

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene



FACULTÉ DE MATHÉMATIQUES

Département de recherche opérationnelle

MÉMOIRE

En vue de l'obtention du Diplôme de MASTER

Recherche Opérationnelle - Modèles et méthodes pour l'ingénierie et la recherche (RO2MIR)

Thème

Optimisation du service de livraison express cas : Yalidine El-djazair

Présenté par : AZZOUNE Leila

BENAISSA Hanafi

Soutenu le 04 Juillet 2023 devant le jury composé de :

Président : CHERGUI Mohamed El-Amine, USTHB

Rapporteur : BOUTICHE Mohamed Amine, USTHB

Examinateuse : AHMIA Ibtissam, USTHB

Invitée : RIHANE Yasmine, Ourquiane

Code Mémoire : 02/RO2MIR/2023

Table des matières

1 Transport et Logistique	3
1.1 Introduction	3
1.2 La logistique	3
1.2.1 Définition de la logistique	3
1.2.2 La chaîne logistique globale	3
1.2.3 L'évolution de la chaîne logistique au cours du temps	4
1.3 L'e-commerce	5
1.3.1 L'e-commerce en Algérie	5
1.3.2 Les défis du e-commerce	5
1.3.2.1 Livraison flexible et rapide	5
1.3.2.2 Livraison à domicile	6
1.3.2.3 Logistique inverse	6
1.4 La logistique du dernier kilomètre	6
1.4.1 Définition du dernier kilomètre	6
1.4.2 L'évolution de la logistique du dernier kilomètre au cours du temps	6
1.4.3 Les défis de la logistique du dernier kilomètre	7
1.4.3.1 L'efficacité logistique	7
1.4.3.2 Congestion des routes	7
1.4.3.3 Durabilité environnementale	7
1.4.4 Les solutions pour la logistique du dernier kilomètre	8
1.4.4.1 Stations à proximité ou points relais	8
1.4.4.2 Logistique Urbaine Collaborative et coopérative	8
1.4.4.3 Optimisation dans le dernier kilomètre	8
1.4.4.4 Véhicules innovant	8
1.5 Conclusion	9
2 Présentation du sujet et problématique	10
2.1 Introduction	10
2.2 Intervenant dans le projet	10
2.2.1 Présentation de Yalidine Service El Djazair	10
2.2.1.1 Services fournis par Yalidine	10
2.2.1.2 Hiérarchie des services de Yalidine	11
2.2.1.3 Capacités opérationnelles de l'entreprise	11
2.2.2 Présentation de Ourquilane	12

2.3	Présentation de la problématique	12
2.3.1	Contexte de la problématique	12
2.3.2	Description du problème	13
2.3.2.1	Objectifs	13
2.3.2.2	Contraintes	14
2.3.2.3	Paramètres du problème	14
2.4	Analyse du problème	15
2.4.1	Revue littéraire	15
2.4.1.1	Recherche opérationnelle	15
2.4.1.2	Recherche opérationnelle et Apprentissage automatique	16
2.4.2	Les concurrents internationaux	16
2.5	Conclusion	17
3	Optimisation combinatoire et problème de tournées de véhicules	18
3.1	Introduction	18
3.2	Optimisation combinatoire	18
3.2.1	Optimisation	18
3.2.1.1	Identifier les variables de décision	18
3.2.1.2	Définir les contraintes	18
3.2.1.3	Déterminer la fonction objectif	18
3.2.2	Optimisation combinatoire	19
3.2.2.1	Optimisation locale et globale	19
3.2.3	Programmation linéaire en nombres entiers (PLNE)	19
3.2.3.1	Définition	19
3.2.3.2	Formule mathématique	19
3.2.3.3	Solution d'un programme linéaire en nombre entiers	20
3.2.4	Complexité	20
3.2.4.1	Complexité algorithmique :	20
3.2.4.2	Complexité d'un problème :	20
3.2.4.3	Les classes de complexité d'un problème	20
3.3	Problème de tournées de véhicules	20
3.3.1	Le problème du voyageur de commerce (TSP)	20
3.3.1.1	Formulation mathématique	20
3.3.2	Le problème de tournées de véhicules (VRP)	21
3.3.2.1	Formulation mathématique :	22
3.3.2.2	Domaines d'application	23
3.3.3	Le problème de tournées de véhicules avec clusters (CluVRP)	23
3.3.3.1	Formulation mathématique	24
3.3.4	Approches de résolution du VRP	25
3.3.4.1	Méthodes exactes	25
3.3.4.2	Méthodes approchées	25
3.3.5	Revue littéraire sur la résolution du (TSP)	27
3.4	Conclusion	28

4 Modélisation et résolution du problème	29
4.1 Introducion	29
4.2 Modélisation et résolution du problème de partitionnement	29
4.2.1 Description du problème de partitionnement	29
4.2.1.1 Modèle mathématique	30
4.2.2 Algorithme de K-means	30
4.2.3 Description du problème de partitionnement avec contrainte de capacité	31
4.2.3.1 Modèle mathématique	31
4.2.3.2 Méthode de résolution	32
4.3 Modélisation et résolution du problème de tournées	33
4.3.1 Évaluation du modèle	34
4.3.2 Algorithme de Colonie de fourmis	34
4.3.3 Algorithme de Golden Ball	38
4.4 Conclusion	42
5 Test et analyse des résultats	43
5.1 Introduction	43
5.2 Comparaison des performances des deux métaheuristiques	43
5.2.1 Configuration expérimentale	43
5.2.1.1 Présentation de python	43
5.2.1.2 Matériel utilisé	44
5.2.1.3 Paramètres de ACS	44
5.2.1.4 Paramètres de GB	44
5.2.2 Analyse des résultats	45
5.3 Analyse des résultats du problème traité	46
5.3.1 Instance du problème	46
5.3.2 Retour d'information de l'entreprise sur les résultats	47
5.3.2.1 Réduction du temps des opérations	49
5.3.2.2 Réduction des ressources humaines	49
5.3.2.3 Amélioration du partitionnement des clients	49
5.3.2.4 Amélioration des tournées	49
5.3.2.5 Amélioration des conditions du travail des livreurs	49
5.3.3 Présentation de l'interface graphique	50
5.4 Conclusion	50
6 Perspectives et conclusion	53
A Annexe	54
A.1 Rapport de visite : Compréhension de la logistique de livraison au sein d'une station de Yalidine	54
A.2 Les API's et les bibliothèques utilisés	55
A.2.1 OpenStreetMap	55
A.2.2 GraphHopper	55
A.2.3 Folium	55
A.2.4 Geopy	56
A.2.5 PyQt5	56

A.2.6 Matplotlib	56
----------------------------	----

Table des figures

1.1	Figure illustrant la chaîne logistique globale	4
1.2	Schéma illustrant le processus d'un achat en ligne	5
2.1	Logo de Yalidine	10
2.2	Hiérarchie des services de Yalidine	11
2.3	Logo du bureau de (R&D) Ourquilane	12
2.4	Schéma modélisant le parcours du colis depuis son expédition à l'arrivée à la station de livraison DS : Delivery station	13
2.5	Schéma modélisant le parcours d'un livreur dans le Last-Mile DS : Delivery station	14
2.6	Exemple des tournées d'une station couvrant quatre communes	15
2.7	Décomposition des problèmes depuis une station de livraison donnée	17
3.1	Un exemple d'une tournée du problème du voyageur de commerce	21
3.2	Un exemple montrant les tournées de véhicules	22
3.3	Figure illustrant une tournée de <i>CluVRP</i>	24
4.1	Décomposition d'un ensemble d'objets en trois clusters	29
4.2	Décomposition d'un ensemble d'objets en trois <i>clusters</i> en utilisant l'algorithme K-means	31
4.3	Figure illustrant une tournée de <i>CluVRP</i> adapté au modèle du problème proposé	33
4.4	Colonie de fourmis	35
4.5	Organigramme de la météaheristique colonie de fourmis	35
4.6	Organigramme de la météaheristique système de colonie de fourmis	37
4.7	Organigramme de la météaheristique GB	38
4.8	Illustration de l'espace des solution dans le cas de $NT = 4$	39
5.1	Logo du langage python	43
5.2	Comparaison des distances moyennes obtenues	45
5.3	Comparaison des meilleures distances trouvées	46
5.4	Comparaisons du temps d'exécution moyen	46
5.5	Emplacement des clients	47
5.6	Partitionnement des clients en clusters	48
5.7	Visualisation des tournées dans chaque cluster	48
5.8	Page d'accueil de l'interface graphique	50
5.9	Sélection du fichier Excel contenant les données géographiques des clients	51
5.10	Exemple du fichier Excel	51

5.11 Affichage des résultats	52
--	----

List of Algorithms

1	Algorithme de k-means	31
2	Algorithme de K-means avec capacité	32
3	Pseudocode de la procédure d'une saison	40
4	Pseudo-code d'un processus d'entraînement	41
5	Procédure d'un match	41

Liste des Abréviations

TSP Traveling salesman problem

ACO Ant colony optimization

ACS Ant colony system

GB Golden ball

CluVRP Clustered vehicle routing problem

VNS Variable neighborhood search

PSO Particle swarm optimization

GA Genetic algorithm

DGA Distributed genetic algorithm

SA Simulated Annealing

TS Tabu search

API Application Programming Interface

Remerciements

Au nom de Dieu le tout miséricordieux.

Nous exprimons notre gratitude envers Dieu, le Tout Miséricordieux, pour nous avoir dotés de la passion et de l'enthousiasme nécessaires pour mener à bien ce travail...

Nous exprimons notre sincère gratitude envers l'ensemble du corps universitaire pour leur précieuse assistance et les connaissances inestimables qu'ils nous ont transmises tout au long de notre parcours académique.

Nous tenons à exprimer notre profonde reconnaissance envers monsieur BOUTICHE Mohamed Amine pour son aide par sa présence, sa confiance et la liberté qu'il nous a accordé durant tout ce projet. Nous lui sommes sincèrement reconnaissants pour son engagement et sa collaboration.

Nous remercions Madame RIHANE Yasmine pour son accompagnement, sa confiance en notre travail, sa bienveillance et son accueil chaleureux au sein du bureau d'étude Ourquilane.

Un grand merci à Halima, pour sa présence positive, son aide, sa gentillesse et les moments joyeux qu'on a partagés avec elle durant ce projet.

Nos remerciements pour les membres du jury monsieur CHERGUI Mohamed El-Amine et madame AHMIA Ibtissam pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner.

Nous tenons à adresser des remerciements particuliers à nos familles et amis, pour leurs innombrables moments de soutien passés et à venir.

Résumé

L'e-commerce a connu une expansion rapide ces dernières années, largement stimulée par la pandémie de Covid-19, ce qui a engendré une augmentation significative des livraisons de colis à domicile. Cette évolution rapide a entraîné l'émergence de nouveaux défis logistiques, notamment en ce qui concerne l'optimisation des tournées de livraison afin d'assurer une efficacité maximale. Ce travail propose une approche en "deux phases" pour résoudre le problème de tournées d'une flotte homogène dans le domaine de la logistique du dernier kilomètre en Algérie. Nous utilisons l'algorithme de k-means pour résoudre le problème de partitionnement, ainsi que deux métahéuristiques, ACS (Ant Colony System) et GB (Golden Ball), pour résoudre le problème de tournées, dans le but de comparer ces différentes méthodes et leurs performances.

Introduction

En comparaison avec d'autres pays qui ont connu une croissance plutôt uniforme dans le développement du commerce électronique au cours des dix dernières années, l'Algérie a connu une croissance exceptionnellement élevée. Cette croissance peut être attribuée à l'émergence de la 3G/4G et des smart-phones pendant cette période. L'e-commerce continue de se développer en Algérie et joue un rôle de plus en plus important dans l'économie du pays. Sa croissance fait augmenter les défis liés à la logistique de livraison des produits achetés aux consommateurs. L'acceptation des défis a créé chez les prestataires de services de la livraison une nécessité absolue d'optimiser leurs opérations afin d'achever une meilleure efficience et de satisfaire au mieux les attentes des clients en matière de délais de livraison.

L'optimisation dans le secteur de la livraison est une approche novatrice en Algérie, qui présente un fort potentiel pour l'avenir. De nombreuses nouvelles entreprises émergent dans le marché de la logistique du dernier kilomètre et ne tarderont pas à adopter des processus visant à améliorer leur efficacité. L'optimisation dans le service de la livraison se concentre principalement sur les défis liés au stockage, à la planification, au tri des colis (dispatching) et aux itinéraires.

Dans ce projet, nous utiliserons des concepts et des algorithmes de recherche opérationnelle pour modéliser et optimiser les processus de tri des colis et des itinéraires au sein du bureau de recherche et développement Ourquiane, avec comme étude de cas spécifique : Yalidine El-djazair.

Description du manuscrit

Dans ce manuscrit, le lecteur a accès à une introduction générale, six chapitres et une annexe contenant tous les outils et les documents supplémentaires qui ont contribué à la réussite de ce projet.

Le **premier chapitre** "*transport et logistique*" présente des notions générales sur la logistique de manière globale, en mettant l'accent sur la logistique du dernier kilomètre et le commerce électronique en particulier. Il aborde les défis spécifiques auxquels l'e-commerce et le domaine de la livraison sont confrontés, et les stratégies adoptées à l'échelle globale pour les surmonter.

