

الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

جامعة محمد البشير الإبراهيمي برج بوعريريج
UNIVERSITÉ MOHAMED EL BACHIR EL IBRAHIMI DE BORDJ BOU ARRERIDJ
كلية الرياضيات و الإعلام الآلي
FACULTÉ DE MATHÉMATIQUES ET D'INFORMATIQUE
قسم الإعلام الآلي
DÉPARTEMENT D'INFORMATIQUE



THÈSE

En vue de l'obtention du grade de Docteur en 3^{ème} cycle

Filière : Informatique

Spécialité : Informatique

Par : **Bilal BENCHARIF**

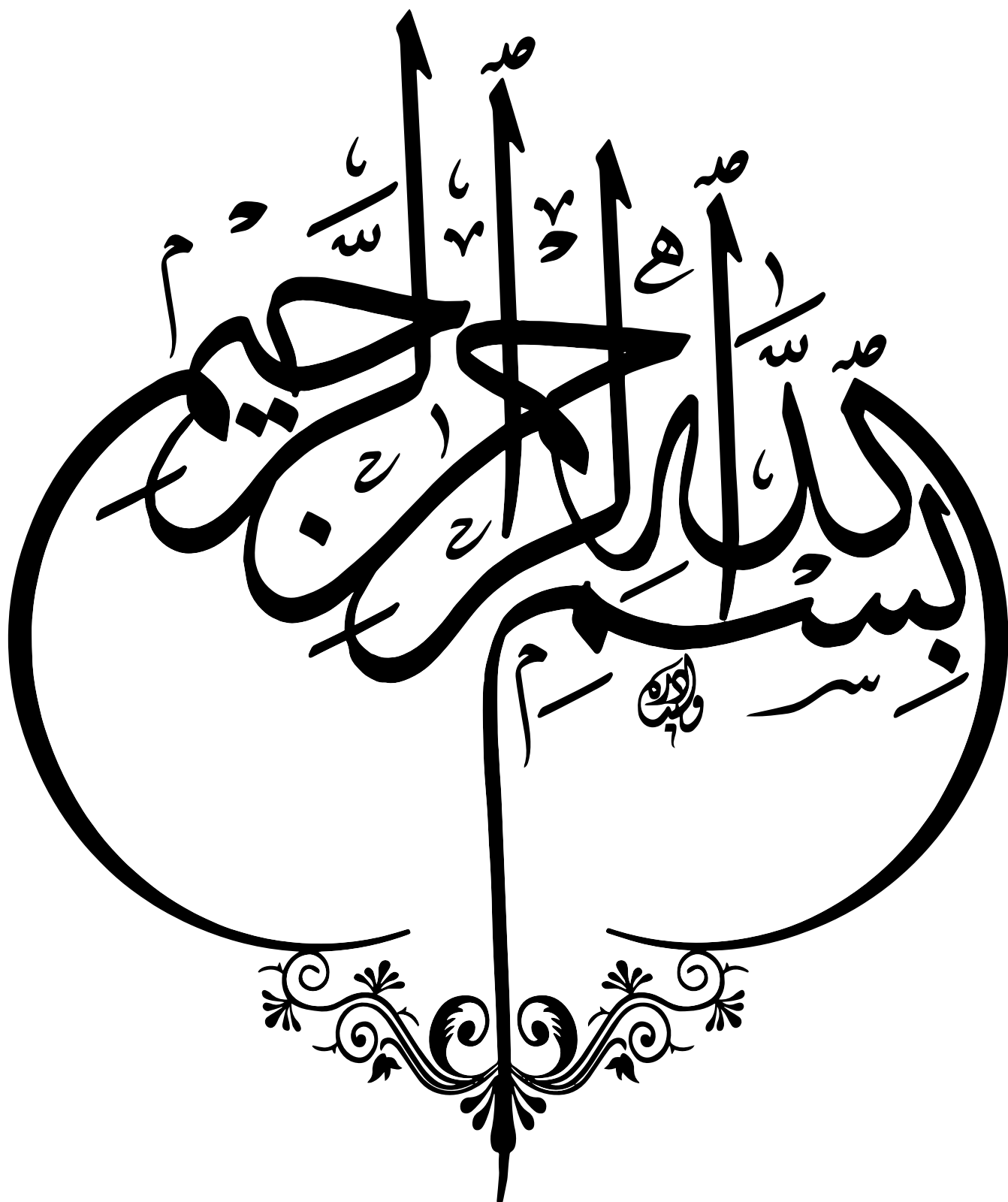
THÈME

**Issues de la simulation basée crowdsourcing pour
l'informatique verte**

Soutenue publiquement, le 01/07/2025 à 16h00, devant le jury composé de :

Mr. Samir AKHROUF	Professeur	à l'U. M'Sila	Président
Mr. Mohamed Amine BEGHOURA	Maître de Conférences A	à l'U. BBA	Directeur de thèse
Mr. Djaafar ZOUACHE	Professeur	à l'U. BBA	Examineur
Mr. Mouhoub BELAZZOUG	Maître de Conférences A	à l'U. BBA	Examineur
Mr. Seddik BESSOU	Maître de Conférences A	à l'U. Setif	Examineur
Mr. Boualem BRAHMI	Maître de Conférences A	à l'U. BBA	Invité

Année Universitaire 2024/2025



Dédicaces

Je tiens à dédier cette thèse,

*À ma chère mère et ma sœur, qui ne sont plus parmi nous, mais qui attendaient avec
impatience ce moment.*

*Leur soutien indéfectible et leur foi en moi ont été une source constante de force, et je
dédie cette réalisation à leur mémoire.*

*À ma chère tante et mon oncle Ali, pour leur soutien, leur amour et leurs sacrifices
tout au long de ma vie. Leur présence et leurs encouragements ont été d'une importance
capitale dans ce parcours. Cette thèse leur est également dédiée.*

*À mon cher beau-frère Fouad, pour son soutien constant et son encouragement.
À mes neveux Anas et Iyad, pour leur affection et leurs sourires qui m'ont toujours
apporté de la joie. Cette thèse est dédiée à eux aussi.*

*À mon ami Tawfiq, pour son amitié sincère, son soutien indéfectible et sa présence
constante tout au long de cette aventure.*

***À mon frère Issam et ma sœur, pour leur soutien sans limites, leurs
encouragements et leur amour tout au long de ce voyage.***

Remerciements

Louange à Allah, le Tout-Puissant, Le Tout-Puissant, pour Sa miséricorde infinie et pour la force qu'Il m'a donnée afin de mener à bien ce travail de thèse. Toutes mes pensées sont dirigées vers Lui, Le Créateur des cieux et de la terre.

Je tiens à exprimer ma plus profonde gratitude à toutes les personnes qui ont contribué à l'aboutissement de ce travail. Leur soutien et leur dévouement ont été une source d'inspiration tout au long de cette recherche.

*Je remercie tout d'abord pour son encadrement exceptionnel et son soutien constant. Sa disponibilité, sa patience et ses conseils précieux ont été des éléments clés pour la réalisation de ce travail. Grâce à ses remarques constructives et à son expertise, j'ai pu affiner mon approche et approfondir les aspects essentiels de ma recherche. **Monsieur BEGHOURA Mohamed Amine**, Maître de Conférences A à l'Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arreridj.*

*Je tiens également à remercier pour la confiance qu'il m'a accordée en me proposant ce sujet. Son soutien et ses conseils avisés ont permis de poser des bases solides et de guider mes premières étapes dans ce travail de thèse. **Monsieur Abdelhak BOUBETRA**, mon premier directeur de recherche, aujourd'hui retraité.*

Je tiens à exprimer ma sincère reconnaissance et mes remerciements aux membres du jury qui ont accepté d'évaluer et d'examiner cette thèse :

*Je remercie particulièrement pour son rôle essentiel en tant que président. Sa rigueur académique, ses analyses approfondies et ses remarques constructives ont grandement contribué à la qualité de cette thèse. Ses commentaires ont été une source précieuse d'amélioration pour ce travail. **Monsieur AKHROUF Samir**, Professeur à l'Université de M'sila, Président du jury.*

*Enfin, je remercie pour son évaluation approfondie et ses remarques éclairées. Son expérience et ses connaissances ont été déterminantes pour améliorer la rigueur scientifique de ce travail. **Monsieur ZOUACHE Djaafar**, Professeur à l'Université de Bordj Bou Arreridj, Examineur.*

*Je remercie également pour ses observations pertinentes et ses conseils techniques qui ont enrichi cette recherche. Ses remarques ont permis de clarifier et de renforcer certains aspects méthodologiques du travail. **Monsieur BELAZZOUG Mouhoub**, Maître de Conférences A à l'Université de Bordj Bou Arreridj, Examineur.*

*Je tiens à adresser mes sincères remerciements à pour ses critiques constructives et ses remarques détaillées. Son expertise et ses suggestions ont permis de perfectionner plusieurs aspects clés de ma thèse. **Monsieur BESSOU Seddik**, Maître de Conférences A à l'Université de Sétif, Examineur.*

*Je remercie sincèrement **Monsieur BRAHMI Boualem**, Maître de Conférences A à l'Université de Bordj Bou Arreridj, Invité. Qui a accepté l'invitation à participer à la discussion et à enrichir le débat scientifique.*

*Je souhaite exprimer ma sincère gratitude à **Monsieur Akram**, ingénieur en informatique au sein de l'entreprise, pour l'assistance technique précieuse qu'il m'a apportée tout au long de la réalisation de ma thèse.*

*Je ne saurais oublier ma famille, qui a toujours été une source de soutien inébranlable. À **ma mère** et à **ma sœur** bien-aimée Que Dieu leur accorde Sa miséricorde, qui m'ont quitté avant de pouvoir assister à l'aboutissement de ce travail, je dédie cette thèse. Leur amour, leurs sacrifices et leurs prières ont été une lumière guidant mon chemin.*

Je tiens également à remercier mon frère et ma sœur pour leur soutien moral et constant. Leur présence et leurs encouragements ont été des piliers essentiels de ma réussite. Leur amour m'a donné la force de persévérer à chaque étape de ce parcours.

Je remercie également mes amis et collègues pour leur soutien scientifique et moral. Leurs discussions enrichissantes et leurs conseils avisés ont été d'une grande aide dans la progression de cette thèse. Leur aide et leur amitié ont été une véritable source de motivation pendant toute la durée de ce travail.

Enfin, je remercie toutes les personnes, proches ou lointaines, qui ont contribué de près ou de loin à la réalisation de ce travail. Leur soutien et leur générosité n'ont pas été oubliés, et je leur suis infiniment reconnaissant.

Que Dieu bénisse tous ceux qui ont contribué à l'aboutissement de ce travail.

Bilal Bencharif, Juillet 2025

Issues de la simulation basée crowdsourcing pour l'informatique verte

Résumé

Dans cette thèse, nous proposons une approche innovante d'optimisation des systèmes logistiques en intégrant des données crowdsourcées en temps réel, issues de l'API Google Distance Matrix. En combinant un algorithme de recherche tabou probabiliste à cette exploitation de données collaboratives, nous avons développé un cadre permettant d'améliorer la gestion des tournées de livraison dans le contexte de l'informatique verte. Nos expérimentations sur des données simulées et réelles révèlent que la vitesse a un impact déterminant sur la consommation énergétique. Par ailleurs, l'analyse de la distance parcourue démontre que celle-ci ne constitue pas à elle seule un indicateur fiable de la consommation : en effet, un trajet plus long n'implique pas nécessairement une consommation énergétique supérieure, notamment lorsqu'il s'accompagne de conditions de circulation optimisées. Ces résultats attestent du potentiel des approches collaboratives pour transformer la logistique durable et ouvrent de nouvelles perspectives pour l'optimisation des systèmes de transport.

Mots Clés : Informatique Verte, Optimisation Logistique, Données Issues du Crowdsourcing, API Google Distance Matrix, Algorithme de Recherche Tabou Probabiliste, Pollution-Routing Problem.

Crowdsourcing-based simulation issues for green IT

Abstract

In this thesis, we propose an innovative approach to optimize logistics systems by integrating crowdsourced real-time data collected via the Google Distance Matrix API. By leveraging a probabilistic tabu search algorithm, we have developed a framework aimed at improving the management of delivery routes in the context of green computing. Experiments conducted on both simulated and real data reveal that speed has a decisive impact on energy consumption. Furthermore, the analysis demonstrates that the traveled distance alone is not a reliable indicator of energy usage—indeed, a longer route does not necessarily imply higher energy consumption, especially when coupled with optimized traffic conditions. These results underscore the potential of collaborative approaches in transforming sustainable logistics and open up new perspectives for optimizing transportation systems.

Keywords : Green IT, Logistics Optimization, Crowdsourced Data, Google Distance Matrix API, Probabilistic Tabu Search Algorithm, Pollution-Routing Problem.

مشكلات المحاكاة المعتمدة على التعهيد الجماعي لتكنولوجيا المعلومات الخضراء

ملخص

في هذه الأطروحة، نقدم نهجاً مبتكراً لتحسين الأنظمة اللوجستية من خلال دمج بيانات التعهيد الجماعي التي تجمع في الوقت الحقيقي عبر استخدام واجهة برمجة التطبيقات لمصفوفة المسافات من جوجل. بالاعتماد على خوارزمية البحث التابو الاحتمالية، قمنا بتطوير إطار عمل يهدف إلى تحسين إدارة جداول التوصيل في سياق المعلوماتية الخضراء. تُظهر التجارب التي أُجريت على بيانات محاكاة وبيانات واقعية أن السرعة تؤثر بشكل حاسم على استهلاك الطاقة. كما يبرز التحليل أن المسافة المقطوعة لا تعد مؤشراً موثوقاً بمفردها لاستهلاك الطاقة، إذ إن الرحلة الأطول قد لا تؤدي بالضرورة إلى زيادة في الاستهلاك، خاصةً عندما تكون مصحوبة بظروف مرور محسنة. تؤكد هذه النتائج إمكانيات النهج التعاوني في تحويل مجال اللوجستيات نحو استدامة بيئية، وتفتح آفاقاً جديدة لتحسين أنظمة النقل.

الكلمات المفتاحية : اتكنولوجيا المعلومات الخضراء، تحسين اللوجستيات، البيانات المستخلصة من التعهيد الجماعي، واجهة برمجة تطبيقات مصفوفة المسافات من جوجل، خوارزمية البحث التابو الاحتمالية، مشكلة التوجيه مع التلوث.

Contents

Dédicaces	1
Remerciements	2
Résumé	5
Table des matières	8
Liste des figures	11
Liste des tableaux	12
Liste des algorithmes	13
Introduction générale	13
I.1 Contexte de la thèse	14
I.2 Objectif et Contributions	16
I.3 Organisation de la thèse	16
1 État de l'Art	18
1.1 Introduction	19
1.2 PRP	20
1.2.1 Modèle microscopique	20
1.2.2 Modèle macroscopique	21
1.3 Modèle hybride	24
1.4 Autres modèles	24
1.5 Le crowdsourcing pour les problèmes de routage	25
1.6 Synthèse des Lacunes de Recherche	26
2 Description du problème	28
2.1 Introduction	29
2.2 Étude de Cas : Distribution Pharmaceutique en Algérie	29
2.3 Définition et formulation du problème VPRPTW	30

2.4	Définition et formulation du problème SG-VCVRP-TW	33
2.5	Modèle de consommation d'énergie	33
3	Méthodologie de la solution	38
3.1	Introduction	39
3.2	Algorithme Tabou Probabiliste	39
3.2.1	Principe général de la recherche taboue	39
3.2.2	Initialisation de l'algorithme	40
3.2.3	Itérations et génération des mouvements	40
3.2.4	Pseudocode de l'algorithme	41
3.2.5	Analyse du fonctionnement étape par étape	42
3.2.6	Composante probabiliste de l'algorithme	42
3.2.7	Renforcement de l'exploration du voisinage par boucle imbriquée . .	43
3.3	Génération des matrices de données	45
3.4	Construction de la solution initiale par insertion gloutonne	46
3.5	Structure de voisinage	47
3.6	Gestion de la liste taboue	49
4	Le Crowdsourcing et son application via les services Google	50
4.1	Introduction	51
4.2	Principes fondamentaux du Crowdsourcing	51
4.2.1	Définition	51
4.2.2	Acteurs du crowdsourcing	51
4.2.3	Typologies principales du crowdsourcing	52
4.3	Intégration du Crowdsourcing dans les systèmes intelligents	52
4.3.1	Rôle du crowdsourcing dans les systèmes intelligents	52
4.3.2	Applications typiques	53
4.3.3	Avantages pour les systèmes intelligents	53
4.3.4	Limites et défis	53
4.4	Avantages et défis du Crowdsourcing	54
4.4.1	Avantages	54
4.4.2	Limites	54
4.4.3	Défis majeurs	54
4.5	Intégration de l'API Google Distance Matrix avec Java	55
4.5.1	Présentation générale	55
4.5.2	Objectifs de l'intégration	55
4.5.3	Apport des données issues du crowdsourcing	56
4.5.4	Code Java : récupération de la matrice de distance et de temps . .	56
4.5.5	Découpage par lots pour respecter les limitations de l'API	57
4.5.6	Exportation vers Excel	58

5	Méthodologie Expérimentale et Analyse des Résultats	60
5.1	Introduction	61
5.2	Présentation du Problème et Contexte Expérimental	61
5.3	Paramètres constants dans l'équation de la consommation énergétique . . .	63
5.4	Synthèse des Résultats et Analyse	64
5.5	Analyse statistique et validation	76
	Conclusion Générale et Perspectives	80
	Bibliographie	83
	Annexe	A

Liste des figures

I.1	Représentation du contexte de recherche.	15
1.1	Classification des Problèmes de Tournées de Véhicules Verts.	19
3.1	Structure de voisinage (1) échange iCROSS (2) échange 1-0. [1]	48
5.1	Emplacements des clients sur la carte.	62
5.2	Longue distance avec des émissions de CO_2 réduites.	70
5.3	CO_2 emissions for Actual Speed vs. Fixed Speed. (a) Fixed Speed at 40 km/h. (b) Fixed Speed at 60 km/h. (c) Fixed Speed at 80 km/h. (d) Fixed Speed at 100 km/h.	75
5.4	CO_2 emissions for real speed vs constant speeds conditions.	75

Liste des tableaux

2.1	Paramètres du modèle de consommation d'énergie du véhicule.	35
2.2	Paramètres complémentaires du modèle de consommation d'énergie du véhicule.	36
5.1	Valeurs typiques attribuées aux paramètres.	63
5.2	Consommation d'énergie, distance et émissions de CO_2 en utilisant la vitesse des services Google Maps.	65
5.3	Consommation d'énergie, distance et émissions de CO_2 en utilisant la vitesse des services Google Maps avec TS simple.	66
5.4	Résultats de la solution initiale en utilisant l'heuristique d'insertion gloutonne.	69
5.5	Cas où une plus grande distance entraîne de moindres émissions.	70
5.6	Consommation énergétique à vitesse constante de 40 km/h.	71
5.7	Consommation énergétique à vitesse constante de 60 km/h.	72
5.8	Consommation énergétique à vitesse constante de 80 km/h.	72
5.9	Consommation énergétique à vitesse constante de 100 km/h.	73
5.10	Comparaison statistique des émissions de CO_2 et de la consommation d'énergie selon les profils de vitesse.	76
5.11	Statistiques de performance pour la première semaine.	78
5.12	Statistiques de performance pour la deuxième semaine.	78
5.13	Statistiques de performance pour la troisième semaine.	79
5.14	Statistiques de performance pour la quatrième semaine.	79

Liste des algorithmes

3.1	Recherche Tabou Probabiliste pour le VPRPTW	41
3.2	Recherche Tabou Probabiliste avec exploration renforcée	44
3.3	Fonction d'échange iCROSS	48

Introduction Générale

I.1 Contexte de la thèse

Le contexte environnemental urgent, porté par la nécessité de diminuer les impacts des systèmes informatiques et logistiques, a élevé indéniablement l'informatique verte et le développement durable au rang de questions clés. L'urgence de réduire les impacts des matières utilisées et de la transition énergétique en cours oblige davantage les porteurs de projets à réinventer leurs stratégies de fonctionnement, afin de rationaliser le recours aux ressources et minimiser l'impact environnemental de leur fonctionnement. Au sein des nombreuses pratiques nouvelles, le crowdsourcing émerge comme une technique innovante de collecte des informations concernant le fonctionnement « temps réel » des systèmes. En intégrant ces données « du terrain », il devient possible de mieux cerner les dynamiques opérationnelles, tout en améliorant la performance des systèmes logistiques et informatiques.

De cette manière, la problématique du Pollution-Routing Problem (PRP) est centrale dans les domaines de la logistique et de l'informatique vertes. Comme cela est schématisé par la Figure I.1, le PRP est inclus dans le domaine de l'optimisation des flux de transport, et le Green Vehicle Routing Problem (Green VRP) en constitue une version spécifique aux tournées de véhicules. La logistique verte ne se limite pas à ces concepts et désigne une vision plus globale des opérations de transport dans la perspective de réduction de l'empreinte carbone.

La notion d'informatique verte (Green IT) surplombe enfin l'ensemble en intégrant l'optimisation des infrastructures informatiques et énergétiques, matérielles ou algorithmiques. Le Green IT se définit comme l'ensemble des pratiques, méthodes et technologies – matériels, logiciels, données et réseaux – visant à réduire l'empreinte carbone et l'impact environnemental des activités humaines. Il s'agit de promouvoir une utilisation responsable et économe des ressources numériques, dans le but d'améliorer l'efficacité énergétique globale des systèmes [2, 3].

On distingue principalement deux dimensions complémentaires :

- Green IT 1.0 – IT for Green: Désigne l'usage des technologies de l'information pour réduire l'empreinte environnementale d'autres secteurs d'activité (logistique, transport, bâtiment, agriculture, etc.).

Exemples:

- Optimisation des tournées logistiques pour réduire la consommation de carburants.
- Gestion énergétique intelligente dans les bâtiments par capteurs et systèmes logiciels.
- Green IT 2.0 – Green for IT: Concerne la réduction de l'impact environnemental des systèmes informatiques eux-mêmes. Cela inclut :

- la réduction de la consommation énergétique des datacenters,
- l’allongement de la durée de vie des équipements,
- la fabrication écoresponsable du matériel,
- le recyclage des déchets électroniques (DEEE) [4].

Cette double approche place le Green IT comme un acteur majeur de la transformation durable des systèmes industriels et logistiques.

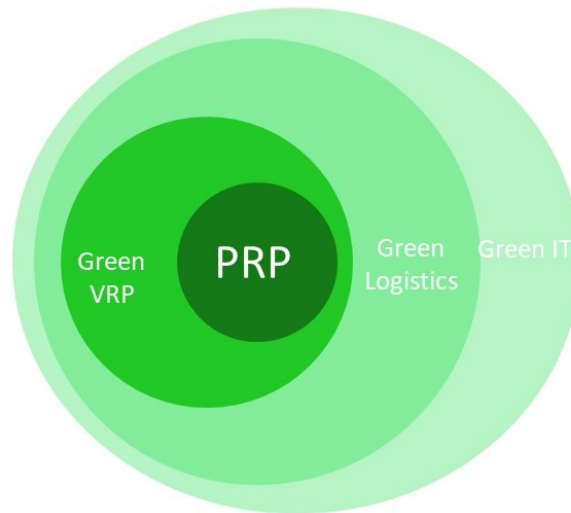


Figure I.1: Représentation du contexte de recherche.

L’intégration de données collaboratives dans des modèles de simulation constitue un pas important pour l’optimisation des processus. En sourçant des données dans des sources variées telles que les APIs de géolocalisation et de plateforme mobilité, on peut réaliser des simulations qui restituent les conditions réelles de fonctionnement des systèmes logistiques. Ce type d’approche permettant de rendre compte de la variabilité du comportement des usagers, des variations du trafic et des conditions environnementales représente un outil puissant au service du diagnostic et de l’optimisation des performances énergétiques des infrastructures.

Dans ce cadre, cette thèse interroge comment l’intégration de données crowdsourcées et l’optimisation algorithmique participent à l’amélioration de l’efficacité énergétique des systèmes logistiques. Pour y répondre, nous proposons un cadre méthodologique novateur, combinant des algorithmes d’optimisation de pointe, parmi lesquels les algorithmes de recherche tabou probabiliste, et des données en temps réel utiles au pilotage opérationnel. Une étude de cas d’optimisation des itinéraires et de gestion des flottes met en lumière comment ces approches permettent de mieux appréhender les complexes relations entre variabilité des vitesses et consommation énergétique.

I.2 Objectif et Contributions

L'objectif principal de cette recherche est de fournir aux décideurs et aux praticiens des outils analytiques et opérationnels pour améliorer l'efficacité énergétique des systèmes tout en favorisant une approche durable. En mettant en lumière les limites des modèles traditionnels et en proposant des pistes d'amélioration, cette thèse ouvre la voie à des solutions plus robustes et adaptées aux contraintes réelles du terrain. L'intégration du crowdsourcing dans des simulations avancées apparaît ainsi comme une réponse prometteuse aux défis de l'informatique verte, en alliant innovation technologique et responsabilité environnementale. Les contributions de cette recherche se déclinent comme suit :

1. **Article scientifique :**

Bilal Bencharif, Mohamed Amine Beghouira, et Emrah Demir. The volume-based pollution-routing problem with time windows: A case study. *Modelling*, 6:6, 2025.

