble donné de n tâches tel que toutes les contraintes d'énergie soient satisfaites.

a. Recharge interdite

Quand la recharge, pendant l'horizon d'optimisation, est interdite ou quand $R = 0$, le problème Minimal ISEC est équivalent au problème de *Bin Packing* avec conflits (BPC) [122, 123].

Le problème BPC est défini par un nombre illimité de boîtes de taille C , un ensemble de n pièces $I = \{1, \dots, n\}$ ayant chacune une taille $c_j \leq C$ et un graphe de conflit $G = (V, E)$. Ce problème consiste à placer les objets dans les boîtes tel que les pièces adjacentes sur le graphe soient affectées à des boîtes distinctes avec comme objectif de minimiser le nombre de boîtes utilisées. L'équivalence entre le problème Minimal ISEC et BPC est définie comme suit. Chaque boîte correspond à une machine. La capacité de chaque boîte C correspond à la capacité B de la batterie de la machine. Chaque tâche i correspond à une pièce de BPC, où la taille c_i de l'objet i correspond à l'énergie e_i nécessaire pour exécuter la tâche i . Deux objets i et j correspondant aux tâches i et j , respectivement, sont en conflit, c'est-à-dire $(i, j) \in E$ si et seulement si les intervalles $[s_i, f_i)$ et $[s_j, f_j)$ se recouvrent ($[s_i, f_i) \cap [s_j, f_j) \neq \emptyset$). Ainsi, la minimisation du nombre de boîtes dans le problème BPC est équivalente à la minimisation du nombre de machines utilisées dans le problème Minimal ISEC.

Si E est un ensemble vide, le problème BPC est équivalent au problème classique de *Bin Packing* (BP), ainsi, le problème BPC est une généralisation du problème BP et il est alors NP-difficile. Le problème BPC a été largement étudié dans la littérature [122–124]. Les approches les plus proposées sont des algorithmes d'approximation [122, 123]. Jansen [124] propose un schéma d'approximation pour le problème BPC. Des bornes supérieures et des bornes inférieures sont proposées dans [125–127]. Une approche exacte basée sur une formulation du problème de couverture par ensembles et résolue par un algorithme *Branch and Price* est présentée dans [126].

b. Recharge autorisée

Dans cette section, nous considérons qu'il est possible de charger les machines pendant l'horizon d'optimisation. Dans ce cas, nous montrons que le problème Minimal ISEC est NP-difficile. Nous supposons que la puissance maximale pour la recharge des machines soit limitée à p_{max} et qu'il n'existe pas de contraintes concernant la capacité du réseau électrique ($R = +\infty$).

Théorème 6.1. *Le problème Minimal ISEC est NP-difficile si la recharge des machines est autorisée.*

Preuve : Pour prouver la complexité du problème Minimal ISEC quand la recharge des machines est autorisée, nous utilisons le problème de Partition qui est connu d'être NP-difficile [128]. Cette preuve est identique à celle proposée pour prouver la complexité de notre problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques.

c. Algorithme d'approximation

Du théorème 6.4.1, le problème Minimal ISEC est NP-difficile. Dans ce qui suit, nous montrons que le problème Minimal ISEC n'admet pas d'algorithmes d'approximation ayant un rapport d'approximation inférieur à $\frac{3}{2}$.

Théorème 6.2. *Il n'existe aucun algorithme d'approximation ayant un rapport d'approximation inférieur à $\frac{3}{2}$ pour le problème Minimal ISEC.*

Preuve : Supposons qu'il existe un algorithme d'approximation A ayant un rapport d'approximation inférieur à $\frac{3}{2}$ pour le problème Minimal ISEC, nous allons alors montrer que, dans ce cas, le problème de Partition pourrait être résolu en un temps polynomial.

Pour une instance donnée I du problème de Partition, nous construisons une instance I' pour le problème Minimal ISEC comme décrit dans la preuve du théorème 6.4.1. Clairement, la réponse au problème de Partition est "oui" si et seulement si les tâches du problème Minimal ISEC sont exécutées sur exactement deux machines. Quand un algorithme A est utilisé pour résoudre le problème Minimal ISEC, il génère le nombre de machines m qui satisfait $m < \frac{3}{2}m^*$, où m^* est la valeur optimale pour l'instance I' . Si $m^* = 2$, nous obtenons $m = 2$, ce qui implique que le problème de Partition a une

réponse "oui". Si $m^* \geq 3$, alors $m \geq m^* \geq 3$, ce qui implique que le problème de Partition n'admet pas de solution. Ainsi, l'algorithme d'approximation A résout le problème de Partition en un temps polynomial, ce qui contredit le fait que le problème de Partition soit NP-difficile. Ainsi, A n'est pas un algorithme d'approximation avec un rapport d'approximation inférieur à $\frac{3}{2}$ pour la résolution du problème Minimal ISEC.

Dans ce qui suit, nous proposons un algorithme d'approximation au problème Minimal ISEC avec une garantie de performance de $\frac{3}{2}$.

Algorithme 5 Algorithme d'approximation pour Minimal ISEC

- 1: Trier les tâches dans l'ordre croissant des heures de début s_j , si des tâches ont la même heure de début, trier les selon l'ordre décroissant des consommations énergétiques e_j . Soit L la liste des tâches obtenue après le tri. Initialement, $m = 1$.
 - 2: Si $L = \emptyset$ alors m est le nombre minimal de machines, sinon, aller à l'étape 3.
 - 3: Exécuter la première tâche i de l'ensemble L sur la machine disponible qui dispose de suffisamment d'énergie pour exécuter i . Si plus qu'une machine existe, choisir la machine ayant une quantité d'énergie maximale. Si une telle machine n'existe pas, alors utiliser une nouvelle machine, $m = m + 1$, mettre à jour la liste L et aller à l'étape 2.
-

Lemme 6.3. *L'algorithme 6.4.1 a une garantie de performance de $\frac{3}{2}$.*

Preuve : Étant donné que le problème Minimal ISEC n'admet pas d'algorithme d'approximation ayant un rapport d'approximation de moins de $\frac{3}{2}$, nous prouvons le résultat du théorème 6.3 en construisant une instance I pour laquelle $m_2 = \frac{3}{2}m^*$, où m_2 est la valeur de la solution générée par l'algorithme 6.4.1 et m^* et la valeur de la solution optimale.

Considérons l'instance suivante : Soit $3n$ le nombre total de tâches où n est un entier positif. Nous considérons n tâches longues L_j , $j = 1, \dots, n$, tel que :

- $s_j = 0$, $f_j = 2$ et $e_j = \frac{B}{3} + 1$; $j = 1, \dots, \frac{n}{2}$
- $s_j = 2$, $f_j = 4$, et $e_j = \frac{B}{3} - 1$; $j = \frac{n}{2} + 1, \dots, n$

et $2n$ tâches courtes H_j , $j = 1, \dots, 2n$, tel que :

- $s_j = 0$, $f_j = 1$, et $e_j = \frac{B}{3}$, $j = 1, \dots, \frac{n}{2}$.
- $s_j = 1$, $f_j = 2$ et $e_j = \frac{B}{3}$, $j = \frac{n}{2} + 1, \dots, n$
- $s_j = 2$, $f_j = 3$ et $e_j = \frac{B}{3}$, $j = n + 1, \dots, \frac{3n}{2}$:
- $s_j = 3$, $f_j = 4$ et $e_j = \frac{B}{3} - 1$, $j = \frac{3n}{2} + 1, \dots, 2n$:

En utilisant l'algorithme 6.4.1, le nombre minimal de machines nécessaires pour exécuter toutes les tâches est égal à $\frac{3n}{2}$, mais, la valeur optimale est n . Ainsi, $m_2 = \frac{3m^*}{2}$.

6.4.2 Maximal ISEC

Étant donné un nombre fixe m de machines identiques, l'objectif de ce problème est de déterminer le nombre maximal de tâches qui peuvent être exécutées sur ces machines tout en vérifiant les contraintes d'énergie. Dans ce qui suit, sauf indication contraire, les résultats sont valides pour les cas où la recharge est autorisée ou interdite.

Dans ce qui suit, nous montrons que le problème Maximal ISEC est polynomial si $m = 1$ et nous prouvons que ce problème devient NP-difficile si $m \geq 2$.

a. Cas $m = 1$

Pour prouver que le problème Maximal ISEC est polynomial si $m = 1$, considérons l'algorithme suivant :

Algorithme 6 Algorithme pour Maximal ISEC

- 1: Trier les tâches selon l'ordre décroissant des dates de début. Si deux tâches ont la même date de début, alors trier ces tâches en fonction de l'ordre croissant de e_j . Soit L la liste des tâches triées et S l'ensemble des tâches exécutées. Initialement, $S = \emptyset$.
 - 2: Si $L = \emptyset$, alors S est le planning optimal des tâches, sinon aller à l'étape 3.
 - 3: Soit j la première tâche dans la liste L .
 - 4: Si la machine est disponible et a suffisamment d'énergie pour exécuter la tâche j , alors, exécuter j sur cette machine et mettre à jour S .
 - 5: Si la machine est disponible mais elle ne dispose pas d'une quantité d'énergie suffisante pour exécuter la tâche j , alors rejeter j .
 - 6: Si la machine n'est pas disponible à l'instant s_j , alors soit k la tâche qui est en cours d'exécution sur cette machine à l'instant s_j . Nous considérons les cas suivants :
 - 3.a. Si $f_j > f_k$ alors rejeter la tâche j et aller à l'étape 2.
 - 3.b. Si toutes les contraintes d'énergie sont vérifiées et nous avons soit $f_j = f_k$ et $e_j < e_k$ soit $f_j < f_k$, alors rejeter la tâche k , exécuter la tâche j , mettre à jour S et aller à l'étape 2.
-

Dans l'algorithme 6.4.2, quand la recharge est autorisée, une nouvelle étape est ajoutée : à chaque intervalle de temps pendant lequel la machine est disponible, nous chargeons la batterie de la machine avec la puissance maximale p_{max} .

Théorème 6.4. L'algorithme 6.4.2 résout le problème Maximal ISEC en $O(n \log n)$.

Preuve : Faisabilité et Complexité. Le planning de tâches obtenu est faisable étant donné qu'à aucun moment la machine n'exécute des tâches qui se recouvrent et les contraintes d'énergie sont toujours satisfaites.

Optimalité. Soit S l'ensemble de tâches créé par l'algorithme ci-dessus et supposons qu'il ne soit pas optimal. Soit S^* un ensemble optimal de tâches et supposons que parmi tous les ensembles optimaux, S^* soit celui qui a le plus grand nombre de tâches de S . Il existe au moins une tâche dans S qui n'appartient pas à S^* . Soit j la première telle tâche en triant les ensembles S et S^* selon l'ordre chronologique. La raison pour laquelle j n'est pas dans S^* est parce qu'il existe au moins une autre tâche dans S qui est en conflit avec j et qui se termine soit plus tôt que j soit en même temps mais elle nécessite moins d'énergie que j . Soit j' une des tâches qui sont en conflit avec j et qui ne sont pas dans S^* . Nous allons montrer maintenant que $S^* \cup \{j'\} \setminus \{j\}$ représente également un ordonnancement optimal des tâches. Pour ce faire, nous prouvons que l'ordonnancement obtenu après substitution est aussi faisable dans n'importe quel cas des trois cas suivants :

- **Cas 1.** $s_{j'} > s_j$: Si $f_{j'} < f_j$, le nouvel ordonnancement est aussi faisable étant donné que j' ne chevauche avec aucune autre tâche dans $S^* \setminus \{j\}$ et les contraintes d'énergie sont satisfaites (la machine peut être chargée avant $s_{j'}$ et après $f_{j'}$, étant donné qu'elle sera libre pendant les intervalles $[s_j, s_{j'}]$ et $[f_{j'}, f_j]$).
Si $f_{j'} = f_j$, le nouvel ordonnancement est aussi faisable vu que $e_{j'} < e_j$. Ainsi, l'exécution de j' sur la machine n'impacte pas l'ordonnancement des tâches de S^* qui commencent après j' .
- **Cas 2.** $s_{j'} = s_j$: L'ordonnancement obtenu est aussi faisable étant donné que les tâches exécutées sur la machine ne se recouvrent pas et les contraintes d'énergie sont satisfaites.

En effet, si la machine ne disposait pas de suffisamment d'énergie pour exécuter j' , j' n'aurait pas été dans S .

- **Cas 3.** $s_{j'} < s_j$: L'ordonnancement obtenu après substitution est également faisable parce que si la machine était occupée pendant l'intervalle de temps $[s_{j'}, s_j]$ ou si elle ne disposait pas de suffisamment d'énergie, j' n'aurait pas été dans S .

Ainsi, nous avons créé un nouvel ordonnancement optimal de tâches en ajoutant une autre tâche de S dans S^* à la place d'une tâche qui n'est pas dans S , ceci contredit l'hypothèse que S^* contient autant de tâches de S que possible. Nous concluons alors que notre algorithme est optimal. Le problème Maximal ISEC peut ainsi être résolu en un temps polynomial en utilisant un algorithme dont la complexité est en $O(n \times \log n)$ si $m = 1$.

b. Cas $m \geq 2$

Dans cette partie, nous nous intéressons au cas où au moins deux machines sont disponibles et nous prouvons que le problème Maximal ISEC devient NP-difficile dans ce cas.

Théorème 6.5. *Le problème Maximal ISEC est NP-difficile dans le cas où la recharge n'est pas autorisée.*

Preuve : Si la recharge n'est pas autorisée, il est facile de remarquer que le problème Maximal ISEC est équivalent à une variante du problème de *Bin Packing* dont l'objectif est de placer un maximum de pièces dans un nombre fixé de boîtes [129]. Plus précisément, étant donné un nombre fixe m de boîtes ayant une taille B et un ensemble L de n pièces ayant une taille e_i , l'objectif est de maximiser le nombre de pièces de l'ensemble L placées dans les m boîtes. Ce problème est NP-difficile [129]. Nous construisons une instance de notre problème comme suit : On dispose de m machines ayant une capacité B . Initialement, toutes les machines sont complètement chargées. Il existe n tâches telles que $\forall i = 1, \dots, n$, la consommation énergétique de la tâche i est e_i , $s_i = i$ et $f_i = i + 1$, $i = 1, \dots, n$. Clairement, une solution optimale pour le problème de *Bin Packing* est aussi une solution optimale pour le problème Maximal ISEC.

Dans le cas où la recharge est autorisée, le problème Maximal ISEC ne peut pas être réduit au problème de *Bin Packing* étant donné qu'à chaque opération de recharge, l'énergie stockée augmente. Dans ce qui suit, nous prouvons que le problème Maximal ISEC reste NP-difficile même si la recharge est autorisée.

Théorème 6.6. *Le problème Maximal ISEC est NP-difficile même si la recharge est autorisée.*

Preuve : Pour prouver que le problème Maximal ISEC est NP-difficile même dans le cas où la recharge est autorisée, nous utilisons une réduction au problème de *Partition*. Étant donné une instance aléatoire du problème de *Partition*, nous construisons une instance (\mathcal{S}) du problème Maximal ISEC avec deux machines. Chaque machine a une capacité B et elle est initialement complètement chargée. À chaque intervalle de temps, une machine i peut être chargée avec une puissance $p_{max} = 1$. Nous supposons qu'il n'existe aucune contrainte concernant la capacité du réseau électrique ($R = +\infty$). Nous considérons un ensemble de $n + 2$ tâches. Les heures de début, les heures de fin et les consommations énergétiques de ces tâches sont données dans le tableau 6.2. Nous supposons que $a_0 = 0$.

Nous prouvons que le problème de *Partition* admet une solution si, et seulement si, toutes les tâches sont exécutées sur exactement deux machines. Le reste de cette preuve est similaire à la preuve du théorème 6.4.1.

TAB. 6.2: Caractéristiques de l'instance (\mathcal{I})

Jobs	s_j	f_j	e_j
$J_i; i = 1, \dots, n$	$\sum_0^{i-1} b_k$	$\sum_0^i b_k$	b_i
J_{n+1}	$2B$	$2B + 1$	B
J_{n+2}	$2B$	$2B + 1$	B

6.4.3 Maximal Weighted ISEC

Dans cette section, nous considérons que chaque tâche a un poids w_i , $i = 1, \dots, n$. L'objectif du problème Maximal Weighted ISEC est d'exécuter un ensemble de tâches sur m machines identiques tel que la somme totale de leurs poids soit maximisée. Étant donné que le problème Maximal ISEC est un cas particulier du problème Maximal Weighted ISEC pour lequel toutes les tâches ont des poids unitaires ($w_i = 1$, $i = 1, \dots, n$), nous obtenons le résultat suivant :

Théorème 6.7. *Le problème Maximal Weighted ISEC est NP-difficile dans le cas où la recharge est autorisée et dans le cas où la recharge est interdite.*

6.5 Conclusion

Dans ce chapitre, nous avons considéré le problème d'ordonnement d'intervalles sous contraintes d'énergie qui représente une nouvelle classe de problèmes d'ordonnement d'intervalles. Ce problème est inspiré de notre problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques.

Nous avons étudié le cas où l'objectif est de déterminer le nombre minimal de machines nécessaires pour exécuter un ensemble de n tâches (Minimal ISEC). Nous avons montré que ce problème est équivalent au problème de *Bin Packing* avec conflits quand la recharge pendant l'horizon de planification est interdite. En revanche, si la recharge est autorisée, nous avons montré que le problème Minimal ISEC est NP-difficile et nous avons proposé un algorithme d'approximation ayant un rapport d'approximation de $\frac{3}{2}$. Ce facteur d'approximation est le meilleur possible pour le problème Minimal ISEC. De plus, nous avons étudié la complexité de deux autres variantes de notre problème où l'objectif est de déterminer le nombre maximal ou le poids maximum des tâches pouvant être exécutées sur un nombre fixe de machines.

Chapitre 7

Heuristiques pour la résolution du problème de tournées de véhicules électriques

Sommaire

7.1 Introduction	90
7.2 Motivation et description du problème	93
7.3 Formulation mathématique	95
7.4 Résolution par heuristiques	100
7.4.1 Représentation d'une solution au problème EVRP-MF	100
7.4.2 Prise en compte des besoins d'énergie des tournées	101
7.4.3 Heuristique constructive	105
7.4.4 Méthodes de recherche locale	107
7.4.5 Recherche locale à grand voisinage (LNS)	115
7.5 Étude comparative	124
7.5.1 Implémentation et jeux de test	124
7.5.2 Choix des paramètres des algorithmes	125
7.5.3 Résultats numériques	125
7.6 Conclusion	130

Ce chapitre est consacré à l'étude d'un nouveau problème de tournées de véhicules électriques qui est inspiré d'un cas d'usage réel. Après la présentation des contraintes et de l'objectif de ce problème, nous détaillerons les méthodes de résolution utilisées et nous exposerons les résultats obtenus sur des instances généralisées de la littérature.

7.1 Introduction

Le problème de tournées de véhicules, ou encore Vehicle Routing Problem (VRP) [75, 130, 131], consiste à déterminer les tournées d'une flotte de véhicules, à coût total minimal, pour servir une liste de clients. Chaque tournée doit commencer et finir dans un dépôt et chaque point à servir doit être visité exactement une fois tout en respectant les contraintes de disponibilité et de capacité des véhicules. Ce problème est une extension du problème de voyageur de commerce qui fait partie de

la classe des problèmes NP-Complets.

Le VRP occupe une place importante dans le domaine de la recherche opérationnelle et il est le sujet d'une recherche soutenue depuis plusieurs années étant donné qu'il permet de modéliser plusieurs applications réelles telles que la collecte des déchets [132], la distribution des matériaux dangereux [133], le ramassage scolaire et les tournées de service comme le cas des métiers La Poste et ERDF.

L'intérêt croissant du VRP et la multitude et la diversité de ses applications, en particulier dans les domaines de la logistique et du transport, ont donné lieu à plusieurs variantes qui se distinguent principalement par les contraintes considérées et les fonctions objectif à optimiser. Parmi les contraintes les plus étudiées figurent les contraintes de capacité des véhicules et les contraintes du temps. Les différentes variantes du VRP sont traditionnellement définies dans la littérature par des sigles. Plusieurs classifications ont été proposées comme celles proposées dans [134] et [135].

Parmi les différentes variantes du VRP, le problème de tournées avec fenêtre de temps [136–138], le problème de tournées avec Backhauls [139] et le problème de tournées avec collecte et livraison [140–143] ont été largement étudiés dans la littérature. Dans [144], les auteurs proposent une classification bibliographique des différentes méta-heuristiques utilisées pour résoudre le problème de tournées de véhicules et ses variantes.

En plus des méta-heuristiques et des méthodes exactes [145, 146], le VRP peut être résolu en utilisant plusieurs types de méthodes approchées telles que :

- les heuristiques de construction des tournées qui calculent généralement des coûts d'insertion des sommets telles que les méthodes d'insertion (exemple : méthode d'insertion à moindre coût) et les méthodes des économies (exemple : algorithme de *Clarke-Wright* [147]).
- les heuristiques d'amélioration des tournées qui consistent à échanger un certain nombre d'arêtes telles que les algorithmes de recherche locale 2-Opt, Or-Opt et les algorithmes proposés par Lin [148, 149].
- les algorithmes en deux phases tels que les algorithmes de type Cluster-First, Route-Second ("groupe en premier, route en second") comme par exemple l'algorithme de *Fisher and Jaikumar* [150], et les algorithmes de type *Route-First Cluster-Second* ("route en premier, groupe en second"). Pour la première famille d'algorithmes en deux phases, des sous-ensembles de sommets voisins sont déterminés et ensuite des tournées sont construites. En revanche, pour la deuxième famille, certaines contraintes sont au début relâchées pour construire une tournée globale. Cette dernière sera par la suite corrigée pour obtenir des tournées réalisables.

Avec le déploiement du véhicule électrique dans les flottes d'entreprises, une nouvelle variante du problème de tournées de véhicules a fait son apparition. Il s'agit du problème de tournées de véhicules électriques qui a commencé à faire l'objet de quelques études depuis quelques années. Ainsi, plusieurs groupes de recherche s'intéressent aujourd'hui à l'étude de ce problème et certaines de ses variantes. Des contraintes telles qu'une capacité d'emport limitée des véhicules ou des fenêtres de temps ont été considérées.

Le problème de tournées de véhicules électriques est encore plus complexe que le problème classique de VRP vu le nombre de contraintes supplémentaires, liées à l'usage et la recharge des véhicules électriques, qu'il doit prendre en compte. La complexité des contraintes considérées varie de la prise en compte d'un seul type de véhicules et de la même technologie de recharge [80] jusqu'à la considération de plusieurs types de batteries [81] et de plusieurs technologies de recharge [83].

À travers l'étude de toutes les variantes étudiées jusqu'à présent, nous remarquons que ces variantes ne reflètent qu'une partie des contraintes liées aux problèmes réels. Le tableau 7.1 présente une comparaison entre les différents problèmes de tournées de véhicules électriques considérés dans

la littérature en fonction des contraintes prises en compte. Concernant les caractéristiques de la flotte des véhicules, seule l'étude proposée dans [82] prend en compte le cas d'une flotte mixte et des véhicules électriques hétérogènes, par contre, elle ne considère qu'un seul type de chargeurs et ne traite pas le cas de la recharge partielle. C'est l'étude proposée par Felipe et al. [83] qui considère plusieurs types de chargeurs et autorise la recharge partielle des véhicules. En revanche, cette étude se limite au cas d'une flotte homogène de véhicules électriques.

TAB. 7.1: Comparaison des problèmes étudiés dans la littérature

Contraintes	Problèmes				
	[78]	[80]	[81]	[82]	[83]
Flotte mixte			✓	✓	
Flotte hétérogène de VEs				✓	
Fenêtres de temps		✓	✓	✓	
Capacité des véhicules		✓	✓	✓	✓
Recharge partielle					✓
Différents chargeurs					✓

Concernant les méthodes de résolution utilisées, une grande partie de ces problèmes a été résolue en utilisant des recherches locales, des recherches adaptatives à grand voisinage [151] et des méthodes de clustering. Le chapitre 3 de ce mémoire propose une étude bibliographique plus détaillée des différentes variantes du problème de tournées de véhicules électriques et des méthodes de résolution utilisées.

Dans ce chapitre, nous considérons le problème de tournées de véhicules électriques et nous prenons en compte la majorité des contraintes exposées dans la littérature ainsi que d'autres contraintes qui n'ont jamais été étudiées auparavant. Nous présentons alors le problème d'optimisation des tournées d'une flotte mixte de véhicules électriques hétérogènes et de véhicules thermiques identiques, noté EVRP-MF.

Ce problème est défini par un ensemble de clients géographiquement dispersés qui doivent être servis par une flotte mixte de véhicules, composée d'une flotte hétérogène de véhicules électriques, ayant des capacités de batterie différentes et des coûts d'exploitation différents, et de véhicules thermiques identiques.

Chaque client est caractérisé par une demande. La somme totale des demandes satisfaites par le même véhicule ne doit pas dépasser sa capacité d'emport maximale. De plus, nous considérons la possibilité de charger les VEs dans les stations de recharge disponibles à l'extérieur du dépôt, pendant les tournées, afin de servir tous les clients à moindre coût.

Chaque station de recharge est caractérisée par une technologie de recharge (recharge lente, accélérée ou rapide). Elle propose l'utilisation de ses bornes de recharge et fixe le coût de ce service. Ainsi, deux stations de recharge proposant la même technologie de recharge, peuvent proposer des coûts différents. Les stations de recharge sont également soumises aux contraintes de fenêtres de temps. Ainsi, si un VE arrive à une station de recharge avant ses horaires d'ouverture, il doit attendre avant d'accéder au service de recharge. Finalement, vu les différentes technologies de batteries, les véhicules électriques ne sont pas nécessairement compatibles avec toutes les technologies de recharge disponibles. Il est également considéré que les VEs peuvent être partiellement chargés à chaque station de recharge visitée. En effet, il peut être intéressant de charger la quantité d'énergie minimale nécessaire pour atteindre un autre point de recharge dans une station dont le coût de recharge est élevé.

L'objectif ici est de minimiser le nombre de véhicules utilisés et de minimiser le coût total des tournées ainsi que le coût de la recharge.

Dans ce chapitre, la section 7.2 décrit notre problème. Nous présentons ensuite un modèle linéaire en nombres mixtes et nous développons plusieurs méthodes de résolution pour générer des solutions de bonne qualité en un temps de calcul raisonnable. Enfin, nous comparons la performance des différentes méthodes sur un ensemble d'instances généralisées de la littérature dans la section 7.5.

7.2 Motivation et description du problème

Dans le cadre du projet Infinidrive, une flotte de véhicules électriques et de véhicules thermiques est disponible pour effectuer un nombre donné de tournées préalablement construites. Même si l'optimisation de l'affectation des tournées et de la recharge des VE permet d'obtenir des gains considérables, avec l'arrivée du véhicule électrique, il est indispensable de proposer une nouvelle réorganisation des tournées pour qu'elles soient plus adaptées à l'usage du véhicule électrique. C'est dans ce but que s'inscrit cette étude qui vise à proposer un outil générique d'optimisation pour la construction des tournées pour une flotte captive tout en considérant un grand nombre de contraintes réelles.

Considérons le cas d'une entreprise de distribution qui s'engage à servir un ensemble de clients géographiquement dispersés pendant un horizon de temps $[0, T]$, en utilisant une flotte captive composée de véhicules électriques hétérogènes et de véhicules thermiques identiques. Cette entreprise dispose d'une infrastructure de recharge composée d'un nombre donné de bornes de recharge disponibles au dépôt. En plus de l'infrastructure de recharge propre à l'entreprise, les VE peuvent être chargés dans des stations de recharge externes afin de mieux satisfaire la demande des clients et réduire le coût total des tournées. Ces stations de recharge externes sont gérées par d'autres entreprises qui acceptent de mutualiser l'usage de leurs infrastructures de recharge afin de rentabiliser leurs coûts d'investissement. Elles fixent le coût de recharge et les fenêtres de temps pendant lesquelles un véhicule externe peut être chargé dans le but de lisser leur courbe de consommation énergétique. Différents types de bornes de recharge peuvent être disponibles. Vu la diversité des modèles des batteries, certains VE peuvent ne pas être compatibles avec tout type d'infrastructure de recharge. Lors des tournées, chaque VE peut revenir au dépôt ou visiter une station de recharge compatible pour être chargé.