Le **deuxième chapitre** "*présentation du sujet et problématique*" concerne la présentation de l'entreprise et de la problématique, qui s'ensuit par une revue littéraire mettant l'accent sur les récentes approches abordées pour traiter des problématiques similaires dans les deux domaines confondues, la recherche opérationnelle et l'apprentissage automatique. Cette section sera ensuite suivie par une décomposition du problème en question.

Le **troisième chapitre** "*Optimisation combinatoire et problème de tournées de véhicules*" succédant la présentation du problème, aborde les concepts fondamentaux qui gravitent autour de l'optimisation dans le domaine de la recherche opérationnelle, nous aborderons également les problèmes de tournées de véhicules et de voyageur de commerce, ainsi que les différentes méthodes utilisées pour les résoudre.

Le **quatrième chapitre** "*Modélisation et résolution du problème*" comme son nom l'indique, se concentre sur la modélisation et la résolution des deux problèmes de partitionnement et de tournées.

Le **cinquième chapitre** "*Test et analyse des résultats*" est consacré à la présentation de la solution proposée pour le problème en question, ainsi qu'une rétroaction de l'entreprise sur les résultats obtenus.

Enfin, le **sixième chapitre** "*Perspectives et conclusion*" conclut le projet par une conclusion générale stipulant les accomplissements et les limites de notre travail.

Chapitre 1

Transport et Logistique

1.1 Introduction

Dans ce chapitre, nous allons présenter un aperçu de la logistique, de la chaîne logistique globale, de l'e-commerce et de la logistique du dernier kilomètre, qui représente le sujet principal de notre travail. Nous soulignons les défis spécifiques rencontrés dans le domaine de l'e-commerce et de la livraison, en mettant en évidence l'influence réciproque entre ces deux domaines. Enfin, nous présenterons les solutions et les stratégies adaptées à l'échelle mondiale pour la logistique du dernier kilomètre.

1.2 La logistique

1.2.1 Définition de la logistique

La logistique est... la gestion de toutes les activités qui facilitent le mouvement et la coordination de l'offre et de la demande dans la création d'une utilité de temps et de lieu.

(Hesketh, Glaskowsky et Ivie, 1973)

La logistique est une fonction qui a pour objectif de mettre à disposition, au moindre coût et avec la qualité requise, un produit, à l'endroit et au moment où la demande existe. Elle concerne toutes les opérations déterminant le mouvement des produits tel que la localisation des usines, des entrepôts, l'approvisionnement, la gestion des stocks, la manutention et la préparation de commandes, le transport et les tournées de livraison.

(L'association française de la logistique Aslog en 2005)

La logistique consiste à disposer les ressources de manière optimale en termes de temps, d'emplacement, de coût et de qualité.

(Chartered Institute of Logistics and Transport (UK), 2012)

La logistique peut se définir comme l'ensemble des activités visant à optimiser les flux physiques, de services et d'informations, depuis la production à la consommation tout en passant par la production, le stockage, le transport, la distribution et la manutention.

1.2.2 La chaîne logistique globale

La chaîne logistique globale comprend tous les acteurs impliqués dans le processus de production et de livraison d'un bien ou d'un service, depuis le fournisseur initial jusqu'au client final, comme le montre la figure 1.2.2. Elle englobe

toutes les étapes qui contribuent à répondre aux demandes des consommateurs, que ce soit de manière directe ou indirecte. Cette chaîne est composée de différents acteurs tels que les fournisseurs, les producteurs, les distributeurs et les clients. Ce processus vise à intégrer toutes les fonctions de la chaîne logistique dans l'objectif de créer une vision globale en considérant l'ensemble de ses maillons comme une seule entité plutôt que de les traiter individuellement.

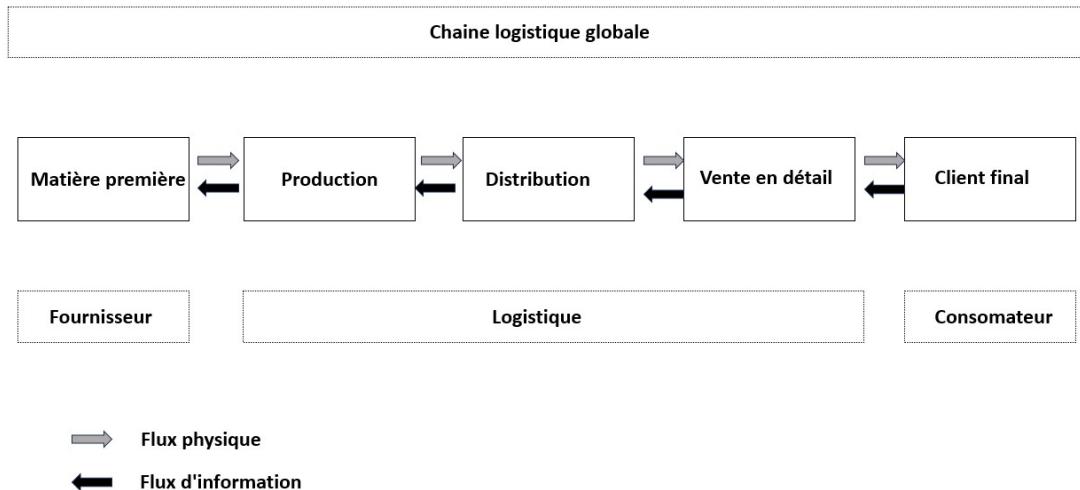


FIGURE 1.1 – Figure illustrant la chaîne logistique globale

1.2.3 L'évolution de la chaîne logistique au cours du temps

Entre les années 1950 et le début des années 1960, la distribution ne faisait l'objet d'aucune planification particulière, avec peu de contrôle sur les flux de marchandises et une faible coordination entre les différents acteurs de la chaîne. Malgré cela, les produits arrivaient néanmoins à destination aux moments et aux endroits souhaités d'une manière ou d'une autre.

Dans les années 1960 et 1970, une prise de conscience a eu lieu quant à l'importance de la liaison entre les différentes composantes de la chaîne logistique. Les années 1970 ont été cruciales pour le développement du concept de la distribution et pour la reconnaissance par les entreprises du besoin d'intégrer un système de gestion de cette dernière.

Au cours des années 1980, en raison de l'augmentation des coûts de distribution, une planification à long terme a été mise en place. Les entreprises ont commencé à prendre en compte les implications financières et opérationnelles de la gestion des flux de produits.

Pendant les années 1990, le processus d'intégration a été développé encore plus loin. Cela incluait non seulement les fonctions internes de l'organisme, mais également les autres fonctions externes de la chaîne, et cela grâce aux avancées de la technologie de l'information à cette époque. Les progrès technologiques ont permis une meilleure communication, une plus grande visibilité des stocks et une coordination accrue entre les différents acteurs de la chaîne logistique.

Dans les années 2000 et 2010, la logistique et la chaîne logistique globale sont finalement devenues des domaines clés pour la réussite de toute organisation. La gestion efficace de la chaîne logistique est devenue un avantage concurrentiel majeur, permettant aux entreprises de répondre rapidement aux demandes des clients, de réduire les coûts et d'améliorer la satisfaction de la clientèle.

Ces évolutions ont conduit à une planification plus sophistiquée, à une coordination étroite entre les acteurs de la chaîne logistique, à l'adoption de technologies avancées telles que l'automatisation et la gestion des stocks en temps réel,

ainsi qu'à une attention accrue portée à la durabilité environnementale et à la responsabilité sociale des entreprises dans le cadre de la logistique globale [3].

1.3 L'e-commerce

Le commerce en ligne, commerce électronique ou e-commerce, est l'échange pécuniaire de biens, de services ou d'informations par l'intermédiaire des réseaux informatiques, notamment Internet.

1.3.1 L'e-commerce en Algérie

Le commerce électronique en Algérie représente un nouveau concept et une toute nouvelle culture de consommation, en comparaison avec les progrès accomplis par les pays avancés ou voisins. Cependant, le passage au commerce électronique peut être considéré comme un véritable moteur de relance pour l'économie algérienne [23]. Toutefois il est important de noter que le développement du e-commerce en Algérie nécessite la mise en place d'une infrastructure solide, notamment en termes d'une connexion Internet fiable, un paiement sécurisé et une logistique efficace. Des efforts doivent être déployés pour renforcer la confiance des consommateurs algériens.

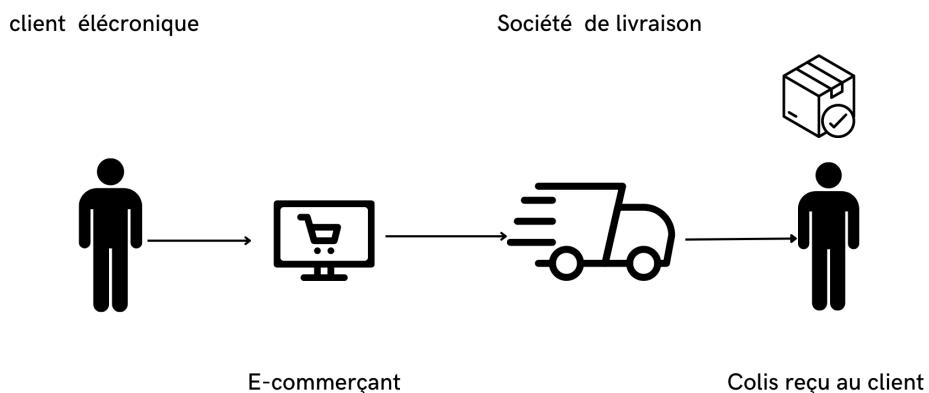


FIGURE 1.2 – Schéma illustrant le processus d'un achat en ligne

1.3.2 Les défis du e-commerce

Le commerce électronique est actuellement en plein essor en tant que canal de marketing. d'après [31] c'est le marché ayant la croissance économique la plus rapide pour différents produits et services destinés aux consommateurs. Nous citons dans ce qui suit les défis majeurs du commerce électronique.

1.3.2.1 Livraison flexible et rapide

Afin de fidéliser leur clientèle, les commerçants en ligne doivent absolument garantir la satisfaction des clients en matière de livraison. Les problèmes liés à la livraison (non-respect des délais, perte ou endommagement des colis, etc.)

demeurent un obstacle pour les consommateurs effectuant des achats en ligne de produits physiques. De plus, les consommateurs accordent de plus en plus d'importance à la flexibilité en ce qui concerne le lieu et les horaires de livraison.

1.3.2.2 Livraison à domicile

La croissance du commerce électronique entraîne un changement majeur dans la structure des chaînes de transport, où les envois massifs destinés aux détaillants sont remplacés par des envois plus fragmentés à destination des consommateurs finaux. Cela oblige les détaillants et les fabricants à revoir leurs systèmes logistiques établis, en passant des envois groupés à des petits colis. La figure 1.2 montre les étapes résultantes d'un achat en ligne.

1.3.2.3 Logistique inverse

Le commerce électronique est connu pour son taux élevé de retours de colis. Dans la plupart des pays, plus de la moitié des acheteurs en ligne ont retourné un achat en ligne [53]. En Algérie, entre 25% et 30% des achats en ligne échouent, principalement en raison du manque d'intégration du paiement électronique (le paiement à la livraison étant le mode de paiement le plus fréquemment utilisé pour régler une facture d'achat en ligne). Par conséquent, il est essentiel, pour le développement de l'e-commerce, de prendre en compte l'efficacité des opérations de logistique inverse.

1.4 La logistique du dernier kilomètre

1.4.1 Définition du dernier kilomètre

La logistique du dernier kilomètre désigne la phase finale de livraison d'un colis d'une entreprise à un consommateur (B2C). Elle commence au niveau du point de préparation de commande et se termine à la destination finale préférée du destinataire (par exemple, à son domicile ou à un point de regroupement/collecte) où les marchandises sont réceptionnées [15].

1.4.2 L'évolution de la logistique du dernier kilomètre au cours du temps

Lors de la Seconde Guerre mondiale, les États-Unis utilisait des stratégies pour assurer le transport rapide et efficace des armes, des munitions, des véhicules et du matériel sur le champ de bataille afin de soutenir les objectifs militaires. Pourtant, les leçons logistiques concernant l'importance de la fonction de livraison pour amener le produit à l'utilisateur final ont été rapidement oubliées à la fin de la guerre.

Pendant les années 1960, les processus de production de masse qui étaient utilisés pour produire des fournitures de guerre ont été rapidement adaptés à la fabrication de biens de consommation. Cependant, les méthodes de distribution efficaces pour livrer les marchandises au point de consommation n'ont pas reçu autant d'importance. Les producteurs et les détaillants ont davantage mis l'accent sur la production de stocks et leur localisation dans des points de vente centraux, minimisant ainsi l'importance du transport pour satisfaire les consommateurs.