2. **Conférence :**

Bilal Bencharif, Mohamed Amine Beghouira, et Emrah Demir. "A Simple Tabu Search with Crowdsourcing Data using Google Services API for Green Logistic: A case study", *2025 7th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, Laghouat, Algeria, 2025.

I.3 Organisation de la thèse

Afin de structurer cette réflexion, cette thèse est organisée comme suit.

Le premier chapitre (1) propose un état de l'art complet qui se concentre sur l'informatique verte, la logistique durable et le PRP. Dans ce chapitre, nous procédons à une classification détaillée des différents types de PRP, en distinguant principalement les modèles microscopiques et macroscopiques, ainsi que les approches hybrides. Cette division permet d'illustrer comment chaque catégorie répond de manière spécifique aux exigences d'optimisation environnementale, en tenant compte à la fois des mesures instantanées et des valeurs moyennes des paramètres influents.

Ce deuxième chapitre (2) permet de décrire la problématique elle-même. Nous présentons d'abord l'étude de cas de la distribution pharmaceutique en Algérie, représentant les difficultés logistiques et environnementales rencontrées par une entreprise parfois à ressources limitées, est ensuite formalisé le problème dans le cadre du *volume-based pollution-routing problem with time windows* (VPRPTW), en intégrant des contraintes de fenêtres temporelles souples, de capacités de véhicules et de réduction de la consommation énergétique. Enfin, est proposé un modèle de consommation d'énergie élaboré, reposant sur des équations précises intégrant les variations dynamiques du comportement du véhicule.

Cette organisation structurée permet de poser une base solide à l'optimisation du routage dans un contexte d'économie et de respect de l'environnement.

Le troisième chapitre (3) de la thèse présente la démarche méthodologique de la solution proposée pour répondre au VPRPTW. Cette démarche préconise d'abord d'utiliser l'algorithme de recherche tabou probabiliste, qui exploite en itérant l'espace de solutions par le biais des mécanismes de mémoire et de stratégies d'évasion des optima locaux. Puis elle formalise la génération des matrices de données (distances, temps et vitesses) grâce à l'API Distance Matrix et la construction d'une solution initiale par heuristique d'insertion gloutonne, avant de définir la structure de voisinage par les échanges iCROSS et 1-0, puis en clôturant par la gestion dynamique de la liste tabou par le biais des mécanismes d'interdiction et d'aspiration pour diversifier la recherche au mieux. Tous ces éléments préfigurent un cadre méthodologique solide pour s'attaquer à un problème NP-difficile et mieux appréhender la recherche de meilleures solutions.

Dans le quatrième chapitre (4), nous décrivons le concept de crowdsourcing, ses modalités d'applications principales et ses différents domaines d'application. Nous accentuons ses intérêts pour collecter des données en temps réel, notamment dans les systèmes logistiques intelligents. L'utilisation des services de Google, notamment Google Maps et Distance Matrix API, est citée comme exemple de sa mise en œuvre à travers cette approche du crowdsourcing.

Le cinquième chapitre (5) est consacré à la méthodologie expérimentale et à l'analyse des résultats. En effet, l'algorithme de recherche tabou probabiliste est appliqué au cas réel de la distribution pharmaceutique en Algérie ; nous comparons les scénarios fondés sur des vitesses fixes, aux scénarios où les vitesses sont celles, réelles, extraites de Google APIs. L'idée est d'étudier l'impact de ces hypothèses sur la consommation d'énergie, les émissions de CO₂ et la performance logistique globale. Les résultats expérimentaux, exposés dans plusieurs tableaux et figures, sont enrichis d'analyse statistique rigoureuse, en particulier via un test ANOVA, qui valident à la fois la significativité des écarts observés et la robustesse de la solution proposée.

Enfin, la conclusion générale (6) présente de manière synthétique les résultats atteints, mais aussi leur portée pratique. Elle présente aussi des perspectives de recherches futures qui portent sur l'amélioration des modèles d'optimisation, en vue d'atteindre une gestion des ressources à la fois plus efficace et plus durable.

Chapitre 1

État de l'Art

1.1 Introduction

Depuis l'introduction du Problème de Tournées de Véhicules (VRP) par Dantzig et Ramser [5], reconnu comme l'un des problèmes logistiques les plus étudiés pour l'optimisation des aspects économiques du transport, plusieurs variantes ont été développées. La littérature scientifique sur le VRP est riche et regroupe de nombreuses déclinaisons classiques. Cette étude porte sur la dimension environnementale du VRP, où l'objectif est de minimiser l'impact écologique tout en répondant aux exigences économiques, sociales et environnementales. Ces considérations sont englobées dans le Problème de Tournées de Véhicules Verts (GVRP).

Dans le GVRP, les véhicules peuvent utiliser deux types de sources d'énergie : les carburants alternatifs (AFV) ou l'électricité, ainsi que les carburants fossiles. Cette revue de la littérature s'intéresse aux différents modèles du GVRP appliqués aux véhicules fonctionnant aux carburants fossiles, connus sous le nom de Problème de Tournées de Véhicules Polluants.

Demir et al. [6] distinguent deux catégories de modèles de consommation de carburant : les modèles microscopiques et les modèles macroscopiques. En nous basant sur cette classification, nous proposons une structuration du PRP selon ces deux approches, comme illustré dans la figure 1.1. Par ailleurs, certains modèles récents intègrent simultanément les véhicules à carburants alternatifs et ceux à carburants fossiles ; ces travaux sont regroupés dans la section consacrée aux modèles hybrides. Enfin, les autres contributions jugées pertinentes sont présentées dans une section distincte intitulée "Autres modèles".

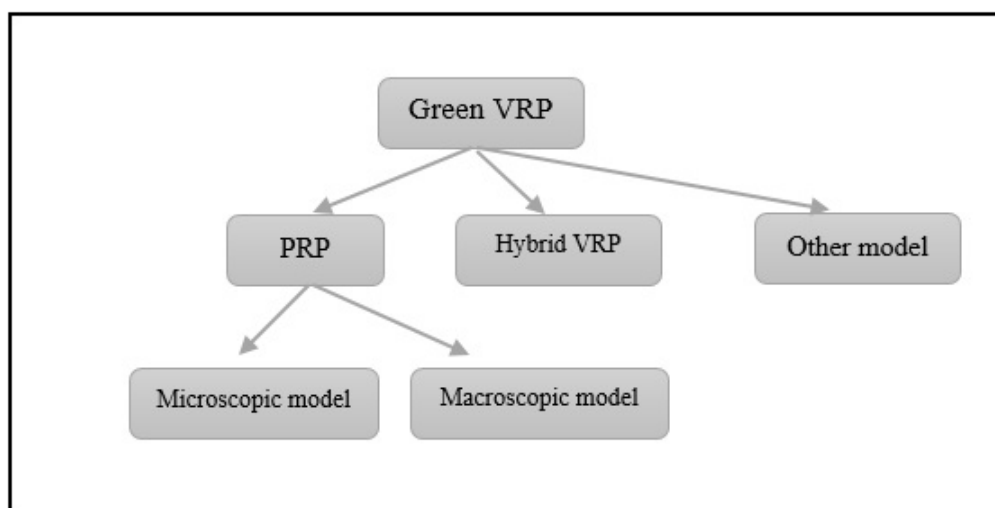


Figure 1.1: Classification des Problèmes de Tournées de Véhicules Verts.

1.2 PRP

Le PRP est une autre variante du VRP à laquelle s'ajoute une préoccupation de la minimisation du coût ou de la distance, et un souci d'impact environnemental, lui-même évalué dans une fonction objective utilisant des modèles mathématiques de la consommation de carburant et d'émissions de gaz à effet de serre (GES). Comme son nom l'indique, le PRP est donc une solution pour l'optimisation d'itinéraire cherchant dans les solutions proposées à réduire la pollution tout en offrant une performance logistique jugée acceptable.

1.2.1 Modèle microscopique

Ces modèles, appelés modèles d'émissions instantanées, permettent d'estimer la consommation énergétique en temps réel (à l'instant t). Leur principal avantage réside dans la précision des résultats obtenus, car ils reposent sur des variables cinématiques instantanées du véhicule, telles que la vitesse et l'accélération, ainsi que d'autres facteurs [6].

Dans le domaine de la logistique, l'optimisation des tournées de véhicules est souvent réalisée en utilisant les temps de trajets ou les distances parcourues comme critères de différenciation. Dans ce cadre-là, Palmer [7] a proposé une étude sur les émissions de CO_2 dans le Problème de Tournées de Véhicules avec Fenêtres de Temps (VRPTW) ; les résultats sont prometteurs puis ils montrent que les émissions de CO_2 peuvent être réduites de plus de 5%, par rapport à une optimisation des temps de trajet. Kono et al. [8] ont présenté une méthode innovante visant à réduire la consommation de carburant qui repose sur l'intégration d'un système de recherche d'itinéraires « verts » prenant en compte certains facteurs qui influencent la consommation d'énergie, comme les données géographiques et le trafic. Urquhart et al. [9] ont résolu le Problème du Voyageur de Commerce (TSP) avec des algorithmes évolutionnaires dont l'objectif est de construire des tournées à faibles émissions de CO_2 .

Bektaş et Laporte [10] ont introduit le PRP comme une extension du VRP classique, impliquant plusieurs véhicules desservant un ensemble de clients. La différence principale réside dans la prise en compte de la vitesse, du poids et d'autres facteurs propres à chaque segment de trajet afin de minimiser la fonction objective, qui est généralement une équation calculant la consommation de carburant et les émissions de gaz à effet de serre. Demir et al. [11] ont proposé une autre application du PRP permettant de calculer la consommation totale de carburant par seconde. Une heuristique de recherche adaptative à large voisinage a été employée, aboutissant à de 10 % des émissions de CO_2 .

Une extension bi-objectif du PRP a été développée par Demir et al. [12]. Le premier objectif vise à minimiser la consommation de carburant, tandis que le second cherche à réduire le temps de conduite. Ces objectifs étant conflictuels, ils ont été traités séparément. Pour résoudre le problème, une méthode de pondération combinée à une optimisation

de la vitesse et une heuristique Adaptive Large Neighborhood Search (ALNS) ont été utilisées. Koç et al. [13] ont introduit une variante du PRP prenant en compte la taille et la composition de la flotte. Cette extension du PRP avec une flotte hétérogène vise à minimiser la somme des coûts fixes des véhicules, des coûts de routage liés aux émissions de CO_2 , ainsi que les coûts de carburant et de main-d'œuvre. Une métaheuristique évolutive hybride a été utilisée pour résoudre ce problème.

Ces dernières années, le PRP a suscité un intérêt croissant. Kargari Esfand Abad et al. [14] ont proposé un modèle bi-objectif pour le PRP avec ramassage et livraison, intégrant la consolidation des expéditions dans un système de cross-docking. Les deux objectifs sont les suivants : minimiser le coût total du système et réduire la consommation totale de carburant des véhicules. Trois algorithmes ont été retenus pour résoudre ce problème : l'algorithme génétique à tri non dominé (NSGA-II), l'algorithme génétique à classement non dominé et l'optimisation par essais particuliers multi-objectifs. Leng et al. [15] ont, quant à eux, présenté un modèle bi-objectif innovant appliqué à la logistique de la chaîne du froid, intégrant à la fois des solutions respectueuses de l'environnement et des décisions de localisation et de routage. Les auteurs ont défini deux objectifs : le premier vise à optimiser simultanément les coûts logistiques et la consommation de carburant, prenant en compte les aspects économiques et environnementaux, tandis que le second porte sur l'amélioration de la satisfaction client en réduisant les pertes de marchandises. Le problème a été résolu à l'aide d'algorithmes évolutifs multi-objectifs.

1.2.2 Modèle macroscopique

Ces modèles estiment la consommation de carburant ou d'énergie en prenant comme valeurs de référence les valeurs moyennes des facteurs influents tels que la vitesse (et d'autres paramètres disponibles tantôt en préalable tantôt en prévision). Ils sont principalement employés dans la gestion de la chaîne logistique écologique lorsqu'il est question de plusieurs variantes du VRP. Notre étude faisant partie de cette section, nous accorderons une attention particulière à l'analyse des travaux antérieurs pertinents, dans le but de mettre en évidence la position de notre contribution dans ce contexte de recherche.

Tavares et al. [16] figurent parmi les précurseurs dans ce domaine, ayant proposé une optimisation des réseaux de collecte et de transport des déchets. Les auteurs ont appliqué un système d'information géographique (SIG) 3D pour minimiser la consommation de carburant en prenant en compte l'angle des routes ainsi que la charge des véhicules. Les résultats obtenus montrent pouvant atteindre 52 % de la consommation de carburant par rapport à la simple minimisation de la distance parcourue. Dans une étude similaire, Zsigraiová et al. [17] ont également utilisé un SIG 3D pour optimiser la collecte des déchets solides municipaux en minimisant la consommation de carburant. Cette approche a pris en compte l'inclinaison des routes ainsi que le poids des véhicules, permettant ainsi de

réduire la distance 3D la plus courte de 29 % et la consommation de carburant de 16 %. La comparaison entre le modèle 3D minimisant la consommation de carburant et celui minimisant la distance a montré une économie de carburant de 8 %, malgré une distance parcourue 1,8 % plus longue.

Figliozzi [18] a introduit une nouvelle variante du VRP appelée problème de tournées de véhicules avec émissions (EVRP), une extension du problème de tournées de véhicules dépendant du temps (TDVRP). L'objectif est de minimiser les émissions et la consommation de carburant en tenant compte des conditions de circulation : fluide, moyennement congestionnée et congestionnée. Les résultats montrent qu'une réduction d'environ 20 % des émissions est possible en optimisant les horaires de départ, ce qui corrobore les résultats du problème de tournées et d'ordonnancement des véhicules étudiés par Maden et al. [19], où l'objectif est de minimiser le temps de trajet total en situation de congestion. Les résultats obtenus indiquent une réduction de 7 % des émissions de CO_2 .

Une extension de l'EVRP avec fenêtres de temps (EVRPTW) a été proposée par Wygonik et Goodchild [20]. Le modèle, développé sous ArcGIS, permet d'analyser les compromis entre trois facteurs : le coût, la qualité du service et les émissions de CO_2 . L'étude révèle une relation stable entre le coût monétaire et les kilogrammes de CO_2 émis, indiquant qu'un kilogramme de CO_2 entraîne une augmentation du coût estimée à 3,5 dollars. En 2012, Omidvar et Tavakkoli-Moghaddam [21] ont introduit un modèle de transport et de tournées de véhicules pour les véhicules à carburants alternatifs (AFV), incluant des véhicules hybrides, électriques et à pile à hydrogène, dans le but de minimiser la consommation énergétique et les émissions de GES. Ce modèle a été testé sur les instances de benchmark de Solomon [22]. Les résultats indiquent que les émissions de CO_2 représentent environ 3 % des coûts totaux.

Un autre travail centré sur la minimisation des émissions dans le TDVRP a été proposé par Saberi et Verbas [23]. Le modèle d'approximation continue présenté facilite la planification stratégique des systèmes de distribution en configuration un-à-plusieurs et permet d'évaluer l'impact des coûts liés aux émissions. L'étude de Xiao et Konak [24] s'intéresse à la réduction de la propagation du CO_2 due à la congestion routière. Pour cela, ils ont proposé le problème de tournées et d'ordonnancement des véhicules dépendant du temps avec optimisation des émissions de CO_2 (TD-VRSP- CO_2), qui a été résolu grâce à une approche hybride combinant un algorithme génétique et une procédure exacte de programmation dynamique.

En 2020, Xiao et al. [25] ont développé une approche ε -précise pour résoudre le PRP dans un cadre d'optimisation continue. De plus, Poonthalir et al. [26] ont mené une étude sur les véhicules publicitaires mobiles (MAV) dans un contexte écologique, où l'objectif est de minimiser à la fois les coûts de tournée et les émissions de carbone. Pour résoudre ce problème, une optimisation hybride par essaims particuliers (HPSO) a été utilisée. Les résultats obtenus suggèrent des émissions allant jusqu'à 50 %.

Fan et al., [27], ont aussi proposé un modèle de programmation en nombres entiers pour un VRP multi-dépôts sur un réseau routier à temps variable, dans le but de minimiser le coût total. Les auteurs s'appuient sur un algorithme génétique hybride (GA), une recherche de voisinage variable (VNS) et sur le modèle MEET (Methodologies for Estimating Air Pollutant Emissions from Transport) pour estimer la consommation de carburant. De même, Gutierrez-Padilla et al., [28], ont décrit le PRPTW à vitesse discrète et proposé un modèle de programmation linéaire mixte qui intègre plusieurs facteurs tels que la pente de la route, les vitesses discrètes, la pollution et les coûts de transport dans le contexte de Colombie. Dutta et al., [29], ont proposé un modèle bi-objectif pour le GVRP résolu par (NSGA-II) prenant en compte la distance et les émissions.

Pour minimiser encore les émissions de CO_2 , plusieurs travaux récents s'intéressent à l'optimisation des vitesses variables des véhicules dans le cadre du GVRP. Yao et al. [30] ont développé un modèle d'optimisation de l'acheminement des véhicules écologiques prenant en compte l'impact de la variabilité de la vitesse des véhicules sur les émissions en intégrant les conditions routières réelles dans le calcul d'émissions. Ils l'ont appliqué à l'ensemble de données standard de Solomon et résolu avec un AG. Dans une autre contribution importante, Shi et Lin [31] ont proposé un modèle d'optimisation bi-objectif pour le GVRP multi dépôts à fenêtres temporelles, portant sur les coûts d'exploitation et les émissions. Ils l'ont résolu avec l'algorithme NSGA-II. Wang et al. [32] se sont davantage intéressés aux effets des restrictions de circulation sur la logistique de distribution, en proposant un nouveau modèle GVRP multi-objectif à fenêtres temporelles souples. Ce modèle vise à minimiser le temps de livraison et les émissions, et les auteurs ont appliqué un algorithme amélioré d'optimisation par colonies de fourmis pour le résoudre.

Liu et al. [33] ont présenté un algorithme de recherche adaptative des grands voisinages (ALNS) pour le GVRP avec fenêtres temporelles (GVRPTW). L'algorithme ALNS utilise la discrétisation temporelle et les contrôles de faisabilité pour déterminer les heures de départ, montrant une performance efficace dans la résolution des instances de petite taille et fournissant des solutions de haute qualité pour les instances plus grandes avec jusqu'à 1 000 clients. Zhang et al. [34] ont étudié l'impact du comportement du conducteur sur la consommation de carburant et les émissions de CO_2 dans le transport routier de marchandises. En utilisant un ensemble de données de plus de 4 000 enregistrements de conduite, l'étude a identifié des comportements clés, tels que les accélérations brutales et les virages, qui ont un impact significatif sur la consommation de carburant. Les auteurs ont introduit un modèle avancé de consommation de carburant qui intègre ces comportements du conducteur. Pour les véhicules électriques, Cataldo-Díaz et al. [35] ont étudié le VRPTW électrique (E-VRPTW), en tenant compte de l'état de charge de la batterie. Leur étude a comparé les méthodes de charge linéaire et non linéaire et a montré que la charge non linéaire réduisait la durée totale de l'itinéraire en minimisant les arrêts de charge inutiles.

Finalement, Lou et al. [36] ont abordé un problème de routage bas carbone intégrant vitesses dépendantes du temps, conditions routières et fenêtres temporelles, et ont proposé un algorithme hybride GA-VNS validé par une étude de cas en Chine. Ferreira et al. [37] ont présenté un modèle GVRP avec contraintes de chargement bidimensionnelles et livraisons fractionnées, résolu par une recherche à voisinage variable. Enfin, Gkyrtis [38] a analysé l'impact du design routier sur la consommation de carburant, montrant que les pentes accentuées peuvent la multiplier jusqu'à 2,5 fois.

1.3 Modèle hybride

Dans cette section, les auteurs ont combiné les véhicules à carburants alternatifs (AFV) et les véhicules à carburant conventionnel.

Tout d'abord, Mancini [39] a proposé un nouveau modèle de programmation linéaire en nombres entiers mixtes (MILP) pour le problème de tournées de véhicules hybrides (HVRP). Dans ce modèle, les véhicules peuvent utiliser à la fois la propulsion électrique et un moteur à carburant traditionnel, avec la possibilité de changer de mode de propulsion à tout moment durant le trajet. En mode électrique, la batterie a une capacité limitée, ce qui implique que les véhicules peuvent nécessiter une recharge dans une station de recharge. Pour résoudre ce problème, une métaheuristique basée sur la recherche à large voisinage (LNS) a été proposée.

Ensuite, Macrina et al. [40] ont développé un nouveau modèle du GVRP dans lequel la flotte inclut à la fois des véhicules conventionnels et électriques. Ces derniers nécessitent une recharge partielle de leur batterie dans les stations disponibles. Le modèle présenté repose sur une consommation énergétique multi-attributs, prenant en compte l'accélération, la décélération, la charge transportée et les pentes des routes. Pour résoudre ce problème, une métaheuristique intégrée dans une recherche à large voisinage a été suggérée.

Finalement, les travaux de Manavizadeh et al. [41] ont abouti à la mise en place d'un nouveau modèle mathématique pour le GVRP à partir d'un parc de véhicules bi-carburants comprenant des véhicules fonctionnant au gaz naturel et des véhicules à essence, dont l'objectif sera d'optimiser les coûts globaux de service, de transport et d'émission de carbone.

1.4 Autres modèles

Cette section présente trois exemples majeurs qui ne s'inscrivent pas dans les classifications précédentes.

L'article de Kara et al. [42] est considéré comme l'un des travaux les plus influents dans ce domaine. Parmi les premiers à intégrer des solutions écologiques aux problèmes de routage, les auteurs ont introduit le problème de tournées de véhicules avec minimisation

de l'énergie (EMVRP). Un modèle de programmation linéaire en nombres entiers mixtes est proposé pour l'EMVRP, qui constitue une extension du problème classique de tournées de véhicules avec capacité (CVRP). La principale différence réside dans la fonction objective, remplacée par une règle physique bien connue stipulant que sur une surface plane, le travail est égal à la force multipliée par la distance. L'objectif de l'EMVRP devient ainsi la minimisation d'une fonction de charge pondérée pour estimer la consommation de carburant. Le modèle proposé a été appliqué à des instances de référence du CVRP afin de comparer les solutions minimisant la distance et celles minimisant l'énergie. L'étude a montré une différence significative entre ces deux approches, révélant que l'optimisation de la distance totale réduit les coûts de parcours de 13% par rapport à l'optimisation énergétique.

Le deuxième travail est celui de Xiao et al. [43], qui ont travaillé sur le taux de consommation de carburant (FCR) ; une attention particulière a été portée au facteur de charge à l'aide d'une fonction dépendant de la charge intégrée au modèle classique du CVRP. Ces auteurs ont utilisé des données statistiques pour construire leurs modèles de régression et ont proposé un modèle d'optimisation mathématique permettant d'estimer la consommation de carburant dans le CVRP (FCVRP) qui a pour objectif de minimiser la consommation de carburant. Pour résoudre le FCVRP, ils ont eu recours à une approche par algorithme de recuit simulé. Les résultats ont montré que le modèle FCVRP permet de réduire, en moyenne, de 5% la consommation de carburant par rapport au modèle CVRP.

Finalement, les chercheurs Demir et al. [44] ont proposé le problème de conception de réseau de services intermodaux verts avec incertitude sur le temps de trajet (GISND-TTU) dont la résolution s'effectue dans une approche stochastique à visée multi-critères pour optimiser plusieurs objectifs (coût, temps de trajet, émission de GES).

1.5 Le crowdsourcing pour les problèmes de routage

Dans la littérature, il est rare de trouver des travaux de recherche qui utilisent le crowdsourcing dans les problèmes de routage dans le contexte que nous avons étudié. C'est la raison pour laquelle nous nous sommes limités à sélectionner des articles proches de notre travail.