Notre problème d'optimisation des tournées d'une flotte mixte de véhicules, ou encore Mixed Fleet Vehicle Routing Problem with Heterogenous Electric Vehicles, noté EVRP-MF, peut être défini par un graphe complet et orienté $G = (V', A)$. V' représente l'ensemble des sommets composé de l'ensemble V de n clients, de l'ensemble F' de sommets fictifs générés pour permettre plusieurs visites à chaque sommet de l'ensemble F des stations de recharge externes $F = \{1, \dots, h\}$ et D' l'ensemble de sommets fictifs générés pour permettre plusieurs visites aux bornes de recharge au dépôt $D = \{1, \dots, \kappa\}$. L'ensemble des arêtes est noté $A = \{(i, j) \mid i, j \in V', i \neq j\}$. Le dépôt est noté soit 0 soit $n + 1$ selon si c'est le premier sommet ou le dernier sommet d'une tournée. L'ensemble $V' \cup \{0\}$ est noté V'_0 , l'ensemble $V' \cup \{n + 1\}$ est noté V'_{n+1} et l'ensemble $V' \cup \{0, n + 1\}$ est noté $V'_{0, n+1}$.

Notre horizon d'optimisation, représentant dans notre cas une journée, est divisé en $H = \frac{T}{\delta}$ périodes de temps équidistantes, $t = 1, \dots, H$, chacune de durée δ , où t représente l'intervalle de temps $[t - 1, t]$.

Une demande q_i est associée à chaque client $i \in V$, ceci représente la quantité de biens qui doit

être livrée à ce client. Un temps de service s_i est associé à chaque client. Chaque arc $(i, j) \in A$ est caractérisé par une distance $d_{i,j}$ et un temps $t_{i,j}$, où $d_{i,j}$ est la distance entre les sommets i et j et $t_{i,j}$ est le temps nécessaire pour parcourir la distance $d_{i,j}$. Quand un arc (i, j) est parcouru par un véhicule électrique, ce dernier consomme une quantité d'énergie $e_{i,j}$ égale à $r \times d_{i,j}$, où r est un coefficient constant de consommation énergétique.

Les bornes de recharge de la station de recharge $f \in F$ sont disponibles pendant l'intervalle de temps $[a_f, b_f] \subset [0, T]$. Par conséquent, le véhicule électrique doit attendre pour être chargé s'il arrive à la station de recharge f avant l'instant a_f .

Nous considérons que les stations de recharge peuvent proposer trois technologies différentes de recharge : (i) recharge normale, (ii) recharge accélérée et (iii) recharge rapide.

Chaque station de recharge $f \in F \cup D$ peut fournir une puissance de recharge maximale p_f (kW) et propose un coût de recharge c_f exprimé en (euros/kWh).

L'entreprise de distribution possède une flotte captive composée de $M_1 = \{1, \dots, m_1\}$ véhicules électriques hétérogènes et $M_2 = \{m_1 + 1, \dots, m_1 + m_2\}$ véhicules thermiques identiques, nécessaires pour servir l'ensemble des clients. Les véhicules électriques diffèrent dans leurs coûts d'exploitation et leurs capacités de batteries.

Chaque véhicule électrique k fonctionne avec une batterie rechargeable caractérisée par une capacité de stockage CE_k (kWh) et un état de charge (SoC_k^0) à l'instant $t = 0$ exprimé comme un quotient entre la quantité d'énergie stockée dans la batterie et la capacité maximale de la batterie CE_k ($0 =$ vide ; $1 =$ pleine). Pour des raisons de sûreté, l'état de charge de la batterie du véhicule électrique k à chaque instant t (SoC_k^t) doit être maintenu dans l'intervalle $[SoC_k^{\text{Min}}, SoC_k^{\text{Max}}]$, où SoC_k^{Min} et SoC_k^{Max} sont les valeurs minimales et maximales admissibles de l'état de charge de la batterie, respectivement.

Chaque véhicule électrique (véhicule thermique) est caractérisé par une capacité maximale Q^{VE} (Q^{VT}) qui représente la quantité maximale de biens que peut transporter ce véhicule. Soit CF^{VE} (CF^{VT}) (euros/ jour) le coût fixe des VEs (VTs). Soit CV_k^{VE} (CV^{VT}) le coût variable (euros/km) des VEs (VTs). Ainsi, si un arc (i, j) est parcouru par un VE k (VT), ceci a un coût d'exploitation noté $cost_{i,j,k}^{\text{VE}}$ ($cost_{i,j}^{\text{VT}}$) et calculé comme suit : $cost_{i,j,k}^{\text{VE}} = d_{i,j} \times CV_k^{\text{VE}}$ ($cost_{i,j}^{\text{VT}} = d_{i,j} \times CV^{\text{VT}}$).

Chaque client $i \in V$ doit être visité, soit par un véhicule électrique soit par un véhicule thermique, exactement une fois pendant l'intervalle de temps $[0, T]$. Chaque station de recharge peut être visitée autant de fois que nécessaire. Pendant la phase de recharge d'un VE dans une station de recharge f , il est supposé que seulement la quantité d'énergie requise est injectée dans la batterie du VE. Ainsi, les véhicules électriques peuvent être partiellement chargés.

Étant donné que nous considérons plusieurs technologies de recharge, nous prenons également en compte les contraintes de compatibilité entre la technologie de recharge et la batterie du VE. Ainsi, lors de la construction de tournées des VEs, seulement les stations de recharge proposant des technologies de recharge compatibles devraient être considérées.

Une solution faisable pour notre problème est composée d'un ensemble de tournées affectées à des véhicules adéquats et un planning de recharge faisable pour les VEs. Une tournée faisable est une séquence de sommets qui satisfait les contraintes suivantes :

- chaque tournée commence et se termine au dépôt ;
- la quantité totale des biens livrés au cours d'une tournée, qui est égale à la somme des demandes q_i des clients servis pendant la tournée, ne doit pas dépasser la capacité maximale du véhicule (Q^{VE} ou Q^{VT}) ;
- la durée totale de chaque tournée, calculée comme la somme de tous les temps de voyage nécessaires pour servir les clients, du temps nécessaire pour charger le véhicule pendant l'intervalle $[0, T]$, du temps de service de chaque client et éventuellement du temps d'attente

du VE s'il arrive à la station de recharge avant ses horaires d'ouverture, ne doit pas dépasser T ;

- pas plus que m_1 VEs et m_2 VTs sont utilisés ;
- chaque client doit être visité une seule fois pendant l'intervalle de temps $[0, T]$;
- les contraintes de recharge sont satisfaites :
 - pendant la phase de recharge, chaque VE doit être chargé en utilisant une borne de recharge compatible ;
 - à chaque station de recharge $f \in F$, un VE ne peut être chargé que pendant ses horaires d'ouverture $[a_f, b_f]$;
 - les contraintes liées à l'état de charge de la batterie doivent être satisfaites ;

Notre objectif est de construire un nombre minimal de tournées tel que tous les clients soient servis par la flotte de véhicules tout en minimisant le coût de recharge des véhicules électriques et le coût total des tournées.

La fonction objectif, mesurée en unités monétaires, consiste à minimiser quatre coûts : (i) le coût du trajet qui dépend du nombre de kilomètres parcourus par chaque véhicule et le coût variable de chaque véhicule, (ii) le coût de recharge des VEs au dépôt et aux stations de recharge externes pendant l'intervalle $[0, T]$, (iii) le coût fixe total des véhicules utilisés et (vi) le coût total généré par le temps d'attente des VEs s'ils arrivent aux stations de recharge avant leurs horaires d'ouverture.

7.3 Formulation mathématique

Dans cette section, nous proposons un modèle linéaire en nombres mixtes (MIP) pour le problème EVRP-MF. Soit F'_k (respectivement, D'_k et D_k) l'ensemble des stations de recharge externes dans F' (respectivement, chargeurs dans D' et chargeurs dans D) qui sont compatibles avec le VE k . Le tableau 7.2 résume les paramètres de notre problème.

Nous introduisons les variables de décision suivantes :

- x_{ijk}^{VE} : variable binaire égale à 1 si le VE $k \in M_1$ parcourt le trajet reliant les sommets $i \in V'_0$ et $j \in V'_{n+1}$ et 0 sinon.
- x_{ijk}^{VT} : variable binaire égale à 1 si le véhicule thermique k ; $k \in M_2$; parcourt le trajet reliant les sommets $i \in V \cup \{0\}$ et $j \in V \cup \{n+1\}$ et 0 sinon.
- y'_{tfk} : variable binaire précisant si le véhicule électrique k est en charge pendant l'intervalle de temps $t \in [0, T]$ à la station de recharge $f \in F'_k \cup D'_k$.
- p'_{tfk} : variable réelle représentant la puissance de recharge du véhicule k pendant la période de temps $t \in [0, T]$ à la station de recharge $f \in F'_k \cup D'_k$ (kW).
- tt_{jk} : variable réelle précisant l'heure d'arrivée du véhicule k ; $k \in M_1 \cup M_2$; au sommet $j \in V'$.

TAB. 7.2: Définition des paramètres de EVRP-MF

$0, n + 1$	0 correspond au sommet de départ des véhicules et $n + 1$ correspond au sommet d'arrivée des véhicules
F'_k	Ensemble des sommets fictifs pour les stations de recharge F compatibles avec le véhicule k
D'_k	Ensemble des sommets fictifs pour les stations de recharge au dépôt D compatibles avec le véhicule k
V	Ensemble des sommets des clients
V'	Ensemble des clients et des stations de recharge : $V' = F' \cup D' \cup V$
V'_0	$V' \cup \{0\}$
V'_{n+1}	$V' \cup \{n + 1\}$
$V'_{0,n+1}$	$V' \cup \{0, n + 1\}$
$[0, T]$	Horizon d'optimisation
δ	Durée d'une période de temps
s_i	Temps de service du client $i \in V$
q_i	Demande associée au client $i \in V$
$d_{i,j}$	Distance entre les sommets i et j
$t_{i,j}$	Temps du trajet entre i et j
r	Ratio de consommation énergétique
$e_{i,j}$	Besoin en énergie pour parcourir le trajet entre les sommets i et j : $e_{i,j} = r \times d_{i,j}$
$[a_f, b_f]$	Fenêtre de temps d'ouverture de la station $f \in F$
p_f	Puissance de recharge maximale fournie par la station f
c_f	Coût de recharge à la station f
M_1	Ensemble de m_1 véhicules électriques hétérogènes
M_2	Ensemble de m_2 véhicules thermiques identiques
CE_k	Capacité de batterie du véhicule électrique k
SoC_k^t	État de charge du véhicule électrique k à la période de temps t
SoC_k^{Min}	État de charge minimal admissible du véhicule électrique k
SoC_k^{Max}	État de charge maximal admissible du véhicule électrique k
Q_k^{VE}	Capacité de transport maximale d'un VE
Q^{VT}	Capacité de transport maximale d'un VT
$CF^{\text{VE}} (CF^{\text{VT}})$	Coût fixe d'un VE (VT)
$CV_k^{\text{VE}} (CV^{\text{VT}})$	Coût variable d'exploitation d'un VE k (CV)
$cost_{i,j,k}^{\text{VE}} (cost_{i,j}^{\text{VT}})$	Coût du trajet de l'arc (i, j) avec le VE k (CV)

- E_{jk}^a : variable réelle spécifiant la quantité d'énergie disponible dans la batterie du véhicule k ; $k \in M_1$; à son arrivée au sommet j (kWh).
- E_{jk}^l : quantité d'énergie disponible dans la batterie du véhicule électrique k ; $k \in M_1$; à son départ du sommet j (kWh).
- l_{jk} : variable binaire représentant la quantité restante de biens dans le véhicule k ; $k \in M_1 \cup M_2$; après avoir visité le sommet j (m^3).
- W_{jk} : le temps d'attente du véhicule électrique k ; $k \in M_1$; à son arrivée à la station de recharge $j \in F'_k \cup D'_k$.

La formulation mathématique (\mathcal{P}) du problème EVRP-MF est comme suit :

$$\begin{aligned} \text{Min } & \sum_{k \in M_1} \sum_{i \in V'_0, j \in V'_{n+1}} \text{cost}_{i,j,k}^{\text{VE}} \times x_{ijk}^{\text{VE}} + \\ & \sum_{k \in M_2} \sum_{i \in V \cup \{0\}, j \in V \cup \{n+1\}} \text{cost}_{i,j,k}^{\text{VT}} \times x_{ijk}^{\text{VT}} + \\ & \sum_{k \in M_1} \sum_{f \in F'_k \cup D'_k} \sum_{t \in [0, T]} \delta \times c_f \times p'_{tfk} + \\ & \sum_{k \in M_1} \sum_{j \in V'} CF^{\text{VE}} \times x_{0jk}^{\text{VE}} + \\ & \sum_{k \in M_2} \sum_{j \in V} CF^{\text{VT}} \times x_{0jk}^{\text{VT}} + \sum_{k \in M_1} \sum_{f \in F'_k \cup D'_k} \beta \times W_{fk} \end{aligned} \quad (7.1)$$

$$\sum_{k \in M_1} \sum_{j \in V'; i \neq j} x_{ijk}^{\text{VE}} + \sum_{k \in M_2} \sum_{j \in V'; i \neq j} x_{ijk}^{\text{VT}} = 1, \forall i \in V \quad (7.2)$$

$$\sum_{k \in M_1} \sum_{j \in V'_{n+1}; i \neq j} x_{ijk}^{\text{VE}} \leq 1, \forall i \in F' \cup D' \quad (7.3)$$

$$\sum_{k \in M_1} \sum_{j \in V'_{n+1}; i \neq j} (x_{ijk}^{\text{VE}} - x_{jik}^{\text{VE}}) = 0, \forall i \in V' \quad (7.4)$$

$$\sum_{k \in M_2} \sum_{j \in V \cup \{n+1\}; i \neq j} (x_{ijk}^{\text{VT}} - x_{jik}^{\text{VT}}) = 0, \forall i \in V \quad (7.5)$$

$$\sum_{j \in V'; i \neq j} x_{0jk}^{\text{VE}} \leq 1, \forall k \in M_1 \quad (7.6)$$

$$\sum_{j \in V; i \neq j} x_{0jk}^{\text{VT}} \leq 1, \forall k \in M_{\text{VT}} \quad (7.7)$$

$$tt_{jk} \geq tt_{ik} + (t_{i,j} + s_i) \times x_{ijk}^{\text{VE}} - T \times (1 - x_{ijk}^{\text{VE}}), \quad \forall i \in V \cup \{0\}, \forall j \in V'_{n+1}; i \neq j, \forall k \in M_1 \quad (7.8)$$

$$tt_{jk} \geq tt_{ik} + (t_{i,j} + s_i) \times x_{ijk}^{\text{VT}} - T \times (1 - x_{ijk}^{\text{VT}}), \quad \forall i \in V \cup \{0\}, \forall j \in V \cup \{n+1\}; i \neq j, \forall k \in M_2 \quad (7.9)$$

$$tt_{jk} \geq tt_{ik} + t_{i,j} \times x_{ijk}^{\text{VE}} + \sum_{t=0}^{t=H} \delta \times y'_{tik} - T \times (1 - x_{ijk}^{\text{VE}}), \quad \forall i \in F'_k \cup D'_k, \forall j \in V'_{n+1}, \forall k \in M_1 \quad (7.10)$$

$$0 \leq tt_{jk} \leq T, \forall j \in V'_{0,n+1}, \forall k \in M_1 \cup M_2 \quad (7.11)$$

$$a_j \leq tt_{jk} + W_{jk} \leq b_j, \forall j \in F'_k \cup D'_k, \forall k \in M_1 \quad (7.12)$$

$$SoC_k^{\text{Min}} \leq \frac{E_{jk}^a}{CE_k} \leq SoC_k^{\text{Max}}, \forall j \in V'_{n+1}, \forall k \in M_1 \quad (7.13)$$

$$SoC_k^{\text{Min}} \leq \frac{E_{jk}^l}{CE_k} \leq SoC_k^{\text{Max}}, \forall j \in V'_{n+1}, \forall k \in M_1 \quad (7.14)$$

$$E_{jk}^a = E_{jk}^l, \forall j \in V, \forall k \in M_1 \quad (7.15)$$

$$E_{jk}^a \leq E_{ik}^l - r \times d_{i,j} \times x_{ijk}^{EV} + CE_k(1 - x_{ijk}^{EV}),$$

$$\forall i \in V'_0, \forall j \in V'_{n+1}, \forall k \in M_1 \quad (7.16)$$

$$E_{jk}^l = E_{jk}^a + \sum_{t=0}^{t=H} \delta \times p'_{tjk}, \forall j \in F'_k \cup D'_k, \forall k \in M_1 \quad (7.17)$$

$$0 \leq p'_{tjk} \leq p_j \times y'_{tjk}, \forall j \in F'_k \cup D'_k,$$

$$\forall t \in [0, T], \forall k \in M_1 \quad (7.18)$$

$$\sum_{t=b_j}^{t=H} y'_{tjk} + \sum_{t=0}^{t=a_j} y'_{tjk} = 0, \forall j \in F'_k \cup D'_k, \forall k \in M_1 \quad (7.19)$$

$$l_{jk} \geq l_{ik} + q_j - Q^{VE} \times (1 - x_{ijk}^{VE}),$$

$$\forall i \in V'_0, j \in V \cup \{n+1\}, i \neq j, \forall k \in M_1 \quad (7.20)$$

$$l_{jk} \geq l_{ik} + q_j - Q^{VT} \times (1 - x_{ijk}^{VT}),$$

$$\forall i \in V \cup \{0\}, j \in V \cup \{n+1\}, i \neq j, \forall k \in M_{VT} \quad (7.21)$$

$$l_{jk} \geq l_{ik} - Q^{VE} \times (1 - x_{ijk}^{VE}),$$

$$\forall i \in V'_{n+1}, j \in F'_k \cup D'_k, i \neq j, \forall k \in M_{VE} \quad (7.22)$$

$$l_{jk} \leq Q^{VE}, \forall j \in V'_{n+1}, \forall k \in M_{VE} \quad (7.23)$$

$$l_{jk} \leq Q^{VT}, \forall j \in V \cup \{n+1\}, \forall k \in M_{VT} \quad (7.24)$$

$$W_{jk} \geq a_j - tt_{jk}, \forall j \in F'_k \cup D'_k, \forall k \in M_1 \quad (7.25)$$

$$x_{ijk}^{VE}; x_{ijk}^{VT}; y_{kt}; y'_{tjk} \in \{0, 1\}; E_{jk}^l; E_{jk}^a; l_{jk}; p_{tk};$$

$$p'_{tjk}; tt_{jk}; l_{jk}; W_{jk} \geq 0, \forall i, j, t, k \in V'_{0,n+1} \quad (7.26)$$

La fonction objectif, mesurée en unités monétaires, consiste à minimiser la somme de quatre coûts : (i) le coût du trajet, (ii) le coût de recharge des véhicules électriques pendant l'intervalle de temps $[0, T]$, (iii) le coût fixe total des véhicules utilisés et (vi) le coût engendré par le temps d'attente ; où β est un coefficient qui estime le coût engendré par un temps d'attente égal à une période de temps. Les contraintes (7.2) assurent que chaque client est visité exactement une fois. Les contraintes (7.3) garantissent que chaque station de recharge est visitée au plus une fois. Les contraintes (7.4) et (7.5) assurent que le nombre d'arcs entrants est égal au nombre d'arcs sortants pour chaque sommet du graphe. Les contraintes (7.6) et (7.7) assurent que chaque véhicule est, au plus, affecté à une tournée. Les contraintes (7.8)-(7.10) établissent le lien entre les heures d'arrivée aux sommets i et j et permettent d'éliminer les sous-tournées. Les contraintes (7.11) assurent que l'heure d'arrivée à chaque sommet doit être dans l'intervalle $[0, T]$ et les contraintes (7.12) assurent que l'heure d'arrivée à n'importe quelle station de recharge doit être dans sa fenêtre de temps d'ouverture. Les contraintes (7.13) et (7.14) assurent que la quantité d'énergie disponible à l'arrivée au sommet j ou au départ de ce sommet ne dépasse jamais l'état de charge maximale admissible et ne va jamais au-dessous de l'état de charge minimal admissible. Les contraintes (7.15) garantissent que la quantité d'énergie disponible à l'arrivée à un client j est égale à la quantité d'énergie disponible au départ de j . Les contraintes (7.16) établissent le lien entre les quantités d'énergie disponibles à l'arrivée du véhicule électrique aux sommets i et j . Les contraintes (7.17) prennent en compte la quantité d'énergie récupérée à la station de recharge j . Les contraintes (7.18) assurent que la puissance

utilisée pour charger un véhicule électrique pendant une période de temps ne dépasse pas la puissance de recharge maximale que peut fournir une borne de recharge dans cette station de recharge. Les contraintes (7.19) interdisent la recharge dans n'importe quelle station de recharge hors ses horaires d'ouverture. Les contraintes (7.20)-(7.24) assurent que la quantité totale des biens à livrer pendant une tournée ne dépasse pas la capacité maximale d'emport des véhicules. Le temps d'attente d'un véhicule k à la station de recharge j est défini par les contraintes (7.25). Finalement, les contraintes (7.26) définissent les domaines de toutes les variables utilisées.

7.4 Résolution par heuristiques

Vu la complexité du problème étudié, nous avons choisi de développer plusieurs méthodes approchées et de les comparer entre elles. Notre choix a porté sur l'implémentation des méthodes basées sur la recherche locale et la recherche locale à grand voisinage.

Dans ce qui suit, nous présentons les différentes heuristiques développées pour résoudre le problème EVRP-MF. Nous exposerons, dans la section 7.4.1, notre représentation d'une solution au problème EVRP-MF avant de détailler, dans la section 7.4.2, trois procédures spécifiques à notre problème et qui permettent de prendre en compte les contraintes d'énergie. Notre algorithme constructif utilisé pour générer des solutions initiales est détaillé dans la section 7.4.3. Ensuite, nous présenterons les différentes heuristiques utilisées pour améliorer les solutions initiales. La section 7.4.4 présente les algorithmes de recherche locale développés et la section 7.4.5 expose notre algorithme de recherche locale à grand voisinage.

7.4.1 Représentation d'une solution au problème EVRP-MF

Une solution réalisable S est représentée comme un ensemble de tournées affectées aux véhicules disponibles tel que chaque tournée commence et se termine au dépôt et elle est composée d'une séquence de clients et de stations de recharge. Chaque solution est stockée dans deux tableaux. Le premier tableau A_{Route} contient les séquences de sommets, hors dépôt, où des indices négatifs sont utilisés pour indiquer le début de chaque nouvelle tournée. La dernière case du tableau contient 0 et marque la fin de la liste des tournées. Le second tableau $A_{Vehicule}$ contient la liste des véhicules affectés aux tournées de A_{Route} . Le tableau 7.3 et la figure 7.1 présentent un exemple d'une solution réalisable du problème EVRP-MF ainsi que sa représentation. Pour cet exemple, trois tournées sont construites, les deux premières tournées sont affectées à des véhicules électriques et la dernière tournée est affectée à un véhicule thermique.

TAB. 7.3: Exemple d'une solution au problème EVRP-MF

Tournées	Ordre	Véhicule
1	0-2-5-8-0	1 (VE)
2	0-3-1-0	2 (VE)
3	0-4-6-9-0	5 (VT)

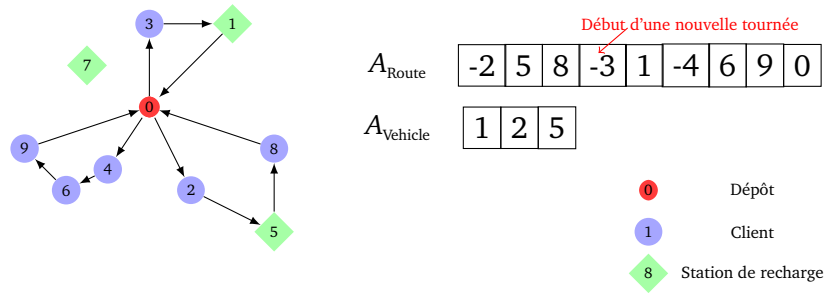


FIG. 7.1: Représentation d'une solution du problème EVRP-MF

7.4.2 Prise en compte des besoins d'énergie des tournées

Dans cette section, nous introduisons quelques procédures spécifiques à notre problème de tournées de véhicules électriques. Elles sont utilisées dans les différentes étapes de nos algorithmes de résolution :

- **Procédure de vérification de faisabilité** : Cette procédure, notée $check_feasibility(route, k)$, est appliquée à une tournée affectée à un véhicule k pour vérifier si elle est réalisable. Elle retourne "vrai" si la tournée est réalisable et "faux" sinon.
- **Procédure d'ajustement de la recharge** : Cette procédure est appliquée à une tournée composée d'un ensemble de clients et un ensemble de bornes de recharge. Elle est utilisée pour calculer la quantité d'énergie à injecter dans la batterie du véhicule électrique à chaque visite d'une station de recharge. Cette fonction n'est appliquée à une tournée donnée $Route$ que si la procédure $check_feasibility(Route, \cdot)$ a déjà retourné "vrai".
- **Procédure d'insertion de chargeurs** : Cette procédure est appliquée à une tournée qui n'est pas réalisable en termes d'énergie. Elle cherche les meilleures stations de recharge à insérer dans cette tournée pour qu'elle devienne faisable tout en générant un coût minimal.

Dans ce qui suit, nous détaillons le fonctionnement des trois procédures citées ci-dessus.

a. Procédure de vérification de faisabilité

La procédure de vérification de faisabilité ($check_feasibility(\cdot, \cdot)$) est utilisée à chaque fois qu'une modification est apportée à une tournée donnée. Elle retourne "vrai" si toutes les contraintes relatives à cette tournée sont vérifiées et "faux" sinon. Elle vérifie alors le volume total transporté, la durée totale de la tournée, la disponibilité d'une quantité d'énergie suffisante à tout moment, le déroulement de la recharge pendant les horaires d'ouverture des stations de recharge et la compatibilité entre le véhicule et les stations de recharge visitées.

Considérons l'exemple de la figure 7.2 où une tournée $Route$, affectée au VE k , est composée, dans l'ordre, des sommets 2, 5, 3, 9 et 6. Le VE k a une capacité de batterie égale à 22 kWh et consomme 0.5 kWh/km. Pour calculer la quantité d'énergie maximale que peut charger le VE k dans la station de recharge compatible $f = 9$, il faut commencer par déterminer le temps maximal Δ_{fk}^{max} que peut passer le véhicule dans cette station de recharge. Pour cela, il faut

calculer l'heure d'arrivée du véhicule à la station de recharge (tt_{fk}) et son heure de départ au plus tard (tt'_{fk}). Si $[tt_{fk}, tt'_{fk}] \cap [a_f, b_f] \neq \{\}$, trois cas se présentent :

- Si $[tt_{fk}, tt'_{fk}] \subset [a_f, b_f]$, $\Delta_{fk}^{max} = tt'_{fk} - tt_{fk}$
- Si $[a_f, b_f] \subset [tt_{fk}, tt'_{fk}]$, $\Delta_{fk}^{max} = b_f - a_f$
- Sinon, $\Delta_{fk}^{max} = \min(tt'_{fk}, b_f) - \max(tt_{fk}, a_f)$

La quantité maximale d'énergie E_{fk}^{max} que peut récupérer le VE à la station f est alors égale à $\min(p_f \times \Delta_{fk}^{max}, CE_k - E_{fk}^a)$. Étant donné que cette fonction ne s'intéresse qu'à la question de faisabilité, elle ne calcule pas la quantité d'énergie exacte à charger à chaque station de recharge.

Pour notre exemple, la fonction $check_feasibility(Route, k)$ retourne "vrai" si toutes les conditions suivantes sont vérifiées :

- $q_2 + q_5 + q_3 + q_6 \leq Q^{VE}$
- $\forall i \in \{2, 5, 3, 9, 6\}, E_{ik}^a \geq 0$
- $E_{fk}^{max} + E_{fk}^a \geq r \times (d_{9,6} + d_{6,0})$
- $t_{0,2} + s_2 + t_{2,5} + s_5 + t_{5,3} + s_3 + t_{3,9} + \Delta_{fk}^{max} + t_{9,6} + s_6 + t_{6,0} \leq T$

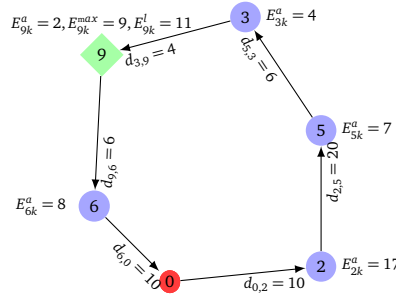


FIG. 7.2: Procédure de vérification de la faisabilité

b. Procédure d'insertion des chargeurs

La Procédure d'insertion des chargeurs (Charger Insertion Method - CIM) peut être utilisée dans les différentes étapes des algorithmes développés afin d'explorer des espaces de recherche plus larges. En effet, au lieu d'exclure certaines solutions qui engendrent une violation des contraintes d'énergie, la CIM essaye de réparer une tournée infaisable en insérant des chargeurs dans cette tournée. Cette procédure cherche les meilleures stations de recharge et les meilleures positions dans la tournée qui engendrent le moindre coût et garantissent la faisabilité de la tournée.