Au début des années 2000, avec l'apparition du commerce électronique, le nombre de personnes utilisant ce dernier ne cesse d'augmenter et les clients exigent une livraison plus rapide et efficace. Les entreprises de transport ont alors commencé à développer des stratégies afin d'optimiser le service de livraison. Les leçons logistiques utilisées par les États-Unis pendant la Seconde Guerre mondiale sont devenues indispensables ([The Last Mile—History Repeating As Supply Managers Face Peak Season – Crane Worldwide Logistics, David Widdifield](#))

La logistique du dernier kilomètre, est aujourd’hui un maillon capital de la chaîne logistique et représente un enjeu important pour le SCM (Supply Chain Management). Cette importance stratégique est liée au développement exponentiel du commerce électronique, dont 90% des produits vendus sont livrés directement chez le consommateur final ou dans un point relais proche de son domicile [32].

1.4.3 Les défis de la logistique du dernier kilomètre

Le e-commerce et, par conséquent, les livraisons urbaines se développent rapidement, et ce développement s'est considérablement accéléré depuis le début de la pandémie du COVID-19. Les entreprises de logistique et de livraison ont dû s'adapter pour répondre à la demande croissante, les livraisons à domicile sont devenues plus courantes, et les clients de plus en plus exigeants pour une livraison rapide et pas chère. L'objectif de la logistique du dernier kilomètre est donc de livrer le plus rapidement possible, avec le meilleur profit possible, tout en satisfaisant le client. La complexité de ce domaine fait que sa croissance rencontre divers défis économiques et environnementaux. Dans ce qui suit nous allons présenter les défis majeurs du dernier kilomètre.

1.4.3.1 L'efficacité logistique

La logistique du dernier kilomètre représente 28% [48] des coûts de la chaîne logistique globale, et cela est dû à la planification requise pour servir un large nombre de clients à leurs adresses individuelles dans des zones urbaines complexes, tout en respectant des contraintes de temps.

La satisfaction du client est un facteur très important, les clients sont de plus en plus exigeant et demandent souvent le suivi en temps réel de leurs colis, ce qui nécessite des investissements dans la technologie, d'infrastructure et d'autres services qui engendrent forcément des coûts.

Les retours et échecs de livraisons sont des nouvelles livraisons mais dans le sens inverse, leur routage nécessite une planification minutieuse afin de ne pas influer sur les délais de la première livraison.

1.4.3.2 Congestion des routes

Les livraisons dans les zones urbaines peuvent être entravé par une circulation dense, ce qui implique des retards dans les délais de livraison, engendrant des coûts supplémentaires. Le Forum économique mondial prévoit que le nombre de véhicules de livraison dans les grandes villes pourrait augmenter de 36% d'ici 2030 [15]. De ce fait les entreprises de livraison vont devoir relever le défi et trouver des solutions durables essentiellement pour les grandes villes étant donné que 54% de la population vit dans les villes [34].

1.4.3.3 Durabilité environnementale

Le transport de marchandise est responsable d'environ 20-25 % des émissions du CO₂, selon une étude du Forum économique mondial, ces émissions sont prévu d'accroître jusqu'à 30% en 2030. En outre, plus les fenêtres de temps sont courtes plus la livraison endommage l'environnement, ce qui rend le défi encore plus complexe [37], étant donné que les consommateurs cherche de plus en plus des livraisons rapides.

1.4.4 Les solutions pour la logistique du dernier kilomètre

Le mode de livraison le plus utilisé pour la livraison de colis est la route, néanmoins d'autres stratégies ont été adoptés par les entreprises pour réduire les effets externes de la logistique du dernier kilomètre. Dans ce qui suit nous citons les stratégies adoptées pour cette dernière.

1.4.4.1 Stations à proximité ou points relais

L'utilisation des stations à proximité est une solution pour la logistique du dernier kilomètre pour amplifier l'efficacité du service. Cette approche est fréquemment adoptée lorsque les clients ne sont pas disponibles à leur domicile pour réceptionner leurs colis, ce qui entraîne une diminution notable des échecs de livraisons. Cette stratégie présente à la fois des avantages économiques et environnementaux. De plus elle offre la possibilité de remplir les points relais pendant la nuit, ce qui contribue à réduire la congestion des routes.

1.4.4.2 Logistique Urbaine Collaborative et coopérative

La logistique collaborative ou "*crowdshipping*" est une stratégie de collaboration qui distribue les tâches de livraison à une masse d'acteurs qui agissent comme des agents de livraison ordinaires, dans le but de réduire les coûts de livraison, accroître la capacité de livraison et de soutenir la durabilité. La mise en relation dans ce service est généralement soutenue par une plateforme en ligne qui offre éventuellement les fonctionnalités d'acheminement et de tarification [46]. Les entreprises de livraison adoptent fréquemment cette stratégie de collaboration quand ils pénètrent dans de nouvelles zones urbaines [7]. La coordination est un élément clé dans ce processus afin de garantir une efficacité collective [20]. Cette approche mène essentiellement à la réduction dans la taille de la flotte de véhicules, notamment le nombre de camions. Cela permet de réduire les émissions de CO₂, de diminuer la congestion des routes et finalement de réduire les coûts de livraison tout en maintenant le même niveau de service.

1.4.4.3 Optimisation dans le dernier kilomètre

L'optimisation de la gestion et de l'acheminement des transports est un domaine de recherche très important pour plusieurs applications dans le domaine de la livraison. Des Algorithmes et des systèmes d'optimisation peuvent être utilisés pour optimiser les acheminements, les ressources, ainsi que les opérations pour un meilleur rapport coûts-efficacité. Dans la revue littéraire du chapitre suivant nous présenterons quelques travaux réalisés dans ce contexte.

1.4.4.4 Véhicules innovant

Au cours des dernières années, différents véhicules innovants ont été utilisés dans le cadre de la logistique du dernier kilomètre, afin d'améliorer l'efficacité et de réduire les émissions du CO₂.

- **Véhicules électriques (VE)** : Fourgonnettes, camions et vélos électriques sont de plus en plus employés pour les livraisons. Ils présentent l'avantage de ne pas nuire à l'environnement, en ne générant aucune émission, et sont également économiquement plus avantageux par rapport aux véhicules traditionnels [38].
- **Vélos de livraison** : Les vélos de livraison, tels que les tricycles et quadricycles, gagnent en popularité pour les courtes distances en milieu urbain.

- **Véhicules de livraison autonomes** : Les véhicules autonomes tels que les drones et les robots sont en développement pour la logistique du dernier kilomètre. Ils sont capables de circuler sur les trottoirs, les pistes cyclables et les zones piétonnes, leur permettant d'atteindre rapidement et efficacement les clients.

1.5 Conclusion

Ce chapitre nous a permis de comprendre que l'optimisation joue un rôle clé dans l'amélioration de la logistique du dernier kilomètre en réduisant ces externalités négatives. En utilisant des techniques d'optimisation avancées, il est possible de concevoir des itinéraires de livraison plus efficaces, de regrouper les envois pour réduire le nombre de véhicules utilisés, d'optimiser les horaires de livraison pour éviter les périodes de congestion, et de minimiser les distances parcourues.

Chapitre 2

Présentation du sujet et problématique

2.1 Introduction

Ce chapitre est consacré à la présentation de l'entreprise Yalidine et ses services. Ensuite, nous examinerons notre problématique en détaillant les contraintes et les objectifs qui nous orientent. Enfin, nous procéderons à une analyse approfondie du problème en effectuant une revue succincte de la littérature sur la logistique du dernier kilomètre.

2.2 Intervenant dans le projet

2.2.1 Présentation de Yalidine Service El Djazair

YALIDINE EL DJAZAIR Service est une entreprise algérienne créée en 2013 spécialisée dans la logistique express, qui assure la collecte et la livraison des colis sur le territoire national.



FIGURE 2.1 – Logo de Yalidine

2.2.1.1 Services fournis par Yalidine

Livraison express (Last-Mile) : Comme son nom l'indique, c'est un mode d'expédition rapide. Ce service concerne l'envoi des plis, colis ou marchandises en urgence, la livraison ne doit pas dépasser 24 heures.

Livraison par VTC : Le colis est livré directement du vendeur au client avec un VTC (véhicule touristique avec chauffeur), elle est utilisée dans le cas où il y a 2 clients (vendeur, receveur) et un seul véhicule, elle cible principalement les magasins et les clients qui ont une petite quantité de colis à livrer avec une courte distance.

Livraison du jour même (Same day) : Pour certains secteurs, la livraison du jour même est une nécessité pour certains produits tels que les produits médicaux, volumineux, fragiles ou autre. les receveurs ne peuvent pas attendre plus qu'un jour et moins supporter un voyage au centres de tri où la casse et la perte peuvent faire des ravages.

Livraison économique : Le colis doit être livré sans délai précis, c'est un mode d'expédition moins rapide que le service Express, ce service concerne l'envoi des colis qui ne sont pas en urgence. Comme son nom l'indique, elle est moins coûteuse que la livraison Express.

Ramassage de colis (Pick up) : Ce service est destiné aux expéditeurs ne souhaitant pas se déplacer aux stations de livraison pour effectuer un envoi de colis. L'entreprise envoie des camions aux adresses des expéditeurs pour ramasser les colis.

Livraison classique : Ce service est dédié aux entreprises qui transportent des marchandises ou colis aux autres entreprises.

2.2.1.2 Hiérarchie des services de Yalidine

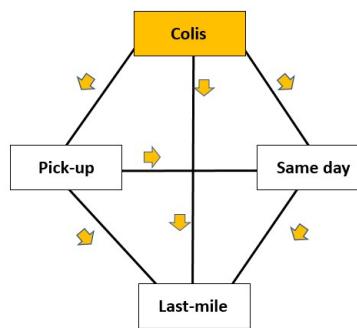


FIGURE 2.2 – Hiérarchie des services de Yalidine

2.2.1.3 Capacités opérationnelles de l'entreprise

En 2020, **Yalidine** détenait :

- 1050 employés
- 115 véhicules dont :
 - 23 motos déployés dans les centres urbains.
 - 30 véhicules légers pour les courses de banlieues et les villes.
 - 50 fourgonnettes pour les courses aux longues distances.
 - 12 fourgons pour les navettes inter-régions.
- 06 centres de tri. Un centre de tri est une station dans laquelle les colis entrent, ils se font trier selon leurs destinations, ils quittent le centre de tri pour aller vers la station de livraison la plus proche de leurs destinations.
- 90 stations de livraison dont 24 stations dans l'ouest, 34 dans l'est, 12 dans le centre et 20 dans le sud. Chaque station couvre un ensemble de communes. Une station de livraison est dans laquelle débute la dernière étape, les colis quittent cette dernière pour aller à la destination finale qui est l'adresse du client final.
- 75 agents coursiers.
- 05 superviseurs qui veillent sur le bon déroulement des opérations sur le terrain.
- 5 agents services client qui sont à l'écoute de la clientèle pour la réservation et le suivi des livraisons.

2.2.2 Présentation de Ourquilane

Ourquilane est un bureau de recherche et développement (**R&D**) créé en 2022 par Yalidine, qui propose des solutions logistiques à ses clients.



FIGURE 2.3 – Logo du bureau de (**R&D**) Ourquilane

2.3 Présentation de la problématique

Lorsqu'un colis est arrivé à la station de la commune la plus proche de sa destination finale, c'est à partir de cet instant que le problème du dernier kilomètre commence, et il ne doit pas s'écouler plus de 24 heures depuis l'instant de l'expédition du colis à l'arrivée du colis au client receveur.

Avant l'étape qui consiste à livrer le colis au consommateur final, le colis suit une étape préliminaire au cours de laquelle il est acheminé vers un centre de tri, tel qu'illustré dans la figure 2.4. Le parcours du livreur à l'étape suivante est illustré dans la figure 2.5

Pour des raisons inattendues le client peut ne pas être dans la mesure de recevoir un colis par la première tentative du livreur à lui livrer le colis, le service de livraison express de Yalidine propose deux autres tentatives de livraison du colis non encore livré par le même livreur. Dans ce contexte, la contrainte liée à la fenêtre de temps ne sera pas prise en considération pour les deux dernières tentatives. Si le colis n'est toujours pas livré après les trois tentatives, le/les colis de ce client sort du système de gestion de livraison, et le colis sera donc compté comme un retour.

Notre contribution se concentre sur l'élaboration d'un algorithme visant à optimiser les tournées de chaque livreur. L'objectif est de proposer une méthode efficace qui permettra d'améliorer l'efficacité globale du processus de livraison.

2.3.1 Contexte de la problématique

Dans cette sous-section, nous mettons l'accent sur l'environnement dans lequel se situe notre problématique afin de mieux la cerner.

— L'entreprise utilise un partitionnement communal de ses clients, où chaque commune est servie par un ou plusieurs livreurs en fonction de la densité de la demande de chaque commune. Ce partitionnement entraîne les conséquences suivantes :

1. Une distribution inégale des colis entre les livreurs. C'est une conséquence qui n'est pas souhaitée par l'entreprise.
2. La donnée de la géo-localisation des clients n'est pas entièrement exploitée.
3. Les mêmes livreurs sont affectés aux mêmes communes tous les jours. Il s'agit ici d'un avantage, car cela engendre une bonne connaissance des routes empruntées par les livreurs. Nous tâcherons de ne pas perdre cet avantage dans la solution que nous allons proposer.