D'après la littérature, la première étude à avoir utilisé la Distance Matrix API de Google est celle de Kiraly et Abonyi en 2010 [45]. Les auteurs y proposent une méthode basée sur les algorithmes génétiques a été proposée pour résoudre le multiple TSP (mTSP), en intégrant la Google Maps API afin de générer dynamiquement des matrices de distances réalistes et d'optimiser la distribution dans des systèmes logistiques à capacité limitée. Liu et Qu [46] présentent un modèle dynamique de congestion de type processus gaussien thématique comme cadre pour exploiter les données participatives et caractériser

simultanément, en temps réel, la dynamique et l'incertitude du trafic. Leurs formulations permettent un routage global non-myopique minimisant à la fois le temps de parcours en cherchant à l'équilibrer efficacement entre exploration et exploitation. Les résultats des études de validation, fondées sur des données de trafic et des réseaux sociaux réelles, démontrent des gains substantiels par rapport à la méthode classique et des résultats quasi-omniscients. Un cas intéressant est présenté par Satyananda [47], proposant une application Web qui exploite les services de Google Maps API pour résoudre un CVRP, en se servant de distances et d'itinéraires « réels » extraits par l'interface cartographique. Cette approche permet d'obtenir une meilleure visualisation interactive et graphique et les itinéraires proposés s'approchent plus des réalités routières et réelles que les résultats d'ordres classiques du CVRP, mais ils apparaissent plus longs en raison de l'asymétrie du trajet réel de transport. Cette approche permet d'atteindre de meilleures précisions permettant ainsi d'optimiser la prise de décision dans un contexte logistique réel. Dans son étude, Kumarage [48] a prouvé que la base de données de temps de trajet issue de l'API Google Distance Matrix consultée sur une plateforme cloud était fiable et économique. À partir de données crowdsourcées validées par des véhicules urbains munis de GPS, il a produit des estimations de flux mais surtout détecté des goulets d'étranglement, évalué des mesures de gestion de trafic urbain (comme celles concernant les voies réservées) et des politiques de gestion par des mesures de gestion de trafic urbain comme la traction des voies réservées.

1.6 Synthèse des Lacunes de Recherche

L'examen approfondi de la littérature sur le Problème de Routage des Véhicules perspectives écologiques met en évidence plusieurs limites méthodologiques. En premier lieu, un très grand nombre de travaux mobilise des modèles de données statiques, y compris des vitesses moyennes ne tenant pas compte des fluctuations du trafic réel. Cette approximation a tendance à diminuer la pertinence des résultats, en particulier lorsque l'on vise à optimiser la consommation de carburant et les émissions polluantes, dans un contexte de circulation variable.

D'autre part, la majorité des approches jugent la consommation d'énergie ou les émissions de GES à partir de la distance parcourue. Pourtant, cet indicateur ne tient pas toujours compte d'autres paramètres influents comme la charge transportée ou les conditions particulières des trajets. En effet, certains travaux engagent ces paramètres, mais de façon insuffisante ou simpliste.

Par ailleurs, il est fréquent que les modèles de consommation d'énergie utilisés soient vieux ou simplistes, car ils ne se basent que sur peu de paramètres, comme la vitesse ou la distance, en passant outre d'autres paramètres aussi importants comme le poids total, le pourcentage de pente des routes, l'état du revêtement, ou même les nombreux

arrêts occasionnés par la circulation en ville. Et pour un modèle, représenter fidèlement la réalité du transport routier est d'autant plus important qu'un modèle simpliste ou erroné ne sera pas pertinent pour optimiser ou inventer de nouveaux modes de transport.

Il ressort de cela une question majeure : comment modéliser de manière plus appropriée l'impact des changements de vitesses et de charges sur la consommation de carburant et des émissions, au lieu de procéder à une simple minimisation des distances ? L'utilisation de données dynamiques et de modèles mieux ajustés aux conditions réelles de transport pourrait faire cette différence et permettre d'aboutir à des choix logistiques verts plus efficaces.

Chapitre 2

Description du problème

2.1 Introduction

Cette section expose la problématique ainsi que ses principales caractéristiques. Tout d’abord, nous introduisons l’étude de cas et définissons formellement le problème. Ensuite, nous proposons un modèle de programmation mixte en nombres entiers et non linéaire pour sa résolution. De plus, la méthode de calcul de la consommation énergétique est intégrée dans la fonction objective.

Ce chapitre est ainsi structuré autour de trois volets complémentaires. Il débute par une étude de cas portant sur la distribution pharmaceutique en Algérie, mettant en évidence les contraintes logistiques et environnementales. Par la suite, une modélisation mathématique du problème est formulée, en tant que problème de tournée de véhicules avec contraintes de volume et de temps (VPRPTW), puis enrichie par une approche verte désignée SG-VCVRP-TW. Enfin, un modèle détaillé de consommation d’énergie est présenté, intégrant les variables physiques et opérationnelles influençant l’efficacité énergétique des tournées.

2.2 Étude de Cas : Distribution Pharmaceutique en Algérie

La distribution des médicaments, maillon essentiel du système de santé, permet de garantir la disponibilité des médicaments pour les patients par l’intermédiaire d’un réseau logistique complexe. Le secteur de la distribution pharmaceutique en Algérie doit faire face à des enjeux majeurs en lien avec la régulation, avec les difficultés logistiques et avec les variations de la demande.

L’approvisionnement en médicaments s’appuie sur un réseau de grossiste-répartiteurs et de pharmacies, avec une forte dépendance aux importations bien que la production nationale se soit développée ces dernières années. Pour permettre une gestion efficace de la chaîne d’approvisionnement s’avèrent nécessaires de disposer d’outils avancés, notamment les algorithmes d’optimisation permettant de réduire les coûts de transport et d’améliorer la disponibilité des produits vierges.

Dans cette étude, nous nous attacherons à un cas pratique de distribution de médicaments en Algérie traitant d’une entreprise possédant une flotte de véhicules homogènes à capacités limitées (poids charge et volume). L’objectif étant de livrer les produits aux clients situés dans différentes villes, en mettant en place des stratégies de routage respectueuses de l’environnement, visant à réduire la consommation d’énergie. Ainsi, l’entreprise doit garantir, pour ses clients, la satisfaction des demandes tout en respectant les contraintes logistiques quant aux ressources disponibles. Nous exploiterons alors des méthodes d’optimisation, en étudiant ici notamment une recherche tabou probabiliste

pour fournir des réponses efficaces aux problèmes du secteur.

La stratégie de distribution actuelle repose sur l'agrégation des commandes des clients chaque nuit pour une livraison le lendemain. Pour cet exercice, chaque véhicule est affecté à un ensemble de clients d'une même ville et doit réaliser son itinéraire quotidiennement, quel que soit le nombre de clients à livrer qui, dans certains cas, peut être réduit à un seul. Cette gestion souffre de plusieurs limites, telles que l'affectation énergétiquement non optimale des ressources utilisées pour réaliser les livraisons, le risque d'insatisfaction clients dans les périodes de préparation pour des livraisons plus tardives dans la journée, ou la difficulté de gérer des livraisons en cas de fort volume des commandes, entraînant un retard (non souhaitable) pour une partie des clients.

La direction de l'établissement a également procédé à une classification de ses clients en tenant compte de leur fidélité, un nombre élevé de transactions permettant une meilleure classification qui donnera un service, une livraison anticipée par rapport aux autres clients. Dans un premier temps, des fenêtres horaires ont été définies par catégorie de chaque client. Ces plages horaires sont déterminées par l'entreprise au regard du critère de fidélisation. En général, les clients sont présents du matin au soir à l'exception de situations particulières, précisées dans chaque catégorie. Dans cette perspective, l'attribution de ces créneaux horaires se fait en fonction de la fidélité du client, avec éventuellement des adaptations selon les disponibilités du client.

D'un autre côté, l'évaluation de la consommation énergétique repose sur un modèle de consommation de carburant dont les facteurs sont notamment étudiés par [49], [50] et [51]. Demir et al. [6] regroupent les différents facteurs en cinq catégories principales : les caractéristiques du véhicule, les conditions de l'environnement, l'état du trafic, le comportement du conducteur et les opérations logistiques. En effet, l'estimation de la consommation énergétique privilégie le plus souvent l'analyse de l'influence du véhicule, du trafic et des conditions de l'environnement, où l'accent est mis sur la distance parcourue par le véhicule, la charge transportée et éventuellement la vitesse ainsi que le mode conducteur. L'hypothèse que ces valeurs soient constantes pendant le trajet engendre des estimations erronées et éloignées de toute réalité opérationnelle.

2.3 Définition et formulation du problème VPRPTW

Nous considérons la formulation du VPRPTW dans laquelle nous disposons d'un ensemble K de véhicules homogènes ayant une capacité de charge Q et une capacité de volume V . Chaque client i est associé à une plage de temps $[e_i, l_i]$, représentant l'instant le plus tôt et le plus tard auquel le déchargement peut commencer. Toutefois, un véhicule est autorisé à commencer le service du client i dès son arrivée, même si celle-ci précède le début de la fenêtre temporelle e_i , en raison de la synchronisation des fenêtres temporelles e_i pour tous les clients dans la matrice de temps, tout en maintenant les contraintes temporelles

sous la forme de deux contraintes 2.13 et 2.14, ce qui rend ce problème un problème de fenêtres temporelles souples.

Le modèle VPRPTW est défini sur un graphe complet $G = (N, A)$, où l'ensemble des nœuds $N = \{0, 1, 2, \dots, n\}$ représente les clients et le dépôt, et l'ensemble des arcs A représente les routes reliant les clients. Nous considérons un ensemble de $n = |N|$ clients numérotés de 1 à n , et le nœud 0 représente le dépôt. De plus, deux sommets auxiliaires numérotés 0 et $n + 1$ représentent respectivement le dépôt pour le début et la fin des tournées.

Nous considérons également la consommation énergétique E , où E_{ij} représente la consommation d'énergie pour parcourir la distance entre le client i et le client j , calculée selon l'équation 2.19. De plus, la matrice des temps de trajet T indique que le temps nécessaire pour aller du client i au client j est t_{ij} , et la distance correspondante est notée d_{ij} . L'ensemble $N_0 = N \setminus \{0\}$ représente l'ensemble des clients sans inclure le dépôt. Les paramètres e_0 et l_0 déterminent respectivement l'heure la plus tôt à laquelle un véhicule peut quitter le dépôt et l'heure limite pour y revenir. Les variables de décision sont définies comme suit.

$$x_{ijk} = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ est utilisé par le véhicule } k \\ 0 & \text{sinon} \end{cases} \quad (2.1)$$

$$w_{ik} = \begin{cases} \text{heure de début de service} & \text{si le client } i \text{ apparaît dans l'itinéraire du véhicule } k \\ 0 & \text{sinon} \end{cases} \quad (2.2)$$

La formulation MINLP pour le VPRPTW est donnée comme suit.

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} E_{ij} x_{ijk} \quad (2.3)$$

La fonction objective 2.3 est soumise aux contraintes suivantes :

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in N_0 \quad (2.4)$$

$$\sum_{k \in K} \sum_{i \in N} x_{ijk} = 1 \quad \forall j \in N_0 \quad (2.5)$$

$$\sum_{i \in N_0} x_{i0k} \leq 1 \quad \forall k \in K \quad (2.6)$$

$$\sum_{j \in N_0} x_{0jk} \leq 1 \quad \forall k \in K \quad (2.7)$$

$$\sum_{i \in N} x_{ipk} = \sum_{j \in N} x_{pj k} \quad \forall p \in N_0, k \in K \quad (2.8)$$

$$\sum_{i \in N} \sum_{k \in K} q_i x_{ijk} \leq Q_k \quad \forall k \in K \quad (2.9)$$

$$\sum_{i \in N} \sum_{k \in K} v_i x_{ijk} \leq V_k \quad \forall k \in K \quad (2.10)$$

$$x_{ijk}(w_{ik} + s_i + t_{ij} - w_{jk}) \leq 0 \quad \forall k \in K, (i, j) \in N_0 \quad (2.11)$$

$$e_i = e_0 \quad \forall i \in N_0 \quad (2.12)$$

$$e_i \sum_{j \in N} x_{ijk} \leq w_{ik} \quad \forall k \in K, i \in N_0 \quad (2.13)$$

$$l_i \sum_{j \in N} x_{ijk} \geq w_{ik} \quad \forall k \in K, i \in N_0 \quad (2.14)$$

$$w_{0k} \geq e_0 \quad \forall k \in K \quad (2.15)$$

$$w_{n+1,k} \leq l_0 \quad \forall k \in K \quad (2.16)$$

$$u_{ik} - u_{jk} + (n - 1)x_{ijk} \leq n - 2 \quad \forall k \in K, (i, j) \in N_0 \quad (2.17)$$

$$x_{ijk} = \{0, 1\} \quad \forall k \in K, (i, j) \in N. \quad (2.18)$$

La fonction objective (2.3) vise à minimiser la consommation totale d'énergie sur l'ensemble des itinéraires. Les contraintes (2.4) et (2.5) garantissent que chaque client est desservi par un seul véhicule. Les contraintes (2.6) et (2.7) imposent qu'un véhicule quitte et revienne au dépôt une seule fois. Les contraintes (2.8) assurent la continuité du trajet en exigeant qu'un véhicule poursuive son itinéraire d'un client à l'autre. Les contraintes (2.9) et (2.10) évitent toute surcharge en veillant à ce que la demande totale ne dépasse pas la capacité du véhicule, tant en poids (Q) qu'en volume (V). La contrainte (2.11) établit le lien entre l'heure de départ d'un véhicule chez un client et celle de son déplacement vers le client suivant. Les contraintes (2.12), (2.13), (2.14), (2.15) et (2.16) garantissent le respect des fenêtres temporelles définies. La contrainte (2.17) empêche la formation de sous-cycles dans la solution. Enfin, la contrainte (2.18) impose que la variable de décision x_{ijk} soit binaire.

2.4 Définition et formulation du problème SG-VCVRP-TW

Dans cette étude, nous utilisons la dénomination (Symmetric Green Volume Capacitated Vehicle Routing Problem with Time Windows) SG-VCVRP-TW ou (Problème Symétrique de Tournée de Véhicules à Capacité Volumique Verte avec Fenêtres de Temps) dans notre travail [52] pour désigner plus précisément la version du problème étudiée dans notre travail [1], précédemment formulée comme VPRPTW.

Cette appellation a pour but d'éviter les ambiguïtés qui viennent avec le terme générique PRP utilisé dans la littérature pour désigner les différentes variantes de problèmes de tournées intégrant des aspects environnementaux. Le terme Green fait référence à l'approche environnementale durable intégrée dans le modèle, qui met l'accent sur l'efficacité énergétique et la diminution de l'empreinte écologique grâce à une gestion optimisée de la capacité volumétrique des véhicules ainsi que des circuits de transport. En ce qui concerne l'adjectif Symmetric, il ne fait pas référence à la symétrie des distances entre des clients mais à l'homogénéité du parc de véhicules qui est constitué de moyens identiques en termes de capacités techniques (volume, charge, consommation, etc.).

On constate donc que le SG-VCVRP-TW présente exactement le même modèle que le VPRPTW de la section précédente, d'un point de vue non seulement de la structure mathématique mais des formulations des contraintes et de la fonction objectif. Dans ce contexte, la nouvelle appellation du problème, introduite dans [52], permet une meilleure prise en compte des spécificités du problème étudié dans le cadre d'une logistique verte et d'une distribution urbaine durable.

Le modèle mathématique associé au SG-VCVRP-TW est donc strictement identique à celui du VPRPTW présenté dans la section 2.3. Pour consulter les détails formels, il est recommandé de se référer à la section mentionnée.

2.5 Modèle de consommation d'énergie

L'équation (2.19) représente la fonction objective visant à minimiser l'énergie consommée, permettant ainsi d'estimer les émissions de CO_2 (ou la consommation de carburant) associées. Nous utilisons un modèle instantané de consommation d'énergie développé par Chaim et Shmerling [53], qui constitue une extension du modèle de Guzzella et Sciarretta [54] en intégrant davantage de paramètres. Cette démarche se caractérise par l'inclusion de plusieurs éléments influençant la consommation d'énergie, en opposition aux autres modèles qui s'appuient sur une cartographie du moteur en faisant l'hypothèse d'une consommation de carburant constante selon les conditions de fonctionnement, ce qui est peu réaliste [53], alors que le modèle que nous proposons intègre cette insuffisance en prenant en compte la consommation spécifique instantanée de carburant et les besoins

énergétiques en temps réel. Ceci lui permet de tenir compte des variations du mode de déplacement.

Les équations 2.19 à 2.31 sont issues des travaux de Ben Chaim et Shmerling [53] ainsi que de Ben-Chaim et al. [55]. Nous avons utilisé ces modèles pour fournir une vision cohérente et intégrée du processus de modélisation.

L'énergie consommée entre le client i et le client j est donnée par :

$$E_{s_{ij}} = \frac{[(E_{1_{ij}} + E_{2_{ij}}) * d_{ij}]}{100} \quad (2.19)$$

Le modèle défini dans l'équation 2.19 repose sur l'hypothèse que le moteur fonctionne selon deux modes principaux. Le premier correspond à un déplacement à vitesse moyenne, où $E_{1_{ij}}$ représente l'énergie nécessaire pour compenser les forces de résistance. Le second mode prend en compte les accélérations ponctuelles, où E_2 correspond à l'énergie cinétique requise, exprimée en joules (J). Les expressions de $E_{1_{ij}}$ et $E_{2_{ij}}$ sont définies respectivement dans les équations 2.20 et 2.21.

$$E_{1_{ij}} = \left(\frac{1}{\eta_T \eta_{P,n_{ij}}} \right) \left(m_{a_{ij}} \cdot g \cdot c_{r_{ij}} + \frac{\rho}{2} \cdot C_D \cdot A_{f_{ij}} \cdot V_{a_{ij}}^2 \right) \cdot S \quad (2.20)$$

$$E_{2_{ij}} = \frac{q m_{a_{ij}}}{\eta_T \eta_e} \sum_l^k \frac{\alpha_l \gamma_{ml} S_l}{\mu_{p_{l_{ij}}} \mu_{n_{l_{ij}}}} \quad (2.21)$$

Ainsi, l'expression complète de $E_{s_{ij}}$ devient :

$$E_{s_{ij}} = \left[\left(\frac{1}{\eta_T \eta_{P,n_{ij}}} \left(m_{a_{ij}} \cdot g \cdot c_{r_{ij}} + \frac{\rho}{2} \cdot C_D \cdot A_{f_{ij}} \cdot V_{a_{ij}}^2 \right) \cdot S + \frac{q m_{a_{ij}}}{\eta_T \eta_e} \sum_l^k \frac{\alpha_l \gamma_{ml} S_l}{\mu_{p_{l_{ij}}} \mu_{n_{l_{ij}}}} \right) * d_{ij} \right] / 100 \quad (2.22)$$

L'ensemble des paramètres utilisés est détaillé dans le tableau 2.1.

Table 2.1: Paramètres du modèle de consommation d'énergie du véhicule.

Paramètre	Description
η_T	Rendement de la transmission.
$\eta_{P,n}$	Rendement du moteur.
m_a	Masse du véhicule.
g	Accélération de la gravité.
c_r	Coefficient de résistance au roulement.
ρ	Densité de l'air.
C_D	Coefficient de résistance aérodynamique du véhicule.
A_f	Surface caractéristique du véhicule.
V_a	Vitesse moyenne du véhicule.
S	Kilométrage du véhicule.
q	Nombre d'accélération dans chaque intervalle d'accélération.
η_e	Rendement maximal du moteur.
α_i	Accélération du véhicule.
γ_{mi}	Facteur de masse du véhicule.
S_i	Distance d'accélération du véhicule.
k	Nombre d'intervalles d'accélération.
μ_{p_i}	Coefficient d'influence du degré d'utilisation de la puissance sur le rendement maximal du moteur.
μ_{n_i}	Coefficient d'influence du régime moteur sur le rendement maximal du moteur.
d	Distance parcourue.

Le rendement du moteur, noté $\eta_{P,n}$, dépend de la puissance utilisée et du régime moteur. Il est défini par :

$$\eta_{P,n} = \eta_e \mu_p \mu_n \quad (2.23)$$

avec les expressions suivantes :

$$\eta_e = 1 / (0.0119531 * g_e) \quad (2.24)$$

$$\mu_p = 0.5968 - 0.1666(P_i/P_e) + 2.4968(P_i/P_e)^2 - 2.1128(P_i/P_e)^3 \quad (2.25)$$

$$\mu_n = 0.7107 + 0.9963(n_i/n_p) - 1.0582(n_i/n_p)^2 + 0.3124(n_i/n_p)^3 \quad (2.26)$$

Où :

$$P_i = (m_a g c_r + 0.5 c_r A_f V_a^2 + m_a a \gamma_m) V_a \quad (2.27)$$

$$P_e = 10^3 P_{max} \left[(n/n_p) + 0.5 (n/n_p)^2 - 0.5 (n/n_p)^3 \right] \quad (2.28)$$

$$n = \frac{9.55 V_a \delta_{ax} \delta_n}{r_d} \quad (2.29)$$

Les paramètres associés sont définis dans le tableau 2.2.

Table 2.2: Paramètres complémentaires du modèle de consommation d'énergie du véhicule.

Paramètre	Description
g_e	Consommation spécifique de carburant.
P_i	Puissance moteur requise pour le mode de mouvement donné.
P_e	Puissance moteur selon les caractéristiques de performance.
n_i	Régime moteur à la vitesse moyenne du véhicule.
n_p	Régime moteur à la puissance maximale du moteur.
P_{max}	Puissance maximale du moteur.
δ_{ax}	Rapport de transmission final.
δ_n	Rapport de transmission dans la boîte de vitesses.
r_d	Rayon de roulement du pneu.

Enfin, les expressions pour c_r et A_f sont :

$$c_r = 0.0136 + 0.40 * 10^{-7} V_a^2 \quad (2.30)$$

$$A_f = 1.6 + 0.00056(m_a - 765). \quad (2.31)$$

Pour simplifier le modèle mathématique et le rendre plus clair, nous incluons la séquence logique. La formule générale de calcul de l'énergie $E_{s_{ij}}$ est présentée dans 2.19 et s'exprime en fonction de $E_{1_{ij}}$, $E_{2_{ij}}$ et de la distance d_{ij} .

Les équations $E_{1_{ij}}$ et $E_{2_{ij}}$ ont ensuite été intégrées dans 2.20 et 2.21. L'équation générale $E_{s_{ij}}$ est obtenue après égalisation des équations $E_{1_{ij}}$ et $E_{2_{ij}}$ dans 2.22. Tous les facteurs et variables mentionnés sont expliqués dans le tableau 2.1.

Dans l'équation 2.23, le coefficient $\eta_{P,n}$, composé de η_e , μ_p et μ_n , a été discuté, et leur méthode de calcul est présentée dans les équations 2.24, 2.25 et 2.26. Pour calculer μ_p ,

nous avons besoin des équations de P_i et P_e , qui sont présentées dans 2.27 et 2.28. De même, pour calculer μ_n , nous avons besoin de l'équation permettant de calculer n , qui est présentée dans 2.29. Tous ces facteurs et variables sont expliqués dans le tableau 2.2.

Enfin, les formules mathématiques permettant de calculer tous les facteurs à partir de c_r et A_f sont fournies dans 2.30 et 2.31.

Chapitre 3

Méthodologie de la solution

3.1 Introduction

Dans la présente section, nous avons l’occasion d’exposer l’algorithme proposé. L’algorithme TS est une métaheuristique basée sur les méthodes de recherche locale (LS) créée par [56] en 1989. Bien qu’elle soit relativement ancienne, elle a prouvé son efficacité et est encore largement utilisée aujourd’hui dans la littérature pour le VRP et ses variantes, car elle présente plusieurs avantages, y compris au regard de sa recherche itérative de l’espace de solutions, de sa méthode de mémoire pour ne pas visiter deux fois une même solution, et de son schéma d’évasion des optima locaux pour approfondir la recherche du minimum global. Le problème étudié étant de complexité NP-difficile, seules les instances de petite taille peuvent être résolues de manière optimale. Étant donné la nouveauté de notre problématique, nous avons développé un algorithme spécifique basé sur la TS afin d’en obtenir une solution efficace.

Parmi les contributions récentes utilisant l’algorithme de TS, celle de Gmira et al. [57] en 2021, a été une source d’inspiration majeure pour le développement de l’algorithme proposé dans notre étude. Donc ils ont proposé un algorithme TS pour résoudre un problème d’acheminement de véhicules avec des temps de parcours dépendant du temps, les résultats de calcul montrent que cette stratégie fournit des solutions quasi-optimales. Une autre étude récente en 2024 a utilisé un TS hybride appelé Simulated Annealing TS pour résoudre un problème d’acheminement de véhicules dépendant du temps avec des fenêtres de temps souples présenté par Liao et Shao [58]. Tlili et al. [59] ont abordé le problème du routage des ambulances avec deux algorithmes dérivés du TS : le TS hybride (HTS) et l’hyperheuristique basée sur le TS (TSHH).

3.2 Algorithme Tabou Probabiliste

3.2.1 Principe général de la recherche taboue

La métaheuristique de recherche taboue (Tabu Search, TS) se classe parmi les méthodes puissantes pour traiter des problèmes d’optimisation combinatoire dont, bien sûr, les problèmes de routage des véhicules (VRP) et leurs variantes. Elle reste d’actualité grâce à sa capacité à examiner censément des espaces de solutions vastes et à prévenir les pièges dans des optima locaux. Contrairement aux méthodes heuristiques plus classiques, la recherche taboue a intégré une mémoire adaptative qui enregistre les solutions préalablement examinées pour empêcher leur réexamen pendant un nombre donné d’itérations, et donc ouvrir la voie à une exploration plus diversifiée du paysage de recherche.