Pour chaque station de recharge f disponible et compatible avec le véhicule sélectionné j et pour chaque position p de la tournée, cette procédure vérifie si l'insertion de f à la position p rend la tournée réalisable. Si oui, elle calcule le coût supplémentaire $increase_cost$ qui sera généré par l'insertion de cette station de recharge dans la position p de la tournée.

Considérons un véhicule avec une capacité de batterie de 22 kWh, qui consomme 0.5 kWh/km et dont le coût opérationnel est 0.1 €/km et le cas d'une tournée composée, dans l'ordre,

des sommets 5, 3, 4 et 6. Nous souhaitons insérer le client 2 dans la position générant le moindre coût de cette tournée (figure 7.3). Si notre étude portait sur le problème classique de VRP, la meilleure façon d'insérer le sommet 2 aurait été de le placer entre les sommets 5 et 3. Toutefois, dans notre cas, l'insertion du client 2 entre les sommets 5 et 3 rend la tournée irréalisable en termes d'énergie (le VE aura besoin de 23 kWh pour effectuer la nouvelle tournée en sa totalité). Au lieu de rejeter cette possibilité d'insertion, nous pouvons choisir d'insérer, en plus du sommet 2, une station de recharge en utilisant la procédure d'insertion des chargeurs qui permet de déterminer le meilleur chargeur à insérer et sa meilleure position dans la tournée. Supposons que la meilleure position d'insertion de chaque station de recharge soit celle représentée en trait interrompu rouge sur la figure 7.3. Pour effectuer la totalité de la tournée, le véhicule doit charger 6 kWh à la station 7 avec un coût $c_7 = 0.1€/kWh$ ou 3 kWh à la station 8 avec un coût $c_8 = 0.22€/kWh$ ou 6 kWh à la station 1 avec un coût $c_1 = 0.15€/kWh$ ou 2 kWh à la station 9 avec un coût $c_9 = 0.2€/kWh$. Si toutes les stations de recharge seront disponibles dès l'arrivée du VE, les coûts d'insertion de ces stations seront calculés comme suit :

- $increase_cost(7) = 20 \times 0.1 + 6 \times 0.1 = 2.6$
- $increase_cost(8) = 14 \times 0.1 + 3 \times 0.22 = 2.06$
- $increase_cost(9) = 16 \times 0.1 + 2 \times 0.2 = 2$
- $increase_cost(1) = 16 \times 0.1 + 6 \times 0.15 = 2.5$

Dans ce cas, c'est le sommet 9 qui sera inséré entre les clients 3 et 4.

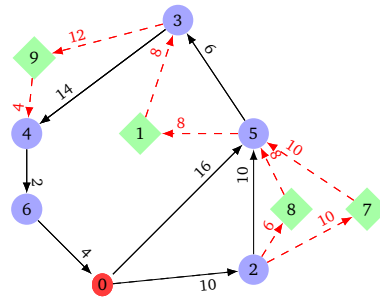


FIG. 7.3: Procédure d'insertion de chargeurs

L'algorithme 7 détaille la procédure d'insertion des chargeurs. Notons par $Insert(r, p, f)$ la fonction qui injecte le sommet f dans la position p de la tournée r et retourne la nouvelle tournée obtenue.

Algorithme 7 Procédure d'insertion des chargeurs

```
1: Entrée : Une tournée  $r$  infaisable affectée au VE  $j$ 
2: Sortie : Une tournée réalisable  $r'$ 
3:  $increase\_cost \leftarrow +\infty, best\_increase\_cost \leftarrow +\infty$ 
4: pour chaque station de recharge  $f$  compatible avec  $j$  faire
5:   pour chaque position  $p$  de  $r$  faire
6:     si  $check\_feasibility(r' = Insert(r, p, f), j) = vrai$  alors
7:       Évaluer le coût d'insertion de  $f$  dans  $p$ 
8:       si  $increase\_cost \leq best\_increase\_cost$  alors
9:          $best\_increase\_cost \leftarrow increase\_cost$ 
10:         $f^* \leftarrow f$ 
11:         $p^* \leftarrow p$ 
12:      fin si
13:    fin si
14:  fin pour
15:  Insérer la station de recharge  $f^*$  dans la position  $p^*$ 
16: fin pour
```

c. Procédure d'ajustement de la recharge

Si la faisabilité d'une tournée donnée est vérifiée, la procédure d'ajustement de la recharge (Charging Adjustment Procedure - CAP) est utilisée pour estimer la quantité minimale d'énergie à injecter dans la batterie du véhicule électrique au dépôt et aux différentes stations de recharge quand la recharge doit être effectuée. Cette procédure permet d'éjecter des chargeurs d'une tournée si la recharge n'est plus indispensable et d'ajuster les quantités d'énergie à injecter en fonction des fenêtres d'ouverture des stations de recharge et des coûts de recharge proposés.

L'idée principale de la CAP est de charger le véhicule électrique avec la quantité maximale d'énergie dans les stations de recharge qui proposent les coûts les plus faibles. Ainsi, si une station de recharge propose des coûts élevés, seulement la quantité d'énergie permettant d'atteindre une autre station de recharge ou le dépôt doit être chargée. Par contre, si une station de recharge propose des coûts faibles, la quantité d'énergie à injecter dans la batterie du véhicule électrique est égale au minimum entre (i) la quantité d'énergie maximale qui peut être injectée tout en vérifiant toutes les contraintes d'énergie, (ii) la quantité d'énergie maximale qui peut être injectée tout en respectant les contraintes liées aux fenêtres d'ouverture des stations de recharge et de la durée maximale des tournées et (iii) l'énergie totale nécessaire pour atteindre le dépôt à la fin de la tournée.

Considérons l'exemple proposé dans la figure 7.4. Afin de servir les clients (2-5-3-4-6), le véhicule électrique a besoin d'être chargé pendant la tournée. Les deux stations de recharge 7 et 1 ont été insérées dans la tournée pour fournir au VE la quantité d'énergie nécessaire pour effectuer la totalité de la tournée. Si $c_1 < c_7$, il pourrait être plus intéressant de charger la quantité d'énergie nécessaire pour atteindre le sommet 1 à la station de recharge 7 et charger la quantité d'énergie nécessaire pour servir le reste des clients au sommet 1, plutôt que de charger le maximum possible au sommet 7 et compléter la recharge au sommet 1.

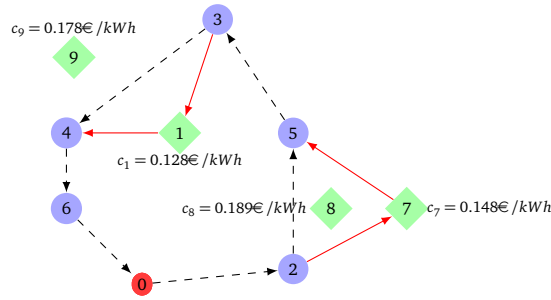


FIG. 7.4: Procédure d'ajustement de la recharge

L'algorithme d'ajustement de la recharge commence par trier les stations de recharge d'une tournée selon l'ordre croissant des coûts de recharge. La liste triée des stations de recharge sera notée *Charger*. À chaque itération, l'algorithme sélectionne la station de recharge proposant le moindre coût, calcule la quantité d'énergie qui doit être injectée dans le véhicule électrique à cette station de recharge et la marque. À la fin de l'algorithme, les stations de recharge correspondant à une quantité d'énergie à injecter nulle sont éliminées. Notons $energy(i, j)$ la quantité d'énergie nécessaire pour servir l'ensemble des clients entre les sommets i et j . Soit $energy_to_inject(i, j, k)$ une variable réelle telle que : $energy_to_inject(i, j, k) = \max(0, energy(i, j) - E_{ik}^a)$. L'algorithme 8 détaille la *Procédure d'ajustement de la recharge*.

Algorithme 8 Procédure d'ajustement de la recharge

- 1: **Entrées** : une tournée r qui est affectée à un VE k
 - 2: **Sorties** : les quantités d'énergie e_i à injecter à chaque station de recharge i
 - 3: Initialement, $\forall i \in r, marked_i \leftarrow false$
 - 4: Construire la liste triée des stations de recharge *Charger*
 - 5: **pour** chaque station de recharge $i \in Charger$ **faire**
 - 6: Trouver la station de recharge j à visiter après i
 - 7: **si** $visited_j = true$ **alors**
 - 8: La quantité d'énergie à injecter à i est $e_i \leftarrow \min(CE_k - E_{ik}^a, energy_to_inject(i, j, k), p_i \times \Delta_{bk}^{max})$
 - 9: **sinon**
 - 10: $e_i \leftarrow \min(CE_k - E_{ik}^a, energy_to_inject(i, 0, k), p_i \times \Delta_{ik}^{max})$
 - 11: $marked_i \leftarrow true$
 - 12: **fin si**
 - 13: Éliminer les stations de recharge ayant $e_i = 0$
 - 14: **fin pour**
-

7.4.3 Heuristique constructive

Pour générer des solutions initiales, nous utilisons un algorithme constructif de tournées et de recharge (Charging Routing Constructive algorithm - CRC) qui commence par une tournée vide et l'étend itérativement jusqu'à ce qu'une solution complète soit construite.

Lorsqu'au moins un client n'est pas encore visité, l'algorithme choisit un VE k selon un critère de choix prédéfini (si aucun VE n'est disponible, un VT peut être sélectionné). Ensuite, il

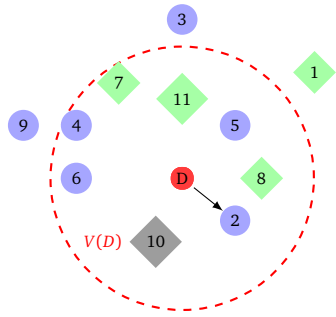
insère itérativement les clients dans la tournée dans la position qui génère le minimum de coût supplémentaire (ce coût est égal à la différence entre le coût de la solution obtenue après l'insertion du noeud et le coût de la solution courante) jusqu'à ce que la contrainte de capacité du véhicule, la contrainte de capacité de la batterie ou la contrainte de temps total ne soit plus vérifiée.

Concernant les insertions des clients dans des tournées affectées à des VEs, celles-ci sont effectuées en anticipant, quand il est possible, toute violation de la contrainte de capacité de batterie par l'insertion de stations de recharge dans la tournée lors de la construction de cette dernière. La meilleure station de recharge est sélectionnée parmi les stations de recharge disponibles et compatibles avec la batterie du VE et qui appartiennent au voisinage $V(i)$ du sommet considéré i qui peut représenter soit un client soit une station de recharge, où $V(i)$ est l'ensemble de tous les sommets appartenant au cercle défini par son centre i et son rayon α ; où α est la distance maximale que peut parcourir le véhicule électrique avec son état de charge actuel. Si plusieurs stations de recharge appartiennent à $V(i)$, nous calculons, pour chaque station de recharge $b \in V(i)$ compatible avec le VE k et dans laquelle le véhicule peut soutirer la quantité totale d'énergie nécessaire pour atteindre le sommet suivant de la tournée, le coût d'insertion de cette station de recharge dans la tournée. Ce coût est calculé comme suit : $cost_increase_b = cost_{i,b,k}^{VE} + cost_{b,n+1,k}^{VE} + c_b \times \min(r \times (d_{i,b} + d_{b,n+1}) - E_{ik}^a, p_b \times \Delta_{bk}^{max}) + \beta \times W_{bk}$, où E_{ik}^a est l'état de charge du véhicule k à son arrivée au sommet i , Δ_{bk}^{max} est le temps maximal que peut passer le véhicule k à la station b en tenant compte des fenêtres d'ouverture de la station et de l'heure de retour au plus tard au dépôt et W_{bk} est le temps d'attente du véhicule k s'il arrive à la station b avant son heure d'ouverture. Le terme $cost_{i,b,k}^{VE} + cost_{b,n+1,k}^{VE}$ représente le coût du trajet entre le sommet i et la borne b d'une part et la borne b et le dépôt $n+1$ d'autre part. Le terme $c_b \times \min(r \times (d_{i,b} + d_{b,n+1}) - E_{ik}^a, p_b \times \Delta_{bk}^{max})$ représente le coût de la recharge à la station de recharge b et le terme $\beta \times W_{bk}$ représente le coût associé au temps d'attente du véhicule.

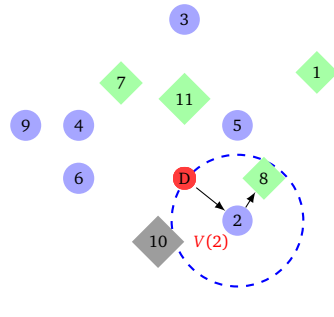
La station de recharge générant le moindre coût sera insérée dans la tournée active. Le calcul de la quantité d'énergie à injecter dans la batterie du VE à chaque station de recharge s'effectue à chaque fois en utilisant la *procédure d'ajustement de la recharge*. Chaque tournée sera fermée et affectée au véhicule sélectionné si une des contraintes de recharge ou de temps n'est pas respectée. Un autre VE sera sélectionné et une nouvelle tournée sera activée.

Si un client ne peut pas être atteint en utilisant un des véhicules électriques ou si aucun VE n'est encore disponible, le client sera affecté au véhicule thermique qui engendrera le moindre coût supplémentaire tout en respectant l'ensemble des contraintes jusqu'à ce que tous les clients soient servis.

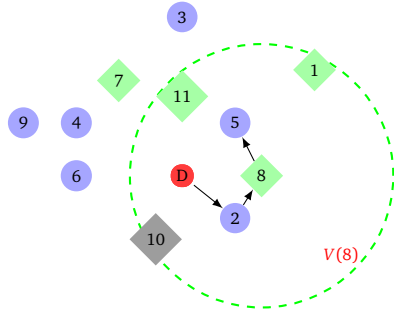
La figure 7.5 explique les étapes principales de l'algorithme CRC. À l'étape initiale, l'ensemble $V(D)$ est composé des clients 2, 4, 5 et 6 et des stations de recharge 7, 8 et 11. Bien que le sommet 10 soit atteignable en termes d'autonomie du véhicule, il n'appartient pas à $V(D)$ puisqu'il n'est pas compatible avec le véhicule électrique sélectionné. Le client 2 est alors ajouté à la tournée active. De même, les sommets 8 et 5 sont ajoutés à cette tournée. À l'étape 4, l'ensemble $V(5)$ est composé d'un seul sommet qui est le dépôt. Le sommet 11 n'appartient pas à l'ensemble $V(5)$ étant donné qu'il ne sera plus disponible à l'arrivée du véhicule (c'est-à-dire, l'heure d'arrivée du VE à la station de recharge 11 sort de sa fenêtre horaire d'ouverture). Si les contraintes de temps le permettent, le VE peut être chargé au dépôt et peut continuer par la suite sa tournée en visitant d'autres clients.



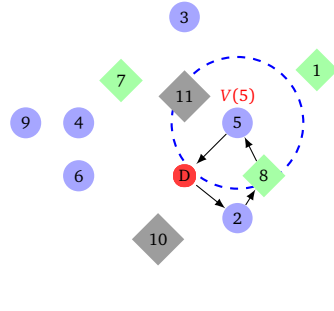
Étape 1 : le VE 1 est un véhicule électrique disponible avec une capacité de batterie maximale $i = D$; $V(D) = \{2, 4, 5, 6, 7, 8\}$;
Ajouter le client 2 à la tournée active : $r_1 := \{D, 2\}$



Étape 2 : $i = 2$; $V(2) = \{8, D\}$; $r_1 := \{D, 2, 8\}$



Étape 3 : $i = 8$; $V(8) = \{5, 1, D\}$ $r_1 := \{D, 2, 8, 5\}$



Étape 4 : $i = 5$; $V(5) = \{D\}$ $r_1 := \{D, 2, 8, 5, D\}$

FIG. 7.5: Les étapes de l'heuristique constructive de tournées et de recharge

7.4.4 Méthodes de recherche locale

Considérons une instance I de notre problème d'optimisation où X est l'ensemble des solutions faisables pour cette instance et $cost : X \rightarrow R$ est une fonction qui retourne le coût de chaque solution dans X . Le voisinage d'une solution $x \in X$ est défini par $N(x) \subset X$. Pour un problème de minimisation, une solution x est un optimum local par rapport à une structure de voisinage N si $cost(x) \leq cost(x'), \forall x' \in N(x)$. Avec ces définitions, il est possible de définir un algorithme de recherche locale. En partant d'une solution initiale x , il cherche la meilleure solution x' dans le voisinage de x . Si $cost(x) \leq cost(x')$, il met à jour la meilleure solution trouvée. Le voisinage de la nouvelle solution est exploré et l'algorithme continue ses itérations jusqu'à ce qu'un critère d'arrêt soit atteint.

L'algorithme 9 décrit le fonctionnement des méthodes de recherche locale.

Algorithme 9 Algorithme de recherche locale

```
1: Entrée Solution initiale  $S_0$ , structure de voisinage  $N$ 
2: Sortie Une meilleure solution  $S$ 
3: Initialement,  $S_0$  et  $amliore \leftarrow vrai$ 
4: tant que  $amliore = vrai$  faire
5:    $amliore \leftarrow faux$ 
6:   pour  $S' \in N(S)$  faire
7:     si  $S'$  est meilleure que  $S$  alors
8:        $S \leftarrow S'$ 
9:        $amliore \leftarrow vrai$ 
10:    fin si
11:  fin pour
12: fin tant que
```

Les algorithmes de recherche locale sont largement utilisés pour résoudre des problèmes d'optimisation combinatoire. Ils peuvent être directement appliqués sur des solutions initiales pour les améliorer comme ils peuvent facilement être intégrés dans des méta-heuristiques. Savelsbergh [152] propose des algorithmes de recherche locale, basés sur l'approche k-interchange, pour résoudre le problème de tournées de véhicules avec fenêtres de temps. Dans [153], les auteurs examinent également les algorithmes de recherche locale pour résoudre ce même problème.

Dans cette section, nous présentons une adaptation de certains algorithmes classiques de recherche locale pour résoudre notre problème de tournées de véhicules électriques. Nous développons également deux autres algorithmes spécifiques à notre problème.

a. Adaptation de la procédure One Point Move ou Relocate

La procédure One Point Move (OPM) ou Relocate cherche à trouver une transformation impliquant un sommet donné (client ou borne) de la solution courante. Elle déplace ce sommet de sa position courante à une nouvelle position. Cette nouvelle position peut être dans la même tournée (transformation "intra-route") ou dans une autre tournée de la solution courante (transformation "inter-route"). Si le changement de position d'un sommet engendre une violation des contraintes d'énergie, la méthode d'insertion de chargeurs pourrait être utilisée pour réparer la nouvelle tournée. Si une transformation acceptable, qui satisfait toutes les contraintes et améliore la solution initiale, est trouvée, alors elle est exécutée.

Considérons l'exemple de la figure 7.6. Les figures à gauche représentent les tournées initiales et les figures à droite représentent les transformations OPM effectuées. Les figures (a) et (b) représentent le cas d'une transformation "inter-route". La meilleure transformation OPM dans ce cas consiste à insérer le sommet 6 après le sommet 5 dans la première tournée, la station de recharge 4 doit être insérée après 6 afin d'assurer le retour du véhicule au dépôt. La station de recharge 7 est éjectée de la deuxième tournée étant donné que le véhicule affecté à cette tournée n'a plus besoin d'être chargé (la tournée devient plus courte en éjectant le sommet 6). Les figures (c) et (d) représentent le cas d'une transformation "intra-route". Dans ce cas, le sommet 5, initialement placé entre le dépôt et le sommet 3, est inséré entre les sommets 3 et 7.

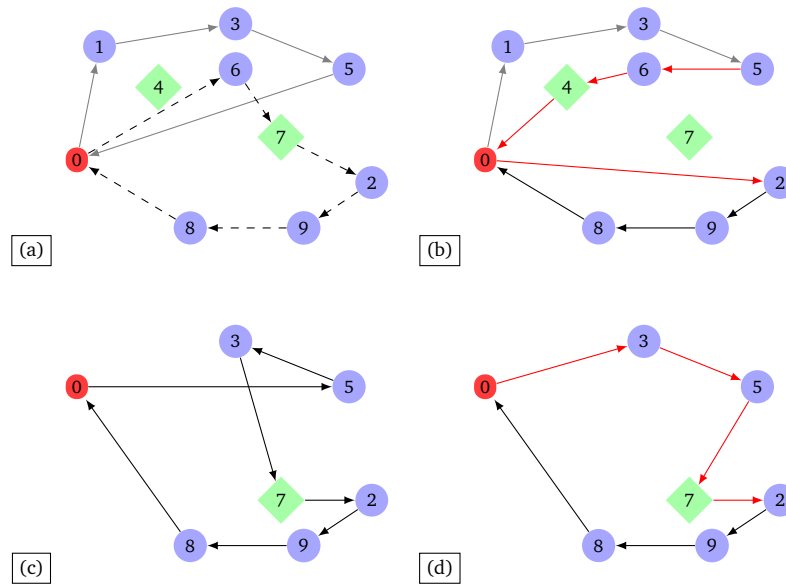


FIG. 7.6: Exemple d'une transformation OPM inter-route et d'une transformation OPM intra-route

L'algorithme 10 détaille la procédure OPM.

Algorithme 10 Algorithme OPM

- 1: **Donnée** : Une solution initiale réalisable S
 - 2: **Sortie** : Une nouvelle solution S'
 - 3: $S' \leftarrow S$; $best_cost \leftarrow cost(S')$
 - 4: Soit $improve \leftarrow true$ et soit $perm$ la liste de tous les sommets de la solution S'
 - 5: **tant que** $improve = true$ **faire**
 - 6: Permuter les éléments de $perm$ aléatoirement
 - 7: **pour** chaque élément i de $perm$ **faire**
 - 8: Evaluer toutes les transformations OPM impliquant le sommet i et choisir la meilleure, notée m
 - 9: **si** la transformation m améliore la solution S' **alors**
 - 10: Effectuer la transformation m sur S'
 - 11: **fin si**
 - 12: **fin pour**
 - 13: **si** $cost(S') < best_cost$ **alors**
 - 14: $best_cost \leftarrow cost(S')$
 - 15: $imp \leftarrow true$
 - 16: **sinon**
 - 17: $imp \leftarrow false$
 - 18: **fin si**
 - 19: **fin tant que**
-

b. Adaptation de la procédure Two Point Move (TPM) ou Swap

Cette procédure essaye de trouver la meilleure transformation TPM, appelée également Swap, impliquant chacun des sommets de la solution courante. Cette procédure consiste à échanger les positions de deux sommets (clients ou bornes) de la même tournée ou de deux sommets de tournées différentes. Elle évalue alors le coût engendré par l'échange des positions de chaque sommet j et de chaque sommet b de la solution courante. Pour définir le type de transformation souhaitée (Inter-route ou Intra-route), nous limitons l'espace de recherche pour chaque sommet. Dans ce cas, seulement les sommets appartenant à l'espace de recherche $search_space(i)$ de i sont considérés.

La figure 7.7 présente un exemple d'une transformation Swap effectuée au sein de la même tournée. Les lignes discontinues représentent la tournée initiale avant la transformation Swap. Les lignes rouges représentent la tournée obtenue en permutant les positions des sommets 2 et 3. À l'issue de cette transformation, la station de recharge 4 ne fait plus partie de la tournée. Elle a été remplacée par la station de recharge 7.

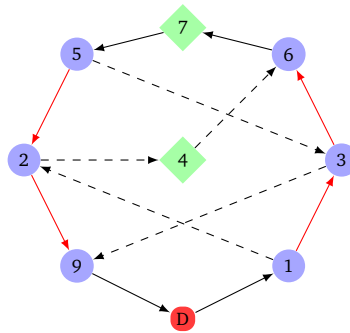


FIG. 7.7: Exemple d'une transformation Swap Intra-route

L'algorithme 11 décrit la procédure Swap.

Algorithme 11 Algorithme Swap

```
1: Donnée : Une solution initiale réalisable  $S$ 
2: Output : Une nouvelle solution  $S'$ 
3:  $S' \leftarrow S$ ;  $best\_cost \leftarrow cost(S')$ 
4: Soit  $imp \leftarrow true$  et soit  $perm$  un tableau contenant tous les sommets de la solution  $S'$ 
5: tant que  $imp = true$  faire
6:   Permuter les éléments de  $perm$  aléatoirement
7:   pour chaque élément  $i$  dans  $perm$  faire
8:     Définir l'espace de recherche du sommet  $i$ , noté  $serach\_space(i)$ 
9:     pour chaque sommet  $j \in serach\_space(i)$  faire
10:      Evaluer la transformation de permutation des positions des sommets  $i$  et  $j$ 
11:     fin pour
12:     Choisir la meilleure transformation Swap impliquant le sommet  $i$ , notée  $m$ 
13:     si  $m$  améliore la solution  $S'$  alors
14:       Appliquer la transformation  $m$  à  $S'$ 
15:     fin si
16:   fin pour
17:   si  $cost(S') < best\_cost$  alors
18:      $best\_cost \leftarrow cost(S')$ 
19:      $imp \leftarrow true$ 
20:   sinon
21:      $imp \leftarrow false$ 
22:   fin si
23: fin tant que
```

c. Adaptation du 2-opt

L'algorithme 2-opt a été proposé par Croes [154] pour résoudre le problème du voyageur de commerce en améliorant la solution initiale. La procédure 2-opt est une approche itérative qui tente d'améliorer une tournée en échangeant deux arcs à chaque itération. Ainsi, à chaque itération de l'algorithme, deux arêtes de la solution courante sont supprimées et les deux portions de tournée obtenues sont reconnectées d'une autre façon pour obtenir une nouvelle solution. Autrement dit, l'algorithme considère les paires d'arêtes (x_i, x_{i+1}) et (x_j, x_{j+1}) et évalue le gain qui peut être obtenu en les remplaçant par les arêtes (x_i, x_j) et (x_{i+1}, x_{j+1}) . Parmi toutes les paires d'arêtes qui permettent d'améliorer la solution courante grâce à une transformation 2-opt, l'algorithme choisit la paire qui permet d'obtenir la meilleure solution possible. Cette procédure est répétée sur toutes les tournées jusqu'à ce qu'aucune amélioration n'est encore possible. La transformation 2-opt peut s'appliquer à des arêtes de la même tournée et à des arêtes de deux tournées différentes.

La figure 7.8 représente un exemple d'une transformation 2-opt. Les figures (a) et (b) représentent le cas d'une transformation intra-route. Dans ce cas, les arêtes (1,3) et (2,4) ont été remplacées par les arêtes (1,2) et (3,4), respectivement. Les figures (c) et (d) représentent le cas d'une transformation inter-route. Dans ce cas, les arêtes (2,10) et (9,6) ont

été remplacées par les arêtes (2,9) et (8,6), respectivement. Étant donné que notre algorithme 2-opt prend en compte la possibilité d'insérer ou de supprimer un chargeur, la station de recharge 10 a été éjectée de la tournée de la figure (d) suite à la transformation 2-opt vu que la recharge à cette station de recharge n'est plus indispensable.

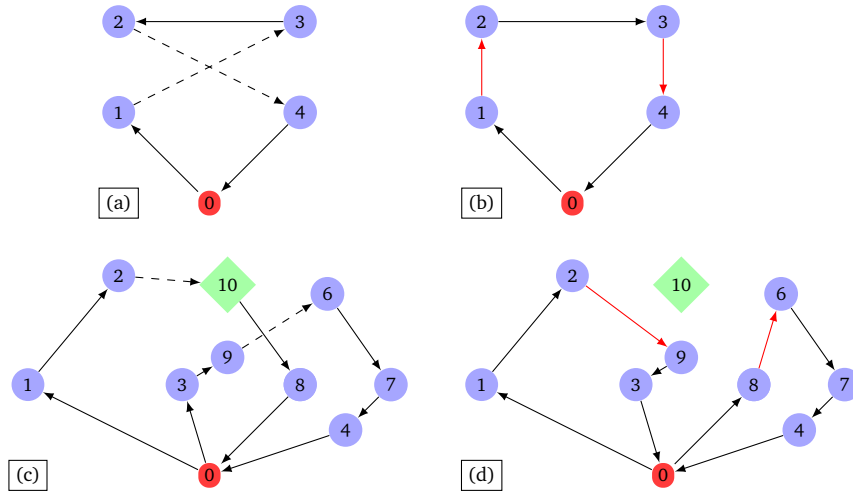


FIG. 7.8: Exemple d'une transformation 2-opt

L'algorithme 12 détaille la procédure 2-opt.