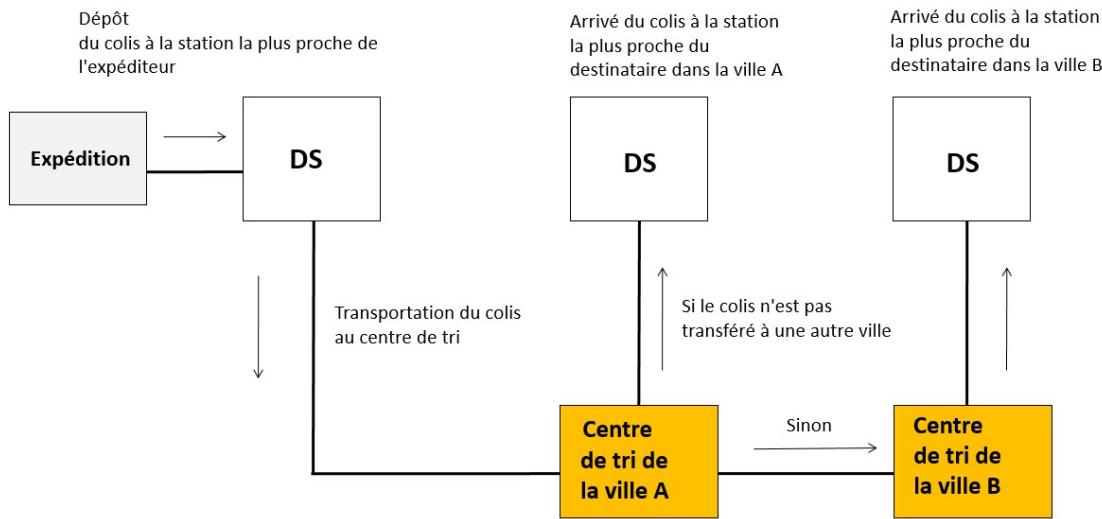


FIGURE 2.4 – Schéma modélisant le parcours du colis depuis son expédition à l'arrivée à la station de livraison
DS : Delivery station

- Les tournées empruntés par les livreurs ne sont pas optimisées, c'est aux livreurs de décider chaque jour leurs tournées.
- Les véhicules sont supposés du même type.
- Le coût lié au recrutement des livreurs est négligé.

Le contexte de la problématique a mené aux questions suivantes :

- Question 1 : Peut-on équilibrer la répartition des colis aux livreurs en améliorant la qualité du partitionnement ?
- Question 2 : Comment satisfaire toute la demande journalière en optimisant les tournées empruntées par les livreurs ?

2.3.2 Description du problème

Dans ce qui suit, nous allons décrire en détails, l'objectif, les contraintes et les paramètres du problème en question. Les informations présentées ci-dessous ont été recueillies lors de notre visite à une des stations de livraison de Yalidine. Dans l'annexe A, le lecteur intéressé trouvera le rapport détaillé de cette visite.

2.3.2.1 Objectifs

- Satisfaire toute la demande journalière en minimisant les distances parcourus pour chaque livreur. Néanmoins la minimisation du nombre de véhicule n'est pas prise en compte, car le coût lié au recrutement des livreurs est négligé.
- Assurer un équilibre dans la répartition des colis entre les livreurs.

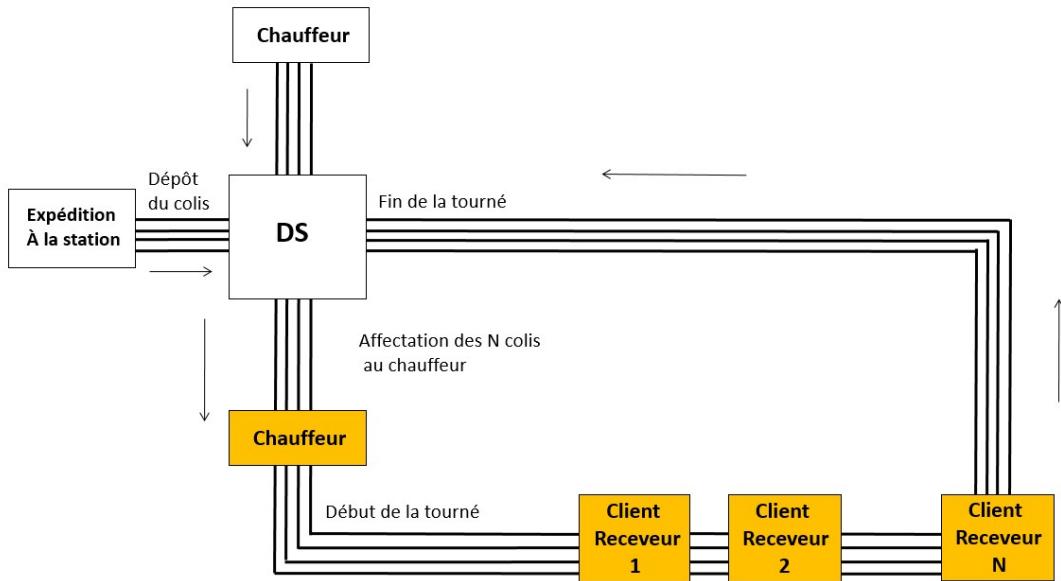


FIGURE 2.5 – Schéma modélisant le parcours d'un livreur dans le Last-MileDS : Delivery station

2.3.2.2 Contraintes

- Nombre moyen de colis à affecter à chaque livreur.
- La demande de chaque client doit être assurée par un seul véhicule.
- Chaque véhicule prend exactement une route.
- Aucun client n'est visité plus d'une fois.
- Tous les véhicules sont supposés du même type (flotte homogène).
- Chaque livreur est affecté à une région donnée.

2.3.2.3 Paramètres du problème

Dans ce qui suit, nous allons présenter les paramètres du problème. Il est important de noter que, pour éviter les problèmes liés aux fluctuations des données temporelles, notamment les matrices de temps de parcours. Une moyenne des colis a été calculée afin de garantir le respect des contraintes liées aux délais de livraison.

- **Station de livraison :** Est désignée par un ensemble de communes couvertes par cette station et un nombre moyen de livreurs affectés à cette station, comme peut le montrer la figure 2.6.
- **Colis :** Est caractérisé par la région, la commune et le quartier de sa destination.
- **Livreur :** Est désigné par un véhicule, une région à laquelle il est affecté.
- **Distances :** Matrice des distances entre les noeuds des quartiers.
- **Client :** La demande du client (Nombre de colis commandé par ce client).

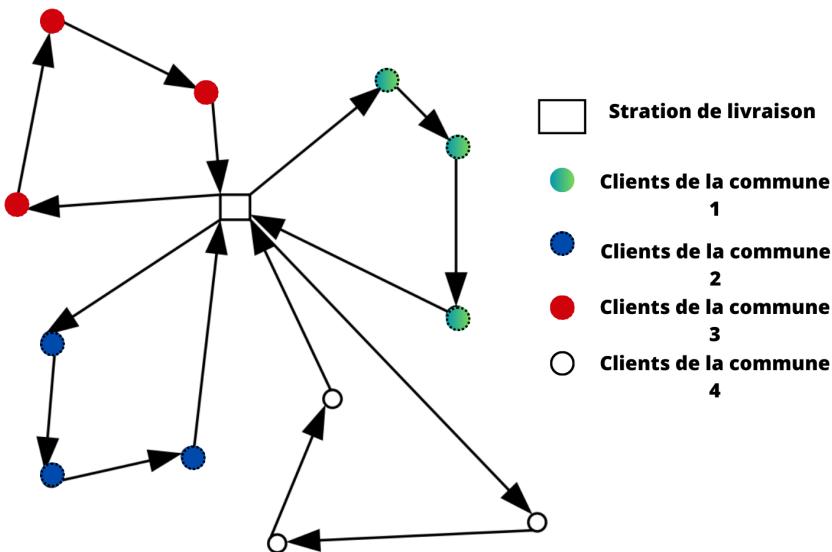


FIGURE 2.6 – Exemple des tournées d'une station couvrant quatre communes

2.4 Analyse du problème

2.4.1 Revue littéraire

Il est nécessaire d'effectuer une revue de la littérature afin d'établir notre problème sur des bases solides. Une revue de la littérature nous permettra de recueillir et d'analyser les travaux de recherche existants sur le sujet, ce qui nous aidera à mieux comprendre le contexte, les problèmes et les solutions proposées dans le domaine de la gestion des livraisons.

2.4.1.1 Recherche opérationnelle

Il existe plusieurs modèles couramment utilisés pour optimiser les itinéraires de livraison. Parmi ces modèles, on peut citer le problème de tournées de véhicule avec capacité (CVRP) [57], le problème de tournées de véhicule avec fenêtres de temps (VRPTW) et le problème de tournées de véhicule avec contraintes de capacité et de fenêtres de temps (CVRPTW). Ces modèles sont fréquemment utilisés pour résoudre les défis spécifiques liés à l'acheminement efficace des marchandises lors de la dernière étape de la livraison.

En raison des défis rencontrés dans la livraison en temps opportun en utilisant l'optimisation des itinéraires des véhicules, **TIWARI et SHARMA** (2023 [56]) ont modélisé le problème en plusieurs problèmes d'optimisation des itinéraires des véhicules avec une contrainte de capacité maximale et aucune contrainte de temps.

Dans l'article (**Li Jiang et al., 2020**) [27], un modèle de livraison a été proposé pour faire face à la pandémie de COVID-19. Le modèle consiste à construire un chemin hamiltonien sur un sous-ensemble de nœuds, où chaque client est affecté au parking le plus proche. L'objectif est de minimiser le coût de la tournée, les distances entre les clients et les parkings, ainsi que les coûts de stationnement. Deux modèles, la formulation basée sur le flux (*Flow-based formulation*) et la formulation basée sur les nœuds (*Node-based formulation*), ont été proposés pour résoudre le problème. Les deux formulations ont été testées sur 108 instances en utilisant CPLEX et ont été comparées à l'algorithme de recherche de voisinage variable (*Variable neighborhood search*). Les résultats ont montré que les deux formulations et l'algorithme VNS ont donné des solutions optimales, mais l'algorithme proposé s'est révélé plus performant en termes de rapidité

d'exécution.

2.4.1.2 Recherche opérationnelle et Apprentissage automatique

Les méthodes d'apprentissage automatique ou encore "*Machine Learning*" peuvent fournir des prédictions plus précises et ont suscité une attention croissante dans le domaine de la recherche opérationnelle. Ces algorithmes trouvent plusieurs applications dans la gestion de la chaîne logistique, notamment dans la Logistique du dernier kilomètre, les principaux sujets traités dans la littérature sont : problèmes d'entrepôts, détection des anomalies, prévision des flux de trafic et de la demande et la satisfaction des clients.

En raison de l'écart entre les itinéraires optimisés et les itinéraires empruntés par les conducteurs en raison de leurs préférences, un modèle d'optimisation de routes hiérarchique, combinant les avantages des approches d'optimisation et d'apprentissage automatique, a été proposé par **SHAO et CHENG (2023)** [55].

Beaucoup d'études ont été faites concernant le sujet de prévision de la demande, les prévisions sont nécessaires pour les opérateurs de transports. Les algorithmes de prévisions de demande sont parmi les algorithmes les plus utilisés dans la gestion de la chaîne logistique. **CHU et al., (2021)** [8] ont proposé une approche d'optimisation basée sur les données qui combine des techniques d'apprentissage automatique avec l'optimisation des itinéraires de véhicules à capacités pour résoudre les problèmes liés à l'incertitude du temps de livraison.

Un autre problème a été débattu qui est le problème de détection des anomalies, un algorithme a été développé par **Rosen et Medvedev (2012)** qui détecte des anomalies dans la trajectoire des véhicules [16].

Une nouvelle approche pour générer des itinéraires a été proposée par **GUO, MO et WANG (2023)** [21]. Elle consiste à prévoir les décisions de changement d'itinéraires qu'un chauffeur peut prendre en se basant sur sa connaissance de la congestion des routes, de la disponibilité de stationnement, des heures de pointe, etc. La méthode proposée repose sur le problème classique du voyageur de commerce (**TSP**) avec une matrice de distances personnalisée qui intègre implicitement les comportements de routage des conducteurs pour les livraisons d'Amazon. Comparé au **TSP** existant, qui vise à minimiser la distance totale parcourue par les véhicules, la matrice de distances a été modifiée et des itinéraires optimaux ont été générés afin de minimiser la distance totale parcourue.

2.4.2 Les concurrents internationaux

Les grandes entreprises telles que DHL, Amazon, UPS et USPS mettent en œuvre des techniques afin d'optimiser leurs services de livraisons.

Amazon a mis au point de nouvelles méthodes visant à optimiser le dernier kilomètre des livraisons. Les chercheurs se sont inspirés du problème de PCVRP (**Prize Collecting Vehicle Routing Problem**). En 2021, Amazon a présenté un nouvel algorithme appelé **Customer Order and Network Density Optimizer (CONDOR)**, capable de résoudre des problèmes plus complexes que le PCVRP. Cet algorithme utilise une combinaison de techniques telles que l'optimisation mathématique, la recherche locale et le CVRP. Grâce à CONDOR, il a été possible d'optimiser les itinéraires de livraison en évaluant les commandes des clients avant leur départ du dépôt. L'objectif était d'identifier des options plus efficaces. ([The science behind grouping package deliveries, Amazon](#))

Amazon utilise la matrice des distances ([Amazon Location Service enables Matrix Routing to optimize route planning](#)) pour l'utiliser comme donnée afin d'optimiser le service de livraison.