Dans les problèmes de routage, au fort caractère combinatoire, qui est à la fois accablé par la complexité des contraintes opérationnelles (fenêtres temporelles, capacités de charge, etc.), les atouts de la recherche taboue sont multiples. Elle permet à la fois une

plus grande souplesse dans la gestion de la structure de voisinage dont on peut jouer pour explorer les nombreuses solutions que peuvent offrir les différentes configurations de routes et de plans de livraison ; sa mise en place de comportements d'interdiction temporaire (tabu tenure) permet d'éliminer les cycles de recherche inopérants et force à diversifier les solutions à adresser ; et, pour finir, elle peut être couplée (hybridation) à des méthodes probabilistes ou à d'autres métaheuristiques (de type recherche locale adaptative ou algorithmes génétiques) pour utilement renforcer les qualités des solutions visées.

Grâce à ces caractéristiques, la recherche taboue reste une approche incontournable pour les entreprises et les chercheurs cherchant à optimiser les coûts et l'efficacité des tournées de véhicules dans des contextes industriels, logistiques et urbains en perpétuelle évolution.

3.2.2 Initialisation de l'algorithme

L'algorithme 3.1 illustre les différentes étapes du processus proposé. Tout d'abord, une phase d'initialisation est réalisée, durant laquelle des matrices de données sur les distances, les temps et les vitesses sont générées à l'aide de l'API Distance Matrix. Ensuite, plusieurs variables sont définies : *maxIteration*, *iteration*, *tabuTenure*, *bestSolution* et *currentSolution*. Ces variables correspondent respectivement au nombre maximal d'itérations de la boucle principale (critère d'arrêt), au compteur d'itérations en cours, à la durée pendant laquelle un arc reste interdit dans la liste taboue, à la meilleure solution rencontrée au cours des itérations et à la solution courante du problème.

3.2.3 Itérations et génération des mouvements

L'algorithme s'arrête lorsque le nombre maximal d'itérations est atteint. Pendant chaque itération, la variable *bestNeighborCost* est initialisée pour stocker la meilleure amélioration de coût parmi toutes les solutions voisines envisageables. Ensuite, *bestMove* est définie pour enregistrer le meilleur déplacement possible parmi toutes les alternatives. Une variable *p* est introduite afin de gérer la probabilité de sélection entre deux stratégies de déplacement : l'échange i-cross et l'échange 1-0. La fonction *allPossibleMoves* permet d'identifier les déplacements admissibles en tenant compte des contraintes de poids, volume, fenêtres temporelles et en s'assurant que le mouvement n'est pas interdit par la liste taboue.

Une fois une stratégie sélectionnée, la fonction *getCost* évalue l'impact de chaque mouvement sur le coût total et enregistre la valeur obtenue dans *neighborCost*. Si cette valeur est inférieure à *bestNeighborCost*, elle est mise à jour et le meilleur déplacement *bestMove* est ajusté en conséquence. Une fois toutes les options évaluées, la liste taboue est mise à jour, puis le déplacement sélectionné est appliqué à la *currentSolution*. Si cette dernière est meilleure que la *bestSolution*, alors cette dernière est remplacée par la nouvelle

solution obtenue.

3.2.4 Pseudocode de l'algorithme

Algorithm 3.1: Recherche Tabou Probabiliste pour le VPRPTW

```
/* Initialisation */
1 Créer les matrices de données (distance, temps et vitesse) en utilisant l'API
  Distance Matrix; Générer une solution initiale faisable  $x$  en utilisant l'insertion
  gloutonne;  $currentSolution \leftarrow x$ ;  $bestSolution \leftarrow currentSolution$ ;
2 while  $iteration < maxIteration$  do
3    $bestNeighborCost \leftarrow \infty$ ;  $bestMove \leftarrow \emptyset$ ; Générer un nombre aléatoire  $p$ 
   /*  $p \in [0, 1]$  */
4   if  $p > 0.8$  then
5     | Utiliser la stratégie d'échange 0-1;
6   else
7     | Utiliser la stratégie d'échange i-cross;
8   end
9   for  $move \in allPossibleMoves$  do
10     $neighborCost \leftarrow getCost(currentSolution, move)$ ;
11    if  $neighborCost < bestNeighborCost$  then
12      |  $bestNeighborCost \leftarrow neighborCost$ ;  $bestMove \leftarrow move$ ;
13      | if  $bestNeighborCost = \infty$  then
14        | | break;
15      | end
16    end
17  end
18  Mettre à jour la liste tabou; Générer un nombre aléatoire  $d$  /*  $d \in [0, 5]$  */
19   $TabuMatrix \leftarrow tabuTenure + d$  /* pour les nouveaux arcs dans  $bestMove$  */
20   $TabuMatrix \leftarrow TabuMatrix - 1$  /* pour chaque ancien arc dans  $bestMove$  */
21   $applyMove()$ ;  $iteration \leftarrow iteration + 1$ ;
    $currentSolution \leftarrow bestNeighborCost$ ;
22  if  $currentSolution \leq bestSolution$  then
23    |  $bestSolution \leftarrow currentSolution$ ;
24  end
25 end
26 return  $bestSolution$ ;
```

3.2.5 Analyse du fonctionnement étape par étape

L'algorithme démarre ainsi par la génération d'une solution initiale via une heuristique d'insertion gloutonne, qui permettra de partir dans l'optimisation. Cette solution est souvent générée après construction des matrices de distances et de temps, et peut servir de point de départ d'exploration de l'espace des solutions, d'une manière sûre.

L'algorithme s'exécute jusqu'à ce qu'un critère d'arrêt soit atteint, la meilleure solution rencontrée est strictement mise à jour à chaque itération. Cependant, la solution courante pour la mise à jour de la meilleure solution est choisie parmi les meilleures solutions voisines, sans passer obligatoirement par sa comparaison avec la solution précédente, ce qui autorise l'exploration libre d'autres configurations, sans être limité par la comparaison avec la qualité des anciennes solutions.

Pour générer les solutions voisines, deux stratégies d'exploration ont été retenues : l'échange iCROSS et l'échange 1-0. Ces deux stratégies permettent d'assurer une bonne diversité de la recherche tout en optimisant dans la durée la solution.

3.2.6 Composante probabiliste de l'algorithme

L'une des originalités de l'algorithme proposé réside dans l'intégration d'une composante probabiliste visant à renforcer la capacité exploratoire de la recherche taboue. Contrairement aux versions classiques de la recherche taboue, qui s'appuient uniquement sur des stratégies déterministes de voisinage, cette version introduit un mécanisme de sélection aléatoire permettant de diversifier les mouvements explorés au cours de l'optimisation.

Concrètement, à chaque itération de l'algorithme, un nombre aléatoire $p \in [0, 1]$ est généré afin de choisir de manière stochastique la stratégie de génération des solutions voisines. Lorsque $p > 0.8$, la structure de voisinage utilisée est l'échange 0-1, qui consiste à déplacer un client d'une tournée vers une autre. Sinon, l'algorithme opte pour l'échange iCROSS, une stratégie plus complexe impliquant l'échange croisé de segments entiers entre deux tournées. Ce choix probabiliste permet à l'algorithme d'éviter une exploration systématique et donc potentiellement myope du voisinage, en favorisant au contraire une alternance flexible et imprévisible entre différentes topologies de solution.

Cette approche s'inspire des principes des métaheuristiques stochastiques telles que les algorithmes évolutionnaires ou le recuit simulé, dans lesquelles la probabilité joue un rôle crucial dans la recherche d'un équilibre entre exploitation (amélioration locale) et exploration (diversification globale). Dans le cadre du VPRPTW, ce mécanisme contribue à réduire les risques de stagnation dans des optima locaux, en augmentant la robustesse de l'algorithme face à la complexité du paysage de solutions.

Ainsi, l'introduction de la composante probabiliste ne se limite pas à un simple choix technique, elle représente une stratégie méthodologique visant à améliorer l'efficacité globale de l'algorithme en enrichissant la dynamique de recherche et en accroissant la variété

des solutions générées tout au long du processus d'optimisation.

3.2.7 Renforcement de l'exploration du voisinage par boucle imbriquée

Dans le cadre de l'algorithme de *Recherche Tabou Probabiliste*, l'amélioration de l'efficacité exploratoire constitue un enjeu central. Afin d'approfondir davantage la recherche dans l'espace des solutions voisines, une extension pertinente de l'algorithme consiste à introduire une boucle **for** imbriquée au sein de chaque itération principale. Cette structure additionnelle permet à chaque cycle de l'algorithme de générer et d'évaluer plusieurs mouvements successifs, élargissant ainsi le champ d'exploration autour de la solution courante avant de sélectionner le déplacement le plus prometteur.

Contrairement à la version de base présentée dans l'Algorithme 3.1, cette nouvelle variante met l'accent sur une exploration locale plus intense à travers un balayage élargi du voisinage à chaque itération. Cela permet à l'algorithme de mieux détecter les améliorations potentielles et de limiter les effets de stagnation dans des optima locaux.

La structure proposée est décrite dans l'Algorithme 3.2, dans lequel la boucle interne permet d'examiner un nombre fixé de voisins (**nExplorations**) à chaque itération, avant

d'appliquer le meilleur mouvement admissible identifié.

Algorithm 3.2: Recherche Tabou Probabiliste avec exploration renforcée

```
/* Initialisation */
1 Créer les matrices de données (distance, temps et vitesse) en utilisant l'API
  Distance Matrix;
2 Générer une solution initiale faisable  $x$  en utilisant l'insertion gloutonne;
3  $currentSolution \leftarrow x$ ;
4  $bestSolution \leftarrow currentSolution$ ;
5 while  $iteration < maxIteration$  do
6    $bestNeighborCost \leftarrow \infty$ ;  $bestMove \leftarrow \emptyset$ ;
7   Générer un nombre aléatoire  $p \in [0, 1]$ ;
8   if  $p > 0.8$  then
9     Utiliser la stratégie d'échange 0-1;
10  else
11    Utiliser la stratégie d'échange iCROSS;
12  end
  /* Boucle imbriquée pour une meilleure exploration du voisinage */
13  for  $i = 1$  to  $nExplorations$  do
14    for  $move \in allPossibleMoves$  do
15       $neighborCost \leftarrow getCost(currentSolution, move)$ ;
16      if  $neighborCost < bestNeighborCost$  et  $move$  non tabou then
17         $bestNeighborCost \leftarrow neighborCost$ ;  $bestMove \leftarrow move$ ;
18      end
19    end
20  end
21  Mettre à jour la liste taboue;
22  Générer un nombre aléatoire  $d \in [0, 5]$ ;
23   $TabuMatrix \leftarrow tabuTenure + d$  /* pour les nouveaux arcs de  $bestMove$  */
24   $TabuMatrix \leftarrow TabuMatrix - 1$  /* décrémenter tous les autres éléments */
25   $applyMove(bestMove)$ ;
26   $currentSolution \leftarrow mise\_jour\_de(currentSolution, bestMove)$ ;
27  if  $getCost(currentSolution) < getCost(bestSolution)$  then
28     $bestSolution \leftarrow currentSolution$ ;
29  end
30   $iteration \leftarrow iteration + 1$ ;
31 end
32 return  $bestSolution$ ;
```

Ce renforcement local par boucle imbriquée permet de balayer plus largement l'espace des configurations voisines, tout en conservant le cadre probabiliste de sélection des mou-

vements. L'algorithme gagne ainsi en robustesse face à la complexité du problème de routage avec fenêtres temporelles, en explorant plus systématiquement les différentes options à chaque étape. Cette approche hybride, combinant intensification locale et diversification probabiliste, constitue une piste prometteuse pour des versions futures plus performantes.

3.3 Génération des matrices de données

Les matrices apparaissent comme une structure de base fondamentale dans de très nombreux algorithmes, non seulement pour organiser, mais pour stocker les données et les traiter, pour les récupérer. Dans les cas des problèmes d'optimisation (comme les problèmes de routage et de planification), les matrices encodent la relation entre chaque élément, par exemple les distances situées entre les points, le temps de déplacement ou les coûts de la transition. Cette performance et cette efficacité dépendent en grande partie de la qualité des données contenues dans ces matrices, le succès des algorithmes repose sur l'effectivité des données qu'elles fournissent.

La mise en place des matrices repose généralement sur des données fiables, mais peut être optimisée lors de sa construction, en fonction de l'exigence posée par le problème étudié. En effet, elles permettent d'organiser l'information dans un cadre pouvant être utilisé par l'algorithme et favorisant la rapidité du processus en permettant d'accélérer le processus de recherche d'une solution optimale.

Ainsi, pour le cas du problème de tournées de véhicules avec fenêtres temporelles (VRP-TW), il est important de connaître les distances et du temps de trajet entre chaque paire de clients. Cependant, l'itinéraire retour peut différer de l'itinéraire aller, entraînant ainsi des matrices asymétriques pour la distance, le temps et la vitesse. En pratique, mesurer ces distances et ces temps de manière exhaustive, en particulier pour un grand nombre de clients, est une tâche fastidieuse et coûteuse, tant en termes de ressources que de temps.

L'API Distance Matrix de Google Maps est un puissant service qui fournit des informations exactes en matière de distances et de temps de trajet entre plusieurs points. Elle permet aux applications et aux algorithmes d'optimisation, et plus précisément à ceux concernant notamment les problèmes de logistique et de routage, d'exploiter des données en temps réel, le tout en tenant compte de la circulation, de la nature du transport (voiture, vélo, marche, etc.), et de l'itinéraire proposé, ce qui permet une estimation fiable du déplacement entre le point d'origine et le point d'arrivée, favorisant des gains significatifs en matière de globalité des déplacements.

Dans cette difficulté, nous utilisons l'API Distance Matrix qui permet de fournir les distances et les temps de trajets entre plusieurs points en utilisant l'itinéraire recommandé [60]. Cette API impose pourtant une limite de 100 éléments par requête, tant du côté du

serveur qu'auprès du client. Le nombre total d'éléments est défini comme le produit du nombre de points de départ par le nombre de destinations, soit jusqu'à 10 origines et 10 destinations par requête. Notre étude se distingue par l'ampleur des données maniées qui donne lieu à 39204 éléments par matrice (distance, temps, vitesse), un volume rarement rencontré dans la littérature.

Une fois les matrices asymétriques de distances et de temps générées pour tous les arcs, nous construisons la matrice des vitesses en appliquant la relation classique : vitesse (v) = distance (d) / temps (t).

3.4 Construction de la solution initiale par insertion gloutonne

L'algorithme de type glouton est une heuristique pour les problèmes d'optimisation combinatoire, comme le VRP. Son principe repose sur le choix local, à chaque étape, de la décision qui, à l'instant où elle est prise, produit la meilleure solution en ne cherchant donc pas a priori à obtenir une solution optimale.

Dans le cadre de la distribution et du transport, cet algorithme trouve régulièrement application dans la construction de la solution initiale réalisable avant d'appliquer d'autres procédures plus affinées par la suite. Ainsi, on attribue progressivement aux clients les tournées de livraison en conservant à chaque étape le choix qui paraît le plus judicieux selon un critère retenu, que ce soit la distance parcourue, le temps de trajet, etc. L'algorithme glouton permet ainsi d'obtenir une solution initiale simple et opérationnelle, bien que souvent encore perfectible, de surcroît, par d'autres méthodes davantage optimales.

Dans la solution initiale retenue, on construit de manière séquentielle les tournées par l'utilisation de l'heuristique d'insertion gloutonne. À chaque itération, l'algorithme cherche à assigner progressivement des clients à la tournée en cours, et la création de chaque tournée prend place de telle manière que le meilleur client est choisi parmi ceux non encore affectés. L'algorithme est stoppé dès qu'une solution acceptée s'est établie après avoir visité tous les clients.

Avec son aptitude à rapidement générer une solution localement réalisable, l'une des principales avantages de l'heuristique gloutonne réside alors dans sa rapidité à la mise en œuvre et sa facilité d'implémentation, ce qui en fait un outil d'une grande vertu pour des problématiques visant à obtenir une solution approximativement correcte dans un temps réduit ! Toutefois, le risque de bloquer à un optimum local peut également être limité par des combinaisons avec d'autres heuristiques ou métaheuristiques.

3.5 Structure de voisinage

Le choix d'une structure de voisinage pour exploiter l'espace des solutions est une étape importante dans la conception de l'algorithme de recherche tabou (TS) [61]. Parmi les différentes stratégies utilisées, l'échange CROSS [62] est l'une des méthodes les plus appropriées et les plus courantes pour les problèmes de tournées de véhicules avec fenêtres de temps. Son principe repose sur l'échange de deux segments appartenant à deux routes différentes, formant ainsi une structure en croix, comme illustré dans la Figure 3.1.

Bräysy et al. [63] ont introduit une extension de cette approche, appelée l'échange iCROSS (inverted CROSS exchange). Contrairement à l'échange CROSS classique, où les segments sont échangés directement, l'échange iCROSS inverse l'ordre des segments avant de les insérer dans les nouvelles routes.

Le processus d'échange repose sur la suppression et la création de certains arcs. Plus précisément, les arcs $(i - 1, i)$ et $(k, k + 1)$ sont supprimés de la première route, tandis que $(j - 1, j)$ et $(l, l + 1)$ sont supprimés de la seconde. Ensuite, afin d'intégrer les segments inversés (k, i) et (l, j) dans leurs nouvelles positions respectives, de nouveaux arcs sont créés : $(i - 1, l)$, $(j, k + 1)$ et $(j - 1, k)$, $(i, l + 1)$, comme illustré à la figure 3.1.

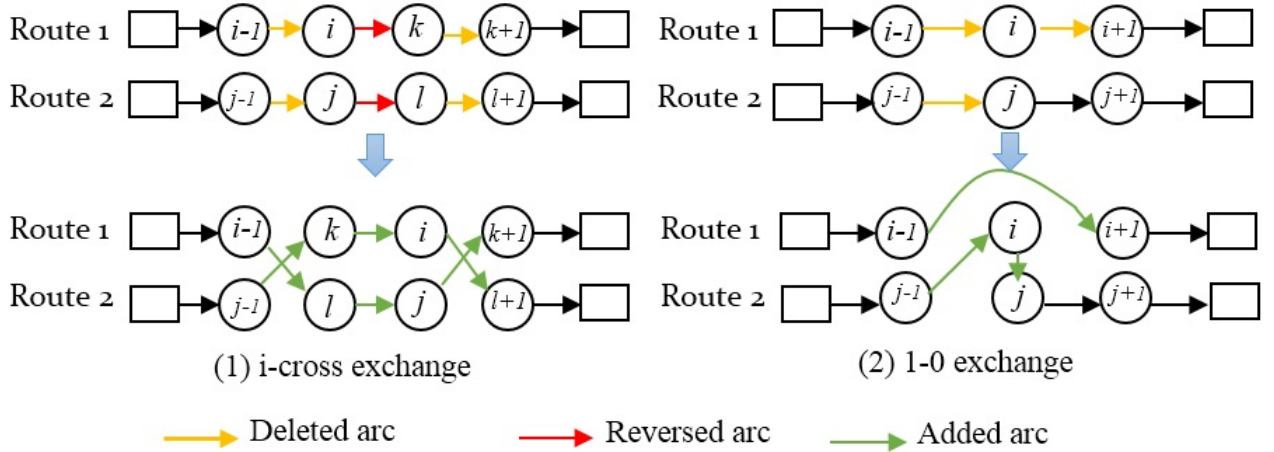
L'algorithme 3.3 précise cette procédure en plusieurs détails. Il est à noter qu'à la différence du nombre de clients échangés dans les segments, qui peut varier de 1 à (longueur de la route - 2), cela permet une certaine souplesse dans la génération des solutions voisines.

Cette méthode s'avère efficace de souplesse donner la chance de visiter un plus grand nombre de solutions dans le voisinage, en permettant de croiser des échanges de manière différente en inversant les segments et en impliquant notamment à traiter l'échange à l'échelle du segment. Ainsi, un échange iCROSS a de meilleures chances d'améliorer la solution obtenue, particulièrement dans un environnement où la contrainte de fenêtres de temps ne permet pas de réaliser des insertions simples. Cela prouve l'utilité d'un tel outil de réinsertion dans le cadre de l'optimisation des tournées de véhicules dans un contexte opérationnel du problème considéré.

Algorithm 3.3: Fonction d'échange iCROSS

```

/* Initialisation */
1  best_cost = Valeur maximale; item1 ← −1, item2 ← −1, item3 ← −1,
   item4 ← −1;
2  for i ← 1 to route1.length − 2 do
3      for k ← i to route1.length − 1 do
4          for j ← 1 to route2.length − 2 do
5              for l ← j to route2.length − 1 do
6                  calculer la nouvelle demande dans route1, route2; calculer le
                     nouveau volume dans route1, route2;
7                  if capacit = vrai et volume = vrai then
8                      mettre à jour route1, route2; calculer le temps total dans
                         route1, route2;
9                      if timeCheck(timeRoute1) = vrai et
                         timeCheck(timeRoute2) = vrai then
10                         calculer old_cost; calculer new_cost;
11                         if cross_cost < best_cost then
12                             item1 ← i, item2 ← k, item3 ← j, item4 ← l;
13                         end
14                     end
15                 end
16             end
17         end
18     end
19 end
20 return item1, item2, item3, item4, best_cost;
    
```


Figure 3.1: Structure de voisinage (1) échange iCROSS (2) échange 1-0. [1]

3.6 Gestion de la liste taboue

La recherche taboue est une métaheuristique d'optimisation combinatoire dont le but est d'explorer intelligemment l'espace des solutions en vue d'échapper aux optima locaux. Les algorithmes gloutons, par exemple sont très exposés au phénomène du blocage dans une solution sous-optimale alors que la recherche taboue utilise la mémorisation et la diversification pouvant lui permettre de continuer à chercher même si la solution actuellement sélectionnée résulte d'une amélioration souhaitée. Parmi ces mécanismes, la liste taboue joue un rôle central dans la gestion des mouvements explorés ainsi que la trajectoire de la recherche.

Dans cette approche, elle prend souvent la forme d'une matrice dont chaque cas représente un arc donné. Lors de l'initialisation, toutes les valeurs sont fixées à zéro. À chaque itération, après la sélection d'une stratégie d'échange – soit iCROSS exchange, soit 0-1 exchange – basée sur une probabilité prédéfinie, une nouvelle solution courante est obtenue, et la mise à jour de la liste taboue est effectuée. Cette mise à jour consiste à décrémenter de 1 les valeurs de tous les arcs où la valeur actuelle est supérieure à zéro, tout en ajoutant une période de tabou (Tabou tenure) aux arcs qui ont été modifiés durant l'itération. Ces arcs spécifiques sont généralement définis selon la structure de voisinage et incluent $(i - 1, l)$, $(j, k + 1)$, $(j - 1, k)$ et $(i, l + 1)$. Voir la figure 3.1.

Pour introduire une variation et éviter que tous les arcs ne soient enlevés en même temps de la liste taboue, on ajoute un nombre aléatoire variant entre 0 et 4 à la durée taboue de chaque arc concerné. Cette méthode garantit une diversification plus poussée et diminue les risques de convergence anticipée vers un optimum localisé. La quête du tabou ne se limite pas à une limitation sévère des mouvements. Elle incorpore un dispositif essentiel nommé critère d'aspiration, qui interdit une démarche taboue si elle mène à une solution meilleure que celle que nous connaissons actuellement.

Grâce à la flexibilité de cette gestion dynamique de la liste taboue qui permet à la recherche tabou de concilier intensification et diversification, de manière à privilégier une meilleure exploration de l'espace des solutions. Cet algorithme permet de contourner habilement les écueils des optima locaux et de motiver correctement la recherche au cœur de problèmes complexes, comme les problèmes de tournées de véhicules, d'ordonnancement, et d'allocation de ressources.

Chapitre 4

Le Crowdsourcing et son application via les services Google

4.1 Introduction

Avec le développement des technologies numériques et l'expansion massive d'Internet, un nouveau type de collaboration et de mobilisation collective a été créé. Ce type est appelé le **crowdsourcing** ou « production participative ». Il est défini comme une méthode permettant de solliciter un grand nombre de personnes, en général via Internet, pour accomplir des tâches autrefois effectuées par un employé ou un sous-traitant.

Le crowdsourcing est rapidement devenu un outil stratégique pour un large éventail de domaines: collecte de données, innovation, résolution de problèmes complexes à indicateurs multiples, cartographie citoyenne, entre autres. Cela est dû au potentiel du crowdsourcing de mobiliser la sagesse des foules, de réduire les coûts et de rendre le traitement des informations à grande échelle plus rapide.

Ce chapitre, en premier lieu, identifiera la base théorique du crowdsourcing, son fonctionnement, les avantages et les inconvénients. Deuxièmement, il abordera un exemple spécifique: l'exploitation du crowdsourcing par les services de Google, en particulier dans le domaine de la collecte de données de transport et de cartographie afin de créer une matrice distance-temps à utiliser dans les algorithmes d'optimisation pour les problèmes de routage.