Algorithme 12 Algorithme 2-opt

```
1: Donnée : Une solution initiale réalisable  $S$ 
2: Output : Une nouvelle solution  $S'$ 
3:  $S' \leftarrow S$ ;  $best\_cost \leftarrow cost(S')$ 
4: Soit  $imp \leftarrow true$  et soit  $perm$  un tableau contenant tous les sommets de la solution  $S'$ 
5: tant que  $imp = true$  faire
6:   Permuter les éléments de  $perm$  aléatoirement
7:   pour chaque élément  $i$  dans  $perm$  faire
8:     Définir l'espace de recherche du sommet  $i$ , noté  $serach\_space(i)$ 
9:     pour chaque sommet  $j \in serach\_space(i)$  faire
10:      Evaluer la transformation de remplacement des arêtes  $(i, i + 1)$  et  $(j, j + 1)$  par les arêtes  $(i, j)$  et  $(i + 1, j + 1)$ 
11:      fin pour
12:      Choisir la meilleure transformation 2-opt impliquant le sommet  $i$ , notée  $m$ 
13:      si  $m$  améliore la solution  $S'$  alors
14:        Appliquer la transformation  $m$  à  $S'$ 
15:      fin si
16:    fin pour
17:    si  $cost(S') < best\_cost$  alors
18:       $best\_cost \leftarrow cost(S')$ 
19:       $imp \leftarrow true$ 
20:    sinon
21:       $imp \leftarrow false$ 
22:    fin si
23: fin tant que
```

d. Adaptation du Or-opt

L'algorithme Or-opt est basé sur l'insertion d'une séquence de sommets, appelée chemin ou path en anglais, entre deux autres sommets. Un chemin est un ensemble de sommets consécutifs servis par le même véhicule, il commence à un sommet donné i et a une longueur prédéfinie len . Cette transformation peut être appliquée au sein de la même tournée ou à des tournées différentes. La figure 7.9 présente une transformation Or-opt inter-route où un chemin d'une longueur $len = 2$, composé des sommets 2 et 6, est supprimé de la tournée 2 (présentée en lignes discontinues) et inséré entre les sommets 5 et 0 de la tournée 1 (présentée en lignes continues). La station de recharge 7 a été éjectée de la tournée 2 étant donné que cette nouvelle tournée est devenue plus courte en supprimant les sommets 2 et 6. En revanche, la borne de recharge 4 a été insérée dans la tournée 1 pour assurer sa faisabilité en termes d'énergie.

e. Concaténation des tournées (ConcT)

Cette procédure peut être appliquée à une tournée qui est affectée à un véhicule électrique dont le coût d'exploitation est élevé ainsi qu'à deux tournées différentes. Dans le premier

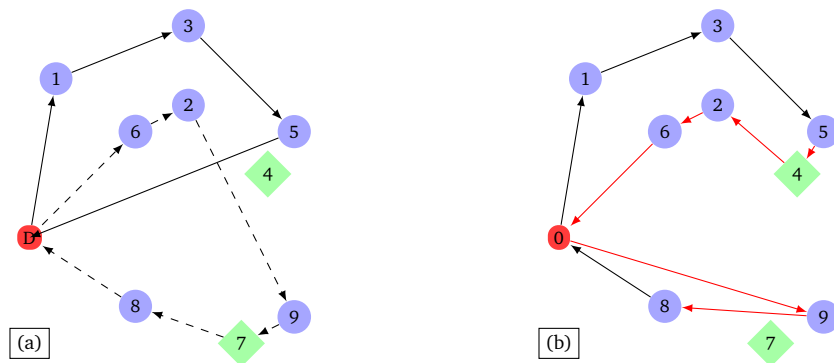


FIG. 7.9: Exemple d'une transformation Or-Opt Inter-route

cas, la procédure de concaténation des tournées sélectionne une tournée qui est affectée à un VE ayant des coûts d'exploitation élevés et une grande capacité de batterie et essaye de l'affecter à un VE moins cher (ayant une capacité de batterie plus petite) tout en considérant la possibilité d'insérer des nouvelles stations de recharge dans la tournée. Dans le second cas, cette procédure sélectionne deux tournées i et j et essaye de les concaténer ensemble. Si la concaténation des tournées n'est pas possible à cause d'une violation des contraintes d'énergie, les possibilités d'affecter les deux tournées à un VE ayant une plus large batterie ou d'insérer un chargeur dans la nouvelle tournée, sont considérées. La figure 7.10 présente un exemple d'une procédure de concaténation de tournées où les tournées (1-3-5) et (6-2-9-8) (figure (a)) sont concaténées grâce à l'insertion de deux stations de recharge 4 et 7 (figure (b)).

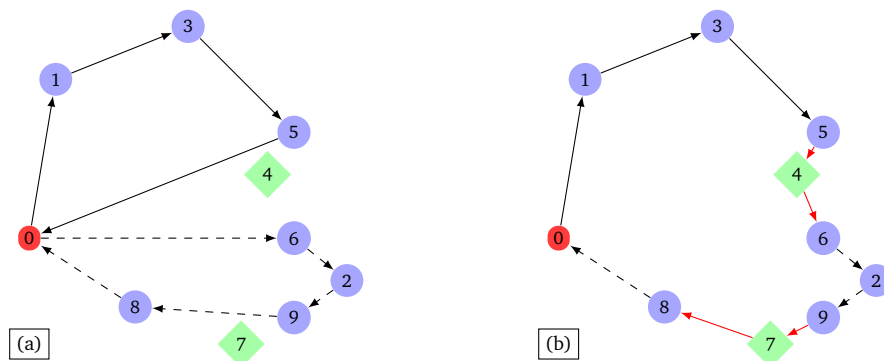


FIG. 7.10: Exemple d'une procédure de concaténation de tournées

f. Réaffectation des tournées (ReaffectT)

Cette procédure choisit une tournée i qui est initialement affectée à un véhicule thermique et essaye d'affecter ses sommets à des VEs tout en considérant la possibilité d'injecter des stations de recharge. Cette procédure peut être utilisée de deux façons : tous les sommets de i sont insérés dans des tournées déjà construites ou quelques sommets de i sont affectés à des tournées déjà construites et le reste de sommets est affecté à un nouveau véhicule électrique. La figure 7.11 présente un exemple de la procédure de réaffectation des tournées. La figure (a) représente une solution réalisable, la tournée présentée en ligne discontinue est initialement affectée à un VT. La figure (b) représente la solution obtenue après l'application de la procédure de réaffectation des tournées. En effet, les sommets de la tournée affectée à

un VT ont été réaffectés aux véhicules électriques utilisés. Le sommet 2 a été injecté dans la tournée constituée des sommets 4, 6 et 9 et pour assurer la faisabilité de la tournée obtenue, la borne de recharge 7 a été insérée entre les sommets 2 et 4. De même, le sommet 8 a été inséré dans la tournée constituée des sommets 1 et 3.

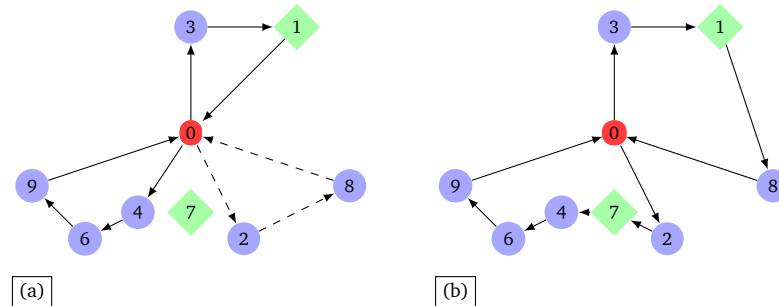


FIG. 7.11: Procédure de réaffectation des tournées

7.4.5 Recherche locale à grand voisinage (LNS)

Dans cette section, nous nous intéressons à la recherche locale à grand voisinage, ou encore Large Neighborhood Search (LNS) en anglais, pour résoudre le problème EVRP-ME. Nous expliquons, dans un premier temps, le principe général de l'algorithme LNS avant de décrire comment nous l'avons utilisé pour résoudre notre problème.

a. Principe de l'algorithme LNS

La méthode de LNS a été proposée dans [155] et elle a ensuite été adaptée par Pisinger et Ropke [156]. Cette méthode appartient à la classe des heuristiques connues sous le nom de "Very Large Scale Neighborhood search algorithms" [157].

L'algorithme de LNS est un algorithme d'amélioration qui part d'une solution initiale et l'améliore progressivement en alternant des phases de destruction et de réparation de la solution courante. La nouvelle solution obtenue peut être acceptée ou rejetée en fonction du critère d'acceptation choisi.

Dans LNS, le voisinage est généralement défini par des méthodes qui sont utilisées pour détruire et réparer la solution courante. Une façon de détruire la solution courante pour un problème de VRP peut consister en une élimination aléatoire d'un certain nombre de clients. Une méthode de réparation peut consister en une réinsertion des clients éjectés en utilisant un algorithme glouton. Étant donné que la méthode de destruction peut détruire une grande partie de la solution, le voisinage de la solution courante peut contenir un grand nombre de solutions, ce qui explique le nom de cette méta-heuristique.

Le degré de destruction de la solution courante est un paramètre important dans l'algorithme LNS. En effet, si seulement une petite partie de la solution est détruite, alors l'heuristique ne sera pas capable d'explorer la totalité de l'espace de recherche. Par contre, si une grande partie de la solution est détruite à chaque itération, l'heuristique aura besoin d'un long temps de calcul pour générer des solutions qui peuvent être de mauvaise qualité. Dans [155], Shaw a proposé d'augmenter progressivement le degré de destruction de la solution courante alors que Ropke et Pisinger [155] ont proposé de choisir un degré de destruction aléatoire à chaque itération. Goel et Gruhn utilisent un algorithme LNS pour la résolution du VRP avec plusieurs emplacements

de collectes et de livraisons. Ils proposent des approches d'amélioration itératives basées sur la recherche de voisinage. Plusieurs extensions de LNS ont été proposées dans la littérature. Une des extensions de LNS les plus utilisées dans la littérature est la Recherche Adaptative à Grand Voisinage (Adaptive large Neighborhood Search-ALNS) [158]. Cette extension étend l'algorithme de LNS en autorisant l'utilisation de plusieurs méthodes de destruction et de réparation de la solution courante au sein de la même recherche. À chaque méthode de destruction et de réparation on associe un poids qui contrôle le nombre de fois chaque méthode est utilisée. Les poids sont ajustés dynamiquement au cours de la recherche. Les méthodes de LNS et d'ALNS ont été utilisées pour résoudre une variété des problèmes d'optimisation tels que les problèmes d'ordonnancement [159] et plusieurs variantes du problème de VRP [160, 161]. La figure 7.12 présente le schéma général de la méthode de LNS.

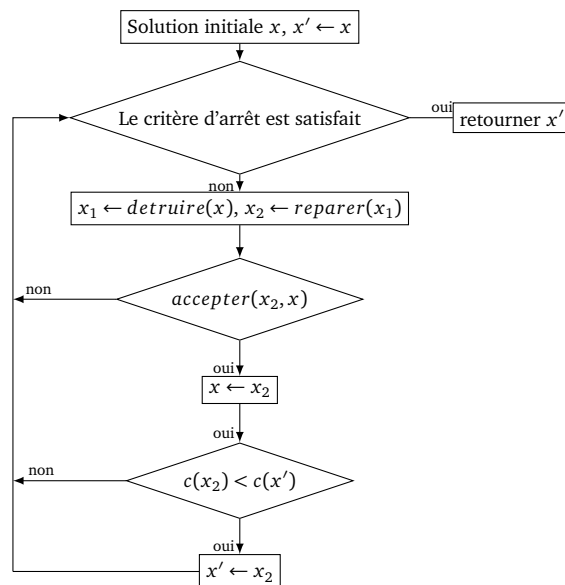


FIG. 7.12: Schéma général de l'algorithme LNS

b. LNS pour EVRP-MF

Pour résoudre le problème EVRP-MF, nous avons développé une recherche locale à grand voisinage. Notre méthode de destruction de la solution, notée *Détruire(.)* consiste à éjecter un ensemble de sommets, choisis aléatoirement, de la solution courante. Notre méthode de reconstruction, appelée également méthode de réparation et notée *Reparer(.)*, consiste à réinsérer les sommets éjectés dans la solution partielle en suivant une stratégie donnée d'insertion.

Pour implémenter les différentes phases de l'algorithme LNS, les paramètres suivants sont utilisés :

- $Iter_1$: paramètre qui contrôle la taille de la boucle principale de LNS ;
- $Iter_2$: paramètre qui précise le nombre de fois les procédures de destruction et de reconstruction doivent être répétées ;
- Num : paramètre qui contrôle la taille de la liste de voisins qui est utilisée dans la procédure de destruction-reconstruction ;

Pour chaque solution faisable S , l'algorithme LNS effectue $Iter_2$ itérations de la procédure de destruction-reconstruction suivante :

- **Détruire(S)** : Un sommet j et un ensemble de $Num - 1$ sommets supplémentaires situés le plus proche possible de j sont sélectionnés d'une façon aléatoire (les sommets sélectionnés peuvent appartenir à des tournées différentes).

Ce voisinage de Num sommets est ensuite éjecté de la solution courante. La phase de destruction peut être appliquée sur des clients et des stations de recharge.

- **Réparer(S)** : Les sommets éjectés sont ensuite réinsérés dans la solution partielle en utilisant une méthode parmi les méthodes d'insertion décrites ci-dessous. Si la solution devient infaisable, une nouvelle station de recharge, qui induit le moindre coût, peut être insérée dans la tournée si les contraintes liées à la compatibilité des stations de recharge avec les véhicules électriques ainsi que les contraintes liées aux fenêtres de temps d'ouverture des stations de recharge sont vérifiées.

S'il n'est pas possible d'insérer les sommets éjectés dans une tournée déjà construite, il est alors possible de créer une nouvelle tournée. Dans ce cas, il est nécessaire d'affecter un véhicule disponible à cette tournée et ajouter le coût fixe de ce véhicule au coût total de la solution.

Quand tous les sommets sont réinsérés dans la solution courante, la solution obtenue est comparée à la solution initiale.

- **Accepter(x)** : Cette fonction décide d'accepter ou de rejeter la nouvelle solution obtenue en fonction du critère d'acceptation choisi. Dans l'algorithme de LNS proposé par Shaw [155], si la nouvelle solution est meilleure que la solution initiale, alors, l'itération suivante continue avec la nouvelle solution. Sinon, l'itération suivante continue avec la solution initiale. D'autres critères permettant d'accepter des solutions non améliorantes comme par exemple le critère d'acceptation du recuit simulé, peuvent être utilisés [158, 162]. Pour cette étude, nous avons décidé de n'accepter que les solutions améliorantes.

Après $iter_1$ itérations, la meilleure solution trouvée est sauvegardée.

Les étapes de destruction et de réparation sont illustrées dans la figure 7.13. La figure à gauche représente une solution réalisable. La figure au milieu représente une solution partielle générée à la fin de la phase de destruction qui a éjecté trois sommets de la solution initiale. La figure à droite montre la solution obtenue suite à la phase de réparation. Durant la phase de réparation, la station de recharge 7 a été insérée entre les sommets 3 et 9 et les stations de recharge 1 et 5 ont été éjectées de la solution étant donné que la recharge à ces deux stations n'est plus indispensable.

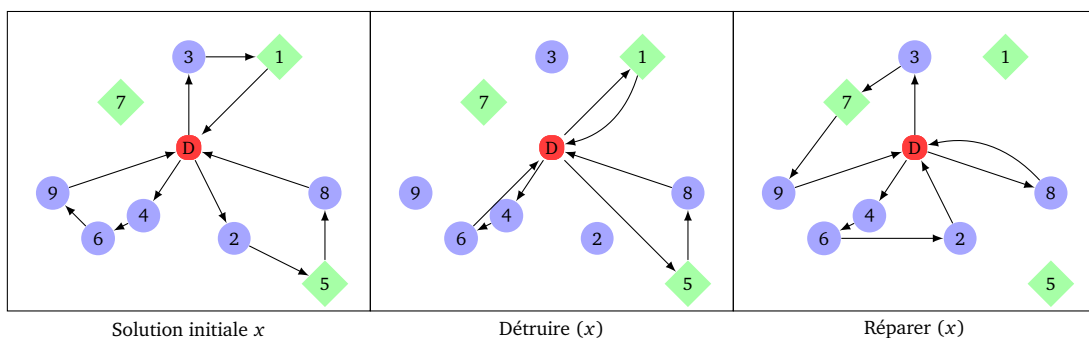


FIG. 7.13: Exemple illustratif des procédures de destruction et de réparation de l'algorithme LNS.

Dans ce qui suit, nous détaillons les trois méthodes d'insertion utilisées lors de la phase de réparation de l'algorithme LNS, à savoir la méthode d'insertion à moindre coût, la méthode d'insertion avec regret et la méthode d'insertion avec score. Pour la méthode d'insertion à moindre coût et la méthode d'insertion avec regret, il est possible de suivre soit la stratégie "First Improvement" (FI) soit la stratégie "Best Improvement" (BI) comme décrit ci-dessous.

Stratégie d'insertion Une recherche locale qui suit la stratégie FI diffère de celle qui suit la stratégie BI dans la façon avec laquelle le voisinage est exploré. Dans la stratégie BI, tout le voisinage est exploré et la meilleure solution est générée, alors que pour la stratégie FI, une solution est sélectionnée aléatoirement de l'ensemble du voisinage. Les algorithmes 13 et 14 décrivent les stratégies BI et FI, respectivement.

Algorithme 13 Best Improvement

- 1: **Entrée** : une solution initiale S
 - 2: **Sortie** : une meilleure solution S'
 - 3: $S' \leftarrow S$
 - 4: **tant que** S' n'est pas un optimum local **faire**
 - 5: Choisir $S' \in V(S)$, tel que $f(S') = \min_{x \in V(S)} f(x)$
 - 6: **si** $f(S') < f(S)$ **alors**
 - 7: $S \leftarrow S'$
 - 8: **fin si**
 - 9: **fin tant que**
-

Algorithme 14 First Improvement

- 1: **Entrée** : une solution initiale S
 - 2: **Sortie** : une meilleure solution S'
 - 3: $S' \leftarrow S$
 - 4: **tant que** S' n'est pas un optimum local **faire**
 - 5: Choisir $S' \in V(S)$ en utilisant un ordre prédéfini
 - 6: **si** $f(S') < f(S)$ **alors**
 - 7: $S \leftarrow S'$
 - 8: **fin si**
 - 9: **fin tant que**
-

i- Méthode d'insertion à moindre coût

La méthode d'insertion à moindre coût, utilisée lors de la phase de réparation de l'algorithme LNS, consiste à insérer les sommets éjectés un à un dans la solution partielle à l'endroit introduisant le coût additionnel le plus faible et respectant toutes les contraintes. Cette méthode insère les sommets un à un sans aucun ordre prédéfini. L'insertion des sommets éjectés est répétée *trial* fois en partant à chaque itération d'un nouvel ordre des sommets à injecter. Une nouvelle solution du problème est alors obtenue en sélectionnant aléatoirement un sommet parmi la liste des sommets éjectés et en l'insérant à la position qui engendre le

moindre coût.

Si l'insertion d'un sommet dans une tournée donnée engendre une violation des contraintes de capacité de véhicule et de temps de service total, cette position ne sera pas retenue. Par contre, si l'insertion d'un sommet dans une position donnée vérifie encore les contraintes de capacité de véhicule et de temps de service total mais engendre une violation des contraintes d'énergie (dans le cas où un véhicule électrique nécessite plus d'énergie pour servir le client), alors la méthode essaye de réparer la solution en insérant des chargeurs dans la tournée tout en vérifiant les contraintes de compatibilité entre le véhicule électrique et les chargeurs et de fenêtres de temps des stations de recharge. À chaque mise à jour des tournées et du planning de recharge, le coût total de la solution est mis à jour. Après *trial* itérations, la méthode d'insertion à moindre coût retourne la meilleure solution obtenue parmi les *trial* solutions générées lors de la phase de réparation de la solution. L'algorithme 15 détaille la méthode d'insertion à moindre coût. Les variables *insert_cost(j)* et *best_insert_cost(j)* représentent, respectivement, le coût supplémentaire et le meilleur coût supplémentaire générés par l'insertion d'un sommet *j* dans la solution partielle. Les variables *solution_cost* et *best_solution_cost* représentent, respectivement, le coût de la solution courante et le coût de la meilleure solution trouvée.

Algorithme 15 Méthode d'insertion à moindre coût

1: **Donnée** : Une solution partielle du EVRP-MF et une liste des sommets éjectés $Eject$
2: **Sortie** : Une solution au EVRP-MF
3: Initialement, $tr \leftarrow 0$, $best_solution_cost \leftarrow \infty$
4: **tant que** $tr < trial$ **faire**
5: Trier la liste $Eject$ aléatoirement
6: **tant que** $Eject \neq \emptyset$ **faire**
7: Choisir un sommet $j \in Eject$
8: $best_insert_cost(j) \leftarrow \infty$, $insert_cost(j) \leftarrow \infty$
9: **pour** chaque position de la tournée **faire**
10: Tester l'insertion de j dans cette position et évaluer $increase_cost(j)$
11: **si** l'insertion est possible et $insert_cost(j) < best_insert_cost(j)$ **alors**
12: $best_insert_cost(j) \leftarrow insert_cost(j)$
13: **sinon**
14: **si** l'insertion satisfait toutes les contraintes sauf celles d'énergie **alors**
15: Insérer une station de recharge b en utilisant la *Méthode d'insertion des chargeurs*
16: **si** $insert_cost(j) + insert_cost(b) < best_insert_cost(j)$ **alors**
17: $best_insert_cost(j) \leftarrow insert_cost(j) + insert_cost(b)$
18: **fin si**
19: **fin si**
20: **fin si**
21: **fin pour**
22: Évaluer le coût d'insertion de j dans une nouvelle tournée
23: **si** $insert_cost(j) < best_insert_cost(j)$ **alors**
24: $best_insert_cost(j) \leftarrow insert_cost(j)$
25: **fin si**
26: **fin tant que**
27: **si** $solution_cost < best_solution_cost$ **alors**
28: $best_solution_cost \leftarrow solution_cost$
29: Sauvegarder le meilleur ordre des sommets insérés
30: **fin si**
31: **si** First Improvement **alors**
32: Aller à l'étape 35
33: **fin si**
34: **fin tant que**
35: Insérer tous les sommets de $Eject$ et éventuellement des stations de recharge dans les positions définies

ii- Méthode d'insertion avec regret

L'inconvénient de la méthode d'insertion à moindre coût est de ne pas maîtriser l'impact de l'insertion d'un sommet donné sur les prochains sommets à insérer. En effet, il est possible que l'insertion d'un sommet donné oblige les sommets difficiles à placer à être insérés au moment

où il reste peu de possibilités d'insertion. Un algorithme développé par Hassin [163] tente de remédier à ce problème en proposant une approche d'insertion avec regret qui permet d'annuler, lors de chaque itération, une ou plusieurs insertions effectuées lors des itérations précédentes. En d'autres mots, au cours de l'algorithme, nous pouvons revenir en arrière en annulant certaines insertions que nous finirons par regretter.

La méthode d'insertion avec regret place les sommets éjectés dans les positions engendrant le moindre coût comme il est le cas pour la méthode d'insertion à moindre coût. En revanche, lors de l'insertion d'un sommet éjecté dans la solution partielle, il est possible d'annuler des insertions précédemment effectuées si ceci permettrait de trouver une meilleure insertion pour le sommet en question.

À chaque étape, le coût de la meilleure insertion du sommet en question et le coût économisé en supprimant un sommet de la solution partielle courante sont calculés. Le coût économisé en éjectant un sommet v est égal au coût de la solution courante incluant le sommet v moins le coût de la nouvelle solution obtenue en éjectant v . Par exemple, soit une solution S composée, entre autres, d'une tournée affectée à un VE k . Cette tournée est constituée, dans l'ordre, des clients u , v et w , le coût d'éjection de v est : $eject_cost(v) = cost_{u,v,k}^{VE} + cost_{v,w,k}^{VE} - cost_{u,w,k}^{VE}$. Le coût de la meilleure insertion d'un client z est égal à la différence entre le coût de la solution obtenue après l'insertion de k dans sa meilleure position et le coût de la solution actuelle. Les suppressions et réinsertions des sommets restent alors temporaires tout au long de l'algorithme et ne deviennent définitives que lorsque tous les sommets éjectés sont de nouveau insérés dans la solution.

La Figure 7.14 donne un exemple illustratif de la méthode d'insertion avec regret. Soit une tournée initiale composée des sommets 1, 2, 3, 4 et 5 ayant les coordonnées (1,-1), (-1,4), (3,4), (1,4) et (-1,-4), respectivement. Supposons que le coût opérationnel du véhicule sélectionné soit égal à 1. Lors de la phase de destruction, quatre sommets (1, 2, 3 et 5) ont été éjectés (Figure (a)). Supposons que le tri aléatoire de la liste des sommets éjectés donne la liste {1, 3, 2, 5}. Le sommet 1 est alors le premier à être réinséré dans la tournée. Il peut être inséré entre le sommet 4 et le dépôt (Figure (b)) avec $insert_cost(1) = 4$. De la même façon, le sommet 3 peut être inséré dans sa meilleure position entre les sommets 4 et 1 avec $insert_cost(3) = 2.38$. Mais, si on calcule le coût d'éjection du sommet 1, on trouve $eject_cost(1) = 4$. On a ainsi $eject_cost(1) > insert_cost(3)$. L'algorithme annule alors l'insertion de 1 et place 3 entre le dépôt et 4 (Figure (c)). En suivant la même logique, le sommet 2 est inséré entre le dépôt et le sommet 4 (Figure (d)). Le sommet 5 a été par la suite inséré entre les sommets 2 et 4 (Figure (e)) et le sommet 1 a été de nouveau inséré entre le sommet 3 et le dépôt (Figure (f)). Tous les sommets éjectés lors de la phase de destruction sont de nouveau réinsérés dans la solution courante. Toutes les insertions deviennent alors définitives.

L'Algorithme 16 décrit la méthode d'insertion avec regret. Soit $trial$ le nombre de fois la méthode d'insertion avec regret doit être répétée pour chaque liste de sommets éjectés. Notons S la solution courante et $best_solution_cost$ le coût de la meilleure solution trouvée.

iii- Méthode d'insertion avec score

La méthode d'insertion avec score est basée sur l'idée d'associer un score à chaque sommet à insérer dans la solution partielle. Cette idée est inspirée du Parallel Regret Algorithm qui a été

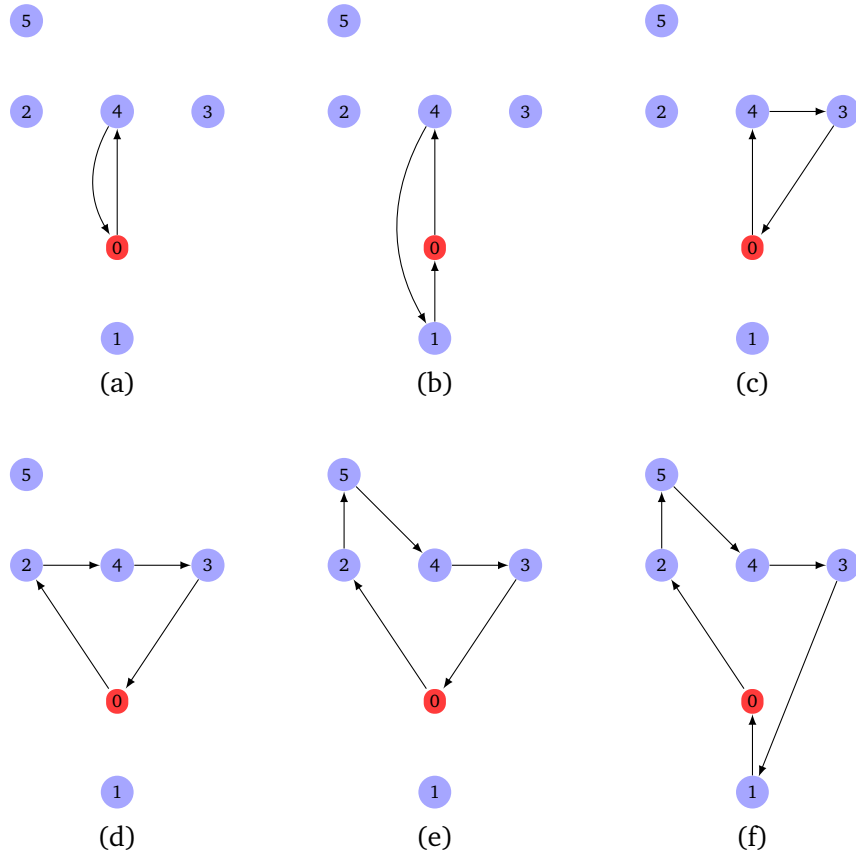


FIG. 7.14: Exemple illustratif de la méthode d'insertion avec regret

introduit dans [164].