DHL a fondé une start-up nommée Greenplan qui a été ensuite devenue une entreprise indépendante en 2022 a crée un algorithme qui optimise les tournées de livraison qui a réduit 20% sur les coûts par rapports aux solutions d'optimisations d'itinéraires standards et il a pu réduire l'émission du CO₂. Greenplan exploite un algorithme qui exploite des méthodes fondées sur l'analyse des données de circulation et des itinéraires passés, ainsi que sur la durée des déplacements. En d'autres termes, Greenplan recourt à des techniques d'apprentissage automatique pour atteindre ses objectifs.. ([GREENPLAN – THE BEST WAY: LOGISTICS EXPERTS LAUNCH POWERFUL ALGORITHM FOR INDIVIDUAL ROUTE OPTIMIZATION](#)).

En 2012, UPS a développé une plateforme appelée ORION (On-Road Integrated Optimization and Navigation). Cette plateforme effectue des mises à jour constantes des itinéraires tout au long de la journée, en prenant en compte des facteurs tels que le trafic, les livraisons, les collectes de colis, etc. Elle utilise un système d'optimisation dynamique. UPSNav, créé par UPS, a été intégré à la plateforme ORION. Il fournit des instructions aux chauffeurs pour les guider sur l'itinéraire optimisé. Cette plateforme est continuellement améliorée afin d'optimiser davantage les opérations de livraison. ([UPS To Enhance ORION With Continuous Delivery Route Optimization](#))

2.5 Conclusion

Après une analyse minutieuse du contexte du problème rencontré, nous avons constaté qu'une décomposition en sous-problèmes est nécessaire afin de s'adapter au mieux à la réalité dans la quelle il réside, et ainsi optimiser notre approche à le résoudre. Ce dernier se décompose en deux sous-problèmes majeurs d'optimisation combinatoire. En premier temps, nous serons amené à résoudre le problème de partitionnement des clients en zones *clustering*, puis en deuxième temps, nous aborderons la résolution du problème de tournée de véhicules avec des clusters, comme illustré dans la figure 2.7.

Il convient de souligner que dans la suite, nous allons proposer un autre partitionnement des clients, différent du partitionnement communal adopté par l'entreprise actuellement, comme décrit dans le chapitre 3. Cette proposition vise à atteindre un équilibre dans la répartition des colis entre les livreurs. Elle permettra également d'éviter les situations où des clients proches les uns des autres ne sont pas servis par le même livreur en raison de leur appartenance à des communes différentes.

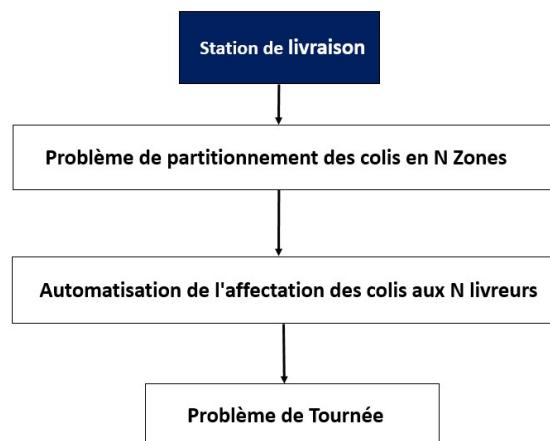


FIGURE 2.7 – Décomposition des problèmes depuis une station de livraison donnée

Chapitre 3

Optimisation combinatoire et problème de tournées de véhicules

3.1 Introduction

L'objectif de ce chapitre est de présenter les principes fondamentaux de l'optimisation combinatoire afin de mieux appréhender le problème. Nous examinerons également les problèmes de tournées de véhicules et du voyageur de commerce, ainsi que les diverses méthodes pour les résoudre.

3.2 Optimisation combinatoire

3.2.1 Optimisation

L'optimisation est une démarche consistante à rendre optimal le fonctionnement d'un système.

En recherche opérationnelle, l'optimisation signifie l'analyse et la résolution des problèmes qui consistent à minimiser ou maximiser une fonction sur un ensemble d'éléments. Modéliser un problème d'optimisation consiste à formuler mathématiquement un problème de décision en décrivant les relations entre les variables et en spécifiant les objectifs à atteindre.

3.2.1.1 Identifier les variables de décision

Les variables sont des quantités inconnues qui influencent la fonction à optimiser dans un problème d'optimisation. Elles peuvent représenter des quantités physiques, des paramètres, des décisions à prendre, etc. Elles peuvent être réelles, binaires ou entières.

3.2.1.2 Définir les contraintes

Les contraintes sont les conditions à respecter pour avoir une solution réalisable. Ce sont les conditions qui limitent les valeurs possibles des variables. Dans un modèle mathématique, les contraintes sont représentées sous forme d'équations ou d'inégalités.

3.2.1.3 Déterminer la fonction objectif

Identifier l'objectif à atteindre et l'écrire sous forme mathématique, elle peut être à maximiser ou à minimiser en fonction du type du problème si c'est un problème de maximisation ou de minimisation.

3.2.2 Optimisation combinatoire

Un problème d'optimisation combinatoire est un type de problème qui vise à trouver la meilleure solution parmi un ensemble fini de solutions réalisables.

Un problème d'optimisation combinatoire est défini à partir d'un ensemble fini E et d'une application $c : S \rightarrow \mathbb{R}$. Il s'agit de déterminer $s \in S$ tel que :

- $S \subset E$
- $f(s) = \sum_{e \in S} c(e)$

3.2.2.1 Optimisation locale et globale

Soit $f(x)$ une fonction objective, S un ensemble de solutions, S' est un sous ensemble de solutions $S' \subset S$.

- **Voisinage** : Le voisinage d'une solution est un sous-ensemble de solutions qu'on peut atteindre à travers une série de transformations données.
- **Optimisation locale** : L'optimisation locale vise à trouver une solution x^l tel que :
$$x^l = \arg \min_{x \in S'} f(x).$$
 [51]
- **Optimisation globale** : L'optimisation globale vise à trouver une solution x^* tel que :
$$x^* = \arg \min_{x \in S} f(x).$$
 [51]

3.2.3 Programmation linéaire en nombres entiers (PLNE)

3.2.3.1 Définition

La programmation linéaire a été introduite par **Dantzig en 1947**. Elle consiste à optimiser une fonction linéaire avec des contraintes linéaires [11].

La programmation linéaire en nombre entiers consiste à optimiser une fonction linéaire avec des contraintes linéaire où les variables de décisions sont des entiers [19].

3.2.3.2 Formule mathématique

La formule mathématique d'un programme linéaire en nombres entiers est défini comme suit :

$$\begin{aligned} & \text{maximiser } Z = c^T x \\ & Ax \leq b^T \\ & x \in \mathbf{N}^n \end{aligned}$$

- x représente le vecteur des variables de décision.
- A représente la matrice des coefficients des contraintes.
- c représente le vecteur des coefficient des variables de décision du vecteur x dans la fonction objectif linéaire.
- b est le vecteur des bornes borne à respecter
- n est la taille du vecteur x

3.2.3.3 Solution d'un programme linéaire en nombre entiers

- Une solution $x = \{x_1, x_2 \dots x_n\}$ est dite réalisable si elle satisfait toutes les contraintes.
- Une solution x^* est dite solution optimale si la fonction objectif z prend sa valeur maximale z^* .

3.2.4 Complexité

3.2.4.1 Complexité algorithmique :

Le terme complexité algorithmique mesure le nombre d'étapes nécessaires à l'algorithme pour résoudre le problème donné. Il évalue l'ordre de comptage des opérations exécutées par un algorithme en fonction de la taille des données d'entrée. Le temps d'exécution d'un programme est lié à la complexité de l'algorithme qu'il met en œuvre [6].

3.2.4.2 Complexité d'un problème :

La complexité du problème est celle du meilleur algorithme qui le résout.

3.2.4.3 Les classes de complexité d'un problème

- **P** : La classe P contient les problèmes qui peuvent être résolus en temps polynomial par une machine déterministe, autrement dit, ils sont faciles à résoudre.
- **NP** : La classe NP (Non déterministe polynomial) contient les problèmes qui peuvent être résolus en temps polynomial par une machine non-déterministe, c'est à dire qu'on peut vérifier rapidement en temps polynomial si une solution candidate est bonne.
- **NP-Difficile** : Un problème est NP-Difficile est un problème vers lequel on peut ramener tout problème de la classe NP par une réduction polynomiale, s'il appartient à la classe NP alors il devient NP-complet. [45]

3.3 Problème de tournées de véhicules

Dans cette section, nous allons donner un aperçu concis des problèmes classiques du TSP (Traveling Salesman Problem), du VRP (Vehicle Routing Problem) et du CluVRP (Clustered Vehicle Routing Problem). Nous mettrons l'accent sur les approches utilisées pour résoudre le VRP et le TSP.

3.3.1 Le problème du voyageur de commerce (TSP)

Le problème du voyageur du commerce (Traveling salesman problem) introduit par **DANTZIG, FULKERSON et JOHNSON** [10] est décrit comme suit : Trouver la plus courte tournée en partant d'une ville, passer par toutes les villes qui doivent être visitées une seule fois, puis revenir au point de départ comme le montre la figure 3.1.

3.3.1.1 Formulation mathématique

En théorie des graphes, le problème du voyageur de commerce consiste à trouver un chemin hamiltonien de coût minimum, dans un graphe complet $G(X, U)$ (c_{ij} est le coût évalué sur chaque arc (i, j) où : $i, j \in U$)
Nous avons les données suivantes :

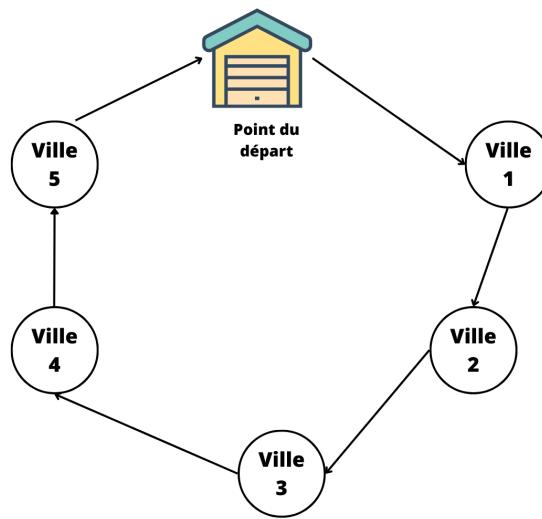


FIGURE 3.1 – Un exemple d'une tournée du problème du voyageur de commerce

$n = |X|$ le nombre de sommets dans un réseau.

c_{ij} = coût de l'arc (i, j)

$$x_{ij} = \begin{cases} 1, & \text{si l'arc } (i, j) \text{ appartient à la tournée} \\ 0, & \text{sinon} \end{cases} \quad (3.1)$$

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

$$\sum_{i=1, (i \neq j)}^n x_{ij} = 1; \quad j = 1, 2, \dots, n \quad (2)$$

$$\sum_{j=1, (i \neq j)}^n x_{ij} = 1; \quad i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S, i \neq j} x_{ij} \leq |S| - 1; \quad \forall S \subseteq X \quad (4)$$

$$x_{ij} \in \{0, 1\}; \quad 1 \leq i, j \leq n$$

- La fonction objectif (1) permet de minimiser le coût de la tournée
- Les deux contraintes (2) et (3) traduisent le fait que chaque ville doit être visitée exactement une seule fois.
- la contrainte (4) interdit les solutions qui sont des sous-tours (des solutions qui ne passent pas par toutes les villes)

3.3.2 Le problème de tournées de véhicules (VRP)

Le problème de tournée de véhicules (Vehicle routing problem) introduit par **TOTH** et **VIGO** [57] Ce problème est considéré comme une extension du problème du voyageur de commerce. Dans le **VRP**, nous disposons de m véhicules et l'objectif est de minimiser le coût total des tournées effectuées par ces véhicules. Chaque véhicule part

du dépôt, parcourt un sous-ensemble de villes et revient ensuite au point de départ. De plus, chaque ville ne peut être visitée qu'une seule fois par un seul véhicule.

VRP prend en compte la demande associée à chaque ville ainsi que la capacité des véhicules, où la somme des demandes des villes visitées au cours d'une tournée ne doit pas dépasser la capacité maximale du véhicule. La figure 3.2 illustre une solution du **VRP**.

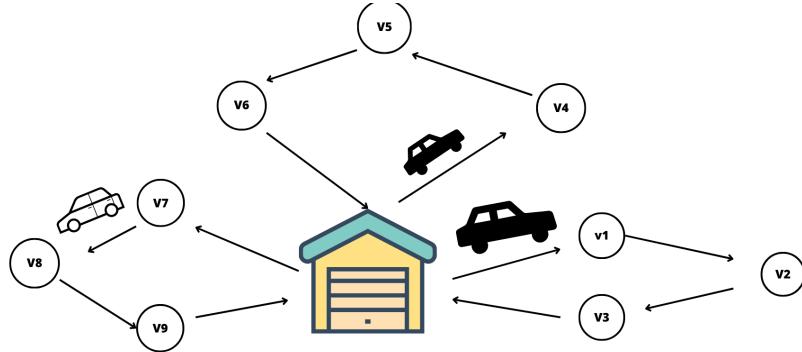


FIGURE 3.2 – Un exemple montrant les tournées de véhicules

3.3.2.1 Formulation mathématique :

Nous avons les données suivantes :

$$i, j \in 0 \dots n$$

$n = |X|$ le nombre de sommets dans un réseau.

c_{ij} = Coût de l'arc (i, j)

m = Le nombre de véhicules.