4.2 Principes fondamentaux du Crowdsourcing

4.2.1 Définition

Le crowdsourcing peut être défini comme un processus impliquant la sous-traitance de certaines tâches qui sont normalement effectuées par un nombre limité d'employés ou d'experts, auquel devraient participer les individus qui contribuent dans la même direction que la mission qui doit être accomplie par un plus grand nombre de signataires de plateformes ouvertes. Howe [64] a pour la première fois utilisé le terme crowdsourcing en 2006 dans un article publié dans Wired Magazine. Le crowdsourcing est formé par les mots « foule » et « externalisation ».

le crowdsourcing permet de recourir à la connaissance collective et à l'intelligence de masse des utilisateurs. Cette méthode peut être utilisée dans de nombreux domaines, notamment pour résoudre des problèmes complexes, cela est illustré par l'improvisation devenue peinture de gueuteau en passant par l'étude de Daren [65].

4.2.2 Acteurs du crowdsourcing

Un projet de crowdsourcing implique généralement trois types d'acteurs principaux :

- **Le demandeur** : une entreprise, une institution, une organisation non gouvernementale, ou même un individu, qui publie une tâche ou un besoin spécifique.

- **La foule (crowd)** : un groupe large et souvent anonyme de contributeurs potentiels, qui peuvent être motivés par des récompenses financières, des incitations sociales, ou l'envie de participer à une cause.
- **La plateforme** : l'intermédiaire numérique qui héberge, facilite et régule les interactions entre le demandeur et les contributeurs.

4.2.3 Typologies principales du crowdsourcing

Le crowdsourcing peut se décliner en plusieurs catégories selon la nature de la tâche et les objectifs recherchés :

- **Crowdsourcing d'idées** : sollicitation de propositions créatives pour l'innovation ou la résolution de problèmes (ex : concours de design).
- **Crowdsourcing de micro-tâches** : division d'un travail complexe en petites tâches simples confiées à un grand nombre d'individus (ex : Amazon Mechanical Turk).
- **Crowdsourcing de données** : collecte massive d'informations via la contribution des utilisateurs (ex : OpenStreetMap).
- **Crowdfunding** : financement participatif où de nombreuses personnes contribuent à financer un projet ou une entreprise.

Ces typologies montrent la diversité d'applications possibles du crowdsourcing, de la créativité à la collecte massive de données.

4.3 Intégration du Crowdsourcing dans les systèmes intelligents

L'intégration du crowdsourcing dans les systèmes intelligents représente une évolution majeure dans la manière dont les données sont collectées, analysées et utilisées pour la prise de décision automatisée. Ainsi, la problématique étudiée dans cette thèse est un système intelligent qui incorpore le crowdsourcing à travers les services de Google, qui sont la pierre angulaire d'algorithmes avancés tels que TS. En combinant la puissance des foules avec les capacités des algorithmes d'IA, il est désormais possible de concevoir des systèmes adaptatifs, réactifs et auto-optimisants.

4.3.1 Rôle du crowdsourcing dans les systèmes intelligents

Le crowdsourcing apporte une richesse de données et de jugements humains dans des contextes où les machines seules peuvent échouer, comme :

- La collecte de données en temps réel (trafic, météo, incidents),
- La validation humaine d’algorithmes (vérification de résultats, apprentissage supervisé),
- L’enrichissement sémantique des systèmes via les annotations, commentaires ou évaluations.

Il devient un composant critique dans les systèmes hybrides [66], où l’humain et la machine coopèrent.

4.3.2 Applications typiques

- **Mobilité intelligente** : Waze utilise les rapports de conducteurs pour ajuster dynamiquement les trajets proposés.
- **Cartographie collaborative** : Google Maps/Local Guides permet la mise à jour rapide d’informations géospatiales.
- **Systèmes de recommandation** : plateformes comme Amazon ou TripAdvisor s’appuient sur les avis massifs pour nourrir leurs moteurs intelligents.
- **Logistique verte et planification dynamique** : intégration de données crowd pour ajuster les tournées en fonction du trafic, comme dans les systèmes VPRT (Vehicle Routing Problems with Real-Time data).

4.3.3 Avantages pour les systèmes intelligents

- Adaptabilité améliorée grâce aux retours utilisateurs en continu,
- Réduction des biais algorithmiques en intégrant une diversité de perspectives,
- Réduction des coûts de collecte de données [64].

4.3.4 Limites et défis

- Fiabilité des données crowd, nécessitant des mécanismes de validation,
- Problèmes de confidentialité et d’éthique (ex. : suivi de localisation),
- Besoin d’architecture technique robuste pour fusionner humain et machine [67].

4.4 Avantages et défis du Crowdsourcing

Le crowdsourcing, en mobilisant l'intelligence collective d'un grand nombre d'individus via des plateformes numériques, offre de nombreux avantages mais présente également des limites et des défis importants à considérer.

4.4.1 Avantages

Les bénéfices du crowdsourcing sont multiples :

- **Collecte rapide, économique et évolutive de données** : Le crowdsourcing permet de recueillir une grande quantité d'informations en peu de temps, à moindre coût et à grande échelle [68–71].
- **Accès à une diversité de compétences et de perspectives** : La variété des contributeurs favorise la créativité, l'innovation, et la résolution de problèmes sous différents angles [71–73].
- **Complémentarité avec les méthodes traditionnelles** : Le crowdsourcing peut enrichir ou améliorer les approches classiques de recherche et d'analyse commerciale [68, 69, 71].

4.4.2 Limites

Malgré ses avantages, le crowdsourcing présente certaines limites :

- **Projets non généralisés** : De nombreux projets de crowdsourcing restent à l'état de pilote et ne passent pas à l'échelle supérieure [69, 70].
- **Hétérogénéité des contributeurs** : La qualité et la pertinence du « crowd » peuvent varier en fonction de la tâche, ce qui influence les résultats [68, 70, 74].
- **Manque de standardisation** : L'absence de normes claires dans la conception des projets peut compromettre la fiabilité et la reproductibilité [74].

4.4.3 Défis majeurs

Pour garantir l'efficacité du crowdsourcing, plusieurs défis doivent être surmontés :

- **Gestion de la qualité** : Cela exige une conception précise des tâches, des mécanismes de contrôle et une sélection adéquate des participants [68, 70, 74, 75].
- **Conception des incitations** : Comprendre et mobiliser des facteurs tels que le plaisir, la satisfaction personnelle, la reconnaissance sociale ou la récompense financière est essentiel [71, 76].

- **Enjeux éthiques et réglementaires** : Il est indispensable de mettre en place des contrôles déontologiques et juridiques pour garantir la protection des participants et la validité des données [69, 70].
- **Défis techniques et de confidentialité** : En particulier dans le crowdsourcing spatial, l'attribution des tâches, la protection de la vie privée et la motivation sont des aspects complexes à gérer [75].

4.5 Intégration de l'API Google Distance Matrix avec Java

Voici dans cette partie de la thèse un retour sur les modalités par lesquelles nous avons opéré l'intégration de l'API Google Distance Matrix grâce à des instructions programmées dans le langage Java dans notre algorithme proposé, ceci en vue de produire automatiquement les matrices de distances et temps de trajets entre le dépôt et l'ensemble des clients, dans le cadre de la problématique de la distribution pharmaceutique sur laquelle nous nous focalisons.

4.5.1 Présentation générale

L'API Distance Matrix proposée par Google a la grande capacité de fournir les durées et distances unitaires, estimées, entre n'importe quelles plusieurs paires de points géographiques (latitude et longitude), choisis selon les méthodes de plusieurs modes de transport différents, en mesure d'améliorer notamment les applications logistiques, les applications d'optimisation des tournées, les applications de calculs réalistes des valeurs de coût de transport. Cet API fournit les distances et durées de trajet en temps réel entre des points géographiques donnés, en tenant compte du trafic. Ces services constituent une forme avancée de crowdsourcing passif, où les données sont collectées de manière implicite à partir du comportement des utilisateurs.

L'utilisation de cette API nécessitera une clé (API Key) fournie par Google et la gestion du nombre de requêtes nécessaires à ces fins, un volume limité étant imposé par l'API.

4.5.2 Objectifs de l'intégration

Les objectifs sont les suivants :

- Générer automatiquement la matrice des distances (en mètres) et des temps (en secondes) entre tous les points du réseau (dépôt + clients).

- Respecter les contraintes imposées par Google (nombre maximum d'origines ou de destinations par requête).
- Exporter les résultats sous forme de fichiers Excel pour une exploitation ultérieure dans l'algorithme d'optimisation.

4.5.3 Apport des données issues du crowdsourcing

Les données issues de Google permettent de :

- Améliorer la précision des matrices de distance et de temps,
- Réagir dynamiquement aux conditions de trafic changeantes,
- Prendre en compte les vitesses moyennes observées en temps réel,
- Réduire la consommation d'énergie en optimisant les trajets sur des critères plus réalistes que la distance géométrique seule.

En effet, celle-ci se fonde d'abord sur la participation de millions d'internautes, se prêtant ou s'associant à l'utilisation de services de géolocalisation de leur position (smartphones Android, services Google Maps, etc.).

4.5.4 Code Java : récupération de la matrice de distance et de temps

Le code suivant permet de récupérer les distances et durées entre une liste d'origines et de destinations.

```
1 public double[][][] distanceMatrix(String[] origins, String[] destinations)
2     throws ApiException, InterruptedException, IOException {
3
4     GeoApiContext context = new GeoApiContext.Builder()
5         .apiKey(API_KEY)
6         .build();
7
8     DistanceMatrixApiRequest req = DistanceMatrixApi.newRequest(context);
9     DistanceMatrix t = req.origins(origins)
10        .destinations(destinations)
11        .mode(TravelMode.DRIVING)
12        .await();
13
14     double[][][] array = new double[origins.length][2][destinations.length
15        ];
16     for(int i = 0; i < origins.length; i++) {
17         for(int j = 0; j < destinations.length; j++) {
```

```

17         array[i][0][j] = t.rows[i].elements[j].distance.inMeters;
18         array[i][1][j] = t.rows[i].elements[j].duration.inSeconds;
19     }
20 }
21 return array;
22 }

```

Listing 4.1: Fonction de récupération de la matrice de distance et de temps

4.5.5 Découpage par lots pour respecter les limitations de l'API

Comme l'API limite le nombre de points par requête (10 points), nous avons conçu une méthode qui divise dynamiquement les requêtes en lots. Cette méthode utilise la fonction précédente pour remplir progressivement les matrices complètes.

```

1 public void CalculerDistanceMatrix(Node[] NodesForRead, int NoOfCustomers,
2     int LimitNumber)
3     throws ApiException, InterruptedException, IOException {
4
5     int NoOfCustomersAndDepot = NoOfCustomers + 1;
6     double rate = NoOfCustomersAndDepot / (double) LimitNumber;
7     int NoOfLoopForEachCustomer = (int) Math.ceil(rate);
8
9     for (int i = 0; i <= NoOfCustomers; i++) {
10         String[] origins = { new LatLng(NodesForRead[i].Lat, NodesForRead[i]
11             .Lng).toString() };
12
13         for (int k = 0; k < NoOfLoopForEachCustomer; k++) {
14             int RestElements = NoOfCustomersAndDepot - (LimitNumber * k);
15             int currentBatchSize = Math.min(LimitNumber, RestElements);
16
17             String[] destinations = new String[currentBatchSize];
18             for (int j = 0; j < currentBatchSize; j++) {
19                 destinations[j] = new LatLng(NodesForRead[j + (LimitNumber
20                     * k)].Lat,
21                     NodesForRead[j + (LimitNumber
22                         * k)].Lng).toString();
23             }
24
25             double[][][] result = distanceMatrix(origins, destinations);
26             for (int j = 0; j < currentBatchSize; j++) {
27                 mapsDistanceMatrix[i][j + (LimitNumber * k)] = result
28                     [0][0][j];
29                 mapsTimeMatrix[i][j + (LimitNumber * k)] = result[0][1][j];
30             }
31         }
32     }
33 }

```

```

28
29     WriteResultToExcel (mapsDistanceMatrix , mapsTimeMatrix);
30 }

```

Listing 4.2: Découpage et traitement par lots

4.5.6 Exportation vers Excel

Après génération, les deux matrices sont exportées vers un fichier Excel grâce à la bibliothèque Apache POI. Ce fichier contient deux feuilles : “Distance Matrix” et “Time Matrix”.

```

1 public void WriteResultToExcel (double [][] mapsDistanceMatrix , double [][]
    mapsTimeMatrix) {
2     HSSFWorkbook workbook = new HSSFWorkbook();
3     HSSFSheet sheet1 = workbook.createSheet("Distance Matrix");
4     HSSFSheet sheet2 = workbook.createSheet("Time Matrix");
5
6     for (int i = 0; i < mapsDistanceMatrix.length; i++) {
7         Row row1 = sheet1.createRow(i);
8         Row row2 = sheet2.createRow(i);
9         for (int j = 0; j < mapsDistanceMatrix[i].length; j++) {
10             row1.createCell(j).setCellValue(mapsDistanceMatrix[i][j]);
11             row2.createCell(j).setCellValue(mapsTimeMatrix[i][j]);
12         }
13     }
14
15     try (FileOutputStream out = new FileOutputStream("DistanceTimeMatrix.
        xls")) {
16         workbook.write(out);
17     } catch (IOException e) {
18         e.printStackTrace();
19     }
20 }

```

Listing 4.3: Exportation des matrices vers un fichier Excel

L’intégration de l’API Distance Matrix de Google nous a permis d’obtenir un modèle plus précis des distances et des temps de trajet réels. Cela est nécessaire pour améliorer la précision des algorithmes de routage et optimiser la consommation d’énergie dans notre étude de cas.

Conclusion

Le crowdsourcing, qui représente une approche coopérative de collecte d’informations et de résolution de problèmes, s’est à présent transformé en un composant crucial des systèmes

intelligents. En exploitant l'intelligence collective à grande échelle, on peut transcender les contraintes classiques en termes de coût, d'accessibilité ou de capacité d'évolution des solutions.

Ce chapitre traite des principes de base du crowdsourcing, de ses différentes formes, et de sa mise en œuvre et de son intégration via l'API Google Distance Matrix en Java. Nous nous concentrons particulièrement sur la mise en œuvre efficace de ce modèle par le biais des services Google, qui ont été utilisés dans notre recherche pour enrichir de manière dynamique, fiable et instantanée les informations relatives à la planification logistique.

Chapitre 5

Méthodologie Expérimentale et Analyse des Résultats

5.1 Introduction

Ce chapitre présente la méthodologie expérimentale adoptée pour évaluer l'efficacité énergétique des tournées de livraison dans le cadre d'un cas réel de distribution pharmaceutique en Algérie. L'approche repose sur l'application d'un algorithme de recherche tabou probabiliste, combinée à des données de trafic réelles extraites via l'API Google Maps.

Afin d'analyser l'impact des différents paramètres de modélisation sur la consommation énergétique, plusieurs scénarios de vitesse ont été testés, incluant des vitesses constantes (40, 60, 80 et 100 km/h) ainsi qu'une vitesse dynamique dérivée des conditions réelles de circulation. L'étude compare les résultats obtenus pour chaque cas, en mettant en évidence les différences de performance énergétique.

De plus, une analyse statistique de type ANOVA à un facteur a été conduite pour vérifier si les différences observées entre les groupes de vitesses sont significatives sur le plan statistique. Cette démarche permet de valider l'hypothèse selon laquelle la variation de la vitesse influence réellement la consommation énergétique, apportant ainsi une justification rigoureuse à l'utilisation de données dynamiques dans les modèles d'optimisation.

5.2 Présentation du Problème et Contexte Expérimental

L'algorithme probabiliste de recherche tabou est appliqué sur des cas réels du cas pratique de la distribution pharmaceutique en Algérie en référence à la Figure 5.1, dans le cadre d'une étude ayant pour but de modéliser et d'optimiser les tournées de livraison d'une entreprise avec pour objectif, la minimisation des coûts de distribution et l'optimisation des processus logistiques. Le cas étudié fait appel à l'utilisation d'une flotte homogène composée de 16 véhicules identiques, ayant chacun les caractéristiques suivantes : charge utile maximale : 1 200 kg volume contenu : 5 mètres cubes. Ces véhicules sont appelés à livrer un total de 198 clients répartis sur cinq instances distinctes correspondant aux cinq jours consécutifs de la semaine, choisis aléatoirement dans le but de représenter la diversité des scénarios de l'année complète en fonction des variations réelles de la demande.

Pour décrire chaque client, plusieurs paramètres sont considérés, la fenêtre temporelle autorisant des horaires de livraison, le temps de service fixé à 10 minutes en moyenne, ainsi que les contraintes de demande en termes de poids (kilogrammes) et de volume (mètres cubes), qui influent sur l'affectation de véhicules pour les livraisons et sur la faisabilité de tournées optimisées par l'algorithme, dont l'exécution s'arrête après 500 itérations avec une durée de tabou (Tabu tenure) fixée à 15 (empêchant ainsi un retour immédiat à des configurations récemment explorées).

Les expérimentations se déroulent sur une machine équipée d'une unité centrale dotée d'un processeur Intel Core i3-4030U à 1,9 GHz, avec 4 Go de RAM, via le développement

et les tests opérationnels au moyen de l'environnement d'exécution Java NetBeans IDE 8.2, fonctionnant sous Windows 7. Ainsi, l'encadrement expérimental permet de garantir la reproductibilité des résultats et d'évaluer les performances de l'algorithme dans un milieu contrôlé, pouvant être le reflet des conditions du monde de l'entreprise, dans lequel il fait l'objet de demandes concurrentielles dans le cadre des contraintes de calcul.



Figure 5.1: Emplacements des clients sur la carte.

Le problème est pris en compte avec des fenêtres temporelles en fonction d'une classification des clients. Celle-ci est fournie par la direction de l'entreprise selon le statut de fidélité des clients. Cette classification permet de définir des créneaux adaptés à leur besoin prioritaire. En règle générale, les clients sont disponibles du matin jusqu'au soir, à l'exception de cas particuliers. Toutefois, pour optimiser la planification des livraisons et réduire les temps d'attente, la plage horaire effective de service est restreinte à l'intervalle 8 h – 14 h.

Afin d'éliminer tout temps d'attente inutile et d'harmoniser l'organisation des tournées, l'heure de disponibilité des clients est fixée à 8 h, garantissant ainsi que toutes les fenêtres temporelles s'ouvrent simultanément. Les horaires de fermeture des fenêtres temporelles sont quant à eux distribués comme suit : 9 h, 11 h, 12 h et 14 h, avec une priorité accordée aux clients les plus fidèles pour bénéficier des créneaux les plus larges. Le but de cette échéance est de composer les contraintes de service avec la satisfaction client. Le dépôt central est ouvert sur une assez large amplitude de temps, 5 h à 20 h, supposant une flexibilité suffisante pour préparer les livraisons et assurer les retours de véhicules.

5.3 Paramètres constants dans l'équation de la consommation énergétique

Tous les paramètres constants qui interviennent dans le calcul d'énergie consommée sont mis en œuvre dans le Tableau 5.1. C'est au moyen de ces valeurs, qui interviennent dans la fonction objective, que l'on peut évaluer la consommation réelle des véhicules, les contraintes physiques et mécaniques du système prenant ainsi une importance considérable dans l'optimisation des tournées, tant pour le choix d'affectation des véhicules que des itinéraires.

En intégrant ces constantes dans le modèle, l'estimation des besoins énergétiques est plus proche de la réalité, puisqu'elle tient compte des conditions de transport et des caractéristiques du véhicule. Le modèle global tente alors de minimiser les coûts d'exploitation tout en cherchant à optimiser les trois contraintes des capacités, des fenêtres et de la consommation énergétique, afin d'optimiser l'efficacité logistique du système.

Table 5.1: Valeurs typiques attribuées aux paramètres.

Paramètre	Valeur
η_T	0.95
g	9.8
ρ	1.165
C_D	0.30
g_e	107
S	10^5
q	14.37
γ_{mi}	1.1
α_i	2.277
S_i	169.45
n_i	62
n_p	5.10^3
P_{max}	94
δ_{ax}	3.5
δ_n	1.2
r_d	0.5
k	4

La vitesse, lorsqu'elle est combinée avec d'autres facteurs, a un impact majeur sur la consommation énergétique des véhicules dans le cadre de l'optimisation des tournées de livraison. En effet, la résistance aérodynamique dépend de la vitesse et la consommation de carburant en est influencée, ce qui a des impacts sur les coûts d'opération.

Les différentes situations de vitesse prises en compte dans l'étude, permettent de mieux comprendre cette relation en estimant l'énergie consommée sur les trajets.

Dans cette étude, trois approches principales sont adoptées pour modéliser la vitesse des véhicules :

- Vitesse constante pour chaque point de livraison : Un même véhicule effectue la totalité de sa mission de tournée à une même et unique vitesse. Ce scénario d'idéalisation permet, d'une part, d'analyser la consommation de combustible dans un état de conditions idéalisées puis, d'autre part enfin de mieux appréhender ou servir de référence comparatif à la suite d'analyses des autres scénarios plus réalistes.
- Vitesse variable au sein d'un même trajet : Chaque véhicule peut adopter des vitesses différentes en fonction des segments de route parcourus, tout en conservant une vitesse fixe sur chaque arc spécifique. Cette approche, plus réaliste, permet de modéliser la variabilité des vitesses en fonction des caractéristiques des routes et des conditions de circulation.
- Vitesse moyenne entre chaque arc : dans ce cas, la vitesse des véhicules sera paramétrée dynamiquement pour chaque segment en fonction des informations sur le trafic et les conditions réellement observées fournies par Google Maps, en référence au comportement des véhicules et repose sur une estimation plus fine de la consommation énergétique.

5.4 Synthèse des Résultats et Analyse

Le Tableau 5.2 présente les résultats obtenus en utilisant les vitesses issues des services Google Maps, ce qui implique une variation dynamique de la vitesse sur chaque arc du réseau routier. Cette approche, bien que plus fidèle à la réalité, introduit une complexité supplémentaire dans le calcul de la consommation énergétique en raison de la fluctuation des vitesses et de la nécessité de recalculer les paramètres de résistance et de consommation à chaque segment de route.

En parallèle, afin de mieux comprendre l'impact d'une vitesse constante sur la consommation énergétique, plusieurs tests ont été menés avec des vitesses fixes tout au long du trajet. La plage de vitesse testée s'étend de 40 à 100 km/h, avec des paliers de 20 km/h (40, 60, 80 et 100 km/h). Les résultats obtenus pour ces scénarios sont détaillés dans les Tableaux 5.6-5.9.

Par exemple, le Tableau 5.6 illustre les résultats obtenus lorsque la vitesse des véhicules est fixée à 40 km/h sur l'ensemble du trajet. Cette configuration permet de mettre en évidence les différences en termes de consommation énergétique par rapport aux autres

vitesse testées. En augmentant progressivement la vitesse jusqu'à 100 km/h, nous pouvons observer l'évolution de la consommation énergétique et identifier les seuils critiques où l'augmentation de la vitesse devient énergétiquement inefficace.

L'analyse comparative de ces différentes approches permet d'évaluer l'impact de la vitesse sur l'optimisation des tournées et la minimisation des coûts énergétiques. Ces résultats sont essentiels pour adapter les stratégies de planification des livraisons et intégrer des modèles prédictifs plus précis, prenant en compte les variations de vitesse en conditions réelles.

Table 5.2: Consommation d'énergie, distance et émissions de CO_2 en utilisant la vitesse des services Google Maps.

Données	Clients	Véhicules utilisés	Distance (km)	Énergie con- sommée (MJ)	Émissions de CO_2 (kg)	Consomm- ation de carburant (L)	Temps CPU (s)
w_1d_1	081	12	5620.62	13375.43	984.43	349.23	45
w_1d_2	065	09	2857.61	4630.76	340.82	120.90	29
w_1d_3	081	14	5721.38	13655.35	1005.03	356.54	40
w_1d_4	064	11	3221.00	6747.51	496.62	176.18	28
w_1d_5	077	13	4410.81	10474.14	770.90	273.48	35
w_2d_1	079	15	6366.49	15248.50	1122.29	398.13	34
w_2d_2	071	10	3333.11	7721.31	568.29	201.60	41
w_2d_3	082	13	4623.28	11083.74	815.76	289.39	61
w_2d_4	075	11	3881.50	9117.64	671.06	238.06	42
w_2d_5	105	16	6373.45	15014.20	1105.06	392.02	81
w_3d_1	069	13	5413.74	12760.86	939.20	333.18	32
w_3d_2	082	12	4160.96	9910.85	729.44	258.77	50
w_3d_3	063	12	5593.45	13330.59	981.13	348.06	26
w_3d_4	050	09	3377.93	7914.78	582.53	206.65	17
w_3d_5	058	10	3901.90	9149.96	673.44	238.90	23
w_4d_1	074	13	5109.84	12218.53	899.28	319.02	31
w_4d_2	061	10	3447.32	8038.35	591.62	209.88	25
w_4d_3	049	08	3630.01	8634.20	635.48	225.44	13
w_4d_4	069	10	4011.79	9562.82	703.82	249.68	93
w_4d_5	080	13	4910.54	11627.58	855.79	303.59	118
Moyenne	71.75	11.7	4498.34	10510.86	773.60	274.44	43.2

Afin de valider la robustesse de l'approche proposée, nous avons présenté une version simplifiée de notre méthode dans un article de conférence scientifique [52]. Cette version reprend la même procédure de résolution détaillée dans le chapitre 3, en conservant la structure de voisinage basée sur l'échange 0-1. Toutefois, contrairement à la version complète développée dans cette thèse, la stratégie d'échange croisé (i-cross Exchange) n'a pas été intégrée. Les résultats obtenus à partir de cette version simplifiée sont résumés dans le Tableau 5.3 permettant ainsi une première évaluation expérimentale de notre approche, dans un cadre réduit mais représentatif du problème réel.