Pour cette méthode, la liste *Eject* est seulement composée de clients. Par contre, il est toujours possible d'éjecter et d'insérer des chargeurs pour réparer la solution. Un score est associé à chaque sommet de la liste *Eject*. Il traduit la difficulté d'insérer le sommet dans la solution courante et il est utilisé pour choisir le sommet j à insérer dans la solution à l'itération suivante. Pour évaluer la difficulté liée à l'insertion d'un sommet, nous avons considéré trois éléments. Pour chaque sommet $j \in Eject$, le score $score(j)$ du sommet j est la somme : (i) de la pénalité $penalty(j, r)$ liée au fait de ne pas insérer le sommet j dans sa meilleure tournée ; (ii) de la distance entre j et la plus proche station de recharge disponible (d_{j, f_c}) ; (iii) du nombre de stations de recharge qui peuvent être atteintes par le véhicule électrique après avoir visité le client j (η_f).

Ainsi, $score(j) = penalty(j, r) + d_{j, f_c} - \eta_f$, où $penalty(j, r)$ est la différence entre le coût d'insertion du sommet j dans la deuxième meilleure position et le coût de son insertion dans la meilleure position. Si un client ne peut être inséré que dans une seule position, la pénalité $penalty(j, r)$ prend une grande valeur pour imposer l'insertion du client dans la meilleure position.

L'Algorithme 17 détaille la méthode d'insertion avec score.

Algorithme 17 Algorithme d'Insertion avec Score

- 1: **Donnée** : Une solution partielle et une liste $Eject$ de sommets éjectés
 - 2: **Sortie** : Une solution au EVRP-MF
 - 3: **pour** chaque client $j \in Eject$ **faire**
 - 4: Calculer les coûts engendrés par l'insertion de j dans chaque position de la tournée en considérant la possibilité d'injecter un chargeur quand nécessaire pour réparer la solution
 - 5: Calculer $score(j)$
 - 6: **fin pour**
 - 7: **tant que** $Eject \neq \emptyset$ **faire**
 - 8: Sélectionner le client j^* ayant une valeur maximale de score
 - 9: Insérer j^* dans la meilleure position
 - 10: Supprimer j^* de $Eject$
 - 11:
 - 12: **pour** chaque $j \in Eject$ **faire**
 - 13: Mettre à jour $score(j)$
 - 14: **fin pour**
 - 15: **fin tant que**
-

Maintenant, nous avons toutes les méthodes pour présenter l'algorithme LNS. Dans ce qui suit, l'Algorithme 18 décrit la méthode de LNS. Dans la ligne 3, la solution courante et la meilleure solution trouvée sont initialisées. Dans les lignes 7 et 8, l'algorithme applique la méthode de destruction et ensuite la méthode de réparation pour obtenir la nouvelle solution S'' . Lors de l'insertion des sommets éjectés dans la solution courante, il est également possible d'injecter des chargeurs en utilisant la *méthode d'insertion des chargeurs*. Dans la ligne 8, la nouvelle solution est évaluée et l'algorithme choisit soit d'accepter ou de rejeter la nouvelle solution obtenue. La ligne 8 vérifie si la nouvelle solution est meilleure que la meilleure solution obtenue jusqu'ici. Le coût total d'une solution S est noté $cost(S)$. La valeur de la fonction objectif de la meilleure solution et la meilleure solution sont mises à jour, si nécessaire, dans les lignes 11 et 12. La ligne 16 retourne la meilleure solution trouvée.

Algorithme 18 Algorithme de Recherche Locale à Grand Voisinage

```
1: Entrée : Une instance du problème
2: Sortie : Une meilleure solution  $S'$ 
3:
4: pour  $d = 0; d < Iter_1$  faire
5:   Générer une solution initiale  $S_0$ 
6:    $S \leftarrow S_0, S' \leftarrow S_0, best\_solution\_cost \leftarrow cost(S_0)$ 
7:   pour  $a = 0; a < Iter_2$  faire
8:     Détruire( $S$ )
9:      $S'' \leftarrow Rparer(S)$ 
10:    si  $Accepter(S'')$  alors
11:       $S \leftarrow S''$ 
12:    fin si
13:    si  $cost(S'') < best\_solution\_cost$  alors
14:       $best\_solution\_cost \leftarrow cost(S'')$ 
15:       $S' \leftarrow S''$ 
16:    fin si
17:  fin pour
18: fin pour
19: Retourner  $S'$ 
```

7.5 Étude comparative

7.5.1 Implémentation et jeux de test

Les algorithmes ont été programmés en C++ et exécutés sur un processeur Intel Xeon E5620 2.4GHz avec 8GB de mémoire RAM. Pour l'évaluation numérique, nous avons adapté des instances de la littérature qui ont été initialement proposées par Solomon [165] et ensuite utilisées par Schneider et al. [80].

Chaque instance est composée de 100 clients et 21 stations de recharge. Ces instances sont divisées en plusieurs classes en se basant sur la distribution géographique des localisations des clients. La classe (R) représente une distribution aléatoire des clients, la classe (C) représente une distribution en clusters et la classe (RC) est un mélange entre les deux premières classes. Chaque classe peut être divisée en une sous-classe avec des petites capacités de véhicules et une sous-classe avec des capacités d'emport des véhicules plus importantes. Nous pouvons alors parler de 6 classes d'instances qui sont R1, C1, R2, C2, RC1 et RC2. Les instances de la même classe diffèrent en fonction des demandes des clients et des caractéristiques des stations de recharge. La technologie de recharge, le coût de recharge et les fenêtres de temps d'ouverture des différentes stations de recharge sont déterminés d'une façon aléatoire tout en s'assurant qu'à chaque période de temps, au moins $n/4$ stations de recharge sont disponibles dont au moins 50% de stations avec des bornes de recharge accélérée. Les bornes de recharge situées au dépôt sont disponibles pendant tout l'horizon de planification. Chaque station de recharge propose d'effectuer soit une recharge normale soit une recharge accélérée. Au dépôt, seulement la recharge normale est possible. Les coûts de la recharge normale varient entre 0.157 et 0.168

(euro/kWh). Les coûts de la recharge accélérée varient entre 0.170 et 0.180 (euro/kWh). La flotte de véhicules est composée de $n/2$ véhicules thermiques, $n/4$ véhicules électriques avec une capacité de batterie de 22 (kWh) et $n/4$ véhicules électriques avec une capacité de batterie de 16 (kWh). Seulement les véhicules avec une capacité de batterie de 22 (kWh) sont compatibles avec la recharge accélérée. Nous considérons que tous les VEs sont complètement chargés au début de l'horizon d'optimisation. Le tableau 7.4 représente les caractéristiques principales des différentes classes de données considérées. Pour notre étude, nous avons fait le choix de diviser par 10 le temps de service des clients et l'horizon de planification afin de se rapprocher le plus possible de notre application réelle.

TAB. 7.4: Caractéristiques des instances

Classe	Coord.	Demande	Capacité d'emport maximale	Temps de service
R1	a	A	200	1
C1	b	B	200	9
RC1	d	C	200	1
R2	a	A	1000	1
C2	c	B	700	9
RC2	d	C	1000	1

7.5.2 Choix des paramètres des algorithmes

Concernant les paramètres comme le nombre maximum d'itérations, nous avons effectué une phase préliminaire de test pour déterminer les valeurs standards de nos paramètres avec comme objectif de trouver le meilleur compromis entre la qualité des solutions générées et le temps de calcul. Le tableau 7.5 présente les valeurs choisies pour les paramètres fixes de notre algorithme LNS.

TAB. 7.5: Paramètres de l'algorithme LNS

Paramètre	Rôle	Valeur
<i>Iter1</i>	nombre maximum d'itérations de l'algorithme	100
<i>Iter2</i>	nombre maximum d'itérations lors de la procédure destruction/réparation	30
<i>TimeOut</i>	temps maximum de calcul	30 (mn)

Concernant les paramètres comme le degré de destruction de la solution, nous avons choisi d'effectuer un grand nombre de tests afin de déterminer la meilleure combinaison possible de paramètres. Ceci fait l'objet de la section suivante.

7.5.3 Résultats numériques

Dans cette section, nous proposons d'étudier la performance de nos algorithmes de recherche locale et de notre algorithme LNS. Nous étudions également l'impact des différents paramètres et des différentes méthodes d'insertion de l'algorithme LNS sur la qualité des solutions obtenues.

a. Performance des algorithmes de recherche locale

Dans cette section, nous présentons les résultats obtenus en utilisant l'algorithme CRC et nos algorithmes de recherche locale. En appliquant les algorithmes One Point Move, Or-Opt et Réaffectation des tournées sur les solutions initiales générées par l'algorithme CRC, nous n'avons remarqué aucune amélioration par rapport à la qualité des solutions obtenues. Ceci peut être expliqué par le fait que les solutions initiales soient des optimum locaux par rapport à ces recherches locales.

Le tableau 7.6 récapitule les résultats obtenus par les algorithmes TPM, 2-opt et Concaténation des tournées. Pour chaque catégorie d'instances, la colonne " S_0 " donne le coût de la solution initiale générée par l'algorithme CRC, la colonne " R_0 " représente le nombre de tournées de la solution initiale, la colonne "Gap" donne l'écart de la solution de recherche locale par rapport à la solution initiale. Ce gap, exprimé en %, est calculé comme suit : $Gap(S_{LS}) = \frac{Obj(S_{CRC}) - Obj(S_{LS})}{Obj(S_{LS})}$. La colonne "CPU" précise le temps de calcul en secondes et les colonnes "VE" et "VT" donnent le nombre de VE et VT utilisés dans les solutions de recherche locale, respectivement.

TAB. 7.6: Résultats des algorithmes de recherche locale

Ins.	CRC		TPM				2-Opt				ConcaT			
	S_0	R_0	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	780.94	28	0.50	8.53	22	6	0.26	73.21	22	6	0.52	212.36	22	6
C2	641.48	25	0.14	8.21	23	2	0.14	71.65	23	2	0.73	268.36	23	2
R1	563.84	19	0.76	10.42	12	7	0.33	70.12	12	7	0.00	124.96	12	7
R2	358.33	9	0.93	8.16	7	2	1.19	70.36	7	2	1.75	159.13	7	2
RC1	833.89	31	0.43	8.5	24	7	0.25	71.22	24	7	4.96	216.89	24	7
RC2	648.59	25	0.14	8.76	23	2	0.25	70.14	23	2	2.08	287.16	23	2
Moy.	637.84	23	0.48	8.76	18	4	0.40	71.12	18	4	1.67	211.48	18	4

Les algorithmes de recherche locale TPM, 2-opt et Concaténation des tournées permettent d'améliorer légèrement la qualité des solutions obtenues. En effet, une amélioration d'environ 0.50% en moyenne est constatée pour les algorithmes TPM et 2-Opt. L'algorithme Concaténation des tournées a permis d'améliorer les solutions initiales de plus de 1.5% en moyenne. Pour tous les algorithmes de recherche locale testés, aucune amélioration par rapport au nombre de véhicules utilisés n'a été constatée.

Les algorithmes TPM et 2-opt semblent être plus efficaces sur les classes d'instance R1 et R2 avec une amélioration d'environ 1% pour R2. En revanche, l'algorithme Concaténation des tournées est plus efficace sur les classes d'instances RC1 et RC2 avec une amélioration d'environ 5% pour la classe RC1. Concernant le temps de calcul, c'est l'algorithme TPM qui semble être le plus rapide avec une moyenne de 8 secondes. L'algorithme Concaténation des tournées nécessite, quant à lui, environ 200 secondes pour générer des solutions.

b. Performance de l'algorithme LNS

Pour évaluer la performance de notre algorithme LNS, nous avons varié les paramètres utilisés et comparé les résultats en fonction de la qualité des solutions obtenues et du temps de calcul. Les paramètres variés sont les suivants :

- Le nombre des sommets éjectés à chaque itération de l'algorithme LNS

- La méthode d’insertion des sommets (Insertion à moindre coût, insertion avec regret et insertion avec score)
- La stratégie suivie lors de l’insertion des sommets (First Improvement et Best Improvement)

Notre objectif ici est d’obtenir la meilleure combinaison de paramètres qui permettrait de générer les meilleures solutions possibles.

i- Impact du degré de destruction sur la qualité de la solution

Pour comprendre l’impact du degré de destruction de l’algorithme LNS sur la qualité de la solution, nous avons développé trois variantes de notre algorithme LNS :

- La première variante LNS1 consiste à fixer la valeur du paramètre *Num* à l’avance et d’utiliser la même valeur tout au long de l’algorithme.
- La deuxième variante LNS2 consiste à varier la valeur du paramètre *Num* d’une façon aléatoire à chaque itération de l’algorithme.
- La troisième variante LNS3 consiste à augmenter la valeur du paramètre *Num* au fur à mesure de l’avancement de l’algorithme. À chaque itération de l’algorithme, cette valeur sera alors incrémentée.

Nous commençons par exposer les résultats obtenus en utilisant la variante *LNS1*. Les tableaux 7.7 et 7.8 présentent les résultats obtenus en variant le nombre de sommets éjectés et en appliquant la méthode d’insertion à moindre coût et la stratégie First Improvement. Pour chaque instance, la colonne "*S₀*" donne le coût de la solution initiale générée par l’algorithme CRC, la colonne "*R₀*" représente le nombre de tournées de la solution initiale, la colonne "Gap" donne l’écart de la solution de LNS par rapport à la solution initiale. Ce gap, exprimé en %, est calculé comme suit : $Gap(S_{LNS}) = \frac{Obj(S_{CRC}) - Obj(S_{LNS})}{Obj(S_{LNS})}$. La colonne "CPU" précise le temps de calcul de LNS en minutes et les colonnes "VE" et "VT" donnent le nombre de VE et VT utilisés dans la solution de LNS, respectivement.

Les résultats des tableaux 7.7 et 7.8 montrent que l’algorithme LNS améliore les solutions initiales de plus de 50% dans la majorité des cas. Nous remarquons que pour les classes d’instances R1 et R2, les améliorations de l’algorithme LNS sont moins importantes que celles pour les autres classes.

Les résultats obtenus prouvent également qu’un degré de destruction élevé permet d’obtenir de bons résultats mais consomme plus de temps de calcul. En effet, une augmentation de 1% du degré de destruction par rapport au nombre total de clients améliore, en moyenne, la qualité des solutions générées de plus de 2%. Le temps de calcul augmente également et passe d’environ 6 minutes pour *Num* = 2 à environ 24 minutes pour *Num* = 10. Concernant le nombre de véhicules utilisés, 23 véhicules ont été utilisés dans les solutions initiales contre seulement 9 véhicules, dont 4 VEs, pour les solutions générées par LNS dans le cas où *Num* = 10. En revanche, si le degré de destruction dépasse 10% du nombre total des clients, les résultats obtenus ont tendance à se dégrader (cas de *Num* = 11 et *Num* = 12) avec un temps de calcul encore plus important. Pour obtenir des solutions de bonne qualité avec un temps de calcul raisonnable, nous concluons alors qu’il est préférable de choisir un degré de destruction qui ne dépasse pas 10% du nombre total de clients.

Concentrons-nous maintenant sur les performances des variantes LNS2 et LNS3. Le tableau 7.9 présente les résultats obtenus en utilisant les variantes LNS2 et LNS3 tout en appliquant la

TAB. 7.7: Résultats de performance de CRC et LNS1 sur des instances généralisées de la littérature (Partie 1)

Ins.	CRC	Num = 2			Num = 3			Num = 4			Num = 5			Num = 6								
		Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT					
C1	780.94	28	44.48	6.77	10	6	49.56	9.01	8	6	53.09	12.11	7	6	53.67	14.88	7	7	57.71	17.88	6	6
C2	641.48	25	73.91	6.44	8	2	91.35	8.77	6	2	87.53	12.22	6	2	95.54	17.55	6	2	96.96	16.33	6	2
R1	563.84	19	15.70	4.66	7	7	12.43	5.77	7	7	16.33	7.11	6	7	15.72	10.66	6	7	15.57	10.88	5	8
R2	358.33	9	14.15	8.55	5	2	21.43	11.11	4	2	21.80	14.00	4	2	25.34	18.44	4	2	31.92	19.88	3	2
RC1	833.89	31	63.86	6.88	8	7	67.36	9.22	7	7	72.68	11.44	6	7	73.54	15.11	6	7	75.57	15.66	5	8
RC2	648.59	25	76.87	6.55	9	2	88.09	9.22	7	2	96.72	12.66	6	2	87.07	14.44	7	2	105.43	12.22	5	2
Moy.	637.84	23	48.16	6.64	8	4	55.04	8.85	7	5	58.02	11.59	6	4	58.48	15.18	6	5	63.86	15.48	5	5

TAB. 7.8: Résultats de performance de LNS1 sur des instances généralisées de la littérature (Partie 2)

Ins.	Num = 7			Num = 8			Num = 9			Num = 10			Num = 11			Num = 12								
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	58.18	21.77	5	7	58.48	21.77	5	7	64.30	22.22	5	7	71.24	26.22	5	7	62.19	29.22	5	7	66.95	35.55	5	7
C2	101.95	19.55	5	2	104.82	21.88	4	2	107.83	19.55	4	2	107.15	25.01	5	2	106.92	24.88	4	2	103.95	34.88	4	2
R1	15.44	13.88	6	8	16.85	12.88	5	8	16.26	13.55	4	8	16.10	17.22	5	8	16.91	21.11	5	8	18.24	22.55	5	8
R2	28.32	20.00	3	2	31.97	19.88	3	2	31.23	20.22	3	2	35.50	24.33	3	2	35.53	25.11	2	2	34.65	30.44	3	2
RC1	75.16	20.00	5	7	75.90	22.22	6	7	77.75	18.66	4	8	77.61	26.77	4	8	75.09	27.88	5	7	77.48	30.50	5	8
RC2	121.74	19.00	4	2	119.45	20.88	5	2	124.03	21.55	4	2	127.45	24.00	4	2	120.34	25.44	4	2	127.30	33.25	3	2
Moy.	66.80	19.03	5	5	67.91	19.92	5	5	70.23	19.29	4	5	72.51	23.93	4	5	69.50	25.61	4	5	71.43	30.66	4	5

TAB. 7.9: Résultats obtenus avec LNS2 et LNS3

Ins.	LNS2				LNS3			
	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	59.24	17.33	6	7	54.38	21.78	7	6
C2	103.89	22.75	5	2	98.44	20.00	5	2
R1	15.42	10.56	5	8	15.46	13.11	5	8
R2	30.29	22.38	3	2	28.90	23.75	3	2
RC1	75.67	15.67	5	8	73.65	18.75	5	8
RC2	119.02	24.50	4	2	111.14	26.50	5	2
Moy.	67.26	18.86	5	5	63.66	20.65	5	5

méthode d'insertion à moindre coût et la stratégie First Improvement. Concernant la variante LNS2, nous choisissons de varier aléatoirement la valeur du paramètre *Num* entre 1 et 10.

En se basant sur les résultats obtenus, nous remarquons que la variante LNS2 permet de générer des bons résultats qui sont très proches de ceux générés par LNS1 dans le cas où *Num* = 8. Le temps de calcul est également légèrement inférieur à celui de LNS1.

Concernant la variante LNS3, les solutions initiales ont été améliorées en moyenne d'environ 64% avec un temps de calcul moyen d'environ 20 minutes.

ii- Impact de la stratégie d'insertion sur la qualité de la solution

Dans ce paragraphe, nous nous intéressons à la qualité des solutions obtenues en utilisant la stratégie "Best Improvement". Le tableau 7.10 résume les résultats obtenus en utilisant la stratégie BI de la méthode d'insertion à moindre coût. Vu que la stratégie BI consomme plus de temps que la stratégie FI, nous avons au début limité le temps de calcul à 30 minutes (deux premières colonnes de la tableau 7.10) et nous l'avons par la suite élargi pour autoriser un temps maximum de calcul d'une heure (deux dernières colonnes du tableau 7.10).

TAB. 7.10: Résultats obtenus pour la stratégie BI

Ins.	Num = 4				Num = 6				Num = 3				Num = 5			
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	44.55	30	9	6	46.33	30	9	6	53.43	58.01	8	6	54.00	60.00	8	6
C2	73.33	30	9	2	75.66	30	9	2	80.92	58.51	8	2	82.36	59.09	7	2
R1	13.80	30	7	7	14.58	30	7	7	16.53	46.33	7	7	17.34	58.83	6	7
R2	10.50	30	6	2	10.52	30	6	2	14.56	59.43	5	2	15.23	60.01	5	2
RC1	57.32	30	9	7	59.45	30	9	7	64.97	57.66	8	7	68.37	60.01	7	7
RC2	68.95	30	9	2	68.95	30	9	2	82.49	58.88	8	2	89.21	59.82	7	2
Moy.	44.74	30	8	4	45.92	30	8	4	52.15	56.47	7	4	54.42	59.63	7	4

Les résultats obtenus prouvent que la stratégie "First Improvement" permet d'obtenir de meilleurs résultats par comparaison à la stratégie "Best Improvement". Pour un temps de calcul maximal de 30 minutes, nous passons d'un gap de 58.08% (respectivement 63.86%) à un gap de 44.74% (respectivement 45.92%) pour *Num* = 4 (respectivement *Num* = 6), soit une dégradation de la qualité des solutions d'environ 15%. Pour un temps de calcul maximal d'une heure, le gap passe de 55.04% (respectivement 58.48) à 52.15% (respectivement 54.42%)

pour $Num = 3$ (respectivement $Num = 5$) si une stratégie BI est appliquée. Les solutions obtenues sont ainsi dégradées d'environ 4% par rapport à celles obtenues en appliquant la stratégie FI et ceci pour un temps de calcul beaucoup plus important. Nous pouvons alors conclure que l'utilisation de la stratégie FI permet d'obtenir des meilleurs résultats pour notre problème.

iii- Impact de la méthode d'insertion sur la qualité de la solution

Dans cette partie, nous nous intéressons à la qualité des solutions obtenues en fonction de la méthode d'insertion choisie lors de la phase de réparation de la solution courante. Nous présentons alors les résultats obtenus pour les méthodes d'insertion avec regret et avec score. Les Tableaux 7.11 et 7.12 récapitulent les résultats obtenus en appliquant la méthode d'insertion avec score. D'après les résultats des tableaux 7.11 et 7.12, nous remarquons une amélioration d'environ 2% dans la qualité des solutions obtenues par comparaison aux solutions obtenues en utilisant la méthode d'insertion à moindre coût. Cette amélioration est accompagnée d'une augmentation de quelques minutes dans le temps de calcul.

Les tableaux 7.13 et 7.14 présentent les résultats obtenus en utilisant la méthode d'insertion avec regret. Les résultats des tableaux 7.13 et 7.14 prouvent que la méthode d'insertion avec regret permet d'améliorer encore plus les solutions initiales par comparaison à la méthode d'insertion à moindre coût et la méthode d'insertion avec score. Pour les cas où $Num = 2$ et $Num = 3$, la méthode d'insertion avec regret génère des solutions qui sont très proches de celles générées en utilisant la méthode d'insertion à moindre coût. Pour les cas où la valeur de Num varie entre 4 et 8, la méthode d'insertion avec regret permet d'améliorer considérablement les solutions générées en utilisant la méthode d'insertion à moindre coût. Cette amélioration atteint une moyenne d'environ 6% pour $Num = 5$. Dans le cas où Num dépasse 9, nous ne remarquons pas une réelle amélioration des solutions générées. De plus, le temps de calcul augmente considérablement et dépasse les 30 minutes.

D'après les différents résultats obtenus, nous pouvons alors conclure que la méthode d'insertion à moindre coût combinée avec la stratégie First Improvement permet d'obtenir des résultats satisfaisants en un temps de calcul raisonnable.

7.6 Conclusion

Dans ce chapitre, nous avons introduit un nouveau problème de tournées de véhicules en considérant une flotte composée de véhicules électriques hétérogènes et de véhicules thermiques identiques. Nous avons également pris en compte plusieurs contraintes inspirées des cas d'usage réels et qui n'ont jamais été considérées dans la littérature.

Pour résoudre ce problème, nous avons développé une heuristique constructive pour générer des solutions initiales. Nous avons par la suite adapté plusieurs algorithmes classiques de recherche locale et introduit deux nouveaux algorithmes pour améliorer les solutions initiales. Ensuite, nous avons proposé un algorithme de recherche locale à grand voisinage. Pour tester les performances des différentes méthodes développées, nous avons généralisé des instances de test de la littérature. D'après les résultats obtenus, l'algorithme de recherche locale à grand voisinage semble être très performant. Pour la majorité des instances, il a, en effet, permis d'améliorer la qualité des solutions générées de plus de 50% par rapport aux solutions initiales et ceci en un temps de calcul raisonnable.