Q = La capacité des véhicules.

q_i = Demande de chaque ville i .

$$x_{ij}^k = \begin{cases} 1, & \text{si l'arc } (i, j) \text{ appartient à la tournée du véhicule } k \\ 0, & \text{sinon} \end{cases} \quad (3.2)$$

$$\text{Minimiser} \quad \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^k \quad (1)$$

$$\sum_{i=0, (i \neq j)}^n \sum_{k=1}^m x_{ij}^k = 1; \quad j = 1, 2, \dots, n \quad (2)$$

$$\sum_{j=0, (i \neq j)}^n \sum_{k=1}^m x_{ij}^k = 1; \quad i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1}^n x_{i0}^k = 1; \quad k = 1 \dots m \quad (4)$$

$$\sum_{j=1}^n x_{0j}^k = 1; \quad k = 1 \dots m \quad (5)$$

$$\sum_{i=0}^n \sum_{j=1}^n q_i x_{ij}^k \leq Q; \quad k = 1 \dots m \quad (6)$$

$$\sum_{k=1}^m \sum_{i \in S} \sum_{j \in S, i \neq j} x_{ij}^k \leq |S| - 1; \quad \forall S \subseteq \{1 \dots n\} \quad (7)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in \{1 \dots n\} \quad \forall k \in 1 \dots m \quad (8)$$

1. La fonction objectif (1) permet de minimiser le coût total des tournées.
2. Les deux contraintes (2) et (3) traduisent que chaque ville doit être visitée une seule fois.
3. les contraintes (4) et (5) indiquent que chaque véhicule doit démarrer du dépôt et revenir au dépôt.
4. la contrainte (6) indique que la demande ne doit pas dépasser la capacité du véhicule.
5. La contrainte (7) interdit les tours qui ne contiennent pas le sommet 0.

3.3.2.2 Domaines d'application

Le problème de tournées de véhicules et le problème du voyageur de commerce ont de nombreuses applications dans plusieurs domaines. Le **VRP** est souvent utilisé dans la logistique, notamment pour optimiser le ramassage, la livraison et la distribution. Nous citons quelques exemples :

- Livraison des colis
- Services d'urgences : Dans les situations d'urgences, les véhicules tel que les ambulances, fourgons d'incendies ...etc, **VRP** peut être utilisé en minimisant le temps de parcours dans le but d'arriver plus tôt.
- Transport des travailleurs : Le VRP peut être utilisé pour ramasser les élèves et les travailleurs dans le but d'arriver le plus tôt possible en réduisant le temps parcouru, il est aussi utilisé dans le transport scolaire.

Le **TSP** est NP-complet [14], tant dis que le **VRP** est NP-difficile [33]. Leurs résolutions par méthode exacte s'avère presque impossible pour les instances de grandes tailles. Des méthodes approchées de complexité polynomiale permettent de donner des solutions de bonne qualité mais ne garantissent pas l'optimalité.

3.3.3 Le problème de tournées de véhicules avec clusters (CluVRP)

Le *CluVRP* a été introduit la première fois par **SEVAUX, SØRENSEN et al.** [54]. le modèle est une variante du classique problème d'optimisation combinatoire *VRP* dans lequel les clients sont regroupés dans différentes zones (*clusters*), le véhicule ne quitte pas un *clusters* pour aller dans un autre avant que ce dernier visite tous les clients du cluster actuel. Il est également supposé qu'une zone est visitée une seule fois et par un seul véhicule. La figure

3.3 illustre une solution du *CluVRP*. Ce problème est plus précisément une généralisation **CVRP**, le **CVRP** est NP-difficile, alors il en est de même pour le **CluVRP** [26].

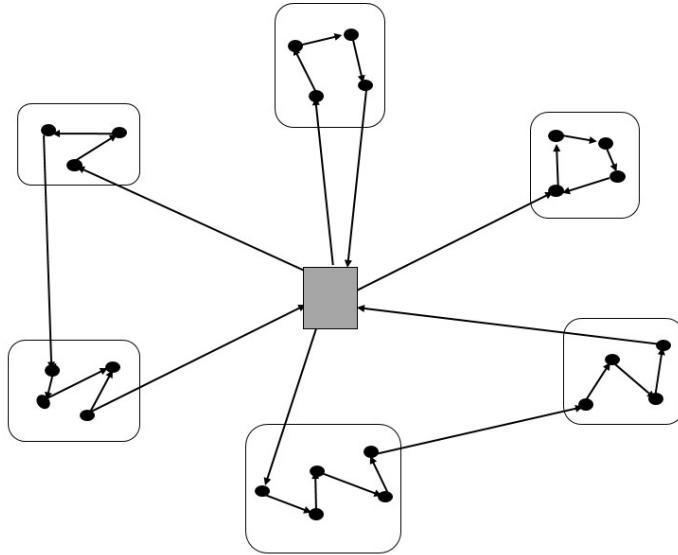


FIGURE 3.3 – Figure illustrant une tournée de *CluVRP*

3.3.3.1 Formulation mathématique

Le *CluVRP* peut être formellement défini en tant qu'un modèle de théorie de graphe.

Soit $G(V, E)$ un graphe non orienté, où $V = \{v_0, v_1, \dots, v_n\}$ est l'ensemble des sommets du graphe et $E = \{(v_i, v_j) | v_i, v_j \in V, v_i \neq v_j\}$. V est partitionné en $k + 1$ sous ensembles appelé clusters ; soit $R = \{R_0, R_1, \dots, R_k\}$ l'ensemble des clusters, l'ensemble R_0 contient uniquement le Dépôt. Deux types d'ensemble d'arêtes inter clusters sont définies $\delta^+(S)$ et $\delta^-(S)$ où $\delta^+(S) = \{(i, j) \in E | i \in S, j \in V \setminus S\}$ et $\delta^-(S) = \{(i, j) \in E | i \in V \setminus S, j \in S\}$ où S est un sous ensemble de V . La demande de chaque sommet est représenté par q_i et la capacité de la flotte de véhicule homogène est noté par Q . $Cr = \{v_i \in V \setminus V_0 : r_i = r\}$ représente l'ensemble des sommets dans un cluster r_i . Finalement Le problème peut être formulé par la formulation suivante [1] :

$$x_{ij}^k = \begin{cases} 1, & \text{si l'arc } (i, j) \text{ appartient à la tournée du véhicule k} \\ 0, & \text{sinon} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{si le sommet } i \text{ est visité par le véhicule k} \\ 0, & \text{sinon} \end{cases}$$

$$\begin{aligned}
 & \text{Minimiser} && \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \quad (1) \\
 & && \sum_{k=1}^m y_{ik} = 1 \quad i = 2, \dots, n \quad (2) \\
 & && \sum_{k=1}^m y_{1k} = m \quad (3) \\
 & && \sum_{j=1}^n x_{ij}^k = \sum_{j=1}^n x_{ji}^k = y_{ik} \quad k = 1 \dots m \quad i \in V \quad (4) \\
 & && \sum_{i=2}^n q_i y_{ik} \leq Q \quad k = 1 \dots m \quad (5) \\
 & && \sum_{i \in S} \sum_{j \in V \setminus S} x_{ij}^k \geq y_{hk} \quad \forall S \subseteq V \setminus V_0 \quad h \in S \quad k = 1 \dots m \quad (6) \\
 & && \sum_{(i,j) \in \delta^+(Cr)} \sum_{k=1}^m x_{ij}^k = \sum_{(i,j) \in \delta^-(Cr)} \sum_{k=1}^m x_{ij}^k = 1 \quad \forall r \in R \quad (7) \\
 & && y_{ik} = y_{jk} \quad \forall (i,j) \in Cr \quad \forall r \in R \quad k = 1 \dots m \quad (8) \\
 & && x_{ij}^k \in \{0, 1\} \quad \forall (i,j) \in \{1 \dots n\} \quad \forall k = 1 \dots m \quad (9) \\
 & && y_{ik} \in \{0, 1\} \quad \forall i \in \{1 \dots n\} \quad \forall k = 1 \dots m \quad (10)
 \end{aligned}$$

- (1) Minimise le coût des tournées de chaque véhicule
- (2) Garantie que chaque sommet est visité une seule fois par un seul véhicule.
- (3) Assure que tous les véhicules passe par le dépôt.
- (4) Cette contrainte force un sommet d'être accédé et quitté par le même véhicule.
- (5) Assure le respect des capacités des véhicules.
- (6) Garantie l'élimination des sous-tours.
- (7) Assure qu'un *cluster* est visité une seule fois par un seul véhicule.
- (8) Présente l'intégrité de l'ordre.

3.3.4 Approches de résolution du VRP

3.3.4.1 Méthodes exactes

Les méthodes exactes en optimisation combinatoire sont des techniques algorithmiques qui garantissent de trouver la solution optimale pour un problème donné. Ces méthodes explorent d'une manière exhaustive l'ensemble des solutions possibles afin de déterminer la meilleure solution selon un critère d'optimisation spécifié. [45]

3.3.4.2 Méthodes approchées

Une méthode approchée, comme son nom l'indique est une méthode qui donne une solution réalisable dont on ne peut prouver l'optimalité, ces méthodes ont été développées dans le but de trouver des solutions rapidement d'où leur utilisation dans le milieu industriel.

Heuristiques

- **Heuristiques de construction** Les heuristiques constructives connues plus généralement sous forme de méthodes gloutonnes, sont des méthodes qui commencent par une solution vide et construisent le reste de cette dernière d'une manière itérative. Ces heuristiques ne fournissent pas de solutions de bonne qualité en général, mais elles sont célèbres pour leur simplicité à les implémenter et pour leur rapidité.
- **Plus proche voisin** Cette heuristique est la plus intuitive, à chaque itération elle permet d'insérer à la solution partielle le sommet le plus proche du dernier inséré, jusqu'à ce que tous les sommets soient affectés à la tournée. L'heuristique est très facile à implémenter et s'exécute rapidement, Néanmoins les solutions qu'elle fournit sont rarement de bonnes qualités.
- **Heuristiques d'amélioration** Les méthodes d'amélioration ou à *recherche locale* consistent à améliorer une solution de départ construite aléatoirement ou bien par une heuristique de construction. L'amélioration est menée par des perturbations de la solution, une perturbation est une *transformation admissible*, un ou plusieurs mouvements appliqués aux sommets, arêtes ou autres, qui génèrent une solution voisine dans le but de améliorer la solution courante. La limite de ces méthodes c'est qu'elles arrêtent l'exploration dès qu'une meilleure solution est trouvée, cette solution est appelée *Optimum locale*.
- **Clarke & Wright** La méthode de Clarke et Wright est largement connue et continue d'être utilisée dans la résolution de problèmes de tournées. Cet algorithme commence par créer une tournée pour chaque client. Ensuite, ces tournées sont fusionnées jusqu'à ce qu'il n'y ait plus de possibilité de réduire le coût de la solution par fusion. Lors de la fusion de deux tournées, le coût est réduit selon un score appelé "saving" (économie) qui est calculé en utilisant le coût des arêtes impliquées dans la fusion. Cette méthode a été initialement développée par **Clarke et Wright** en 1964 [9].

Méthodes en deux phases

Ces méthodes divisent le problème de tournées de véhicules en deux sous-problèmes, *partitionnement* et *routing*, il existe deux possibilités dans ce type de méthodes, la première **cluster-first-route-second** [47] dans laquelle les sommets sont en premier temps partitionnés en *cluster* selon un critère de densité ou de minimisation de distance entre les sommets, ensuite dans chaque *cluster* une séquence de sommets sera déterminée. tandis que dans la deuxième possibilité **Route-first-cluster-second** [4] il s'agit ici de transformer une solution du **TSP** en une solution de **VRP**, une tournée géante du **TSP** contenant tous les sommets est créée tout en minimisant les coûts liés à la distance, et à partir de cette tournée des *clusters* sont déterminés.

Métaheuristiques

Contrairement aux heuristiques qui sont spécifiques à un problème particulier. Les métaheuristiques sont conçues pour s'adapter au problème étudié afin d'explorer efficacement l'espace des solutions possibles. Leur structure leur permet d'échapper aux optimums locaux et ainsi garantir des solutions d'une meilleure qualité. Un autre concept crucial dans les métaheuristiques est l'intensification et la diversification. Ces deux forces déterminent d'une manière significative le comportement d'une métaheuristique, elles sont à la fois opposées et complémentaires [5]. La notion de diversification se réfère à l'exploration de l'espace de recherche, tant que que l'intensification implique l'exploitation d'une zone de recherche prometteuse. Les métaheuristiques sont généralement classées en deux types

majeurs : les méthodes à base de trajectoire et les méthodes à base de population.