Table 5.3: Consommation d'énergie, distance et émissions de CO_2 en utilisant la vitesse des services Google Maps avec TS simple.

Données	Clients	Véhicules utilisés	Distance (km)	Énergie con- sommée (MJ)	Émissions de CO_2 (kg)	Consomma- tion de carburant (L)	Temps CPU (s)
w_1d_1	081	12	5608.45	12951.81	953.25	338.16	19
w_1d_2	065	09	2939.61	6587.89	484.86	172.00	12
w_1d_3	081	14	5820.16	13795.24	1015.33	360.18	19
w_1d_4	064	10	3136.25	6477.17	476.72	169.11	11
w_1d_5	077	12	4518.94	10194.37	750.30	266.17	18
w_2d_1	079	15	6523.34	15073.22	1109.38	393.55	17
w_2d_2	071	10	3466.48	8130.68	598.41	212.28	16
w_2d_3	082	13	5040.14	11491.78	845.79	300.04	19
w_2d_4	075	11	3920.28	8241.70	606.58	215.18	16
w_2d_5	105	15	6770.96	15910.04	1170.97	415.40	29
w_3d_1	069	13	5505.13	12602.10	927.51	329.03	13
w_3d_2	082	13	4422.65	8494.97	625.23	221.80	19
w_3d_3	063	12	5839.76	12708.42	935.33	331.81	11
w_3d_4	050	09	3383.48	6798.53	500.37	177.50	09
w_3d_5	058	09	3946.88	8401.40	618.34	219.35	10
w_4d_1	074	11	4968.83	11621.98	855.37	303.44	16
w_4d_2	061	10	3419.50	7695.81	566.41	200.93	12
w_4d_3	049	08	3636.73	8201.16	603.60	214.12	07
w_4d_4	069	11	4264.07	9979.39	734.48	260.55	11
w_4d_5	080	13	4946.58	10088.81	742.53	263.41	18
Moyenne	71.75	11.5	4603.91	10272.32	756.04	268.20	15.1

Une analyse comparative des performances expérimentales obtenues entre la version simplifiée (0-1 exchange) et la version complète (0-1 exchange et i-cross exchange) met en évidence plusieurs écarts significatifs, détaillés dans les Tableaux 5.3 et 5.2. Bien que les deux versions exploitent la même source de données de vitesse issue des services Google Maps, la différence réside essentiellement dans la stratégie de recherche locale : la version simplifiée repose uniquement sur un voisinage 0-1 exchange, tandis que la version complète intègre en plus la stratégie i-cross exchange, permettant une exploration plus riche de l'espace des solutions.

Sur l'ensemble des 20 instances, la version complète présente une consommation énergétique moyenne légèrement plus élevée (10510.86 MJ contre 10272.32 MJ), de même pour les émissions de CO_2 (773.60 kg contre 756.04 kg), ainsi que pour la consommation de carburant (274.44 L contre 268.20 L). Cependant, ces différences doivent bien évidemment être relativisées, car elles s'expliquent notamment par une réduction du nombre total de la distance parcourue (4498.34 km en moyenne contre 4603.91 km) et révèlent en quelque sorte une optimisation des tournées qui va encore plus loin que précédemment. L'approche utilisée pour déterminer les résultats consiste à sélectionner les meilleurs résul-

tats après avoir exécuté l'algorithme plusieurs fois ou un certain nombre d'itérations, ce qui peut s'expliquer par le fait que le nombre d'itérations dans la version simple est deux fois plus élevé que dans la première version, 20 fois contre 10 fois. En d'autres termes, la probabilité d'obtenir de meilleurs résultats est plus élevée lorsque l'algorithme est exécuté plus souvent. Néanmoins, les résultats sont assez proches.

En revanche, la principale différence se manifeste au niveau du temps de calcul, où la version complète demande en moyenne près de trois fois plus de temps CPU (43.2 s contre 15.1 s). Ceci est dû à l'utilisation d'un ordinateur plus performant que celui utilisé précédemment et à la version simple de TS qui utilise moins de ressources.

Le Tableau 5.2 présente les résultats obtenus en utilisant les vitesses fournies par Google Maps pour chaque segment de trajet. La première colonne du tableau représente l'ensemble des données, structuré en quatre sections, où chaque semaine est identifiée par la lettre w , suivie de la lettre d qui désigne le jour correspondant. Cette structuration permet une analyse comparative des variations quotidiennes de la consommation énergétique, des distances parcourues et des émissions de CO_2 . L'un des éléments clés de cette étude est la relation entre l'énergie consommée, la distance parcourue et les émissions de CO_2 . À partir de la quantité d'énergie consommée, nous pouvons estimer les émissions de CO_2 ainsi que la consommation de carburant. Le calcul des émissions de CO_2 repose sur l'équation suivante :

$$CO_2Emission(kg) = \frac{Energyconsumed(MJ) * 73.6}{1000} \quad (5.1)$$

où 73.6 g/MJ représente le facteur d'émission du carburant diesel, conformément aux travaux de Jokiniemi [77]. Cette équation permet d'obtenir une estimation fiable des émissions polluantes générées par les trajets effectués.

Une observation immédiate concerne le nombre de véhicules utilisés par jour. Ce nombre varie entre 8 et 16 véhicules, avec une moyenne de 12 véhicules par jour. La réduction du nombre de véhicules utilisés représente une optimisation significative en termes de rentabilité économique et d'impact environnemental, puisqu'une flotte plus restreinte entraîne une réduction de la consommation de carburant et des émissions polluantes.

Concernant la distance parcourue, le minimum observé est de 2857.61 km pour le jeu de données w_1d_2 , avec des émissions minimales de 340.82 kg de CO_2 . À l'opposé, la distance maximale enregistrée est de 6373.45 km pour w_2d_5 , avec une émission de 1105.06 kg de CO_2 . Cependant, il est intéressant de noter que l'émission la plus élevée n'a pas été observée sur ce trajet, mais plutôt sur w_2d_1 , où 1122.29 kg de CO_2 ont été émis pour une distance légèrement inférieure de 6366.49 km. L'écart entre ces deux cas est de 6.96 km en distance et 17.23 kg en émissions de CO_2 , ce qui souligne une observation essentielle : une distance plus longue ne signifie pas nécessairement une consommation de carburant plus élevée.

Cette variation s'explique par plusieurs facteurs influençant la consommation énergétique, notamment la vitesse moyenne et le poids transporté. Une vitesse plus élevée peut augmenter la résistance aérodynamique et donc la consommation de carburant, tandis que le poids des véhicules influe directement sur l'énergie requise pour le déplacement. En d'autres termes, la gestion optimale de la vitesse et de la charge transportée peut jouer un rôle crucial dans la réduction de la consommation énergétique et des émissions polluantes.

En somme, ces résultats mettent en évidence la complexité des facteurs influençant la consommation d'énergie et les émissions de CO_2 , démontrant qu'une approche purement basée sur la distance parcourue est insuffisante. Une gestion intelligente des vitesses, des itinéraires et du nombre de véhicules pourrait conduire à des réductions significatives de l'impact environnemental et des coûts opérationnels.

Afin d'évaluer la performance de notre algorithme proposé, la Recherche Tabou (TS), nous l'avons comparé à une solution initiale obtenue à l'aide de l'heuristique d'insertion gloutonne, dont les résultats sont présentés dans le Tableau 5.4. L'heuristique d'insertion gloutonne génère une solution faisable mais sous-optimale en insérant de manière itérative l'option la plus avantageuse disponible dans la solution courante, sans exploration approfondie du paysage des solutions possibles. Cette approche permet une construction rapide, mais elle conduit généralement à des solutions de qualité inférieure.

À l'inverse, l'algorithme de Recherche Tabou (TS) exploite une mémoire adaptative et une stratégie d'exploration plus souple, ce qui lui permet de surmonter les solutions localement optimales en naviguant intelligemment dans l'espace de recherche. Grâce à sa capacité à éviter les cycles et à explorer des solutions diversifiées, la TS affine la solution initiale et améliore significativement l'optimisation des résultats.

La comparaison entre les deux approches démontre l'efficacité et la robustesse de la Recherche Tabou pour optimiser des critères tels que le nombre de véhicules utilisés, la distance parcourue, la consommation énergétique et les émissions de CO_2 . L'impact de cette amélioration ne se limite pas uniquement à la réduction des coûts opérationnels, mais contribue également à une diminution de l'empreinte environnementale, élément essentiel dans un contexte où l'optimisation énergétique est un enjeu majeur.

Table 5.4: Résultats de la solution initiale en utilisant l’heuristique d’insertion gloutonne.

Données	Clients	Véhicules	Distance (km)	Énergie (MJ)	Émissions de CO_2 (kg)	Carburant (L)	Temps CPU (s)
w_1d_1	081	24	13348.95	32379.21	2383.11	845.41	1
w_1d_2	065	19	8486.14	20574.13	1514.26	537.18	1
w_1d_3	081	29	14616.02	35603.85	2620.44	929.60	1
w_1d_4	064	20	9659.96	23598.09	1736.82	616.14	1
w_1d_5	077	27	13592.63	33045.57	2432.15	862.81	1
w_2d_1	079	27	15748.61	38397.90	2826.09	1002.56	1
w_2d_2	071	21	10356.94	25041.73	1843.07	653.83	1
w_2d_3	082	25	14313.21	34790.21	2560.56	908.36	1
w_2d_4	075	23	11625.92	28364.57	2087.63	740.59	1
w_2d_5	105	31	17795.65	43269.05	3184.60	1129.74	1
w_3d_1	069	25	12765.76	30896.53	2273.98	806.70	1
w_3d_2	082	23	11528.64	30003.80	2208.28	783.39	1
w_3d_3	063	23	12286.37	30003.80	2208.28	783.39	1
w_3d_4	050	19	9364.14	22759.15	1675.07	594.23	1
w_3d_5	058	18	10143.91	24720.02	1819.39	645.43	1
w_4d_1	074	25	12765.76	30896.53	2273.98	806.70	1
w_4d_2	061	22	10042.51	24451.66	1799.64	638.42	1
w_4d_3	049	16	8770.96	21157.53	1557.19	552.42	1
w_4d_4	069	21	11436.48	27915.39	2054.57	728.86	1
w_4d_5	080	24	13868.12	33742.63	2483.46	881.01	1
Moyenne	71.75	23.15	12091.18	29395.23	2163.49	767.50	1

L’analyse des résultats obtenus met en évidence une relation non linéaire entre la distance parcourue et les émissions de CO_2 . Contrairement à l’hypothèse intuitive selon laquelle une augmentation de la distance entraîne systématiquement une hausse des émissions, les données montrent plusieurs cas où des itinéraires plus longs aboutissent à une réduction des émissions. Ce phénomène peut être expliqué de manière générale, et non spécifiquement pour cette étude, par plusieurs facteurs liés aux caractéristiques de la route, au trafic et à la gestion des véhicules.

L’un des éléments déterminants expliquant cette relation non linéaire réside dans la formulation de l’équation utilisée pour calculer l’énergie consommée, où des variables telles que la vitesse et le poids du véhicule jouent un rôle essentiel. En effet, la consommation énergétique ne dépend pas uniquement de la distance parcourue, mais aussi des conditions de conduite. Par exemple, une vitesse plus élevée sur certaines sections de route peut induire une surconsommation de carburant, tandis qu’une vitesse plus stable et modérée favorise une meilleure efficacité énergétique. De même, la charge transportée par le véhicule influence directement la consommation : un itinéraire plus court mais avec des véhicules fortement chargés peut engendrer une consommation plus importante que des trajets plus longs effectués avec des véhicules moins chargés.

L’analyse des résultats montre plusieurs exemples où cette tendance se manifeste clairement. Par exemple, dans le cas de w_2d_1 , une augmentation de 197 km a conduit à une réduction de 13,7 kg des émissions de CO_2 , ce qui confirme que la consommation de carburant ne dépend pas uniquement de la distance, mais aussi de la vitesse et du poids du véhicule. De manière similaire, le cas de w_1d_1 illustre une augmentation minime de 5,77

km, mais qui a permis une diminution de 7,23 kg des émissions. Ces résultats suggèrent que l'efficacité énergétique ne se limite pas à une réduction mécanique des kilomètres parcourus, mais qu'elle repose sur un compromis entre distance, vitesse, poids et conditions de circulation. Comme illustré dans la Figure 5.2, le chemin A, bien que plus long que le chemin B, entraîne une quantité d'émissions inférieure, ce qui démontre que la distance seule n'est pas un indicateur suffisant pour évaluer l'impact environnemental d'un trajet.

Table 5.5: Cas où une plus grande distance entraîne de moindres émissions.

Données	Distance (km)	Véhicules utilisés	Émissions de CO_2 (kg)	Consommation de carburant (L)	Différence de distance (km)
w_1d_1	5620.62	12	984.43	349.23	+5.77
	5614.85	12	991.66	351.79	
w_1d_2	2857.61	09	340.82	120.90	+87.41
	2770.20	09	472.84	167.74	
w_1d_5	4410.81	13	770.90	273.48	+120.47
	4290.34	12	778.69	276.24	
w_2d_1	6563.49	15	1108.59	393.27	+197
	6366.49	15	1122.29	398.13	
w_3d_2	4243.08	14	718.78	254.99	+82.12
	4160.96	12	729.44	258.76	
w_4d_2	3447.32	10	591.62	209.87	+6.73
	3440.59	10	595.94	211.41	

Par ailleurs, la gestion du nombre de véhicules utilisés joue également un rôle clé dans l'optimisation des émissions. Dans certains cas, l'adoption d'une solution impliquant un nombre réduit de véhicules a permis d'obtenir des résultats intéressants. C'est notamment le cas de w_1d_5 et w_3d_2 , où un véhicule de moins a permis une meilleure répartition des charges, tout en limitant l'impact global des émissions. Cette stratégie se révèle particulièrement bénéfique sur le plan économique, en réduisant les coûts d'exploitation liés à l'entretien et à la consommation de carburant.

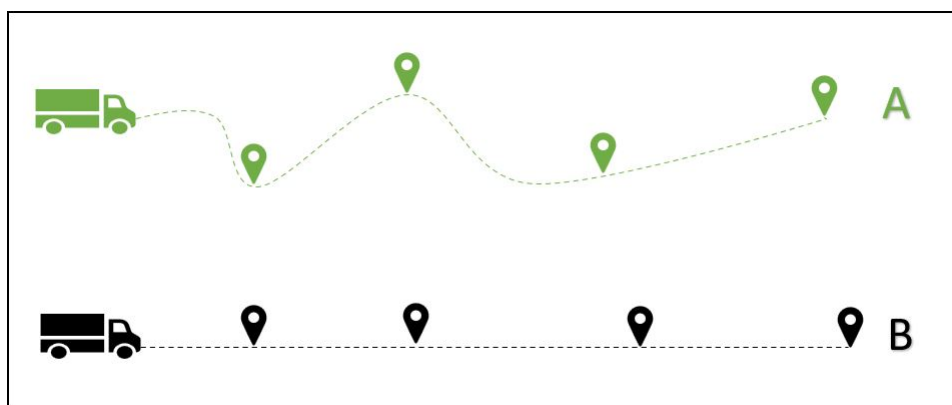


Figure 5.2: Longue distance avec des émissions de CO_2 réduites.

Cette analyse vise à évaluer l'impact de la vitesse sur l'estimation de la consommation d'énergie et des émissions en comparant les résultats obtenus à l'aide de vitesses constantes avec ceux issus des vitesses réelles fournies par les services Google Maps. Dans cette étude,

nous désignons la vitesse extraite de Google Maps sous le terme de « vitesse réelle » (*real speed*).

Les résultats obtenus pour différentes vitesses constantes le long du trajet sont présentés dans les tableaux 5.6-5.9 et les figures correspondantes 5.3 (a)-(d). Dans cette configuration expérimentale, chaque solution proposée est générée en supposant une vitesse constante sur l'ensemble du trajet, simulant ainsi un scénario d'optimisation avec vitesse fixe. Cette approche permet d'évaluer l'influence de cette hypothèse sur les résultats obtenus. Elle diffère d'une approche plus simple consistant à proposer une solution unique et à en déduire ultérieurement les émissions correspondantes à différentes vitesses constantes.

Afin de quantifier l'écart entre les émissions calculées avec une vitesse constante et celles obtenues avec la vitesse réelle, nous définissons un indicateur de déviation exprimé en pourcentage. Il est calculé comme la différence entre les émissions de CO_2 réelles (celles basées sur la vitesse réelle) et les émissions estimées à vitesse constante, divisée par ces dernières, puis multipliée par 100.

Table 5.6: Consommation énergétique à vitesse constante de 40 km/h.

Données (Set)	Dist. (km)	Véhic. Util.	Énergie (MJ)	CO_2 (Kg)	Carburant (L)	CPU (s)	Énergie réelle (MJ)	CO_2 réel (Kg)	Carburant réel (L)	Écart (%)
w_1d_1	5642.71	12	11838.95	871.35	309.11	46	13473.69	991.66	351.79	-13.81
w_1d_2	2783.96	9	5855.80	430.99	152.89	32	2783.96	481.03	170.65	-11.61
w_1d_3	5731.49	14	11973.54	881.25	312.63	42	13703.32	1008.56	357.79	-14.45
w_1d_4	3107.87	10	6531.20	480.70	170.53	28	6610.35	486.52	172.59	-1.21
w_1d_5	4418.20	13	9264.00	681.83	241.88	37	10433.15	767.88	272.41	-12.62
w_2d_1	6525.39	14	13663.22	1005.61	356.74	34	15531.40	1143.11	405.51	-13.67
w_2d_2	3573.22	10	7477.35	550.33	195.23	33	8299.67	610.85	216.70	-11.00
w_2d_3	4627.46	11	9664.85	711.33	252.35	51	10964.87	807.01	286.29	-13.45
w_2d_4	3838.31	11	8074.71	594.30	210.83	39	8926.19	656.97	233.06	-10.55
w_2d_5	6381.20	15	13345.55	982.23	348.45	87	15049.28	1107.63	392.93	-12.77
w_3d_1	5449.92	13	11399.69	839.02	297.64	25	12842.20	945.19	335.31	-12.65
w_3d_2	4168.20	13	8733.31	642.77	228.02	49	9778.21	719.68	255.31	-11.97
w_3d_3	5587.20	12	11650.14	857.45	304.18	26	13222.87	973.20	345.24	-13.50
w_3d_4	3382.73	9	7076.37	520.82	184.76	16	7943.63	584.65	207.41	-12.26
w_3d_5	3872.83	9	8092.68	595.62	211.30	24	9149.52	673.40	238.89	-13.06
w_4d_1	4952.59	12	10392.00	764.85	271.33	36	11752.32	864.97	306.85	-13.09
w_4d_2	3443.30	10	7208.84	530.57	188.22	25	8072.29	594.12	210.76	-11.98
w_4d_3	3615.53	9	7566.71	556.91	197.56	15	8509.64	626.31	222.18	-12.46
w_4d_4	4079.69	11	8536.02	628.25	222.87	27	9664.53	711.31	252.34	-13.22
w_4d_5	4984.64	12	10429.24	767.59	272.30	43	11807.56	869.04	308.29	-13.22
Moyenne	4508.32	11.45	9438.71	694.69	246.44	35.75	10425.93	781.15	292.63	-12.13

Table 5.7: Consommation énergétique à vitesse constante de 60 km/h.

Données (Set)	Dist. (km)	Véhic. Util.	Énergie (MJ)	CO_2 (Kg)	Carburant (L)	CPU (s)	Énergie réelle (MJ)	CO_2 réel (Kg)	Carburant réel (L)	Écart (%)
w_1d_1	5636.35	12	13237.69	974.29	345.63	46	13476.19	991.85	351.86	-1.77
w_1d_2	2826.25	9	6648.30	489.31	173.58	31	6648.95	489.36	173.60	-0.01
w_1d_3	5788.16	14	13550.65	997.33	353.80	40	13937.85	1025.83	363.91	-2.78
w_1d_4	3163.70	10	7438.79	547.49	194.22	27	7143.73	525.78	186.52	4.13
w_1d_5	4290.34	12	10075.37	741.55	263.06	39	10580.08	778.69	276.24	-4.77
w_2d_1	6466.96	14	15166.28	1116.24	395.99	33	15400.13	1133.45	402.09	-1.52
w_2d_2	3402.77	10	7983.10	587.56	208.44	34	7969.26	586.54	208.07	0.17
w_2d_3	4661.77	12	10910.65	803.02	284.87	54	11128.57	819.06	290.56	-1.96
w_2d_4	3832.75	11	9021.43	663.98	235.55	38	9126.58	671.72	238.29	-1.15
w_2d_5	6318.48	15	14803.72	1089.55	386.52	81	14847.04	1092.74	387.65	-0.29
w_3d_1	5422.55	13	12705.61	935.13	331.74	23	12782.41	940.79	333.74	-0.60
w_3d_2	4223.01	13	9905.50	729.04	258.63	50	9678.40	712.33	252.70	2.35
w_3d_3	5593.45	12	13074.27	962.27	341.36	27	13330.59	981.13	348.06	-1.92
w_3d_4	3359.40	9	7875.72	579.65	205.63	16	7897.77	581.28	206.20	-0.28
w_3d_5	3858.41	10	9034.79	664.96	235.90	25	8997.78	662.24	234.93	0.41
w_4d_1	5027.43	13	11808.95	869.14	308.33	35	12095.83	890.25	315.82	-2.37
w_4d_2	3444.04	10	8074.00	594.25	210.81	23	8092.21	595.59	211.28	-0.22
w_4d_3	3622.06	9	8489.77	624.85	221.66	14	8586.17	631.94	224.18	-1.12
w_4d_4	4092.52	12	9594.97	706.19	250.52	33	9521.35	700.77	248.60	0.77
w_4d_5	5037.8	13	11802.06	868.63	308.15	53	11949.69	879.50	312.00	-1.24
Moyenne	4503.41	11.65	10560.08	777.22	275.72	36.10	10659.53	784.54	278.32	-0.71

Table 5.8: Consommation énergétique à vitesse constante de 80 km/h.

Données (Set)	Dist. (km)	Véhic. Util.	Énergie (MJ)	CO_2 (Kg)	Carburant (L)	CPU (s)	Énergie réelle (MJ)	CO_2 réel (Kg)	Carburant réel (L)	Écart (%)
w_1d_1	5638.27	12	14494.39	1066.79	378.44	53	13500.26	993.62	352.49	7.36
w_1d_2	2783.87	9	7169.01	527.64	187.18	32	6537.91	481.19	170.70	9.65
w_1d_3	5755.86	14	14746.59	1085.35	385.03	37	13744.99	1011.63	358.88	7.29
w_1d_4	3217.73	11	8278.39	609.29	216.15	26	7668.76	564.42	200.28	7.95
w_1d_5	4401.29	12	11309.85	832.41	295.30	36	10473.98	770.88	273.47	7.98
w_2d_1	6666.64	15	17106.44	1259.03	446.64	34	15913.02	1171.20	415.48	7.50
w_2d_2	3461.20	11	8884.82	653.92	231.98	42	8144.48	599.43	212.65	9.09
w_2d_3	4694.28	12	12022.79	884.87	313.91	52	11131.76	819.30	290.65	8.00
w_2d_4	3994.40	12	10289.88	757.34	268.67	36	9343.91	687.71	243.97	10.12
w_2d_5	6359.29	16	16305.50	1200.08	425.73	80	14447.91	1063.37	377.23	12.86
w_3d_1	5479.71	13	14048.39	1033.96	366.80	33	12862.23	946.66	335.83	9.22
w_3d_2	4216.18	12	10827.00	796.87	282.69	47	9957.02	732.84	259.97	8.74
w_3d_3	5539.75	11	14174.27	1043.23	370.09	29	13211.42	972.36	344.95	7.29
w_3d_4	3302.28	9	8471.18	623.48	221.18	15	7794.55	573.68	203.51	8.68
w_3d_5	3861.92	10	9895.98	728.34	258.38	22	9115.11	670.87	237.99	8.57
w_4d_1	5108.06	13	13126.64	966.12	342.73	30	12111.83	891.43	316.24	8.38
w_4d_2	3443.30	10	8836.64	650.38	230.72	22	8016.82	590.04	209.32	10.23
w_4d_3	3615.51	9	9275.60	682.68	242.18	14	8508.41	626.22	222.15	9.02
w_4d_4	4029.88	11	10339.62	761.00	269.96	30	9570.52	704.39	249.88	8.04
w_4d_5	5061.95	14	12979.81	955.31	338.90	41	11998.52	883.09	313.28	8.18
Moyenne	4531.57	11.80	11629.14	855.90	303.63	35.55	10702.67	787.72	279.45	8.71

Table 5.9: Consommation énergétique à vitesse constante de 100 km/h.