TAB. 7.1.1: Résultats de performance de LNS avec la méthode d'insertion avec score (Partie 1)

Ins.	Num = 2			Num = 3			Num = 4			Num = 5			Num = 6			Num = 7								
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	46.71	7.22	8	6	50.41	11.33	8	6	55.92	19.62	7	6	54.14	21.12	7	7	57.97	20.05	6	7	57.72	25.34	6	7
C2	70.13	7.01	9	2	81.19	12.43	6	2	89.13	15.72	6	2	100.12	20.22	5	2	98.32	22.65	5	2	98.89	21.12	6	2
R1	15.02	5.12	7	7	14.67	6.35	7	7	16.01	12.11	6	7	15.32	12.43	7	7	16.07	16.71	5	8	16.01	16.73	5	8
R2	13.93	10.73	5	2	19.21	16.43	5	2	20.99	16.38	4	2	26.13	22.73	3	2	28.32	22.68	4	2	28.89	22.14	3	2
RC1	64.11	7.21	8	7	70.45	12.01	7	7	73.01	14.74	7	7	71.32	19.43	6	7	74.32	23.12	6	7	74.01	21.36	6	7
RC2	80.19	6.98	8	2	91.11	11.99	6	2	100.14	18.42	5	2	101.63	25.24	5	2	108.21	22.43	5	2	119.54	24.82	4	2
Moy:	48.34	7.37	8	4	54.50	11.75	6	4	59.20	16.16	6	4	61.44	20.19	6	4	63.86	21.27	5	5	65.84	21.91	5	5

TAB. 7.1.2: Résultats de performance de LNS avec la méthode d'insertion avec score (Partie 2)

Ins.	Num = 8			Num = 9			Num = 10			Num = 11			Num = 12							
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	57.99	23.49	8	6	61.32	30.22	6	7	68.57	23.34	4	7	62.17	32.11	5	7	67.15	38.22	4	7
C2	101.43	25.13	9	2	103.15	24.66	5	2	108.01	23.33	4	2	108.33	33.56	4	2	106.21	31.13	4	2
R1	14.18	14.54	7	7	17.63	27.73	4	8	15.91	21.57	5	8	16.12	28.76	3	9	17.12	32.43	4	8
R2	31.03	27.55	5	2	33.16	32.45	2	2	30.35	23.66	3	2	36.23	30.22	2	2	37.03	35.34	2	2
RC1	74.89	26.24	8	7	75.99	22.45	6	7	77.32	28.12	5	7	76.55	32.22	5	8	76.89	40.55	5	8
RC2	120.13	32.77	8	2	126.34	27.34	4	2	121.66	25.47	5	2	121.09	33.23	4	2	130.67	42.18	3	2
Moy:	66.60	24.95	8	4	69.59	27.47	4	5	70.30	24.24	4	5	70.08	31.68	4	5	75.51	36.64	4	5

TAB. 7.13: Résultats de performance de LNS avec la méthode d'insertion avec regret (Partie 1)

Ins.	Num = 2			Num = 3			Num = 4			Num = 5			Num = 6			Num = 7								
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	51.50	8.00	8	6	51.45	14.66	8	6	55.92	19.62	7	6	55.34	21.12	7	7	59.03	23.19	6	7	56.10	28.91	6	7
C2	69.43	7.87	9	2	90.49	14.12	6	2	91.39	17.93	6	2	103.77	23.19	5	2	101.17	25.28	5	2	96.17	23.72	6	2
R1	14.54	5.33	7	7	16.72	9.00	7	7	16.12	14.23	6	7	15.14	14.28	7	7	16.17	16.87	5	8	16.32	17.98	5	8
R2	13.83	14.00	5	2	17.78	18.62	5	2	20.72	20.64	4	2	29.69	25.87	3	2	25.72	24.98	4	2	29.02	25.62	3	2
RC1	64.75	8.33	8	7	73.09	14.66	7	7	74.30	18.61	7	7	70.53	22.91	6	7	72.47	24.98	6	7	74.11	23.81	6	7
RC2	81.08	8.62	8	2	98.35	16.87	6	2	102.64	22.01	5	2	113.23	27.17	5	2	109.51	25.67	5	2	117.65	26.71	4	2
Moy.	49.19	8.69	8	4	57.98	14.66	6	4	60.18	18.84	6	4	64.62	22.42	6	4	64.01	23.50	5	5	64.89	24.46	5	5

TAB. 7.14: Résultats de performance de LNS avec la méthode d'insertion avec regret (Partie 2)

Ins.	Num = 8			Num = 9			Num = 10			Num = 11			Num = 12							
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	57.50	29.13	8	6	58.46	33.17	6	7	64.35	36.87	4	7	63.17	42.98	5	7	70.60	46.19	4	7
C2	99.78	26.78	9	2	102.05	34.01	5	2	109.15	32.85	4	2	118.62	46.18	4	2	116.88	41.89	4	2
R1	17.98	16.51	7	7	18.11	29.18	4	8	15.51	25.17	5	8	16.20	36.63	3	9	16.85	38.19	4	8
R2	31.83	29.78	5	2	39.49	35.17	2	2	29.99	32.17	3	2	37.24	40.17	2	2	39.20	45.83	2	2
RC1	74.75	28.98	8	7	75.84	32.89	6	7	77.04	31.89	5	7	78.13	40.84	5	8	74.33	48.18	5	8
RC2	121.76	35.62	8	2	131.89	41.19	4	2	118.32	30.06	5	2	122.70	48.35	4	2	142.11	52.37	3	2
Moy.	67.27	27.80	8	4	70.97	34.27	4	5	69.06	31.50	4	5	72.68	42.53	4	5	76.66	45.44	4	5

Algorithme 16 Méthode d'insertion avec regret

1: **Entrée** : Une solution partielle S_0 et une liste $Eject$ de sommets éjectés
2: **Sortie** : Meilleure solution trouvée S'
3: $S \leftarrow S_0$, $best_solution_cost \leftarrow +\infty$, $node_to_eject \leftarrow -1$, $max_eject_cost \leftarrow -\infty$
4: **pour** ($tr = 0$; $tr < trial$) **faire**
5: Trier aléatoirement la liste $Eject$
6: **pour** ($j = 0$; $j < Num$) **faire**
7: Trouver la meilleure position p^* pour insérer $Eject[j]$ et éventuellement la meilleure station de recharge f^*
8: $max_eject_cost \leftarrow -\infty$
9: **pour** ($k = 0$; $k < j$) **faire**
10: **si** $Eject[k] \neq p^*$ et $eject_cost(Eject[k]) > max_eject_cost$ **alors**
11: $node_to_eject \leftarrow Eject[k]$
12: $max_eject_cost \leftarrow eject_cost(Eject[k])$
13: **fin si**
14: **fin pour**
15: **si** $node_to_eject = -1$ ou $insert_cost(Eject[j]) \geq max_eject_cost$ **alors**
16: Insérer $Eject[j]$ dans p^*
17: Insérer éventuellement f^*
18: **sinon**
19: Éjecter $node_to_eject$
20: Insérer $Eject[j]$ dans p^* ainsi qu'éventuellement f^*
21: $Eject[j] \leftarrow node_to_eject$
22: $j \leftarrow j - 1$
23: **fin si**
24: **fin pour**
25: **si** $cost(S) < best_solution_cost$ **alors**
26: $best_solution_cost \leftarrow cost(S)$
27: $S' \leftarrow S$
28: **fin si**
29: **si** First Improvement **alors**
30: Aller à l'étape 32
31: **fin si**
32: **fin pour**
33: Retourner S'

Chapitre 8

Approches méta-heuristiques basées sur LNS

Sommaire

8.1 Introduction	134
8.2 ILS avec LNS intégré	135
8.2.1 Principe de ILS	135
8.2.2 ILS avec LNS intégré pour EVRP-MF	137
8.3 Recherche à Voisinage Variable avec LNS intégré	138
8.3.1 Principe de VNS	139
8.3.2 VNS avec LNS intégré pour EVRP-MF	140
8.4 Étude expérimentale	141
8.4.1 Performance de l'algorithme ILS	141
8.4.2 Performance de l'algorithme VNS	145
8.4.3 Étude comparative	147
8.5 Conclusion	150

Ce chapitre représente la suite logique du chapitre précédent. Il évoque le même problème mais propose des approches méta-heuristiques hybrides pour le résoudre.

8.1 Introduction

La littérature scientifique propose une grande variété de méthodes pour la résolution des problèmes d'optimisation combinatoire. À part l'utilisation des méthodes exactes et des méta-heuristiques, une autre approche de résolution pourrait consister à créer des algorithmes hybrides. Les algorithmes hybrides ont commencé à prouver leur efficacité et ils ont permis de trouver de bons résultats pour plusieurs problèmes d'optimisation combinatoire. Il existe plusieurs types d'hybridations possibles :

- Hybridation Recherche Locale/méthodes évolutives : La recherche Locale permet d'explorer efficacement les régions prometteuses détectées par les méthodes évolutives. Les algorithmes mémétiques [166] hybrident, par exemple, les algorithmes génétiques avec des méthodes de recherche locale.

- Hybridation de plusieurs méta-heuristiques : Cette hybridation peut s'effectuer de deux façons différentes. D'une part, des méta-heuristiques complètes peuvent être exécutées séquentiellement. Par exemple, une solution d'un algorithme génétique peut être utilisée comme une solution de départ pour une recherche locale. Pour résoudre le problème d'ordonnancement des horaires d'examens, Azimi [167] propose deux méthodes hybrides. La première méthode utilise la solution générée par une recherche tabou comme une solution initiale pour un algorithme de colonie de fourmis. La deuxième méthode choisit la meilleure solution générée par un algorithme de colonies de fourmis et l'utilise comme une solution de départ pour une recherche tabou. D'autre part, des méta-heuristiques complètes peuvent travailler en parallèle et coopérer pour générer des solutions optimales pour un problème donné. Par exemple, Hammami et Ghédira [168] font coopérer une recherche tabou et un recuit simulé pour résoudre le problème de partition des graphes. Pour cette méthode, les algorithmes utilisés échangent des informations portant, par exemple, sur la meilleure solution trouvée, et ceci à intervalles réguliers.
- Hybridation Méta-heuristique/méthode exacte : Par exemple, Applegate et al. [169] développent une méthode séquentielle pour résoudre le problème du voyageur de commerce. Dans un premier temps, une Recherche Locale est exécutée pour créer un ensemble de solutions et, ensuite, un algorithme exacte est utilisé pour trouver la solution optimale.

Pour notre cas, les résultats des expérimentations du chapitre précédent prouvent l'efficacité de l'algorithme de recherche locale à grand voisinage dans la résolution de notre problème par comparaison aux méthodes de recherche locale classiques. En effet, notre algorithme LNS a amélioré les solutions initiales d'au moins 50% pour la plupart des instances. Ce qui nous amène à l'idée de concevoir des méthodes hybrides avec un algorithme LNS intégré.

Dans ce chapitre, nous proposons d'étudier la performance de plusieurs méta-heuristiques hybrides basées sur LNS. Dans la section 8.2, nous exposerons notre méthode de Recherche Locale Itérative avec LNS intégré. La section 8.3 sera consacrée à l'étude de l'algorithme de Recherche Locale à Voisinage Variable avec LNS intégré. Finalement, la section 8.4 propose une étude comparative des différentes méta-heuristiques hybrides développées.

8.2 ILS avec LNS intégré

8.2.1 Principe de ILS

La méthode de Recherche Locale Itérative est connue d'être simple, facile à implémenter, robuste et très efficace. Elle a été utilisée pour résoudre un grand nombre de problèmes tels que le problème de tournées de véhicules et plusieurs de ses variantes. Ibaraki et al. [170] proposent un algorithme ILS pour le problème de tournées de véhicules avec fenêtres de temps. Pour traduire les contraintes de fenêtres de temps liées aux clients, des fonctions convexes de pénalité sont utilisées. L'algorithme ILS incorpore un algorithme de programmation dynamique qui est utilisé pour déterminer les horaires de passage chez les clients dans le but de minimiser la pénalité sur le temps. Hashimoto et al. [171] étudient une extension du problème de tournées de véhicules avec fenêtres de temps en considérant que les durées et les coûts des trajets sont des fonctions dépendantes du temps. Un algorithme ILS qui utilise des adaptations des méthodes 2-opt*, cross exchange et Or-opt a été proposé. Penna et al. [172] étudient le problème de tournées de véhicules dans le cas d'une flotte hétérogène. Pour résoudre ce

problème, un algorithme ILS utilisant une méthode de descente à voisinage variable, avec un ordre aléatoire des voisinages, dans sa phase de recherche locale, a été développé.

L'idée essentielle de la méthode de ILS consiste à concentrer la recherche non pas sur l'espace complet de solutions, mais sur un sous-espace défini par les solutions qui sont localement optimales pour un critère d'optimisation donné. Cette méthode permet d'améliorer itérativement une solution initiale en alternant une phase de perturbation de la solution courante et une phase de recherche locale. La phase de recherche locale (intensification) permet d'atteindre rapidement un optimum local et la phase de perturbation (diversification) permet d'échapper des optima locaux. Un critère d'acceptation est utilisé pour contrôler la balance entre la diversification et l'intensification. Un point critique dans la mise en place d'un algorithme ILS consiste alors à trouver un compromis entre l'intensification de la recherche aux alentours des meilleures solutions possibles et la diversification de la recherche de façon à explorer de nouvelles zones contenant potentiellement de meilleures solutions.

L'algorithme ILS utilise cinq procédures : (i) génération de la solution initiale ; (ii) une recherche locale qui améliore la solution initialement générée ; (iii) un mécanisme de perturbation qui génère un nouveau point de départ en perturbant la solution générée par la recherche locale ; (iv) un critère d'acceptation qui spécifie si la nouvelle solution doit être acceptée ou pas et (v) un critère d'arrêt qui spécifie quand la procédure de ILS doit s'arrêter.

Après avoir construit une solution initiale, la recherche locale est appliquée à cette solution pour générer un nouvel optimum local S^* . Une perturbation est ensuite appliquée à cet optimum local pour générer une nouvelle solution. Enfin, la recherche locale est de nouveau appliquée sur la solution courante pour générer un nouvel optimum local. Cette nouvelle solution doit passer par un certain critère d'acceptation pour devenir la nouvelle solution courante. Généralement, si la solution obtenue est meilleure que la solution S^* , alors cette nouvelle solution devient le nouveau point de départ de l'algorithme. Ce processus est répété jusqu'à ce que le critère d'arrêt soit satisfait. La Figure 8.1 présente le schéma général de l'algorithme de recherche locale itérative.

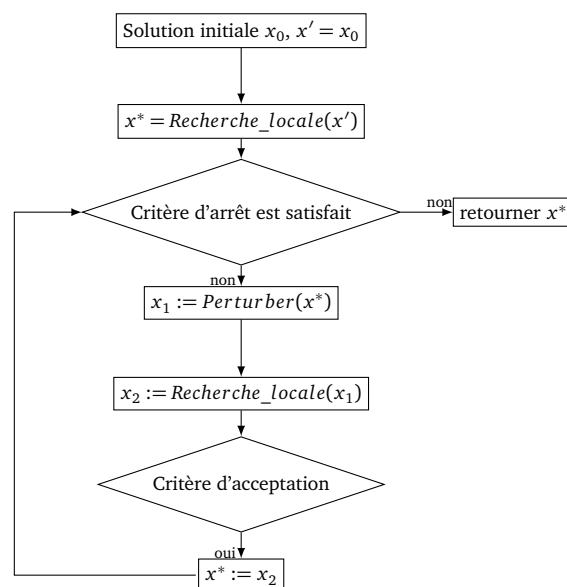


FIG. 8.1: Schéma général de l'algorithme de ILS

8.2.2 ILS avec LNS intégré pour EVRP-MF

Dans cette section, nous proposons un algorithme ILS à départ multiple qui exécute un nombre prédéfini d'itérations et à chaque itération, une nouvelle solution initiale est générée. La solution initiale obtenue est améliorée en utilisant un algorithme de recherche locale itérative. Dans ce qui suit, une description détaillée de toutes les procédures de l'algorithme ILS est présentée.

a. Génération de la solution initiale

Pour générer une nouvelle solution initiale à chaque itération, l'algorithme CRC est utilisé. Afin d'obtenir plusieurs points de départ pour ILS, nous varions notre critère de sélection des véhicules lors de l'exécution de l'algorithme CRC. Les véhicules peuvent alors être triés en fonction du coût ou de la capacité de leurs batteries. Ils peuvent également être sélectionnés aléatoirement.

b. Recherche locale

La procédure de recherche locale de notre algorithme ILS utilise un algorithme LNS. Selon nos recherches bibliographiques, aucune étude antérieure n'a utilisé un algorithme ILS avec une recherche locale basée sur LNS pour résoudre un problème de tournées de véhicules. L'algorithme LNS incorporé dans la phase de recherche locale utilise la méthode d'insertion à moindre coût et la stratégie "First Improvement".

c. Mécanisme de perturbation

La solution générée par la procédure de recherche locale est perturbée pour éviter d'être bloqué dans un minimum local. Le mécanisme de perturbation utilise aussi un algorithme de LNS mais il explore un espace de voisinage plus large que celui exploré dans la procédure de recherche locale. La phase de perturbation est composée des étapes suivantes :

- Éjecter une liste aléatoire de $Num_{perturb}$ sommets tel que $Num_{perturb} > Num$.
- Réinjecter les sommets éjectés d'une façon aléatoire en utilisant la méthode d'insertion à moindre coût. Nous avons choisi d'appliquer la méthode d'insertion à moindre coût étant donné qu'elle permet d'obtenir des solutions de bonne qualité en un temps de calcul raisonnable.

L'intensité de la perturbation qui est ici liée à la taille $Num_{perturb}$ de la liste des noeuds à éjecter est un paramètre important pour notre algorithme ILS. Dans la partie expérimentale de ce chapitre, nous étudierons le lien entre ce paramètre et la qualité des solutions obtenues.

d. Critère d'acceptation

Pour échapper à l'optimum local courant, même les solutions qui n'améliorent pas la meilleure solution obtenue peuvent être acceptées. Notre critère d'acceptation est basé sur le mécanisme d'acceptation des solutions non-améliorantes utilisé par l'algorithme Record-to-Record [173]. Durant l'exécution de la procédure ILS, toute solution est acceptée si son coût est inférieur à $(1 + Deviation) \times Record$, où $Record$ est la valeur de la meilleure solution obtenue jusqu'à présent et $Deviation$ est un paramètre qui spécifie le degrés autorisé de dégradation de la meilleure

solution. Initialement, la valeur de *Record* est égale au coût de la solution initiale. Durant les différentes itérations de l'algorithme, *Record* prend la valeur de la meilleure solution obtenue.

e. Critère d'arrêt

Le critère d'arrêt de notre algorithme est défini par deux conditions :

- Le nombre total d'itérations atteint le nombre maximal d'itérations autorisé max_{Iter}
- Le nombre d'itérations sans amélioration de la meilleure solution obtenue atteint le nombre maximal max_{impr}

Maintenant, nous avons toutes les procédures nécessaires pour décrire notre algorithme ILS. Dans ce qui suit, l'algorithme 19 décrit l'algorithme ILS en détail. Le paramètre *Max_Restart* spécifie le nombre maximum d'itérations à exécuter partant d'une nouvelle solution initiale générée par l'algorithme CRC.

Algorithme 19 Algorithme ILS

```

1: Entrée : Un graphe  $G = (V', A)$  et un ensemble de  $m_1 + m_2$  véhicules
2: Sortie : Un ensemble de tournées affectées à au plus  $m_1 + m_2$  véhicules
3:  $Record \leftarrow +\infty$ ,  $impr \leftarrow 0$ ,  $restart \leftarrow 0$ ,  $n_{Iter} \leftarrow 0$ 
4: tant que  $restart < Max\_Restart$  faire
5:   CRC( $S_0$ )
6:    $S_1 \leftarrow LNS(S_0)$ 
7:    $record \leftarrow cost(S_1)$ 
8:   tant que  $n_{Iter} < max_{Iter}$  et  $impr < max_{impr}$  faire
9:      $S' \leftarrow Perturbation(S_1)$ 
10:     $S'_1 \leftarrow LNS(S_1)$ 
11:    si  $cost(S'_1) < (1 + Deviation) \times record$  alors
12:       $S_1 \leftarrow S'_1$ 
13:    fin si
14:    si  $cost(S'_1) < Record$  alors
15:       $Record \leftarrow cost(S'_1)$ ,  $impr \leftarrow 0$ 
16:    sinon
17:       $impr \leftarrow impr + 1$ 
18:    fin si
19:     $n_{Iter} \leftarrow n_{Iter} + 1$ 
20:  fin tant que
21:   $restart \leftarrow restart + 1$ 
22: fin tant que

```

8.3 Recherche à Voisinage Variable avec LNS intégré

La recherche à voisinage variable a été introduite par Mladenović et Hansen [26]. Plusieurs de ses variantes ont été utilisées pour résoudre différents problèmes d'optimisation combinatoire.

La plus simple de ces variantes est la descente à voisinage variable (Variable Neighborhood Descent) qui représente la version multi-voisinage de la recherche locale classique. Cette variante consiste à appliquer un opérateur de recherche locale sur la solution initiale jusqu'à obtenir un optimum local, l'algorithme continue ensuite en appliquant une autre structure de voisinage sur la solution courante jusqu'à atteindre un optimum local par rapport à cette nouvelle structure. L'algorithme s'arrête quand il atteint une solution qui est l'optimum local par rapport à toutes les structures de voisinage considérées. Dans [174], un algorithme VND est utilisé pour résoudre le problème de collecte de déchets avec des dépôts mobiles. Hetz et Mittaz [175] proposent d'utiliser le VND pour résoudre le CARP (Capacitated Arc Routing Problem). Hu et Raidl [176] proposent une autre version de l'algorithme VND dans laquelle l'ordre des voisinages à explorer est défini d'une façon dynamique lors de la recherche. Chen et al. [177] utilisent la descente à voisinage variable itérative pour résoudre le problème de tournées de véhicules avec contraintes de capacité. Dans [178], un algorithme de VNS est utilisé pour résoudre le problème de tournées de véhicules avec fenêtres de temps.

8.3.1 Principe de VNS

Comme la Recherche Locale Itérative, la méthode de Recherche à Voisinage Variable, ou Variable Neighborhood Search-VNS, repose sur deux méthodes complémentaires : (1) la recherche locale dont l'objectif est d'améliorer la solution courante et (2) les perturbations qui permettent d'élargir l'espace de recherche exploré. Ces deux méthodes sont généralement appelées Intensification et Diversification.

Contrairement aux méthodes présentées jusqu'ici, l'algorithme de VNS utilise méthodiquement plusieurs types de voisinages.

Soit N_k , $k = 1, \dots, k_{max}$ l'ensemble de toutes les structures de voisinage qui seront explorées durant la phase de recherche.

Les différentes étapes de VNS sont présentées dans l'algorithme 20.

Algorithme 20 Algorithme de VNS

1: **Donnée** Soit N_k l'ensemble des structures de voisinage pour $k = 1, \dots, k_{max}$, soit S_0 une solution initiale réalisable
2: **Sortie** Une meilleure solution S_1
3: Initialement, $S_1 \leftarrow S_0$
4: **tant que** La condition d'arrêt n'est pas encore satisfaite **faire**
5: $k \leftarrow 1$
6: **tant que** $k \leq k_{max}$ **faire**
7: *Perturbation* : Générer aléatoirement une solution S' du k^{ime} voisinage de S_1 ($S' \in N_k(S_1)$)
8: *Recherche locale* : Appliquer une recherche locale à S' , soit S'' le nouvel optimum local obtenu
9: *Move Or Not* :
10: **si** $Cost(S'') < Cost(S_1)$ **alors**
11: $S_1 \leftarrow S''$, $k \leftarrow 1$
12: **sinon**
13: *Changement de voisinage* : $k \leftarrow k + 1$
14: **fin si**
15: **fin tant que**
16: **fin tant que**

8.3.2 VNS avec LNS intégré pour EVRP-MF

Dans cette section, nous détaillons les différentes composantes de notre algorithme VNS. Concernant la génération des solutions initiales, nous utilisons toujours notre algorithme CRC. Dans ce qui suit, nous présentons les procédures de recherche locale et de perturbation utilisées.

a. Recherche Locale

La recherche locale pour l'algorithme de VNS consiste à envisager plusieurs structures de voisinage pour améliorer la qualité de la solution obtenue. Pour chaque structure de voisinage, une succession de transformations de la solution courante est effectuée. La solution courante est remplacée par une meilleure solution qui appartient à son voisinage. Le passage d'une structure de voisinage à une autre s'effectue lorsqu'il n'est plus possible de trouver une solution améliorante dans le voisinage considéré de la solution courante.

Si le voisinage de la solution courante est complètement exploré, la recherche locale assure l'obtention d'un optimum local pour la transformation utilisée. Cependant, cette notion d'optimum local reste relative à une transformation spécifique. Il est alors possible qu'un optimum local pour une transformation donnée ne soit pas un optimum local pour une autre transformation. Le choix des structures de voisinage représente alors une partie importante de l'algorithme de recherche à voisinages variables.

L'ordre des structures de voisinage à explorer par l'algorithme de VNS peut être soit aléatoire soit basé sur une règle donnée, par exemple, les voisinages peuvent être ordonnés du voisinage le plus petit au voisinage le plus large.

Dans notre cas, nous introduisons un algorithme VNS qui n'utilise que des structures de voisinage basées sur l'algorithme LNS. Chaque structure de voisinage représente ainsi un algorithme LNS ayant un degré de destruction prédéfini. Par exemple, nous pouvons définir trois structures de voisinage N_1 , N_2 et N_3 tel que N_1 est un algorithme LNS avec $Num = 2$, N_2 est un algorithme LNS avec $Num = 3$ et N_3 est un algorithme LNS avec $Num = 4$.

b. Perturbation

La perturbation consiste à modifier modérément la meilleure solution connue au lieu de partir d'une solution entièrement aléatoire.

Avant d'appliquer une recherche locale sur la solution courante, l'algorithme de VNS applique une perturbation sur la meilleure solution connue et la recherche locale sera par la suite appliquée sur cette nouvelle solution obtenue.

La perturbation a comme objectif de générer, aléatoirement, une solution S' à partir du k^{ime} voisinage de S . Le mouvement de perturbation doit correspondre à un mouvement aléatoire dans le voisinage avec un degré plus grand que celui utilisé par la méthode de recherche locale. Le choix de l'intensité de la perturbation à appliquer est alors important. Si l'intensité est trop faible, uniquement un petit voisinage de la solution sera exploré. Si, par contre, l'intensité est importante, les caractéristiques de la meilleure solution courante seront perdues et la solution obtenue ne sera pas nécessairement meilleure qu'une solution aléatoire.

Pour fixer l'intensité de la perturbation à appliquer, un paramètre K'_k peut être utilisé. L'algorithme 21 représente un algorithme général de perturbation.

Algorithme 21 Algorithme de Perturbation

- 1: **Input** Une solution S , une structure de voisinage N_k et le paramètre k
 - 2: **Output** Une solution perturbée S'
 - 3: **tant que** $i \leq K'_k$ **faire**
 - 4: Choisir aléatoirement $x' \in N_k(S)$
 - 5: $S \leftarrow S'$
 - 6: **fin tant que**
-

Dans notre cas, nous appliquons, pour chaque structure de voisinage N_k , un algorithme LNS dont le degré de destruction est supérieur à celui utilisé dans la phase de recherche locale.

Notre algorithme VNS s'arrête si le nombre maximal d'itération max_{Iter} est atteint ou si le temps de calcul maximum est dépassé.

8.4 Étude expérimentale

Dans cette section, nous proposons d'étudier les performances de nos méta-heuristiques hybrides. Nous testons alors les méthodes développées sur les instances généralisées de la littérature qui ont été exposées dans le chapitre précédent.

8.4.1 Performance de l'algorithme ILS

Afin de mieux évaluer la performance de notre algorithme ILS avec LNS intégré et de mieux comprendre l'apport de l'hybridation ILS/LNS, nous avons choisi de développer une autre

variante de l'algorithme ILS (ILS/2-opt) qui utilise un algorithme 2-opt dans la phase de recherche locale à la place d'un algorithme LNS. L'algorithme 2-opt est un algorithme de recherche locale puissant qui permet de trouver des solutions de haute qualité dans les zones de recherche pouvant potentiellement contenir des solutions optimales.

Le mécanisme de perturbation utilisé pour la variante ILS/2-opt est le même que celui décrit dans le paragraphe 8.2.2 avec $Num_{perturb} = 6$.

Nous exposons alors, dans ce qui suit, les différents résultats obtenus.

a. Performance de ILS avec LNS intégré

Avant d'exposer les résultats obtenus en utilisant notre algorithme ILS avec LNS intégré, nous exposons, dans le tableau 8.1 les différents paramètres utilisés.

TAB. 8.1: Paramètres utilisés pour ILS

Paramètre	Rôle	Valeur
max_{Iter}	Nombre maximal d'itérations	50
max_{impr}	Nombre maximal d'itérations sans amélioration	10
$TimeOut$	Temps de calcul maximal	60 (mn)

Concernant l'algorithme LNS utilisé au sein de ILS, nous avons choisi la méthode d'insertion à moindre coût et la stratégie "First Improvement". Nous avons fixé la valeur de $Iter1$ à 50 et la valeur de $Iter2$ à 30.

Pour évaluer la performance de notre algorithme ILS, nous avons varié les paramètres utilisés et comparé les résultats en fonction de la qualité des solutions obtenues et du temps de calcul. Les paramètres variés sont les suivants :

- Le nombre des sommets éjectés lors de la phase de recherche locale
- Le nombre des sommets éjectés lors de la phase de perturbation
- La valeur du paramètre *Deviation*

i- Impact de l'intensité de perturbation

L'intensité de la perturbation a un impact important sur la qualité des solutions obtenues. Pour comprendre cet impact, nous fixons au début le nombre de noeuds éjectés lors de la phase de recherche locale et nous varions ensuite le degré de perturbation.

Les tableaux 8.2 et 8.3 récapitulent les résultats obtenus en variant l'intensité de perturbation de l'algorithme ILS.

D'une part, nous pouvons remarquer qu'en fixant le degré de perturbation et en variant le nombre de noeuds éjectés Num lors de la recherche locale, la qualité des solutions obtenues s'améliore en augmentant la valeur de Num , ce qui confirme les résultats obtenus dans le chapitre précédent. En effet, pour $Num_{perturb} = 8$, nous passons d'une amélioration moyenne de 70.77% dans le cas où $Num = 2$ à une amélioration de 71.33% si $Num = 4$.