- **Diversification** La diversification est une notion qui fait référence à l'identification des zones prometteuses de l'espace de recherche des solutions, et cela peut se faire en utilisant différentes stratégies d'exploration, telles que l'utilisation des opérations de voisinage. La diversification permet d'élargir l'espace de recherche, ce qui aide la métaheuristique à échapper aux optimum locaux et à aboutir à des solutions de meilleure qualité.
- **Intensification** L'intensification est la notion qui fait référence à l'exploitation d'une zone prometteuse en intensifiant la recherche pour des solutions de meilleure qualité en utilisant souvent des méthodes de recherche locale. Cependant, cela peut également entraîner un risque de convergence prématurée vers des optimums locaux.

Hybridation des méthodes

L'hybridation des méthodes est une approche de résolution des problèmes d'optimisation combinant deux ou plusieurs méthodes afin de renforcer à la fois la diversification et l'intensification de l'espace de recherche, dans le but d'améliorer la qualité des solutions obtenues.

3.3.5 Revue littéraire sur la résolution du (TSP)

Dans [40] Différentes méthodes, telles que la *recherche taboue* [17], le *recuit simulé* [58], *l'algorithme génétique* [25] et *l'algorithme de colonies* de fourmis[12], ont été évaluées sur diverses instances du problème, allant de 51 à 262 sommets. Pour les petites instances, ces méthodes ont produit des solutions assez proches de la valeur optimale, cependant pour les instances plus grandes, l'algorithme génétique a montré une performance nettement inférieure. La recherche taboue et le recuit simulé ont produit des résultats similaires, mais l'ACO a donné les meilleures performances parmi toutes les méthodes testées.

Dans [2] Une comparaison entre *l'Evolution Strategy* [39], *Recherche taboue* [17] et *VNS* [24] à été conduite sur 6 instances, résultant que la méthode VNS performe mieux que la recherche taboue et (l'Evolution Strategy) en terme de qualité de solution mais converge plus lentement que ces deux dernières.

[61] Les auteurs dans cette étude comparative ont testé trois métaheuristiques classiques (ACO, GA et PSO [30]) sur un problème comportant 10 noeuds et un autre comportant 50 noeuds. Les résultats ont montré que l'ACO offre les meilleures solutions avec une variation de solution relativement faible sur une dizaine d'itérations. Cependant, le temps d'exécution de l'ACO est fortement influencé par la taille du problème, avec le temps d'exécution le plus long lorsque le nombre de villes est de 50. Le GA fournit une meilleure solution que PSO, et sa durée d'exécution est moins influencée par la taille du problème. Par conséquent, le GA est probablement plus efficace lorsque la taille du problème devient plus grande. Même si PSO offre la moins bonne solution, sa durée d'exécution est relativement très courte.

[28] Cette étude comparative, a confirmé que la nouvelle métaheuristique Golden Ball [42] peut rivaliser avec les méthodes les plus utilisés pour résoudre le **TSP** comme La recherche taboue et le recuit simulé évolutif.

Une autre étude comparative a été menée [18] pour comparer huit métaheuristiques, deux classiques (SA et TS), deux évolutives (GA et algorithme memétique [22]), et quatre inspirées de la biologie (ACO, colonies d'abeilles [41], algorithme des lucioles et algorithme [59] de recherche des coucous [62]), il a été observé que les approches inspirées de la biologie ont tendance à conduire à de meilleures solutions que les autres méthodes. Plus particulièrement, l'algorithme de recherche des Coucous a produit de meilleures résultats en termes de longueur de la tournée.

[44] Dans cette étude, trois métaheuristiques, à savoir l'algorithme génétique, l'algorithme de colonie de fourmis et le recuit simulé, ont été comparées en termes d'efficacité. Le temps d'exécution et la qualité des solutions fournies ont été pris en compte pour évaluer chaque approche. En outre, les métriques du programme, telles que la complexité de McCabe, l'effort de développement et les lignes de code, ont été calculées pour compléter l'étude comparative. Les résultats ont montré que le recuit simulé était facile à mettre en œuvre, tandis que l'algorithme génétique était acceptable en termes de solutions mais manquait de stabilité et était plus compliqué à mettre en œuvre que les deux autres méthodes. Cependant, l'algorithme génétique a donné le meilleur temps d'exécution, avec une moyenne de 35% plus rapide que le recuit simulé et 60% plus rapide que l'algorithme de colonie de fourmis. Enfin, la méthode de la colonie de fourmis a produit les meilleures solutions, avec une amélioration de 4,2% par rapport aux autres minimums. Bien que le temps de traitement ne soit pas le meilleur, l'implémentation de l'algorithme de colonie de fourmis était relativement simple, environ 1,7 fois plus lent en moyenne que l'algorithme génétique.

Dans [43] Golden Ball encore une fois a été comparer avec Le GA et le DGA [60] (l'algorithme génétique distribué), il a été montré que la méthode de GB a fournis les meilleurs résultats que les deux autres métaheuristiques sur toute les dix-huit instances testé.

3.4 Conclusion

Suite à la modélisation établie dans le chapitre 2, il est clair que nous allons adopter une méthode en deux phases pour résoudre notre problème, connue sous le nom de "Cluster-first-route-second" (regrouper d'abord, puis planifier les itinéraires). Dans le chapitre suivant nous allons d'abord modéliser et résoudre le problème de partitionnement des clients, puis passer à la modélisation du problème de tournées par le **CluVRP** en changeant une seule contrainte, et le résoudre en utilisant la résolution du TSP de chaque *cluster*.

Chapitre 4

Modélisation et résolution du problème

4.1 Introducion

Au cours de ce chapitre, nous avons décomposé notre problème en deux sous-problèmes distincts. Le premier sous-problème consiste à résoudre le problème de partitionnement en tenant compte de la contrainte de capacité. Le deuxième sous-problème concerne le problème de VRP avec des clusters, où les clients sont regroupés en clusters.

4.2 Modélisation et résolution du problème de partitionnement

4.2.1 Description du problème de partitionnement

L'objectif de l'analyse de "clusters" est de regrouper les objets en "clusters" de telle manière que les objets au sein d'un même "cluster" soient plus similaires entre eux que les objets provenant d'autres "clusters". En d'autres termes, on cherche à créer des groupes où les objets partagent des caractéristiques communes et sont plus proches les uns des autres en termes de similarité [13]. Cette idée est illustrée dans la figure 4.1.

Dans notre situation, le partitionnement peut être adapté pour créer des régions où les emplacements des clients sont considérés comme les sommets, sont aussi proches les uns des autres que possible. L'objectif est de regrouper les clients en fonction de leur proximité géographique, en formant des clusters ou des sous-ensembles compacts.

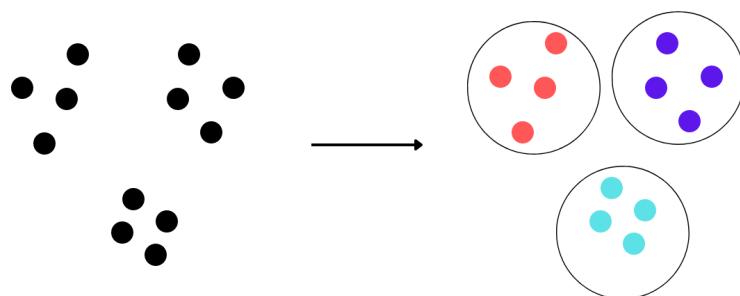


FIGURE 4.1 – Décomposition d'un ensemble d'objets en trois clusters

4.2.1.1 Modèle mathématique

Le modèle mathématique suivant a été présenté par RAO [49], l'objectif de ce modèle est de minimiser les distances entre les sommets dans chaque cluster.

Les données suivantes sont utilisées dans le modèle :

d_{ij} : la distance entre les deux clients i et j

n : nombre de clients

K : nombre de clusters

Variable de décision :

$$x_{ik} = \begin{cases} 1, & \text{si le client } i \text{ est affecté au cluster } k \\ 0, & \text{sinon} \end{cases} \quad (4.1)$$

$$\text{Minimiser } z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} \sum_{k=1}^K x_{ik} x_{jk} \quad (1)$$

$$\sum_{k=1}^K x_{ik} = 1; \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^n x_{ik} \geq 1; \quad k = 1, 2, \dots, K \quad (3)$$

$$x_{ik} \in \{0, 1\}; \quad i \in \{1 \dots n\}, k \in \{1 \dots K\}$$

- La fonction objective (1) permet de minimiser les distances entre les clients dans le cluster.
- La contrainte (2) indique que chaque client est affecté exactement à un seul cluster.
- La contrainte (3) indique que chaque cluster doit contenir au moins un client.

4.2.2 Algorithme de K-means

Il existe plusieurs méthodes de *clustering* telles que K-means [36] et K-medoids [29] qui sont couramment utilisées pour résoudre des problèmes de regroupement de données. Toutefois, parmi ces méthodes, K-means est la plus populaire et la plus largement utilisée.

K-means a été proposé par MacQueen en 1967, La méthode cherche à partitionner un ensemble de données en k *clusters* où chaque point de données appartient à un *cluster* dont le centre (appelé centroïde) est le plus proche. L'objectif est de minimiser les distances entre les sommets et leurs centres dans chaque *cluster*.

L'algorithme est le plus utilisé pour :

- Faible temps de calcul à cause de sa complexité linéaire qui est $O(nkd)$ où n est le nombre d'objets, k est le nombre de *clusters* et d est la dimension de l'espace de définition des objets.
- Son efficacité même avec de grands ensembles de données, car il peut facilement être parallélisé pour traiter des données en parallèle.
- Les résultats sont très faciles à interpréter. K-Means génère des descriptions de *cluster* sous une forme minimisée pour maximiser la compréhension des données.

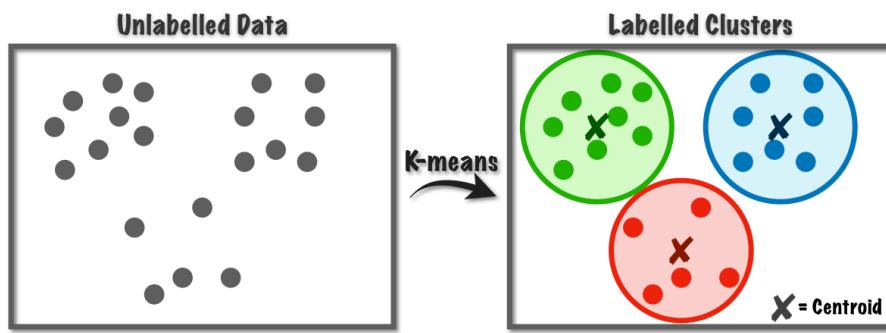


FIGURE 4.2 – Décomposition d'un ensemble d'objets en trois *clusters* en utilisant l'algorithme K-means

Algorithm 1 Algorithme de k-means

1. Initialisation : Choisir K centres de manière aléatoire et les considérer comme des centres.
 2. Attribution des clients : Assigner chaque client au cluster dont le centroïde est le plus proche. Ceci peut être fait en calculant la distance euclidienne entre chaque point de données et chaque centroïde de cluster.
 3. Calcul des nouveaux centres de cluster : Recalculer les centres de cluster en tant que moyenne de tous les clients appartenant à ce cluster.
 4. Répéter les étapes 2 et 3 jusqu'à ce que les centres de cluster convergent vers une solution stable ou les clients ne changent plus de cluster.
-

4.2.3 Description du problème de partitionnement avec contrainte de capacité

Le problème de partitionnement avec contrainte de capacité divise les sommets du graphe non orienté en plusieurs *clusters* disjoints de sorte que la somme des poids des nœuds dans chaque *cluster* respecte la limite de capacité.

[35] Le problème peut être utilisé dans le problème de tournées de véhicules pour diviser les villes en zones en fonction de leur emplacement et de leurs besoins en termes de capacité, cela permet de garantir que les véhicules ne dépassent pas leur capacité maximale pendant la tournée de livraison.

Le modèle mathématique suivant est une extension du modèle mathématique précédent avec une contrainte de capacité tenue en compte.

4.2.3.1 Modèle mathématique

d_{ij} : La distance entre les deux clients i et j

n : Nombre de clients

K : Nombre de clusters

q_i : La quantité de demande dans chaque sommet i

Q : la quantité à ne pas dépasser dans chaque cluster

Variable de décision :

$$x_{ik} = \begin{cases} 1, & \text{si le client } i \text{ est affecté au cluster } k \\ 0, & \text{sinon} \end{cases} \quad (4.2)$$

$$\text{Minimiser } z = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} \sum_{k=1}^K x_{ik} x_{jk} \quad (1)$$

$$\sum_{k=1}^K x_{ik} = 1; \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^n x_{ik} \geq 1; \quad k = 1, 2, \dots, K \quad (3)$$

$$\sum_{i=1}^n x_{ik} q_i \leq Q \quad k = 1, 2, \dots, K \quad (4)$$

$$x_{ik} \in \{0, 1\}; \quad i \in \{1, \dots, n\}, k \in \{1, \dots, K\}$$

La contrainte (4) signifie que la somme des demandes des clients dans un cluster ne doit pas dépasser quantité Q

Évaluation du modèle :

Nombre de variables

- Variable du type $x_{ik} = n \cdot K$

Nombre total de variables : $n \cdot K$

Nombre de contraintes

- Contrainte du type (2) : n

- Contrainte du type (3) : K

- Contrainte du type (4) : K

Nombre total de contraintes : $n + 2 \cdot K$

Complexité du problème : Le problème de partitionnement avec contrainte de capacité est considéré comme un problème d'optimisation combinatoire NP-Difficile [63]. Cela signifie que le temps nécessaire pour trouver une solution optimale augmente de manière exponentielle avec la taille de l'instance du problème, d'où la nécessité de le résoudre par des méthodes approchées.