Données (Set)	Dist. (km)	Véhic. Util.	Énergie (MJ)	CO_2 (Kg)	Carburant (L)	CPU (s)	Énergie réelle (MJ)	CO_2 réel (Kg)	Carburant réel (L)	Écart (%)
w_1d_1	5646.87	12	15640.35	1151.13	408.36	52	13434.97	988.81	350.78	16.42
w_1d_2	2821.28	9	7823.56	575.81	204.27	32	6651.44	489.55	173.67	17.62
w_1d_3	5772.04	14	15932.02	1172.60	415.98	36	13765.96	1013.17	359.42	15.74
w_1d_4	3203.10	10	8882.53	653.75	231.92	23	7576.33	557.62	197.82	17.24
w_1d_5	4290.09	12	11881.63	874.49	310.23	34	10578.36	778.57	276.20	12.32
w_2d_1	6247.80	14	17286.02	1272.25	451.33	33	14863.36	1093.94	388.08	16.30
w_2d_2	3443.10	11	9521.98	700.82	248.62	36	7952.05	585.27	207.63	19.74
w_2d_3	4644.49	12	12820.13	943.56	334.73	49	11049.71	813.26	288.50	16.02
w_2d_4	3904.82	11	10841.24	797.92	283.06	35	9032.30	664.78	235.83	20.03
w_2d_5	6334.36	16	17501.08	1288.08	456.95	88	14887.27	1095.70	388.70	17.56
w_3d_1	5426.04	13	14992.27	1103.43	391.44	26	12788.83	941.26	333.91	17.23
w_3d_2	4131.23	13	11430.76	841.30	298.45	49	9498.48	699.09	248.00	20.34
w_3d_3	5698.75	13	15709.08	1156.19	410.16	34	13486.00	992.57	352.11	16.48
w_3d_4	3345.07	9	9247.55	680.62	241.45	15	7872.77	579.44	205.56	17.46
w_3d_5	3875.85	10	10701.97	787.67	279.42	23	9130.76	672.02	238.40	17.21
w_4d_1	5043.38	13	13967.21	1027.99	364.68	31	11932.35	878.22	311.55	17.05
w_4d_2	3448.26	10	9535.22	701.79	248.96	24	7982.79	587.53	208.43	19.45
w_4d_3	3615.53	9	9995.48	735.67	260.98	14	8509.64	626.31	222.18	17.46
w_4d_4	4136.49	12	11437.47	841.80	298.63	30	9891.91	728.04	258.27	15.63
w_4d_5	5087.65	13	14055.24	1034.47	366.98	64	11918.60	877.21	311.19	17.93
Moyenne	4505.81	11.80	11960.14	917.07	325.33	36.40	10640.19	783.12	272.81	17.26

Afin d'évaluer l'impact de la vitesse sur la prédiction de la consommation d'énergie et des émissions, nous analysons les variations des résultats à différentes vitesses, en utilisant à la fois les données issues des services de Google Maps et des vitesses fixes dans l'étude. La vitesse obtenue via les services de Google Maps est désignée comme *la vitesse réelle* dans notre analyse.

Les résultats obtenus pour des vitesses constantes le long des trajets sont présentés dans les tableaux 5.6-5.9 et illustrés dans les figures 5.3 (a)-(d). Dans cette expérience, chaque solution proposée est générée en supposant une vitesse constante sur tout le trajet, simulant ainsi un scénario d'optimisation avec vitesse fixe. Les résultats de l'algorithme sont directement liés à la fonction objective, où la vitesse est l'un des paramètres influents. Cette approche diffère de l'approche consistant à proposer une seule solution optimisée et à en déduire ensuite les émissions correspondantes pour différentes vitesses constantes.

Pour quantifier la différence entre les émissions obtenues, nous calculons l'écart relatif entre les émissions de CO_2 estimées avec la vitesse réelle et celles obtenues avec une vitesse constante. Cet écart est défini comme la différence entre les émissions réelles de CO_2 et les émissions calculées avec vitesse constante, divisée par ces dernières, puis multipliée par 100 afin d'exprimer la variation en pourcentage.

L'analyse intégrée des quatre tableaux ainsi que des données issues des services Google Maps permet de dégager plusieurs constats pertinents sur l'impact de la vitesse sur la consommation d'énergie et les émissions de CO_2 . À 40 km/h, les résultats montrent une variabilité importante avec une déviation moyenne de 12,13%, oscillant entre 1,21% et 14,45% (Tableau 5.6). Cette variabilité élevée peut s'expliquer par les caractéristiques particulières de la circulation à basse vitesse, telles que les arrêts fréquents et les redé-

marrages, qui introduisent une incertitude plus grande dans la modélisation des consommations énergétiques et des émissions. Ces fluctuations, observées quotidiennement (cf. Figure 5.3 (a)), soulignent la sensibilité du modèle aux conditions de circulation instables à basse vitesse.

En comparaison, à 60 km/h, la précision des prédictions s'améliore considérablement. Le Tableau 5.7 indique des déviations beaucoup plus faibles, avec une moyenne de 0,71% et des extrêmes allant de 0,01% à 4,77%. Ce faible écart suggère que la modélisation est mieux calibrée pour des vitesses modérées, où les conditions de circulation sont généralement plus stables et prévisibles. Par exemple, dans l'ensemble de données w_1d_5 , bien qu'une différence d'émission de 37,14 kg ait été relevée, la Figure 5.3 (b) montre que la vitesse fixe et la vitesse réelle sont pratiquement identiques, ce qui renforce la fiabilité du modèle à cette vitesse.

Lorsque l'on examine des vitesses plus élevées, à 80 km/h et 100 km/h, les écarts entre les valeurs théoriques et réelles deviennent significativement plus prononcés. À 80 km/h (Tableau 5.8), les déviations varient de 7,29% à 12,86% avec une moyenne de 8,71%, tandis qu'à 100 km/h (Tableau 5.9), elles s'étendent de 12,32% à 20,34%, avec une moyenne de 17,26%. En particulier, l'ensemble de données w_3d_2 à 100 km/h présente une différence d'émission remarquable de 142,24 kg, dépassant le seuil critique de 20%. Les Figures 5.3 (c) et (d) illustrent clairement ces écarts importants, mettant en lumière la dégradation de la précision du modèle à mesure que la vitesse s'éloigne de la valeur optimale.

Les données de vitesse réelle obtenues via Google Maps (Tableau 5.2) offrent une dimension complémentaire essentielle à cette étude. En superposant ces résultats aux valeurs fixes des vitesses de 40, 60, 80 et 100 km/h dans la Figure 5.4, on constate que les émissions calculées à 60 km/h se rapprochent étroitement des émissions réelles. Cependant, lorsque la vitesse s'écarte de cette valeur, les écarts deviennent significatifs, avec une augmentation ou une diminution de la vitesse conduisant à des variations proportionnelles, positives ou négatives, respectivement. Ce constat suggère que la modélisation des émissions et de la consommation énergétique est optimisée autour de 60 km/h, alors qu'elle perd en précision pour des vitesses inférieures ou supérieures.

Dans l'ensemble, ces résultats soulignent une relation directe entre la vitesse fixe utilisée dans la modélisation et la précision des estimations énergétiques et environnementales. La concordance maximale à 60 km/h, et la détérioration progressive des estimations lorsque la vitesse s'éloigne de cette valeur, démontrent l'importance d'intégrer des données de vitesse réelle dans les modèles de simulation. Pour des applications pratiques telles que l'optimisation des itinéraires et la réduction de l'impact environnemental, il apparaît indispensable d'ajuster les modèles afin de prendre en compte les variations de vitesse réelles, notamment en s'appuyant sur des services comme Google Maps. L'amélioration de la calibration des modèles selon différentes plages de vitesse représente ainsi une voie prometteuse pour obtenir des prédictions plus fiables et guider des politiques de mobilité

durable.

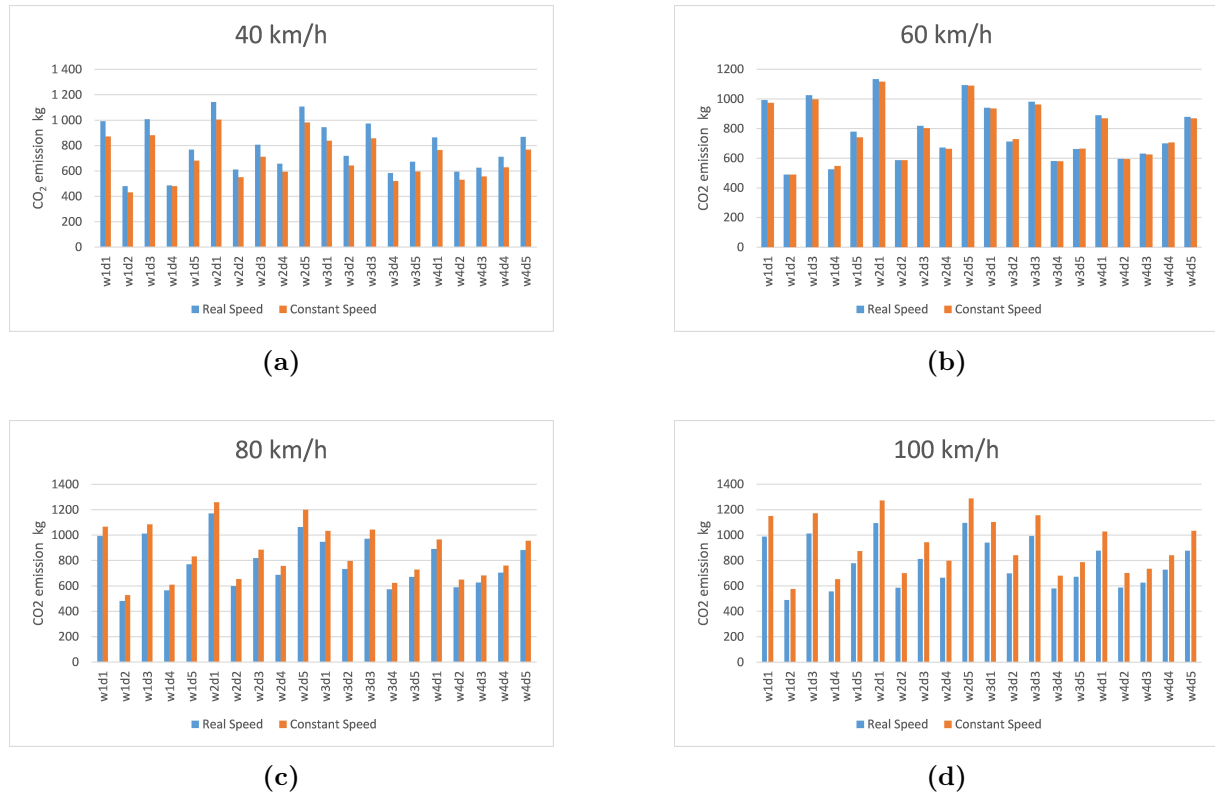


Figure 5.3: CO₂ emissions for Actual Speed vs. Fixed Speed. (a) Fixed Speed at 40 km/h. (b) Fixed Speed at 60 km/h. (c) Fixed Speed at 80 km/h. (d) Fixed Speed at 100 km/h.

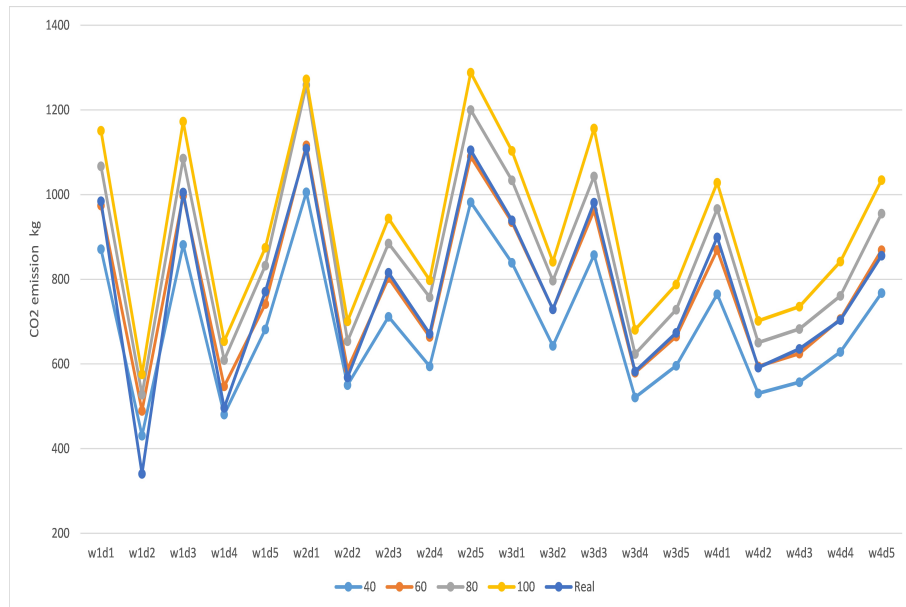


Figure 5.4: CO₂ emissions for real speed vs constant speeds conditions.

5.5 Analyse statistique et validation

Une analyse de variance unidirectionnelle (ANOVA à un facteur) a été réalisée afin de valider les différences observées en matière d'émissions et de consommation d'énergie entre différents profils de vitesse. L'ANOVA à un facteur est une méthode statistique permettant de déterminer s'il existe des différences significatives entre les moyennes de plusieurs groupes. Dans notre étude, les profils de vitesse constituent ces groupes comparés.

En appliquant cette méthode, nous cherchons à établir si les variations observées des émissions et de la consommation d'énergie sont attribuables aux changements de profil de vitesse ou si elles résultent simplement d'une variabilité aléatoire. Un test ANOVA est considéré comme statistiquement significatif lorsque la valeur de p est inférieure à 0,05 et que la statistique F est suffisamment élevée, indiquant ainsi une hétérogénéité marquée entre les groupes. L'adoption de cette approche rigoureuse permet non seulement de mettre en évidence les différences observées, mais aussi de démontrer leur robustesse statistique.

Afin d'examiner si la moyenne des émissions et de la consommation d'énergie varie de manière significative entre les quatre profils de vitesse étudiés (40 km/h, 60 km/h, 80 km/h et 100 km/h), nous procédons à une analyse de variance unidirectionnelle. Cette analyse permet d'évaluer l'impact du facteur vitesse sur la consommation énergétique et les émissions polluantes, contribuant ainsi à une meilleure compréhension des mécanismes influençant l'efficacité énergétique et l'empreinte environnementale du transport.

- Hypothèse nulle (H_0) : Il n'existe aucune différence significative dans les émissions ou la consommation d'énergie entre les profils de vitesse.
- Hypothèse alternative (H_1) : Il existe une différence significative dans les émissions ou la consommation d'énergie entre au moins deux profils de vitesse.

Les probabilités associées aux valeurs p :

- Si la valeur $p < 0,05$, nous rejetons l'hypothèse nulle et concluons qu'il existe des différences significatives dans les émissions entre les profils de vitesse.
- Si la valeur $p \geq 0,05$, nous ne rejetons pas l'hypothèse nulle, indiquant l'absence de différences significatives.

Table 5.10: Comparaison statistique des émissions de CO_2 et de la consommation d'énergie selon les profils de vitesse.

Paramètre	F-statistique	P-value
Émissions de CO_2 (kg)	4.75	0.0043
Consommation d'énergie (MJ)	3.41	0.0216

Les résultats obtenus, présentés dans le Tableau 5.10, montrent une F-statistique de 4,75 et une p-valeur de 0,0043 pour les émissions de CO_2 et une F-statistique de 3,41 avec une p-valeur de 0,0216 pour la consommation d'énergie. Ces résultats indiquent clairement qu'il existe des différences significatives entre les profils de vitesse, confirmant ainsi le rôle déterminant de la vitesse dans l'optimisation énergétique et la réduction des impacts environnementaux. En effet, les variations observées démontrent que des vitesses plus élevées tendent généralement à augmenter les émissions et la consommation énergétique, soulignant ainsi la nécessité d'une gestion optimale de la vitesse pour minimiser l'empreinte écologique des véhicules.

Afin d'évaluer la robustesse de l'algorithme proposé, nous avons réalisé dix exécutions indépendantes en utilisant différentes graines pour le générateur de nombres aléatoires, appliquées à des vitesses réelles, afin d'examiner la stabilité de notre approche non déterministe. L'analyse des résultats a été menée en calculant la moyenne et l'écart type (**S D**) de la fonction objectif, qui correspond à l'énergie consommée, sur l'ensemble de ces exécutions. Les tableaux 5.11 à 5.14 présentent, pour chacune des semaines, des statistiques détaillées incluant la distance parcourue, le nombre de véhicules utilisés, la consommation d'énergie, les émissions de CO_2 , la consommation de carburant ainsi que le temps de calcul CPU.

Les pourcentages d'écart type (% **S D**), variant entre 1,06 % et 4,04 %, mettent en évidence la stabilité de la méthode dans l'optimisation de la consommation énergétique. Ces variations relativement faibles témoignent de la cohérence des performances obtenues, renforçant ainsi la fiabilité de l'approche proposée face à des conditions de simulation variées. Cette robustesse est particulièrement importante dans un contexte où l'optimisation des trajectoires et des vitesses joue un rôle clé dans la réduction des émissions polluantes et l'amélioration de l'efficacité énergétique.

En conclusion, ces analyses statistiques révèlent non seulement que la vitesse influe de manière significative sur les émissions et la consommation énergétique, mais également que l'algorithme utilisé est robuste et fiable dans son optimisation, comme en témoignent les faibles variations relatives observées entre les différentes exécutions. Cette approche rigoureuse, combinant des tests d'hypothèses via l'ANOVA et une évaluation approfondie de la performance algorithmique, offre une validation solide des résultats obtenus. De plus, l'intégration de données de vitesses réelles, par le biais des services Google Maps, permet d'enrichir la modélisation et d'affiner les estimations pour une meilleure prise en compte des conditions de circulation réelles.

Ainsi, l'ensemble de ces analyses souligne l'importance de la vitesse comme facteur clé influençant la consommation énergétique et les émissions polluantes, tout en attestant de la fiabilité de l'approche méthodologique proposée pour l'optimisation de ces paramètres dans le cadre de politiques de mobilité durable.

Table 5.11: Statistiques de performance pour la première semaine.

Jeu de données	Statistiques	Distance	Nombre de véhicules utilisés (km)	Énergie consommée (MJ)	CO ₂ Émissions (kg)	Consommation de carburant (L)	Temps CPU (s)
w_1d_1	Moyenne	5761.03	12.50	13427.78	988.28	350.59	69.70
	Écart-type	119.16	0.50	277.74	20.44	7.25	3.63
	% Écart-type	2.07	4.00	2.07	2.07	2.07	5.21
w_1d_2	Moyenne	2898.26	9.20	6755.25	497.18	176.37	42.80
	Écart-type	99.81	0.42	232.64	17.12	6.07	1.75
	% Écart-type	3.44	4.58	3.44	3.44	3.44	4.09
w_1d_3	Moyenne	5858.84	14.60	13655.77	1005.06	356.54	58.90
	Écart-type	82.36	0.92	191.97	14.13	5.01	2.66
	% Écart-type	1.41	6.28	1.41	1.41	1.41	4.52
w_1d_4	Moyenne	3400.53	11.10	7925.94	583.35	206.94	38.60
	Écart-type	125.09	0.70	291.56	21.46	7.61	2.46
	% Écart-type	3.68	6.31	3.68	3.68	3.68	6.37
w_1d_5	Moyenne	4673.25	14.20	10892.38	333.51	284.40	54.10
	Écart-type	183.96	0.75	428.77	142.90	11.19	4.09
	% Écart-type	3.94	5.27	3.94	3.94	3.94	7.55

Table 5.12: Statistiques de performance pour la deuxième semaine.

Jeu de données	Statistiques	Distance	Nombre de véhicules utilisés (km)	Énergie consommée (MJ)	CO ₂ Émissions (kg)	Consommation de carburant (L)	Temps CPU (s)
w_2d_1	Moyenne	6823.90	15.30	15905.12	1170.62	415.28	50.70
	Écart-type	267.10	1.35	622.56	45.82	16.25	4.82
	% Écart-type	3.91	8.79	3.91	3.91	3.91	9.50
w_2d_2	Moyenne	3571.91	11.10	8325.39	612.74	217.37	53.70
	Écart-type	75.61	0.70	176.25	12.97	4.60	3.16
	% Écart-type	2.12	6.31	2.12	2.12	2.12	5.89
w_2d_3	Moyenne	4772.45	12.30	11123.61	818.69	290.43	74.60
	Écart-type	50.36	0.46	117.37	8.64	3.07	4.92
	% Écart-type	1.06	3.73	1.06	1.06	1.06	6.60
w_2d_4	Moyenne	3864.91	10.90	9008.33	663.01	235.20	57.00
	Écart-type	109.94	0.54	256.24	18.86	6.69	1.90
	% Écart-type	2.84	4.94	2.84	2.84	2.84	3.33
w_2d_5	Moyenne	6629.84	17.00	15452.80	1137.32	403.46	129.00
	Écart-type	138.77	0.89	323.44	23.81	8.45	9.82
	% Écart-type	2.09	5.26	2.09	2.09	2.09	7.61

Table 5.13: Statistiques de performance pour la troisième semaine.

Jeu de données	Statistiques	Distance	Nombre de véhicules utilisés (km)	Énergie consommée (MJ)	CO ₂ Émissions (kg)	Consommation de carburant (L)	Temps CPU (s)
w_3d_1	Moyenne	5536.61	13.30	12904.70	949.78	336.93	35.00
	Écart-type	100.16	0.64	233.44	17.18	6.09	1.41
	% Écart-type	1.81	4.81	4.04	1.81	1.81	4.04
w_3d_2	Moyenne	4251.63	13.20	9909.69	729.35	305.21	72.30
	Écart-type	102.76	0.98	239.52	17.63	138.50	4.65
	% Écart-type	2.42	7.42	2.42	2.42	2.42	6.43
w_3d_3	Moyenne	5695.89	12.40	13275.97	977.11	346.63	36.30
	Écart-type	108.42	0.49	252.70	18.60	6.60	2.41
	% Écart-type	1.90	3.95	1.90	1.90	1.90	6.64
w_3d_4	Moyenne	3450.15	9.90	8041.59	591.86	209.96	21.20
	Écart-type	89.10	0.70	207.67	15.28	5.42	1.40
	% Écart-type	2.58	7.07	2.58	2.58	2.58	6.60
w_3d_5	Moyenne	3976.47	9.90	9268.35	682.15	241.99	31.60
	Écart-type	68.72	0.54	160.18	11.79	4.18	1.91
	% Écart-type	1.73	5.44	1.73	1.73	1.73	6.04

Table 5.14: Statistiques de performance pour la quatrième semaine.

Jeu de données	Statistiques	Distance	Nombre de véhicules utilisés km	Énergie consommée MJ	CO ₂ Émissions kg	Consommation de carburant L	Temps CPU s
w_4d_1	Moyenne	5166.98	13.40	12043.18	886.38	314.44	49.90
	Écart-type	205.11	0.80	478.07	35.19	12.48	1.58
	% Écart-type	3.97	5.97	3.16	3.97	3.97	3.16
w_4d_2	Moyenne	3506.75	10.40	8173.53	601.57	213.40	31.00
	Écart-type	35.05	0.49	81.69	6.01	2.13	1.84
	% Écart-type	1.00	4.71	1.00	1.00	1.00	5.95
w_4d_3	Moyenne	3666.00	9.50	8544.70	628.89	223.10	20.70
	Écart-type	48.59	0.50	113.26	8.34	2.96	1.10
	% Écart-type	1.33	5.26	1.33	1.33	1.33	5.31
w_4d_4	Moyenne	4123.14	11.80	9610.20	707.31	250.91	44.20
	Écart-type	98.42	0.75	229.39	16.88	5.99	3.74
	% Écart-type	2.39	6.34	2.39	2.39	2.39	8.45
w_4d_5	Moyenne	5186.20	15.20	12087.97	889.67	315.61	63.70
	Écart-type	77.98	1.17	181.75	13.38	4.74	2.83
	% Écart-type	1.50	7.67	1.50	1.50	1.50	4.44

Conclusion Générale et Perspectives

Les résultats que nous avons obtenus montrent clairement que la combinaison des données en temps réel obtenues par crowdsourcing, fournies par les services Google Maps, permet un meilleur suivi des dynamiques opérationnelles et énergétiques que les méthodes traditionnelles habituelles en la matière, par exemple, le calcul de la distance par la méthode euclidienne. À l'inverse, la plus grande longueur des trajets peut donner, dans certains contextes, une consommation d'énergie inférieure, d'où l'importance de travailler avec des modèles tenant compte de la multitude des facteurs pouvant influencer. Ces observations renforcent l'idée que la thèse selon laquelle les données crowdsourcées pourraient être un bon instrument d'aide à la décision en informatique verte.