D'autre part, en fixant la valeur de Num et en augmentant le degré de perturbation, nous remarquons également une amélioration dans la qualité des solutions obtenues. Par exemple, pour $Num = 2$, nous passons d'une amélioration d'environ 58% dans le cas où $Num_{perturb} = 6$ à une amélioration de plus de 70% si $Num_{perturb} = 10$. Toutefois, il est très compliqué de déterminer les meilleures valeurs pour les deux paramètres Num et $Num_{perturb}$ vu l'aspect

TAB. 8.2: Résultats de performance de ILS en fonction de l'intensité de perturbation (Partie 1)

Ins.	(2,6)			(2,7)			(2,8)			(2,9)			(2,10)			(3,3)								
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	60.94	39.63	7	6	65.54	42.11	7	6	66.22	42.22	7	6	66.49	44.44	7	6	68.09	55.89	7	6	56.95	24.22	8	6
C2	83.30	40.25	8	2	88.17	41.00	7	2	109.55	45.71	6	2	107.85	45.50	6	2	106.53	52.13	6	2	79.25	26.75	9	2
R1	21.15	29.89	6	7	21.74	28.33	6	7	23.06	34.25	6	7	22.60	45.44	6	7	22.85	37.00	6	7	18.61	19.89	7	7
R2	20.50	48.75	5	2	25.97	51.88	5	2	30.49	43.57	4	2	29.50	44.50	4	2	28.80	54.50	4	2	17.63	39.38	6	2
RC1	70.70	34.22	7	7	75.32	35.00	7	7	80.58	40.22	6	7	81.93	43.11	6	7	82.74	45.11	6	7	71.60	22.33	7	7
RC2	96.12	45.25	7	2	104.46	46.00	6	2	114.73	45.88	6	2	117.51	48.13	6	2	119.94	51.25	6	2	89.28	30.63	7	2
Moy:	58.78	39.66	7	4	63.51	40.72	6	4	70.77	41.97	6	4	70.98	45.18	6	4	71.41	49.31	6	4	55.55	27.20	7	4

TAB. 8.3: Résultats de performance de ILS en fonction de l'intensité de perturbation (Partie 2)

Ins.	(3,4)			(3,5)			(3,6)			(3,7)			(3,8)			(4,8)								
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT				
C1	58.04	32.89	8	6	58.64	37.22	7	6	58.09	35.78	8	6	62.04	42.56	7	6	67.06	48.44	7	6	68.52	56.00	7	6
C2	83.99	43.13	8	2	89.42	36.5	7	2	88.57	39.88	8	2	93.51	47.63	7	2	110.48	54.25	6	2	111.51	59.38	6	2
R1	23.16	24.67	7	7	23.61	26.56	6	7	19.19	25.78	7	7	22.86	30.78	6	7	22.59	31.67	6	7	21.98	43.33	6	7
R2	28.48	50.13	5	2	29.36	50.88	5	2	31.81	49.25	5	2	23.85	51.00	5	2	29.06	56.75	5	2	29.32	58.75	5	2
RC1	74.21	36.11	7	7	74.94	34.67	7	7	73.52	34.44	7	7	77.97	40.56	7	7	80.75	39.78	6	7	80.62	52.33	6	7
RC2	86.56	37.38	8	2	85.36	42.63	7	2	99.32	42.88	7	2	101.97	49.50	6	2	115.23	50.13	7	2	116.06	60.00	6	2
Moy:	56.74	37.39	7.2	4.3	58.72	38.08	7	4	60.08	38.00	7	4	63.70	43.67	6	4	70.86	46.84	6	4	71.33	54.97	6	4

aléatoire des méthodes de recherche locale et de perturbation utilisées.

Concernant le temps de calcul, nous remarquons qu'il dépend de la somme $Num + Num_{perturb}$. En effet, il passe de 27 minutes dans le cas où $Num = 3$ et $Num_{perturb} = 3$ à environ 55 minutes dans le cas où $Num = 2$ et $Num_{perturb} = 10$.

Pour l'étude de l'impact de la valeur de la déviation sur la qualité des solutions obtenues, nous avons choisi de fixer la valeur de Num à 2 et la valeur de $Num_{perturb}$ à 8.

ii- Impact du paramètre Deviation

Le tableau 8.4 présente les résultats obtenus en variant la valeur de la déviation pour le cas où $Num = 2$ et $Num_{perturb} = 8$.

TAB. 8.4: Impact de la variation de la valeur de la déviation

Ins.	Deviation = 0.05				Deviation = 0.1				Deviation = 0.2				Deviation = 0.3			
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	66.06	41.67	7	6	66.48	45.44	7	6	66.72	45.89	8	6	63.67	43.56	7	6
C2	108.34	45.88	6	2	110.13	50.88	7	2	109.49	56.13	7	2	99.69	42.13	7	2
R1	20.42	29.11	6	7	21.39	31.67	6	7	21.97	33.33	6	7	20.10	29.22	7	7
R2	30.63	52.63	5	2	30.29	52.13	5	2	31.73	54.88	4	2	22.80	50.25	5	2
RC1	78.38	38.78	7	7	76.77	40.78	7	7	78.89	44.89	6	7	77.49	37.44	6	7
RC2	118.10	50.63	6	2	181.26	42.25	6	2	119.18	49.50	6	2	105.89	50.00	6	2
Moy.	70.32	43.11	6	4	70.53	43.86	6	4	71.33	47.44	6	4	64.94	42.10	6	4

En acceptant des solutions non améliorantes, nous remarquons une amélioration au niveau de la qualité des solutions générées. En effet, si on autorise une déviation de 20%, les solutions initiales sont améliorées d'en moyenne 71.33% contre seulement 70.77% dans le cas où la déviation est nulle. En revanche, si la valeur de la déviation dépasse 20%, une dégradation de la qualité des solutions obtenues est remarquée. En effet, pour une déviation de 30%, l'amélioration par rapport aux solutions initiales n'est plus que de 64.94%.

b. Performance de ILS/2-opt

Le tableau 8.5 résume les résultats obtenus en utilisant la variante ILS/2-opt.

Dans un premier temps, nous comparons les résultats générés par ILS/2-opt avec ceux obtenus par l'algorithme LNS dans le cas où $Num = 6$. Nous remarquons alors que la variante ILS/2-opt a permis d'améliorer les solutions initiales d'environ 67% en moyenne contre une amélioration d'environ 63% pour LNS. De plus, ILS/2-opt a permis de privilégier l'usage des véhicules électriques. Pour un total de 10 véhicules utilisés, ILS/2-opt a généré une solution utilisant 6 VEs et 4 VTs alors que la solution générée par LNS utilise 5 VEs et 5 VTs.

Dans un second temps, nous remarquons que notre variante ILS/2-opt génère des résultats compétitifs avec ceux générés par LNS pour les cas où $Num \leq 8$.

En comparant les résultats obtenus en utilisant notre algorithme ILS avec LNS intégré avec ceux de l'algorithme ILS/2-opt, nous remarquons que l'hybridation ILS/LNS a permis d'obtenir des meilleurs résultats dans la plupart des cas. En effet, l'algorithme ILS/2-opt a, en moyenne, amélioré les solutions initiales de 67.59% contre une amélioration qui atteint 71.41% pour ILS avec LNS intégré.

TAB. 8.5: Résultats de ILS/2-opt

Ins.	ILS/2-opt			
	Gap	CPU	VE	VT
C1	63.60	66.13	7	6
C2	107.35	60.58	6	2
R1	24.04	62.15	6	7
R2	25.51	66.98	5	2
RC1	78.96	65.23	6	7
RC2	106.05	62.17	6	2
Moy.	67.59	63.87	6	4

8.4.2 Performance de l'algorithme VNS

Dans cette section, nous proposons d'étudier la performance de notre algorithme VNS. Pour ce faire, nous développons plusieurs variantes de VNS et nous comparons les différents résultats obtenus.

Avant d'exposer les résultats obtenus, nous exposons, dans le tableau 8.6 les différents paramètres utilisés.

TAB. 8.6: Paramètres utilisés pour ILS

Paramètre	Rôle	Valeur
max_{Iter}	Nombre maximal d'itérations	50
$TimeOut$	Temps de calcul maximal	60 (mn)

a. Variantes de VNS

Dans cette section, nous présentons les différentes variantes de VNS développées. Nous définissons principalement 3 variantes : VNS1, VNS2 et VNS3. Pour VNS1, seulement des structures de voisinage basées sur LNS sont utilisées. Pour VNS2, seulement les 6 structures de voisinage définies dans le chapitre précédent sont utilisées. Pour VNS3, un mix de toutes les structures de voisinage peut être utilisé.

Pour chaque variante de VNS, nous testons trois types de perturbation : Pr^{Rd} , Pr^{SRd} et Pr^{MRd} . La perturbation Pr^{Rd} consiste à appliquer un algorithme LNS d'une façon complètement aléatoire. Autrement dit, lors de cette perturbation, les noeuds à éjecter sont choisis aléatoirement et sont par la suite insérés aléatoirement dans la solution partielle. La perturbation Pr^{SRd} consiste à choisir les noeuds à éjecter aléatoirement et les injecter ensuite dans les meilleures positions. Cette perturbation correspond alors à notre algorithme LNS avec la méthode d'insertion à moindre coût. La perturbation Pr^{MRd} consiste à alterner les deux autres types de perturbation Pr^{Rd} et Pr^{SRd} .

Nous présentons, dans ce qui suit, les résultats obtenus en utilisant les différentes variantes de VNS.

b. Performance de VNS1

Dans ce paragraphe, nous présentons les résultats obtenus en utilisant la variante VNS1. Nous testons deux cas qui diffèrent en fonction des structures de voisinage choisies. Pour les deux cas, nous fixons $k_{max} = 3$ et nous utilisons la perturbation Pr^{SRd} avec $Num = 8$.

Pour le premier cas, noté $VNS1_LNS234$, nous utilisons trois structures de voisinage qui utilisent un algorithme LNS avec $Num = 2$ pour N_1 , $Num = 3$ pour N_2 et $Num = 4$ pour N_3 .

Pour le deuxième cas, noté $VNS1_LNS567$, nous utilisons trois structures de voisinage qui utilisent un algorithme LNS avec $Num = 5$ pour N_1 , $Num = 6$ pour N_2 et $Num = 7$ pour N_3 . Les tests sont effectués avec $K'_k \in \{1, 2, 3\}$ et les meilleurs résultats obtenus sont sauvegardés à chaque fois.

Le tableau 8.7 présente les résultats obtenus pour la variante VNS1.

TAB. 8.7: Résultats de la variante VNS1

	VNS1_LNS234				VNS1_LNS567			
	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	67.05	55.23	7	6	64.23	49.65	7	6
C2	101.77	41.18	7	2	97.08	38.12	7	2
R1	26.43	49.78	5	7	25.74	54.69	6	7
R2	32.82	43.12	4	2	27.62	56.78	4	2
RC1	82.85	46.82	6	7	77.59	58.46	7	7
RC2	110.15	46.12	6	4	96.97	57.78	7	2
Moy.	70.17	47.04	6	4	64.87	51.54	6	4

D'après les résultats du tableau 8.7, la variante $VNS1_LNS234$ semble être plus performante que $VNS1_LNS567$ avec une amélioration d'environ 70% par rapport aux solutions initiales contre seulement 64% pour la variante $VNS1_LNS567$. La variante $VNS1_LNS234$ permet également de réduire le nombre de véhicules utilisés avec une moyenne de 6 véhicules électriques et 4 véhicules thermiques. Le temps de calcul de $VNS1_LNS234$ est également légèrement inférieur à celui de $VNS1_LNS567$. Pour VNS3, nous utiliserons alors les mêmes structures de voisinage que celles utilisées dans $VNS1_LNS234$.

c. Performance de VNS2

Dans ce paragraphe, nous présentons les résultats obtenus en utilisant la variante VNS2. Notre algorithme VNS utilise alors 6 structures de voisinage ordonnées comme suit : OPM, TPM, 2-Opt, Or-Opt, Réaffectation des tournées et Concaténation des tournées. Concernant la phase de perturbation, pour chaque structure de voisinage, un nombre prédéfini de sommets est éjecté puis réinséré dans la solution. Ce nombre doit être supérieur au nombre de sommets impactés par chaque type de transformation. Par exemple, pour l'algorithme OPM, un seul sommet est considéré à chaque itération. La perturbation correspondante à cette structure de voisinage consiste alors à éjecter 2 sommets et les réinsérer par la suite dans la solution courante. Concernant l'intensité de la perturbation, nous varions la valeur de K'_k de 1 à 3 pour chaque k et nous répertorions à chaque fois la meilleure solution trouvée.

Le tableau 8.8 présente les résultats obtenus en utilisant la variante VNS2.

TAB. 8.8: Résultats de la variante VNS2

	P_r^{SRD}				P_r^{MRD}				P_r^{RD}			
	Gap	CPU	VE	VT	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	67.35	52.13	7	6	11.09	56.14	18	5	10.47	53.28	18	5
C2	109.79	56.23	6	2	21.48	55.09	19	2	19.55	55.16	19	2
R1	20.86	35.48	6	7	3.18	59.48	9	5	2.22	51.97	10	5
R2	28.84	55.38	5	2	5.49	58.12	6	2	4.69	54.23	7	2
RC1	76.64	54.67	7	7	14.82	58.74	20	5	13.47	58.12	20	5
RC2	89.23	48.88	8	2	20.48	59.87	21	2	18.11	52.79	21	2
Moy.	65.45	50.46	6	4	12.76	57.91	16	4	11.42	54.25	16	4

Les résultats du tableau 8.8 prouvent que la méthode de perturbation P_r^{SRD} est la plus adaptée à notre problème. En effet, en utilisant la perturbation P_r^{SRD} , la variante VNS2 a permis d'améliorer les solutions initiales de plus de 65% contre une amélioration de seulement 12.76% en cas d'utilisation de la perturbation P_r^{MRD} et de 11.42% si la perturbation P_r^{RD} est utilisée. De plus, nous avons constaté que les perturbations P_r^{MRD} et P_r^{RD} consomment beaucoup de temps en essayant de réinsérer le sommet éjecté dans une position aléatoire réalisable de la solution partielle. Cette difficulté à trouver une position aléatoire est liée au fait que notre algorithme n'accepte que les solutions réalisables. En plus, notre problème prend en compte un grand nombre de contraintes qui réduisent le nombre de solutions réalisables dans l'espace de recherche. Pour le reste de notre étude expérimentale, nous n'exposerons alors que les résultats obtenus en appliquant la perturbation P_r^{SRD} .

Même si la variante VNS2 permet d'améliorer les solutions initiales de plus de 65%, elle ne dépasse pas la performance de notre algorithme ILS.

Si nous comparons la variante VNS1 et la variante VNS2, nous remarquons que le fait d'utiliser plusieurs structures de voisinage basées sur un algorithme LNS a permis d'améliorer la qualité des solutions obtenues.

c. Performance de VNS3

Dans ce paragraphe, nous présentons les résultats obtenus en utilisant la variante VNS3. Pour VNS3, nous testons deux cas : (i) le cas où toutes les structures de voisinage sont triées en fonction de la taille du voisinage exploré comme il est le cas pour les variantes VNS1 et VNS2 ($VNS3^{Sq}$) et (ii) le cas où à chaque itération une nouvelle structure de voisinage est choisie aléatoirement ($VNS3^{Rd}$). Le tableau 8.9 résume les résultats obtenus pour $VNS3^{Rd}$ et $VNS3^{Sq}$. Les résultats du tableau 8.9 prouvent que la variante $VNS3^{Sq}$ est plus performante que la variante $VNS3^{Rd}$. En effet, l'algorithme $VNS3^{Sq}$ a permis d'améliorer les solutions initiales de plus de 70% alors que la variante $VNS3^{Rd}$ génère un pourcentage d'amélioration moyenne d'environ 60%.

Parmi tous les algorithmes VNS testés, c'est l'algorithme utilisant toutes les structures de voisinage proposées ($VNS3^{Sq}$) qui a généré les meilleurs résultats.

8.4.3 Étude comparative

Dans cette section, nous proposons une étude comparative des différentes méthodes développées pour résoudre notre problème. Nous exposons alors, pour chaque méthode, les

TAB. 8.9: Résultats de la variante VNS3

	VNS3 Rd				VNS3 ^{Sq}			
	Gap	CPU	VE	VT	Gap	CPU	VE	VT
C1	55.23	46.00	9	6	69.92	54.56	8	6
C2	102.07	35.88	6	2	110.59	60.28	5	2
R1	18.57	38.00	7	7	22.93	37.50	8	7
R2	22.09	42.38	5	2	34.24	60.83	4	2
RC1	74.16	41.22	7	7	81.43	54.98	8	7
RC2	85.21	37.50	8	2	111.46	61.27	8	2
Moy.	59.56	40.16	7	4	71.76	54.90	7	4

meilleurs résultats obtenus en variant ses différents paramètres.

Le tableau 8.10 présente, pour chaque méthode, les meilleurs résultats obtenus. Les colonnes (LNS), (ILS) et (VNS) présentent les meilleures valeurs trouvées par les méthodes LNS, ILS et VNS, respectivement. La colonne (Best) donne la meilleure valeur de la fonction objectif trouvée parmi toutes les solutions générées par les différentes méthodes. La ligne Gap présente le gap moyen par rapport aux solutions initiales.

TAB. 8.10: Meilleurs résultats obtenus

	LNS	ILS	VNS	Best
c101_21	473.39	436.56	450.41	436.56
c102_21	439.18	457.18	429.93	429.93
c103_21	476.88	462.52	470.50	462.52
c104_21	462.99	458.33	443.91	443.91
c105_21	426.75	457.33	429.89	426.75
c106_21	466.20	463.92	460.78	460.78
c107_21	452.52	463.16	452.81	452.52
c108_21	450.68	454.32	462.21	450.68
c109_21	447.06	454.33	443.68	443.68
c201_21	302.94	291.81	305.00	291.81
c202_21	300.15	296.06	300.15	296.06
c203_21	304.74	277.06	287.23	277.06
c204_21	274.45	292.39	264.24	264.24
c205_21	292.68	292.72	288.53	288.53
c206_21	317.07	296.08	300.95	296.08
c207_21	277.42	285.72	301.90	277.42
c208_21	292.12	298.56	306.60	292.12
r101_21	464.07	451.62	431.54	431.54
r102_21	460.80	418.81	430.51	418.81
r103_21	462.43	454.92	444.49	444.49
r104_21	467.64	443.70	439.30	439.30
r105_21	466.25	438.84	426.60	426.60
r106_21	475.87	445.44	454.51	445.44
r107_21	458.31	437.97	447.82	437.97
r108_21	465.10	439.74	424.67	424.67
r109_21	468.05	439.24	357.35	357.35

	LNS	ILS	VNS	Best
r201_21	255.35	274.42	255.32	255.32
r202_21	254.11	264.44	243.36	243.36
r203_21	247.74	274.29	247.74	247.74
r204_21	245.51	272.98	261.05	245.51
r205_21	251.30	255.30	261.28	251.30
r206_21	255.74	276.48	274.32	255.74
r207_21	250.44	275.76	263.45	250.44
r208_21	245.47	260.67	271.04	245.47
rc101_21	464.08	452.97	430.72	430.72
rc102_21	472.50	438.60	448.24	438.60
rc103_21	440.23	433.13	440.23	433.13
rc104_21	457.67	432.57	435.41	432.57
rc105_21	456.54	433.14	453.47	433.14
rc106_21	463.45	433.08	444.77	433.08
rc107_21	453.79	446.63	451.06	446.63
rc108_21	472.93	438.64	434.90	434.90
rc109_21	469.40	449.36	433.31	433.31
rc201_21	253.38	261.24	306.59	253.38
rc202_21	272.00	293.01	302.50	272.00
rc203_21	274.81	302.01	267.01	267.01
rc204_21	263.02	292.06	318.98	263.02
rc205_21	261.07	279.61	312.26	261.07
rc206_21	272.98	293.80	313.29	272.98
rc207_21	254.46	268.24	254.46	254.46
rc208_21	275.87	272.91	275.87	272.91
Moy.	371.17	368.31	365.81	355.78
Gap	76.67	78.04	79.26	84.31

En se basant sur les résultats du tableau 8.10, nous pouvons tirer les conclusions suivantes :

- L’algorithme LNS améliore les solutions initiales de plus de 76% et il reste l’algorithme le plus rapide parmi les différentes méthodes développées.
- L’algorithme ILS améliore les solutions initiales d’environ 78%. L’algorithme VNS a le meilleur pourcentage d’amélioration avec une moyenne d’environ 79%.
- L’algorithme LNS réussit à trouver la meilleure solution pour 15 instances parmi les 51 instances testées. LNS semble être particulièrement performant pour la classe R2. Il génère, en effet, les meilleurs résultats pour 6 instances parmi les 8 instances de cette classe. L’algorithme LNS permet ainsi de générer de bons résultats pour des instances de données caractérisées par une distribution géographique aléatoire des clients. Il a néanmoins plus de difficulté à résoudre des instances avec une distribution en clusters des clients.
- L’algorithme ILS génère les meilleurs résultats pour 16 instances et l’algorithme VNS réussit à trouver les meilleures solutions pour 20 instances.
- L’algorithme ILS semble être particulièrement performant pour la classe RC1 où il trouve les meilleures solutions pour 6 instances parmi les 9 instances de cette classe. Les phases

de destruction et de réparation de l'algorithme ILS permettent ainsi une résolution plus efficace des instances qui ne sont pas complètement aléatoires.

- L'algorithme VNS semble pouvoir mieux s'adapter aux différentes classes de données et fournir des résultats satisfaisants.

8.5 Conclusion

Dans ce chapitre, nous avons développé un algorithme ILS avec LNS intégré et un algorithme VNS avec LNS intégré. Nous avons ensuite évalué la performance des différents algorithmes hybrides développés sur des instances généralisées de la littérature. Même si ces algorithmes ont permis d'améliorer les résultats générés par notre algorithme LNS dans la majorité des cas, ce dernier reste particulièrement performant sur les instances de données caractérisées par une distribution aléatoire des clients.

Chapitre 9

Conclusion générale et perspectives

Le véhicule électrique semble être, de nos jours, une solution adéquate pour réduire les émissions des gaz à effet de serre et limiter la dépendance énergétique de notre société. En revanche, le déploiement des véhicules électriques, surtout dans le cadre industriel, nécessite la mise en place des infrastructures de recharge et l'adaptation de l'activité industrielle en fonction de l'autonomie réelle de ce véhicule.

Cette thèse s'inscrit dans le cadre du projet national de déploiement des véhicules électriques "Infinidrive". L'objectif de ce projet est de développer des outils d'aide à la décision pour l'optimisation de l'usage et de la recharge des flottes captives tout en assurant un bilan économique et écologique positif. Notre application de référence est alors l'optimisation conjointe de l'affectation et de la recharge des véhicules électriques dans un contexte industriel. Dans le cadre de ce mémoire, nous nous sommes intéressés à l'étude des problèmes de planification de la recharge et d'optimisation des tournées dans le cas d'une flotte captive selon différents points de vue. Ces problèmes ont commencé à susciter l'intérêt des chercheurs depuis quelques années vu la grande importance accordée au déploiement des véhicules électriques et des infrastructures de recharge surtout dans le cadre industriel.

La revue de littérature que nous avons effectuée dans ce domaine nous a confirmé que les problèmes étudiés dans le cadre de ce mémoire n'ont jamais été étudiés auparavant.

Nous avons commencé nos travaux de recherche par la proposition d'un modèle linéaire en nombres mixtes pour le problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques. Nous avons par la suite proposé une méthode exacte et deux heuristiques pour résoudre ce problème. Toutes les méthodes développées ont été testées sur des instances réelles fournies par le Groupe La Poste et d'autres aléatoires.

En plus de l'étude de notre application de référence, nous avons proposé plusieurs extensions à notre problème de base et nous avons varié les scénarios d'usage.

Pour compléter cette partie de recherche, nous avons effectué une étude théorique de notre problème de base. Nous avons alors considéré un nouveau problème d'ordonnement d'intervalles sous contraintes d'énergie et nous nous sommes intéressés à la complexité et l'approximabilité de plusieurs variantes de ce problème.

Finalement, nous avons étendu le problème de tournées de véhicules électriques considéré dans la littérature en ajoutant plusieurs contraintes réelles afin de proposer des outils d'optimisation industriellement exploitables. Pour résoudre ce problème, nous avons commencé par proposer des algorithmes de recherche locale et un algorithme de recherche locale à grand voisinage. Même si les algorithmes classiques de recherche locale n'ont pas permis d'obtenir de bons résultats, l'algorithme de recherche locale à grand voisinage (LNS) a

généralisé des solutions très satisfaisantes, ce qui nous a poussés à proposer des méta-heuristiques hybrides avec LNS intégré. Toutes les méthodes développées ont été testées sur des instances généralisées de la littérature.

Tout au long de ces trois années de recherche, nous avons essayé d'aborder différents problèmes liés à l'usage du véhicule électrique et de varier les méthodes utilisées pour les résoudre. En revanche, vu que ce domaine de recherche est relativement récent, il reste encore plusieurs pistes à explorer. Comme perspectives, nous proposons d'étudier les extensions et les pistes suivantes :

- Par rapport à notre problématique de référence, il est possible de proposer un algorithme hybride qui combine notre méthode exacte et notre heuristique globale. Nous pouvons également faire coopérer, d'une façon séquentielle, notre méthode exacte (ou nos heuristiques) avec des heuristiques d'amélioration ou des méta-heuristiques.
- Concernant le problème d'ordonnement d'intervalles sous contraintes d'énergie, beaucoup de travail peut encore être fait. En effet, l'introduction des contraintes énergétiques dans les problèmes d'ordonnement ouvre des pistes de recherche académiques multiples et met l'accent sur diverses applications réelles de ces problèmes. Il reste encore à approfondir ce thème de recherche qui n'est qu'à ses débuts en proposant de nouvelles approches heuristiques et méta-heuristiques et des études analytiques comme d'autres approches à garantie de performance.
- Finalement, concernant notre problème de tournées de véhicules électriques, il est possible de simplifier le problème étudié afin de pouvoir comparer les résultats obtenus avec les résultats de la littérature. De plus, d'autres méthodes de résolution peuvent encore être testées. Il est également possible d'étendre notre problème en ajoutant les contraintes de fenêtres de temps pour les clients.

Bibliographie

- [1] Pour un futur [simple] du véhicule électrique. 2015.
- [2] Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3) :563–581, 1977.
- [3] Scott Kirkpatrick. Dg jr., and mp vecchi. *Optimization by simulated annealing*. *Science*, 220(4598) :671–680, 1983.
- [4] Scott Kirkpatrick. Optimization by simulated annealing : Quantitative studies. *Journal of statistical physics*, 34(5-6) :975–986, 1984.
- [5] Vladimír Černý. Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1) : 41–51, 1985.
- [6] RW Eglese. Simulated annealing : a tool for operational research. *European journal of operational research*, 46(3) :271–281, 1990.
- [7] IH Osman and CN Potts. Simulated annealing for permutation flow-shop scheduling. *Omega*, 17(6) :551–557, 1989.
- [8] Peter JM Van Laarhoven, Emile HL Aarts, and Jan Karel Lenstra. Job shop scheduling by simulated annealing. *Operations research*, 40(1) :113–125, 1992.
- [9] Wen-Chyuan Chiang and Robert A Russell. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63(1) : 3–27, 1996.
- [10] Fred Glover. Tabu search-part i. *ORSA Journal on computing*, 1(3) :190–206, 1989.
- [11] Fred Glover, James P Kelly, and Manuel Laguna. Genetic algorithms and tabu search : hybrids for optimization. *Computers & Operations Research*, 22(1) :111–134, 1995.
- [12] Michel Gendreau and Jean-Yves Potvin. Tabu search. In *Search methodologies*, pages 165–186. Springer, 2005.
- [13] Alain Hertz and Dominique de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4) :345–351, 1987.
- [14] Holland John. *Holland, adaptation in natural and artificial systems*. 1992.
- [15] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.
- [16] Nidamarthi Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3) :221–248, 1994.

- [17] Alberto Colomi, Marco Dorigo, Vittorio Maniezzo, et al. Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142. Paris, France, 1991.
- [18] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *BioSystems*, 43(2) :73–81, 1997.
- [19] Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. Macs-vrptw : A multiple colony system for vehicle routing problems with time windows. In *New ideas in optimization*. Citeseer, 1999.
- [20] E-G Talbi, Olivier Roux, Cyril Fonlupt, and Denis Robillard. Parallel ant colonies for the quadratic assignment problem. *Future Generation Computer Systems*, 17(4) :441–449, 2001.
- [21] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. *Iterated local search*. Springer, 2003.
- [22] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search : Framework and applications. pages 363–397, 2010.
- [23] Matthijs Den Besten, Thomas Stützle, and Marco Dorigo. Design of iterated local search algorithms. In *Applications of Evolutionary Computing*, pages 441–451. Springer, 2001.
- [24] Thomas Stützle. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, 174(3) :1519–1539, 2006.
- [25] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12) :3281–3290, 2009.
- [26] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11) :1097–1100, 1997.
- [27] Pierre Hansen and Nenad Mladenović. *Variable neighborhood search*. Springer, 2005.
- [28] Pierre Hansen and Nenad Mladenovic. *A tutorial on variable neighborhood search*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal, 2003.
- [29] Teodor Gabriel Crainic, Michel Gendreau, Pierre Hansen, and Nenad Mladenović. Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, 10(3) :293–314, 2004.
- [30] Cedric Avanthay, Alain Hertz, and Nicolas Zufferey. A variable neighborhood search for graph coloring. *European Journal of Operational Research*, 151(2) :379–388, 2003.
- [31] Michael Polacek, Richard F Hartl, Karl Doerner, and Marc Reimann. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of heuristics*, 10(6) :613–627, 2004.
- [32] Olli Bräysy. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, 15(4) :347–368, 2003.
- [33] Vera C Hemmelmayr, Karl F Doerner, and Richard F Hartl. A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3) :791–802, 2009.
- [34] Jari Kytöjoki, Teemu Nuortio, Olli Bräysy, and Michel Gendreau. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research*, 34(9) :2743–2757, 2007.

- [35] J. Tomic and W Kempton. Using fleets of electric-drive vehicles for grid support. *J. Power. Sources*, 168 :459–468, 2007.
- [36] W. Kempton and S.E. Letendre. Electric vehicles as a new power source for electric utilities. *Transport Res.*, D 2 :157–175, 1997.
- [37] W. Kempton and J. Tomic. Vehicle-to-grid power implementation : from stabilizing the grid to supporting large-scale renewable energy. *J. Power Sources*, 144 :280–294, 2005.
- [38] W. Kempton and J. Tomic. Vehicle-to-grid power fundamentals : calculating capacity and net revenue. *J. Power Sources*, 144 :268–279, 2005.
- [39] C. Hutson, G.K. Venayagamoorthy, and K.A. Corzine. Intelligent scheduling of hybrid and electric vehicle storage capacity in a parking lot for profit maximization in grid power transactions. 2008.
- [40] W. Shi and V.W.S. Wong. Real-time vehicle-to-grid control algorithm under price uncertainty. In *SmartGridComm*, 2011.
- [41] Friric Nicolas, Marie Devillers, Jules Bodineau, and Jmie Cohen. Un rau intelligent pour vcules ctriques le smart grid : une solution face rgence des vcules ctriques ?
- [42] Charles S ReVelle and Ralph W Swain. Central facilities location. *Geographical analysis*, 2(1) :30–42, 1970.
- [43] Trevor S Hale and Christopher R Moberg. Location science research : a review. *Annals of Operations Research*, 123(1-4) :21–35, 2003.
- [44] Xueping Li, Zhaoxia Zhao, Xiaoyan Zhu, and Tami Wyatt. Covering models and optimization techniques for emergency response facility location and planning : a review. *Mathematical Methods of Operations Research*, 74(3) :281–310, 2011.
- [45] Reza Zanjirani Farahani, Nasrin Asgari, Nooshin Heidari, Mahtab Hosseininia, and Mark Goh. Covering problems in facility location : A review. *Computers & Industrial Engineering*, 62(1) :368–407, 2012.
- [46] Richard Church and Charles R Velle. The maximal covering location problem. *Papers in regional science*, 32(1) :101–118, 1974.
- [47] Liang Feng, Shaoyun Ge, and Hong Liu. Electric vehicle charging station planning based on weighted voronoi diagram. pages 1–5, 2012.
- [48] Shaoyun Ge, Liang Feng, and Hong Liu. The planning of electric vehicle charging station based on grid partition method. pages 2726–2730, 2011.
- [49] Jee E Kang and WW Recker. An activity-based assessment of the potential impacts of plug-in hybrid electric vehicles on energy and emissions using 1-day travel data. *Transportation Research Part D : Transport and Environment*, 14(8) :541–556, 2009.
- [50] Nora Touati-Moungla and Vincent Jost. Combinatorial optimization for electric vehicles management. *Journal of Energy and Power Engineering*, 6(5) :738–743, 2012.
- [51] Inês Frade, Anabela Ribeiro, Gonçalo Gonçalves, and António Pais Antunes. Optimal location of charging stations for electric vehicles in a neighborhood in lisbon, portugal. *Transportation research record : journal of the transportation research board*, 2252(1) : 91–98, 2011.
- [52] Joana Cavadas, Gonçalo Homem de Almeida Correia, and João Gouveia. A mip model for locating slow-charging stations for electric vehicles in urban areas accounting for driver tours. *Transportation Research Part E : Logistics and Transportation Review*, 75 : 188–201, 2015.

- [53] Hisatomo Hanabusa and Ryota Horiguchi. A study of the analytical method for the location planning of charging stations for electric vehicles. pages 596–605, 2011.
- [54] Sara Mehar and Sidi Mohammed Senouci. An optimization location scheme for electric charging stations. 1 :1–5, 2013.
- [55] Haris Gavranović, Alper Barut, Gürdal Ertek, Orkun Berk Yüzbaşıoğlu, Osman Pekpostalcı, and Önder Tombuş. Optimizing the electric charge station network of eşarj. *Procedia Computer Science*, 31 :15–21, 2014.
- [56] Yanqing Li, Ling Li, Jing Yong, Yuhai Yao, and Zhiwei Li. Layout planning of electrical vehicle charging stations based on genetic algorithm. In *Electrical Power Systems and Computers*, pages 661–668. Springer, 2011.
- [57] Long Jia, Zechun Hu, Yonghua Song, and Zhuowei Luo. Optimal siting and sizing of electric vehicle charging stations. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, pages 1–6. IEEE, 2012.
- [58] Y.W. Wang. An optimal location choice model for recreation-oriented scooter recharge stations. *Transport. Res. Part D : Transport and Environment*, 12(3) :231–237, 2007.
- [59] Y.W. Wang. Locating battery exchange stations to serve tourism transport. *Transport. Res. Part D : Transport and Environment*, 13(3) :193–197, 2008.
- [60] Y.W. Wang and C.C. Lin. Locating multiple types of recharging stations for battery-powered electric vehicle transport. *Transport Res.*, Part E 58 :76–87, 2013.
- [61] Andy Ip, Simon Fong, and Elaine Liu. Optimization for allocating bev recharging stations in urban areas by using hierarchical clustering. In *Advanced Information Management and Service (IMS), 2010 6th International Conference on*, pages 460–465. IEEE, 2010.
- [62] Kristien Clement-Nyns, Edwin Haesen, and Johan Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *Power Systems, IEEE Transactions on*, 25(1) :371–380, 2010.
- [63] Lingwen Gan Ufuk Topcu Steven Low. Optimal decentralized protocol for electric vehicle charging. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) pp 5798-5804*, 2011.
- [64] M. Ramezani, M. Graf, and H. Vogt. A simulation environment for smart charging of electric vehicles using a multi-objective evolutionary algorithm. In *Information and Communication on Technology for the Fight against Global Warming*, volume ICT-GLOW 2011, LNCS 6868, page pp. 5663, 2011.
- [65] J. Lee, G.L. Park, H.Y Kwak, and H. Jeon. Design of an energy consumption scheduler based on genetic algorithms in the smart grid. In *Computational Collective Intelligence. Technologies and Applications*, volume ICCCI 2011, Part I, LNCS 6922, page 438447, 2011.
- [66] Kristien Clement, Edwin Haesen, and Johan Driesen. Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids. In *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, pages 1–7. IEEE, 2009.
- [67] Valentin Robu, Sebastian Stein, Enrico H Gerding, David C Parkes, Alex Rogers, and Nicholas R Jennings. An online mechanism for multi-speed electric vehicle charging. pages 100–112, 2012.
- [68] Saeid Bashash, Scott J. Moura, Joel C. Forman, and Hosam K. Fathy. Plug-in hybrid electric vehicle charge pattern optimization for energy cost and battery longevity. *Journal of Power Sources*, 196 :541–549, 2010. doi : 10.1016/j.jpowsour.2010.07.001.

- [69] Kevin Mets, Tom Verschueren, Wouter Haerick, Chris Develder, and Filip De Turck. Optimizing smart energy control strategies for plug-in hybrid electric vehicle charging. pages 293–299, 2010.
- [70] A. Artmeier, J. Haselmayr, M. Leucker, and M. Sachenbacher. The optimal routing problem in the context of battery-powered electric vehicles. In *Workshop CROCS at CPAIOR-10, 2nd International Workshop on Constraint Reasoning and Optimization for Computational Sustainability*, 2010.
- [71] Ning Shi. Constrained shortest path problem. *Automation Science and Engineering, IEEE Transactions on*, 7(1) :15–23, 2010.
- [72] Martin Sachenbacher, Martin Leucker, Andreas Artmeier, and Julian Haselmayr. Efficient energy-optimal routing for electric vehicles. In *Twenty-fifth AAAI conference on artificial intelligence*, 2011.
- [73] Jochen Eisner, Stefan Funke, and Sabine Storandt. Optimal route planning for electric vehicles in large networks. In *AAAI*, 2011.
- [74] Timothy M Sweda and Diego Klabjan. Finding minimum-cost paths for electric vehicles. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, pages 1–4. IEEE, 2012.
- [75] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 2001.
- [76] C. Bloch et PChatonnay W.Ramdane cherif, M. Haj Rachid. New notation and classification scheme for vehicle routing problems. *RAIRO*, DOI 10.1051/ro/2014030 : To appear, 2014.
- [77] Ryan G Conrad and Miguel Andres Figliozzi. The recharging vehicle routing problem. In *Proceedings of the 2011 Industrial Engineering Research Conference*, 2011.
- [78] S. Erdogan and E. Miller-Hooks. A green vehicle routing problem. *Transport. Res., Part E* 48 :100–114, 2012.
- [79] Xiangning Lin, Jinwen Sun, Shengfang Ai, Xiaoping Xiong, Yunfei Wan, and Dexian Yang. Distribution network planning integrating charging stations of electric vehicle with v2g. *International Journal of Electrical Power & Energy Systems*, 63 :507–512, 2014.
- [80] Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle routing problem with time windows and recharging stations. Technical report, University of Kaiserslautern, Germany, 2012.
- [81] Dominik Goeke, Michael Schneider, and DB Schenker Endowed Assistant Professorship. Routing a mixed fleet of electric and conventional vehicles. Technical report, Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), 2014.
- [82] Gerhard Hiermann, Jakob Puchinger, and Richard F Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. Technical report, Working Paper. Accessed July 17, 2014. URL : http://prolog.univie.ac.at/research/publications/downloads/Hie_2014_638.pdf, 2014.
- [83] Ángel Felipe, M Teresa Ortuño, Giovanni Righini, and Gregorio Tirado. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E : Logistics and Transportation Review*, 71 : 111–128, 2014.

- [84] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6) :369–395, 2004.
- [85] Tao Zheng, Gang Wu, Guang-Hong Liu, and Qing Ling. *Multi-objective nonlinear model predictive control : Lexicographic method*. INTECH Open Access Publisher, 2010.
- [86] J.J. Brijnesh and F. Wysotski. The maximum weighted clique problem and hopfield network. In *Proceedings European Symposium on Artificial Neural Networks*, 2004.
- [87] K. Yamaguchi and S. Masuda. A new exact algorithm for the maximum weight clique. In *the 23rd International Technical Conference on Circuits/Systems, Computers and Communications*, 2008.
- [88] R. Garey and S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness, A Series of Books in the Mathematical Sciences*. W. H. Freeman and Company, 1979.
- [89] E. Balas and C. S. Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19 :247–253, 1989.
- [90] B. Vaidyanathan and R. K Ahuja. Fast algorithms for specially structured minimum cost flow problems with applications. *Oper. Res.*, 58 Issue 6 :1681–1696, 2010.
- [91] J.B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Oper. Res.*, 41(2) : 338–350, 1993.
- [92] Lingwen Gan, Ufuk Topcu, and Steven H Low. Stochastic distributed protocol for electric vehicle charging with discrete charging rate. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE, 2012.
- [93] A. W.J. Kolen, J. K. Lenstra, C. H. Papadimitriou, and F. C.R. Spijksma. Interval scheduling : A survey. *Naval Research Logistics (NRL)*, 54 :530–543, 2007.
- [94] M.Y. Kovalyov, C.T. Ng, and T.C.E. Cheng. Fixed interval scheduling : Models, applications, computational complexity and algorithms. *Eur. J. Oper. Res.*, 178 :331–342, 2007.
- [95] GB Dantzig and DR Fulkerson. Minimizing the number of tankers to meet a fixed schedule. *The Basic George B. Dantzig*, page 91, 2003.
- [96] Ilya Gertsbakh and Helman I Stern. Minimal resources for fixed and variable job schedules. *Operations Research*, 26(1) :68–85, 1978.
- [97] Udaiprakash I Gupta, Der-Tsai Lee, and Joseph Y.-T. Leung. An optimal solution for the channel-assignment problem. 1979.
- [98] L.G. Kroon, M. Salomon, and L. N. Van Wassenhove. Exact and approximation algorithms for the tactical fixed interval scheduling problem. *Operations Research*, 45 (4) :624–638, 1997.
- [99] Randeep Bhatia, Julia Chuzhoy, Ari Freund, and Joseph Seffi Naor. Algorithmic aspects of bandwidth trading. In *Automata, Languages and Programming*, pages 751–766. Springer, 2003.
- [100] VR Dondeti and Hamilton Emmons. Fixed job scheduling with two types of processors. *Operations Research*, 40(1-supplement-1) :S76–S85, 1992.
- [101] Qiwei Huang and Errol Lloyd. Cost constrained fixed job scheduling. In *Theoretical Computer Science*, pages 111–124. Springer, 2003.

- [102] Diego BC Faneyte, Frits CR Spieksma, and Gerhard J Woeginger. A branch-and-price algorithm for a hierarchical crew scheduling problem. *Naval Research Logistics (NRL)*, 49(8) :743–759, 2002.
- [103] Klaus Jansen. An approximation algorithm for the license and shift class design problem. *European Journal of Operational Research*, 73(1) :127–131, 1994.
- [104] E.M. Arkin and E.B. Silverberg. Scheduling with fixed start and end times. *Discrete Appl. Math.*, 18 :1–8, 1987.
- [105] K.I. Bouzina and H. Emmons. Interval scheduling on identical machines. *J. Global Optim.*, 9 :379–393, 1996.
- [106] Ulrich Faigle and Willem M Nawijn. Note on scheduling intervals on-line. *Discrete Applied Mathematics*, 58(1) :13–17, 1995.
- [107] M.C. Carlisle and E.L. Lloyd. On the k-coloring of intervals. *Discrete Appl. Math.*, 59 :225–235, 1995.
- [108] E. Arkin, M. Silverberg, and B. Ellen. Scheduling jobs with fixed start and end times. *Discrete Applied Mathematics*, 18(1) :1–8, 1987.
- [109] Peter Brucker and Lars Nordmann. Thek-track assignment problem. *Computing*, 52(2) :97–122, 1994.
- [110] Chi To Ng, Tai Chiu Edwin Cheng, Andrei M Bandalouski, Mikhail Y Kovalyov, and Sze Sing Lam. A graph-theoretic approach to interval scheduling on dedicated unrelated parallel machines. *Journal of the Operational Research Society*, 65(10) :1571–1579, 2013.
- [111] S. O.Krumke, C. Thielen, and S. Westphal. Interval scheduling on related machines. 38, 38 :1836–1844, 2011.
- [112] L.G. Kroon, M. Salomon, and L.N. Van Wassenhove. Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82(1) :190–205, 1995.
- [113] E. Angelelli and C. Filippi. On the complexity of interval scheduling with a resource constraint. *Theoretical Computer Science*, 412(29) :3650–3657, 2011.
- [114] Mark Bartlett, Alan M Frisch, Youssef Hamadi, Ian Miguel, S Armagan Tarim, and Chris Unsworth. The temporal knapsack problem and its solution. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 34–48. Springer, 2005.
- [115] Andreas Darmann, Ulrich Pferschy, and Joachim Schauer. Resource allocation with time intervals. *Theoretical Computer Science*, 411(49) :4217–4234, 2010.
- [116] M. Y. Kovalyov, C.T. Ng, and T.C.E. Cheng. Fixed interval scheduling : Models, applications, computational complexity and algorithms. *European Journal of Operational Research*, 178 :331–342, 2007.
- [117] R. Lipton. Online interval scheduling. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 302–311. Society for Industrial and Applied Mathematics, 1994.
- [118] J. Sgall. On-line scheduling. *Lecture Notes in Computer Science*, 1442 :196–231, 1998.
- [119] Sally A Goldman, Jyoti Parwatikar, and Subhash Suri. Online scheduling with hard deadlines. *Journal of Algorithms*, 34(2) :370–389, 2000.

- [120] Gerhard J Woeginger. On-line scheduling of jobs with fixed start and end times. *Theoretical Computer Science*, 130(1) :5–16, 1994.
- [121] Ulrich Faigle, Walter Kern, and Willem M. Nawijn. A greedy on-line algorithm for the track assignment problem. *Journal of Algorithms*, 31(1) :196–210, 1999.
- [122] K. Jansen and S. Ohring. Approximation algorithms for time constrained scheduling. *Information and Computation*, 132(2) :85–108, 1997.
- [123] L. Epstein and A. Levin. On bin packing with conflicts. *SIAM J. Optim.*, 19(4) :1270–1298, 2008.
- [124] K. Jansen. An approximation scheme for bin packing with conflicts. *Journal of Combinatorial Optimization*, 3(4) :363–377, 1999.
- [125] A. Khanafer, F. Clautiaux, and E. Talbi. New lower bounds for bin packing problems with conflicts. *European Journal of Operational Research*, 206 :281–288, 2010.
- [126] A.E. Fernandes-Muritiba, M. Iori, E. Malaguti, and P. Toth. Algorithms for the bin packing problem with conflicts. *INFORMS Journal on Computing*, 22(3) :401–415, 2009.
- [127] M. Gendreau, G. Laporte, and F. Semet. Heuristics and lower bounds for the bin packing problem with conflicts. *Computers and Operations Research*, 31(3) :347–358, 2004.
- [128] R. Garey and S.E. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. A Guide to the Theory of NP-Completeness, A Series of Books in the Mathematical Sciences., W.H. Freeman, 1979.
- [129] E. G. Jr. Coffman, J. Y. T. Leung, and D. W. Ting. Bin packing : Maximizing the number of pieces packed. *Acta Informatica*, 9(3) :263–271, 1978.
- [130] Gilbert Laporte. The vehicle routing problem : An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3) :345–358, 1992.
- [131] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil. *The Vehicle Routing Problem : Latest Advances and New Challenges : latest advances and new challenges*, volume 43. Springer Science & Business Media, 2008.
- [132] Dang Vu Tung and Anulark Pinnoi. Vehicle routing–scheduling for waste collection in hanoi. *European Journal of Operational Research*, 125(3) :449–468, 2000.
- [133] Konstantinos G Zografos and Konstantinos N Androutsopoulos. A heuristic algorithm for solving hazardous materials distribution problems. *European Journal of Operational Research*, 152(2) :507–519, 2004.
- [134] Martin Desrochers, Jan Karel Lenstra, and Martin WP Savelsbergh. A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, 46(3) :322–332, 1990.
- [135] Lawrence Bodin and Bruce Golden. Classification in vehicle routing and scheduling. *Networks*, 11(2) :97–108, 1981.
- [136] Brian Kallehauge, Jesper Larsen, Oli BG Madsen, and Marius M Solomon. *Vehicle routing problem with time windows*. Springer, 2005.
- [137] Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part i : Route construction and local search algorithms. *Transportation science*, 39(1) :104–118, 2005.
- [138] Jean-Yves Potvin and Samy Bengio. The vehicle routing problem with time windows part ii : genetic search. *INFORMS journal on Computing*, 8(2) :165–172, 1996.

- [139] Charlotte Diane Jacobs-Blecha and Marc Goetschalckx. *The Vehicle Routing Problem with Backhauls : Properties and Solution Algorithms*. Material Handling Research Center, Georgia Institute of Technology, 1992.
- [140] Yvan Dumas, Jacques Desrosiers, and Francois Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1) :7–22, 1991.
- [141] Hokey Min. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A : General*, 23(5) :377–386, 1989.
- [142] Fermín Alfredo Tang Montané and Roberto Diéguez Galvao. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3) :595–619, 2006.
- [143] The Jin Ai and Voratas Kachitvichyanukul. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 36(5) :1693–1702, 2009.
- [144] Michel Gendreau, Jean-Yves Potvin, Olli Bräumlaysy, Geir Hasle, and Arne Løkketangen. *Metaheuristics for the vehicle routing problem and its extensions : A categorized bibliography*. Springer, 2008.
- [145] Ulrich Blasum and Winfried Hochstättler. Application of the branch and cut method to the vehicle routing problem. Technical report, Universität zu Köln, 2000.
- [146] Marshall L Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research*, 42(4) :626–642, 1994.
- [147] G u Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4) :568–581, 1964.
- [148] Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2) :498–516, 1973.
- [149] Shen Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal, The*, 44(10) :2245–2269, 1965.
- [150] Marshall L Fisher and Ramchandran Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2) :109–124, 1981.
- [151] Roman Lutz, Uwe Schöning, and Dipl-Inf Gunnar Völkel. *Adaptive Large Neighborhood Search*. PhD thesis, Ulm, Universität Ulm, Bachelorarbeit, 2014, 2015.
- [152] Martin WP Savelsbergh. Local search in routing problems with time windows. *Annals of Operations research*, 4(1) :285–305, 1985.
- [153] Olli Bräysy and Michel Gendreau. Route construction and local search algorithms for the vehicle routing problem with time windows. *SINTEF Applied Metaheuristics*, pages 1–28, 2001.
- [154] Georges A Croes. A method for solving traveling-salesman problems. *Operations research*, 6(6) :791–812, 1958.
- [155] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint ProgrammingCP98*, pages 417–431. Springer, 1998.
- [156] David Pisinger and Stefan Ropke. Large neighborhood search. In *Handbook of metaheuristics*, pages 399–419. Springer, 2010.

- [157] Ravindra K Ahuja, Özlem Ergun, James B Orlin, and Abraham P Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1) : 75–102, 2002.
- [158] Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4) : 455–472, 2006.
- [159] Emilie Danna and Laurent Perron. Structured vs. unstructured large neighborhood search : A case study on job-shop scheduling problems with earliness and tardiness costs. In *Principles and Practice of Constraint Programming–CP 2003*, pages 817–821. Springer, 2003.
- [160] Glaydston Mattos Ribeiro and Gilbert Laporte. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3) :728–735, 2012.
- [161] Eric Prescott-Gagnon, Guy Desaulniers, and Louis-Martin Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4) :190–204, 2009.
- [162] Gerhard Schrimpf, Johannes Schneider, Hermann Stamm-Wilbrandt, and Gunter Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2) :139–171, 2000.
- [163] Refael Hassin and Ariel Keinan. Greedy heuristics with regret, with application to the cheapest insertion algorithm for the tsp. *Operations Research Letters*, 36(2) :243–246, 2008.
- [164] Mauro Dell’Amico, Michele Monaci, Corrado Pagani, and Daniele Vigo. Heuristic approaches for the fleet size and mix vehicle routing problem with time windows. *Transportation Science*, 41(4) :516–526, 2007.
- [165] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2) :254–265, 1987.
- [166] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts : Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826 :1989, 1989.
- [167] Zahra Naji Azimi. Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163(2) :705–733, 2005.
- [168] Moez Hammami and Khaled Ghédira. Cosats : A new cooperation model between simulated annealing and tabu search for the k-graph partitioning problem. In *Soft Computing as Transdisciplinary Science and Technology*, pages 863–873. Springer, 2005.
- [169] David Applegate, Robert Bixby, William Cook, and Vasek Chvátal. *On the solution of traveling salesman problems*. Rheinische Friedrich-Wilhelms-Universität Bonn, 1998.
- [170] Toshihide Ibaraki, Shinji Imahori, Koji Nonobe, Kensuke Sobue, Takeaki Uno, and Mutsunori Yagiura. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics*, 156(11) :2050–2069, 2008.
- [171] Hideki Hashimoto, Mutsunori Yagiura, and Toshihide Ibaraki. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, 5(2) :434–456, 2008.

- [172] Puca Huachi Vaz Penna, Anand Subramanian, and Luiz Satoru Ochi. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2) :201–232, 2013.
- [173] Feiyue Li, Bruce Golden, and Edward Wasil. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(9) :2734–2742, 2007.
- [174] Alberto Del Pia and Carlo Filippi. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2) :125–141, 2006.
- [175] Alain Hertz and Michel Mittaz. A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. *Transportation science*, 35(4) :425–434, 2001.
- [176] Bin Hu and Günther R Raidl. Variable neighborhood descent with self-adaptive neighborhood-ordering. In *Proceedings of the 7th EU/MEeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics*. Citeseer, 2006.
- [177] Ping Chen, Hou-kuan Huang, and Xing-Ye Dong. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, 37(2) :1620–1627, 2010.
- [178] Louis-Martin Rousseau, Michel Gendreau, and Gilles Pesant. Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of heuristics*, 8(1) :43–58, 2002.

Résumé

Le véhicule électrique est actuellement au coeur des alternatives énergétiques qui permettent de faire face à la croissance du coût du carburant et au réchauffement climatique. En revanche, l'autonomie limitée des batteries des véhicules électriques et l'indisponibilité d'un nombre suffisant de bornes de recharge représentent des enjeux majeurs auxquels se trouvent confrontés les utilisateurs. Le déploiement des véhicules électriques doit alors passer par la conception et l'expérimentation des outils d'aide à la décision pour une gestion optimisée et adaptée de l'écosystème du véhicule électrique. C'est dans ce contexte que s'inscrit cette thèse qui vise à fournir des outils d'aide à la décision pour l'optimisation des usages, de la recharge et des tournées des véhicules électriques dans le cadre industriel.

Dans un premier temps, nous proposons d'étudier le problème d'optimisation conjointe de l'affectation et de la recharge des véhicules électriques. L'objectif de ce problème est de maximiser l'usage des véhicules électriques et minimiser les coûts de recharge tout en prenant en compte les contraintes d'ordre opérationnel et technique. Pour résoudre ce problème, nous proposons une méthode exacte et deux heuristiques. Nous comparons les performances de ces méthodes sur des instances réelles et d'autres aléatoires. Nous exposons ensuite plusieurs extensions au problème de base en intégrant de nouvelles fonctions objectif et de nouvelles contraintes. Nous étudions par la suite notre problème de point de vue ordonnancement et nous proposons une étude de complexité et des algorithmes d'approximation avec garantie de performance pour le problème d'ordonnancement d'intervalles sous contraintes d'énergie. Finalement, nous nous intéressons à un nouveau problème de construction de tournées pour une flotte mixte de véhicules électriques et thermiques. Pour résoudre ce problème, nous proposons des heuristiques et des méta-heuristiques hybrides et nous comparons les performances des différentes méthodes sur des instances généralisées de la littérature.

Mots-clés: Véhicule électrique, Optimisation, Recharge, Tournée.

Abstract

Electric Vehicles may decrease transportation-related emissions and provide for less dependence on foreign oil. However, electric vehicles are still facing many weaknesses related to the high purchase prices, limited battery range and scarce charging infrastructure. The deployment of electric vehicles must then involve the design and the deployment of charging infrastructures.

Within this study, the overall objective is to provide enhanced optimization methods and decision tools for electric vehicles assignment, charging and routing that are relevant to different real-world constraints. Firstly, we propose to study the joint scheduling and optimal charging of electric vehicles problem. This problem consists in assigning a set of already constructed routes to the available electric and conventional vehicles and in, simultaneously, optimizing the electric vehicles charging costs while ensuring that all constraints are satisfied. The objective of this problem is to maximize the use of EVs and to minimize charging costs. Secondly, we propose different extensions to our baseline problem and we vary the objective functions and the considered constraints. Moreover, our problem can be seen as a fixed interval scheduling problem with complementary constraints of energy. We propose then to study the complexity and the approximability of many variants of this new problem. Finally, we consider a new variant of the electric vehicle routing problem with a heterogenous fleet of vehicles and we propose different heuristics and metaheuristics to solve it. We test the different solving methods on benchmark instances and we evaluate the efficiency of each method.

Keywords: Electric vehicles, Optimization, Recharge, Routing.