Cette étude, comme toutes les autres, présente des limites et des lacunes, ainsi, l'absence de comparaisons systématiques avec d'autres méthodes d'optimisation ne permet pas, pour l'instant, de conclure à la supériorité de l'approche proposée. De plus, certains facteurs supposés influents comme les variations de topographie ou le comportement des conducteurs n'y sont pas intégrés, limitant la généralisation des résultats. Autant de points qui rendent compte de la complexité des systèmes réels et de la nécessité de poursuivre les efforts de recherche pour produire des modèles de plus en plus robustes et représentatifs.

Il existe de nombreuses pistes de recherche pour l'avenir et de nombreux domaines d'amélioration qui doivent être explorés plus en profondeur. Il conviendrait d'étendre la portée de l'étude en incorporant d'autres variables environnementales et opérationnelles dans le modèle, en particulier dans l'équation objective, telles que le terrain (pente de la route) ou les indicateurs de trafic en temps réel, afin de mieux rendre compte de la diversité des conditions d'exploitation. Par ailleurs, l'exploration de nouvelles approches d'algorithmes d'optimisation métaheuristiques, telles que celles fondées sur un algorithme génétique, SPO, etc., ou sur une combinaison de ces méthodes d'optimisation (mémo-tique, etc..) pourrait améliorer les solutions sélectionnées. Les recherches les plus récentes introduisent actuellement des modèles d'intelligence artificielle dans l'optimisation et la programmation des routes à des fins économiques ou environnementales, et les grands modèles de langage (LLM) sont les plus courants dans cette section, ce qui appelle à réfléchir à l'intégration de l'intelligence artificielle à des fins environnementales, qui est une lacune scientifique très importante pour laquelle nous préconisons fortement de réfléchir à cette question, bien qu'il s'agisse d'un grand défi parce qu'il nécessite des ressources considérables. Enfin, la conception d'une plateforme de simulation en temps réel exploitable dynamiquement grâce aux données fournies par les foules pourrait apporter des vues intéressantes aux décideurs et donc d'optimiser les stratégies de l'information verte et de la logistique durable.

En conclusion, cette thèse contribue de manière significative à la compréhension des interactions entre crowdsourcing et développement durable dans le domaine de l'informatique verte. Les travaux présentés ouvrent la voie à de nouvelles approches permettant d'optimiser

la consommation énergétique tout en prenant en compte la complexité des systèmes réels, et constituent une base solide pour des recherches ultérieures visant à favoriser une transition vers des pratiques plus respectueuses de l'environnement.

Bibliographie

- [1] Bilal Bencharif, Mohamed Amine Beghoura, and Emrah Demir. The volume-based pollution-routing problem with time windows: A case study. *Modelling*, 6:6, 1 2025.
- [2] San Murugesan. Harnessing green it: Principles and practices. *IT professional*, 10(1):24–33, 2008.
- [3] Alemayehu Molla. Organizational motivations for green it: Exploring green it matrix and motivation models. In *the Pacific Asia Conf. on Information Systems*, 2009.
- [4] Jason Dedrick. Green is: Concepts and issues for information systems research. *Communications of the Association for Information Systems*, 27(11):173–184, 2010.
- [5] G. B. Dantzig and J. H. Ramser. The Truck Dispatching Problem. *Management Science*, 6(1):80–91, 1959.
- [6] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.
- [7] Andrew Palmer. *The development of an integrated routing and carbon dioxide emissions model for goods vehicles. Ph.D. Dissertation, School of Management, Cranfield University.* PhD thesis, 2007.
- [8] Toshiaki Kono, Takumi Fushiki, Katsuya Asada, and Kenjiro Nakano. Fuel consumption analysis and prediction model for "ECO" route search. In *15th World Congress on Intelligent Transport Systems and ITS America Annual Meeting 2008*, volume 5, pages 3022–3030, New York NY, United States, 2008.
- [9] Neil Urquhart, Cathy Scott, and Emma Hart. Using an evolutionary algorithm to discover low CO2 tours within a travelling salesman problem. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6025 LNCS, pages 421–430. Springer, Berlin, Heidelberg, 2010.
- [10] Tolga Bektaş and Gilbert Laporte. The Pollution-Routing Problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

- [11] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the Pollution-Routing Problem. *European Journal of Operational Research*, 223(2):346–359, dec 2012.
- [12] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. The bi-objective Pollution-Routing Problem. *European Journal of Operational Research*, 232(3):464–478, 2014.
- [13] Çağrı Koç, Tolga Bektaş, Ola Jabali, and Gilbert Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, dec 2014.
- [14] H. Kargari Esfand Abad, Behnam Vahdani, M. Sharifi, and F. Etebari. A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system. *Journal of Cleaner Production*, 193:784–801, aug 2018.
- [15] Longlong Leng, Jingling Zhang, Chunmiao Zhang, Yanwei Zhao, Wanliang Wang, and Gongfa Li. A novel bi-objective model of cold chain logistics considering location-routing decision and environmental effects. *PLoS ONE*, 15(4):e0230867, 2020.
- [16] Gilberto Tavares, Zdena Zsigraiova, Viriato Semiao, and Maria Da Graça Carvalho. A case study of fuel savings through optimisation of MSW transportation routes. *Management of Environmental Quality: An International Journal*, 19(4):444–454, 2008.
- [17] Zdena Zsigraiová, Gilberto Tavares, Viriato Semiao, and Maria de Graça Carvalho. Integrated waste-to-energy conversion and waste transportation within island communities. *Energy*, 34(5):623–635, 2009.
- [18] Miguel Figliozzi. Vehicle routing problem for emissions minimization. *Transportation Research Record*, 2197(2197):1–7, 2010.
- [19] W. Maden, R. Eglese, and D. Black. Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society*, 61(3):515–522, 2010.
- [20] Erica Wygonik and Anne Goodchild. Using a GIS-Based Emissions Minimization Vehicle Routing Problem with Time Windows (EVRPTW) Model to Evaluate CO2 Emissions and Cost Trade-offs in a Case Study of an Urban Delivery System. Technical report, 2011.
- [21] A. Omidvar and R. Tavakkoli-Moghaddam. Sustainable vehicle routing: Strategies for congestion management and refueling scheduling. In *2012 IEEE International Energy Conference and Exhibition, ENERGYCON 2012*, pages 1089–1094. IEEE, 2012.

- [22] Marius M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problems With Time Window Constraints. *Operations Research*, 35(2):254–265, 1987.
- [23] Meead Saberi and I. Ömer Verbas. Continuous approximation model for the vehicle routing problem for emissions minimization at the strategic level. *Journal of Transportation Engineering*, 138(11):1368–1376, 2012.
- [24] Yiyong Xiao and Abdullah Konak. A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem. *Journal of Cleaner Production*, 167:1450–1463, nov 2017.
- [25] Yiyong Xiao, Xiaorong Zuo, Jiaoying Huang, Abdullah Konak, and Yuchun Xu. The continuous pollution routing problem. *Applied Mathematics and Computation*, 387:125072, dec 2020.
- [26] G. Poonthalir, R. Nadarajan, and M. Senthil Kumar. Hierarchical optimization of green routing for mobile advertisement vehicle. *Journal of Cleaner Production*, 258:120661, jun 2020.
- [27] Houming Fan, Yueguang Zhang, Panjun Tian, Yingchun Lv, and Hao Fan. Time-dependent multi-depot green vehicle routing problem with time windows considering temporal-spatial distance. *Computers and Operations Research*, 129, 5 2021.
- [28] M V Gutiérrez-Padilla, D Morillo-Torres, and G Gatica. A novel mathematical model for a discrete speed pollution routing problem with time windows in a colombian context**this study was financed by the regular project of dgi at the unab. *IFAC-PapersOnLine*, 54:229–235, 2021. 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021.
- [29] Joydeep Dutta, Partha Sarathi Barma, Anupam Mukherjee, Samarjit Kar, Tanmay De, Dragan Pamučar, Šarūnas Šukevičius, and Giedrius Garbinčius. Multi-objective green mixed vehicle routing problem under rough environment. *Transport*, 37(1):51–63, 2022.
- [30] Zhiying Yao, Haiqing Cao, Zhenliang Cui, Yuru Wang, and Ning Huang. Research on urban distribution routes considering the impact of vehicle speed on carbon emissions. *Sustainability (Switzerland)*, 14, 12 2022.
- [31] Yuhe Shi and Yun Lin. Bi-objective optimization for multi-depot green vehicle routing problem with time windows considering carbon emission. *Journal of Physics: Conference Series*, 2583:012008, 9 2023.
- [32] Heng Wang, Menghan Li, Zhenyu Wang, Wei Li, Tianjiao Hou, Xianyi Yang, Zhenzhen Zhao, Zhenfeng Wang, and Tong Sun. Heterogeneous fleets for green vehicle

- routing problem with traffic restrictions. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):8667–8676, 2023.
- [33] Yiming Liu, Baldacci Roberto, Jianwen Zhou, Yang Yu, Yu Zhang, and Wei Sun. Efficient feasibility checks and an adaptive large neighborhood search algorithm for the time-dependent green vehicle routing problem with time windows. *European Journal of Operational Research*, 310:133–155, 2023.
- [34] Zhuowu Zhang, Emrah Demir, Robert Mason, and Carla Di Cairano-Gilfedder. Understanding freight drivers’ behavior and the impact on vehicles’ fuel consumption and co2e emissions. *Operational Research*, 23:59, 2023.
- [35] Cristian Cataldo-Díaz, Rodrigo Linfati, and John Willmer Escobar. Mathematical models for the electric vehicle routing problem with time windows considering different aspects of the charging process. *Operational Research*, 24:1, 2023.
- [36] Ping Lou, Zikang Zhou, Yuhang Zeng, and Chuannian Fan. Vehicle routing problem with time windows and carbon emissions: a case study in logistics distribution. *Environmental Science and Pollution Research*, 2024.
- [37] Kamyla Maria Ferreira, Thiago Alves de Queiroz, Pedro Munari, and Franklina Maria Bragion Toledo. A variable neighborhood search for the green vehicle routing problem with two-dimensional loading constraints and split delivery. *European Journal of Operational Research*, 2024.
- [38] Konstantinos Gkyrtis. Theoretical considerations from the modelling of the interaction between road design and fuel consumption on urban and suburban roadways. *Modelling*, 5:737–751, 6 2024.
- [39] Simona Mancini. The Hybrid Vehicle Routing Problem. *Transportation Research Part C: Emerging Technologies*, 78:1–12, may 2017.
- [40] Giusy Macrina, Gilbert Laporte, Francesca Guerriero, and Luigi Di Puglia Pugliese. An energy-efficient green-vehicle routing problem with mixed vehicle fleet, partial battery recharging and time windows. *European Journal of Operational Research*, 276(3):971–982, aug 2019.
- [41] Neda Manavizadeh, Hamed Farrokhi-Asl, and Stanley Frederick W.T. Lim. A New Mathematical Model for the Green Vehicle Routing Problem by Considering a Bi-Fuel Mixed Vehicle Fleet. *Journal of Optimization in Industrial Engineering*, 13(2):165–183, jul 2020.

- [42] Imdat Kara, Bahar Y. Kara, and M. Kadri Yetis. Energy minimizing vehicle routing problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4616 LNCS:62–71, 2007.
- [43] Yiyong Xiao, QiuHong Zhao, Ikou Kaku, and Yuchun Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers and Operations Research*, 39(7):1419–1431, 2012.
- [44] Emrah Demir, Wolfgang Burgholzer, Martin Hrušovský, Emel Arıkan, Werner Jammerneegg, and Tom Van Woensel. A green intermodal service network design problem with travel time uncertainty. *Transportation Research Part B: Methodological*, 93:789–807, nov 2016.
- [45] András Király and János Abonyi. A Google Maps based novel approach to the optimization of multiple Traveling Salesman problem for limited distribution systems. *Acta Agraria Kaposváriensis*, 14(3):1–14, feb 2010.
- [46] Siyuan Liu and Qiang Qu. Dynamic collective routing using crowdsourcing data. *Transportation Research Part B: Methodological*, 93:450–469, 2016.
- [47] Darmawan Satyananda and Riwayat Artikel. Google Map API service for VRP Solving Application. *Seminar Nasional Integrasi Matematika dan Nilai Islami*, 1(1):240–245, jul 2017.
- [48] Sakitha Kumarage. *Use of Crowdsourced Travel Time Data in Traffic Engineering Applications*. PhD thesis, 2018.
- [49] S Ardekani and Ezra Hauer. Traffic impact models. *Chapter 7 in Traffic Flow Theory*,, 1992.
- [50] R. Eglese and D. Black. Optimizing the routing of vehicles. *Green Logistics: Improving the Environmental Sustainability of Logistics*, edited by McKinnon, A., Cullinane, S., Browne, M., Whiteing, A., Kogan Page Ltd., pages 215–228, 2010.
- [51] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357, 2011.
- [52] Bilal Bencharif, Mohamed Amine Beghouara, and Emrah Demir. A simple tabu search with crowdsourcing data using google services api for green logistic: A case study. *7th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, Laghouat, Algeria, pages 0–0, 2025.
- [53] Michael Ben Chaim and Efraim Shmerling. A model of vehicle fuel consumption at conditions of the EUDC. *International Journal of Mechanics*, 7(1):10–17, 2013.

- [54] Lino Guzzella and Antonio Sciarretta. Vehicle Propulsion Systems. *Vehicle Propulsion Systems*, 2013.
- [55] Michael Ben-Chaim, Efraim Shmerling, and Alon Kuperman. Analytic modeling of vehicle fuel consumption. *Energies*, 6(1):117–127, 2013.
- [56] Fred Glover. Tabu search—part I. *ORSA Journal on computing*, 1(3):190–206, 1989.
- [57] Maha Gmira, Michel Gendreau, Andrea Lodi, and Jean Yves Potvin. Tabu search for the time-dependent vehicle routing problem with time windows on a road network. *European Journal of Operational Research*, 288(1):129–140, jan 2021.
- [58] Xiongxiang Liao and Yongyun Shao. Optimization algorithm for the time-dependent vehicle routing problem with softtime windows: A research based on mixed integer model and mixed tabu search. *Mathematical Foundations of Computing*, pages 0–0, 2024.
- [59] Takwa Tlili, Sirine Ben Nasser, Francisco Chicano, and Saoussen Krichen. Tabu search-based hyper-heuristic for solving the heterogeneous ambulance routing problem with time windows. *International Journal of Intelligent Transportation Systems Research*, 22(2):446–461, 2024.
- [60] Google developers. Get Started | Distance Matrix API | Google Developers, 2020, Accessed May 23, 2022.
- [61] Michel Gendreau. An Introduction to Tabu Search. In *Handbook of Metaheuristics*, pages 37–54. Kluwer Academic Publishers, feb 2006.
- [62] Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186, 1997.
- [63] Olli Bräysy. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.
- [64] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 2006.
- [65] Daren C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75–90, 2008.
- [66] Alexander J. Quinn and Benjamin B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, page 1403–1412, New York, NY, USA, 2011. Association for Computing Machinery.

- [67] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, page 1301–1318, New York, NY, USA, 2013. Association for Computing Machinery.
- [68] J. Goodman and Gabriele Paolacci. Crowdsourcing consumer research. *Journal of Consumer Research*, 44:196–210, 2017.
- [69] Kerri Wazny. “crowdsourcing” ten years in: A review. *Journal of Global Health*, 7, 2017.
- [70] Kerri Wazny. Applications of crowdsourcing in health: an overview. *Journal of Global Health*, 8, 2018.
- [71] C. Parvanta, Yannig Roth, and H. Keller. Crowdsourcing 101. *Health Promotion Practice*, 14:163 – 167, 2013.
- [72] Larissa Hammon and Hajo Hippner. Crowdsourcing. *Business Information Systems Engineering*, 4:163 – 166, 2012.
- [73] Allan Afuah and Christopher L. Tucci. Crowdsourcing as a solution to distant search. *Academy of Management Review*, 37:355–375, 2012.
- [74] Fábio R. Assis Neto and Celso A. S. Santos. Understanding crowdsourcing projects: A systematic review of tendencies, workflow, and quality management. *Inf. Process. Manag.*, 54:490–506, 2018.
- [75] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and C. Shahabi. Spatial crowdsourcing: a survey. *The VLDB Journal*, 29:217 – 250, 2019.
- [76] Ella Segev. Crowdsourcing contests. *Eur. J. Oper. Res.*, 281:241–255, 2020.
- [77] Tapani Jokiniemi. Greenhouse gas emissions from Direct combustion of various fuels. Technical report, 2011.

Annexe

Annexe 1: Données utilisées dans l'étude

Les données utilisées dans le cadre de cette étude sont disponibles en ligne et peuvent être téléchargées à partir du lien suivant :

- **CustomersData.xls** : fichier contenant les informations relatives aux clients (coordonnées, fenêtres temporelles, volumes, etc.).
- **DistanceTimeMatrix.xls** : matrice contenant les distances et les temps de parcours entre les différents points du réseau.

Lien de téléchargement :

<https://github.com/Bilal-df/ResearchData>

Annexe 2: Génération automatique des matrices de distance et de temps

Cette annexe présente le code Java développé pour interroger l'API Google Distance Matrix et générer automatiquement deux matrices : l'une pour les distances (en mètres), et l'autre pour les temps de parcours (en secondes), entre un ensemble de points (clients, dépôts, etc.). Le code permet également l'exportation des matrices dans un fichier **.xls** lisible par Excel, à l'aide de la bibliothèque **Apache POI**.

Fonctionnalités principales du code

- **Initialisation du contexte Google Maps API** : connexion sécurisée via une clé API.
- **Requête à Distance Matrix API** : obtention des distances et des temps entre des points spécifiés.

- **Traitement des réponses** : les distances (en mètres) et durées (en secondes) sont extraites directement à partir des objets Java fournis par la bibliothèque Google Maps, sans manipulation directe du format JSON.
- **Construction des matrices** : stockage structuré dans un tableau 3D de type `double[][][]`.
- **Exportation vers Excel** : génération de deux feuilles, l'une pour les distances et l'autre pour les temps.

Bibliothèques utilisées

- Google Maps Java Client : pour les appels à l'API Distance Matrix.
- Apache POI (HSSFWorkbook) : pour la création et l'écriture de fichiers Excel au format `.xls`.

Exemple de points utilisés

Dans l'exemple inclus dans la méthode `main()`, les villes suivantes sont utilisées à titre de démonstration (les coordonnées réelles des clients sont confidentielles):

- Origines : Setif, Jijel
- Destinations : Constantine, Bordj Bou Arreridj

Code source

Le code complet est inclus dans le fichier `CreationDistanceMatrix.java` et peut être adapté selon les besoins de l'utilisateur, notamment en modifiant la liste des adresses ou les paramètres de transport.

F.1 Code source de `CreationDistanceMatrix.java`

```
1 import java.io.DataOutputStream;
2 import java.io.EOFException;
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.ObjectInputStream;
8 import java.io.*;
9
10
```

```
11 import com.lynden.gmapsfx.shapes.Polyline;
12 import com.lynden.gmapsfx.shapes.PolylineOptions;
13 import com.lynden.gmapsfx.javascript.object.MVCArray;
14 import com.lynden.gmapsfx.javascript.object.LatLng;
15
16 import com.google.maps.RoadsApi;
17 import com.google.maps.DirectionsApi.RouteRestriction;
18 import com.google.maps.DirectionsApi;
19 import com.google.maps.DirectionsApiRequest;
20 import com.google.maps.GeocodingApi;
21 import com.google.maps.GeocodingApiRequest;
22 import com.google.maps.GeoApiContext;
23 import com.google.maps.model.DirectionsRoute;
24 import com.google.maps.model.DirectionsResult;
25 import com.google.maps.model.DirectionsStep;
26 import com.google.maps.model.GeocodedWaypoint;
27
28 import com.google.maps.DistanceMatrixApi;
29 import com.google.maps.DistanceMatrixApiRequest;
30 import com.google.maps.GeoApiContext;
31 import com.google.maps.GeocodingApi;
32 import com.google.maps.errors.ApiException;
33 import com.google.maps.model.DistanceMatrix;
34 import com.google.maps.model.ElevationResult;
35 import com.google.maps.ElevationApi;
36 import com.google.maps.model.DirectionsLeg;
37 import com.google.maps.model.EncodedPolyline;
38 import com.google.maps.model.GeocodingResult;
39 import com.google.maps.model.LatLng;
40 import com.google.maps.model.TrafficModel;
41 import com.google.maps.model.TravelMode;
42 import java.util.ArrayList;
43 import java.util.List;
44 import java.util.Vector;
45 import java.net.*;
46 import java.util.concurrent.TimeUnit;
47
48 import org.apache.poi.hssf.usermodel.HSSFCell;
49 import org.apache.poi.hssf.usermodel.HSSFRow;
50 import org.apache.poi.hssf.usermodel.HSSFSheet;
51 import org.apache.poi.hssf.usermodel.HSSFWorkbook;
52 import org.apache.poi.ss.usermodel.Cell;
53 import org.apache.poi.ss.usermodel.CellType;
54 import org.apache.poi.ss.usermodel.Row;
55 import org.apache.poi.ss.usermodel.Workbook;
56 import org.apache.poi.xssf.usermodel.XSSFSheet;
57 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
```

```
58 import org.apache.poi.util.*;
59
60 public class GoogleDistanceMatrix {
61
62     // Cle API Google (remplacer par votre propre cle)
63     private static final String API_KEY = "YOUR_GOOGLE_API_KEY";
64
65     // Fonction pour obtenir la matrice des distances et des temps
66     public double[][][] distanceMatrix(String[] origins, String[]
        destinations) throws Exception {
67         // Initialisation du contexte avec la cle API
68         GeoApiContext context = new GeoApiContext.Builder().apiKey(API_KEY)
            .build();
69
70         // Envoi de la requete a l'API Distance Matrix
71         DistanceMatrix result = DistanceMatrixApi.newRequest(context)
72             .origins(origins)
73             .destinations(destinations)
74             .mode(TravelMode.DRIVING) // Mode de transport : Voiture
75             .await(); // Attendre la reponse
76
77         // Stockage des distances et des durees dans un tableau
78         double[][][] array = new double[origins.length][2][destinations.
            length];
79         for (int i = 0; i < origins.length; i++) {
80             for (int j = 0; j < destinations.length; j++) {
81                 // Distance en metres et duree en secondes
82                 array[i][0][j] = result.rows[i].elements[j].distance.
                    inMeters;
83                 array[i][1][j] = result.rows[i].elements[j].duration.
                    inSeconds;
84             }
85         }
86         return array;
87     }
88
89     // Fonction pour ecrire les resultats dans un fichier Excel
90     public void WriteResultToExcel(double[][] mapsDistanceMatrix, double
        [][] mapsTimeMatrix) {
91         try {
92             // Creation d'un fichier Excel avec Apache POI
93             HSSFWorkbook workbook = new HSSFWorkbook();
94             Sheet sheet1 = workbook.createSheet("Distance Matrix");
95             Sheet sheet2 = workbook.createSheet("Time Matrix");
96
97             // Remplir la feuille de distances
98             for (int i = 0; i < mapsDistanceMatrix.length; i++) {
```

```
99         Row row = sheet1.createRow(i);
100         for (int j = 0; j < mapsDistanceMatrix[i].length; j++) {
101             row.createCell(j).setCellValue(mapsDistanceMatrix[i][j]
102                 );
103         }
104     }
105
106     // Remplir la feuille des temps
107     for (int i = 0; i < mapsTimeMatrix.length; i++) {
108         Row row = sheet2.createRow(i);
109         for (int j = 0; j < mapsTimeMatrix[i].length; j++) {
110             row.createCell(j).setCellValue(mapsTimeMatrix[i][j]);
111         }
112     }
113
114     // Sauvegarde du fichier Excel
115     FileOutputStream out = new FileOutputStream(new File("
116         DistanceTimeMatrix.xls"));
117     workbook.write(out);
118     out.flush();
119     out.close();
120
121     System.out.println("Excel avec les matrices ecrit avec succes")
122     ;
123 } catch (IOException e) {
124     e.printStackTrace();
125 }
126
127 // Exemple d'utilisation
128 public static void main(String[] args) {
129     try {
130         // Definir les origines et destinations
131         String[] origins = {
132             new LatLng(37.7749, -122.4194).toString(), // Exemple :
133             Setif
134             new LatLng(34.0522, -118.2437).toString() // Exemple :
135             Jijel
136         };
137
138         String[] destinations = {
139             new LatLng(36.1699, -115.1398).toString(), // Exemple :
140             Constantine
141             new LatLng(40.7128, -74.0060).toString() // Exemple :
142             Bordj Bou Arreridj
143         };
144     }
```

```
139      // Creation de l'instance de GoogleDistanceMatrix
140      GoogleDistanceMatrix gdm = new GoogleDistanceMatrix();
141
142      // Recuperation des matrices de distances et de temps
143      double[][][] result = gdm.distanceMatrix(origins, destinations)
144          ;
145
146      // Sauvegarde des resultats dans un fichier Excel
147      gdm.WriteResultToExcel(result[0], result[1]);
148
149      } catch (Exception e) {
150          e.printStackTrace();
151      }
152 }
```

SourceCode/CreationDistanceMatrix.java