4.2.3.2 Méthode de résolution

Le problème de partitionnement avec contrainte de capacité peut être résolu en utilisant la méthode de K-means avec capacité. Cette méthode est une extension de la méthode de K-means, à la différence qu'elle ajoute une contrainte de capacité à la formulation du problème. Ainsi, lors de l'attribution des points aux *clusters*, la capacité de chaque *cluster* est également prise en compte pour s'assurer que la somme des demandes des sommets n'excède pas la capacité maximale du *cluster*.

Algorithm 2 Algorithme de K-means avec capacité

1. Initialisation : Choisir K premiers clients qui ont le plus de demandes et les considérer comme des centres.
 2. Attribution des clients : Assigner chaque client au *cluster* dont le centroïde est le plus proche, et la somme des demandes des sommets ne dépasse pas la capacité. Ceci peut être fait en calculant la distance euclidienne entre chaque point de données et chaque centroïde de cluster.
 3. Calcul des nouveaux centres de cluster : Recalculer les centres de cluster en tant que moyenne de tous clients appartenant à ce cluster.
 4. Répéter les étapes 2 et 3 jusqu'à ce que les centres de cluster convergent vers une solution stable ou les clients ne changent plus de cluster.
-

4.3 Modélisation et résolution du problème de tournées

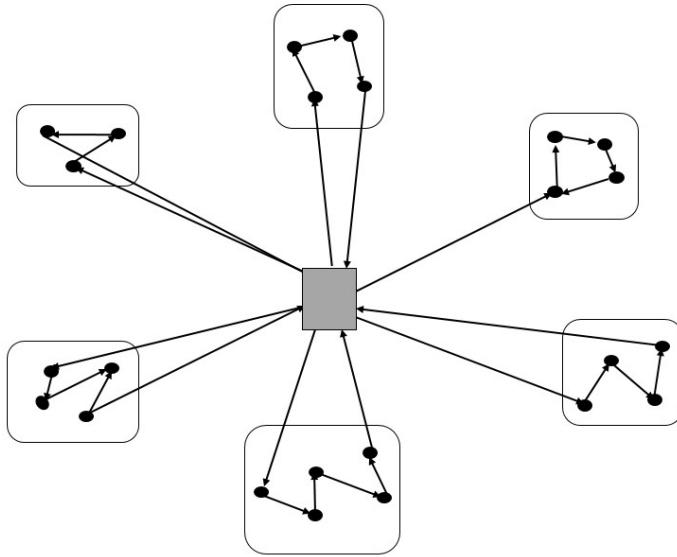


FIGURE 4.3 – Figure illustrant une tournée de *CluVRP* adapté au modèle du problème proposé

Après une analyse détaillée de notre problème et une revue littéraire qui concerne les problèmes de tournées de véhicules nous avons constaté qu'il peut être modélisé en tant qu'un problème de tournées de véhicule avec *cluster CluVRP* en modifiant la contrainte (7). Dans notre cas de figure le véhicule ne peut pas visiter plus qu'une zone. Cela nous amène à mettre l'égalité de la contrainte à zéro tout en écartant le *cluster* contenant le dépôt, ce qui va empêcher le véhicule de visiter plusieurs zones. La formulation mathématique du problème alors devient :

$$\text{Minimiser} \quad \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \quad (1)$$

$$\sum_{k=1}^m y_{ik} = 1 \quad i = 2, \dots, n \quad (2)$$

$$\sum_{k=1}^m y_{1k} = m \quad (3)$$

$$\sum_{j=1}^n x_{ij}^k = \sum_{j=1}^n x_{ji}^k = y_{ik} \quad k = 1 \dots m \quad i \in V \quad (4)$$

$$\sum_{i=2}^n q_i y_{ik} \leq Q \quad k = 1 \dots m \quad (5)$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij}^k \geq y_{hk} \quad \forall S \subseteq V \setminus V_0 \quad h \in S \quad k = 1 \dots m \quad (6)$$

$$\sum_{(i,j) \in \delta^+(Cr)} \sum_{k=1}^m x_{ij}^k = \sum_{(i,j) \in \delta^-(Cr)} \sum_{k=1}^m x_{ij}^k = 0 \quad \forall r \in R \setminus R_0 \quad (7)$$

$$y_{ik} = y_{jk} \quad \forall (i,j) \in Cr \quad \forall r \in R \quad k = 1 \dots m \quad (8)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i,j) \in \{1 \dots n\} \quad \forall k = 1 \dots m \quad (9)$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in \{1 \dots n\} \quad \forall k = 1 \dots m \quad (10)$$

4.3.1 Évaluation du modèle

Nombre de variables

- Variable du type $y_{ik} : n \times m$
- Variable du type $x_{ij}^k : n^2 \times m$
- Nombre totale de variables : $n \times m + n^2 \times m$

Nombre de contraintes

- Contrainte du type (2) : $n - 1$
- Contrainte du type (3) : 1
- Contrainte du type (4) : $n \times m$
- Contrainte du type (5) : m
- Contrainte du type (6) : $m \times 2^{n-1} \times |S|$
- Contrainte du type (7) : $(r - 1)$
- Contrainte du type (8) : $m \times r \times \frac{n^2 - 3n + 2}{2!}$
- Nombre totale de contraintes : $n + r - 1 + m \times (n + 1 + r \times \frac{n^2 - 3n + 2}{2!} + 2^{n-1} \times |S|)$

Nous avons conclu que la résolution du problème de tournées ne fait pas l'objet d'utilisation de méthode exactes.

Il est important de souligner qu'il n'existe pas de heuristique /métaheuristique "idéale" pour résoudre un problème donné, car l'efficacité de chaque algorithme dépend de l'instance du problème spécifique et de sa mise en oeuvre. En pratique il est souvent utile d'expérimenter plusieurs métaheuristiques et de comparer leur résultats sur la même instance. De ce fait et d'après la revue littéraire présentée dans le chapitre précédent, nous avons opté pour tester les deux métaheuristiques, algorithme de "Golden Ball" et l'algorithme de "système de colonie de fourmis". Nous avons choisi ces deux méthodes pour observer et comprendre le comportement qu'elles adoptent lorsqu'elles sont appliquées aux mêmes instances. Notre motivation principale était la curiosité de comparer et d'analyser les résultats obtenus par ces deux approches.

4.3.2 Algorithme de Colonie de fourmis

L'algorithme de colonie de fourmis (Ant Colony Optimization) a été créé par Dorigo et Di Caro en 1999 [12]. Cette méthode s'inspire du comportement des fourmis lorsqu'elles trouvent de la nourriture. Elles sécrètent instinctivement du phéromone, ce qui permet aux autres fourmis de suivre la trace jusqu'à la source de nourriture.

Dans l'algorithme ACO appliqué au **TSP**, un ensemble de solutions est considéré comme des fourmis artificielles. Les solutions les plus prometteuses laissent du phéromone sur les arêtes, ce qui favorise la sélection de ces solutions lors de la recherche des prochaines solutions 4.4.

La loi de transition :

La fourmi k dans la ville i choisit la ville j selon la loi de transition suivante :

$$p_{ij}^k = \begin{cases} \frac{\tau(i,j)^\alpha \eta(i,j)^\beta}{\sum_{u \in J_k(i)} \tau(i,u)^\alpha \eta(i,u)^\beta}, & \text{si } j \in J_k(i) \\ 0, & \text{sinon} \end{cases}$$

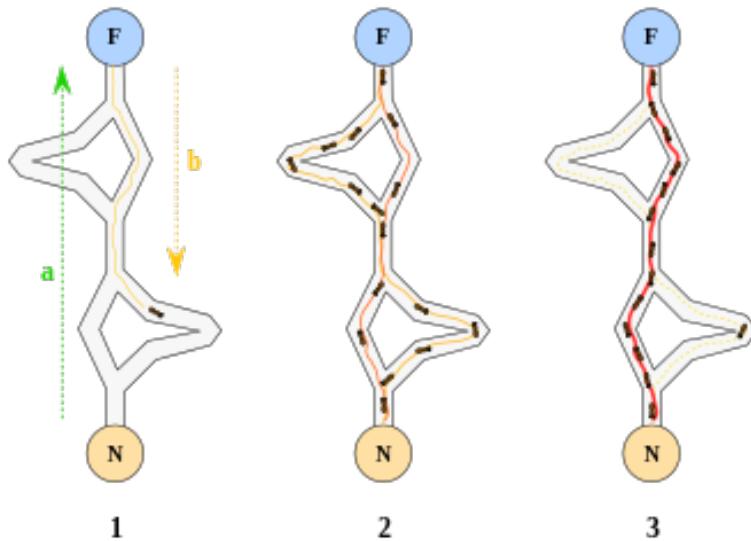


FIGURE 4.4 – Colonie de fourmis

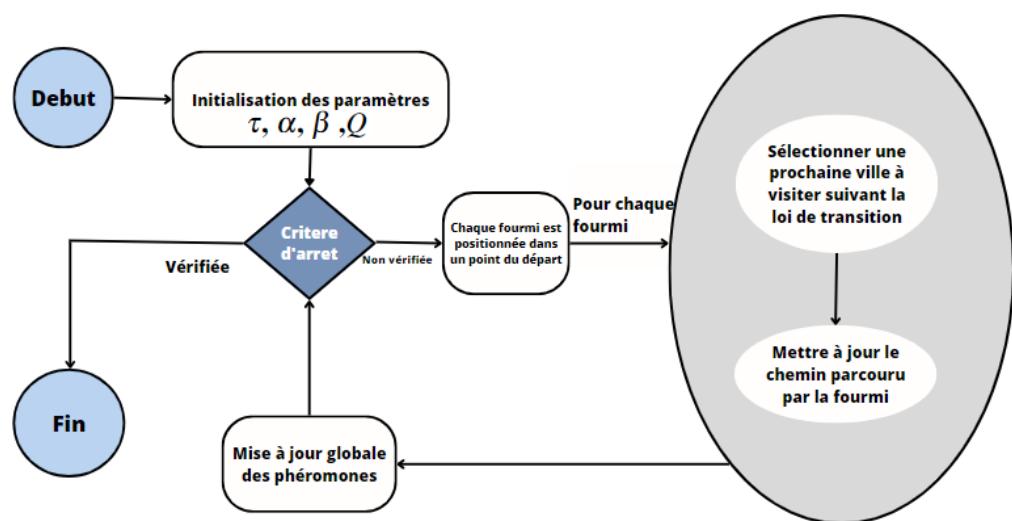


FIGURE 4.5 – Organigramme de la météahuristique colonie de fourmis

- $J_k(i)$ est l'ensemble de voisinage du client i .
- $\tau(i, j)$ est la quantité du phéromone de l'arc (i, j) .
- $\eta(i, j)$ est l'inverse de $d(i, j)$ qui représente la visibilité de l'arête (i, j) , où $\eta(i, j) = \frac{1}{d(i, j)}$, d est la matrice des distances.
- α et β sont les deux paramètres principaux qui contrôlent l'algorithme, $0 \leq \alpha$ contrôle l'influence du phéromone τ , $1 \leq \beta$ contrôle l'influence de la visibilité η

Mise à jour globale des phéromones Une fois que toutes les fourmis ont construit leurs tours, le phéromone est mis à jour selon la règle suivante :

$$\tau(i, j) = (1 - \rho) * \tau(i, j) + \sum_{m=1}^k \Delta\tau(i, j)^k$$

$$\Delta\tau(i, j)^k = \begin{cases} \frac{Q}{L_k}, & \text{si } (i, j) \text{ appartient à La tournée faite par la fourmi } k \\ 0, & \text{sinon} \end{cases}$$

Où $0 \leq \rho \leq 1$ est un paramètre de réglage du phéromone qui évite l'accumulation illimitée des traces de phéromones et permet à l'algorithme d'oublier les mauvaises décisions prises précédemment. En effet, si un arc n'est pas choisi par les fourmis, sa valeur de phéromone associée diminue de façon exponentielle avec le nombre d'itérations. L_k est la distance de la tournée faite par la fourmi k , Q est une constante.

Système de colonie de fourmis

L'algorithme du système de colonie de fourmis (*Ant colony system*) est une variante de l'algorithme de colonie de fourmis. ACS diffère de ACO dans trois points suivants [12] :

1. il exploite l'expérience de recherche accumulée par les fourmis plus fortement que l'ACO fait grâce à l'utilisation d'une règle de choix d'action plus agressive.
2. La mise à jour des phéromone n'a lieu que sur les arcs appartenant à la meilleure tournée à ce jour.
3. Chaque fois qu'une fourmi utilise un arc (i, j) pour se déplacer du client i au client j , Une mise à jour local du phéromone sur l'arc est effectuée

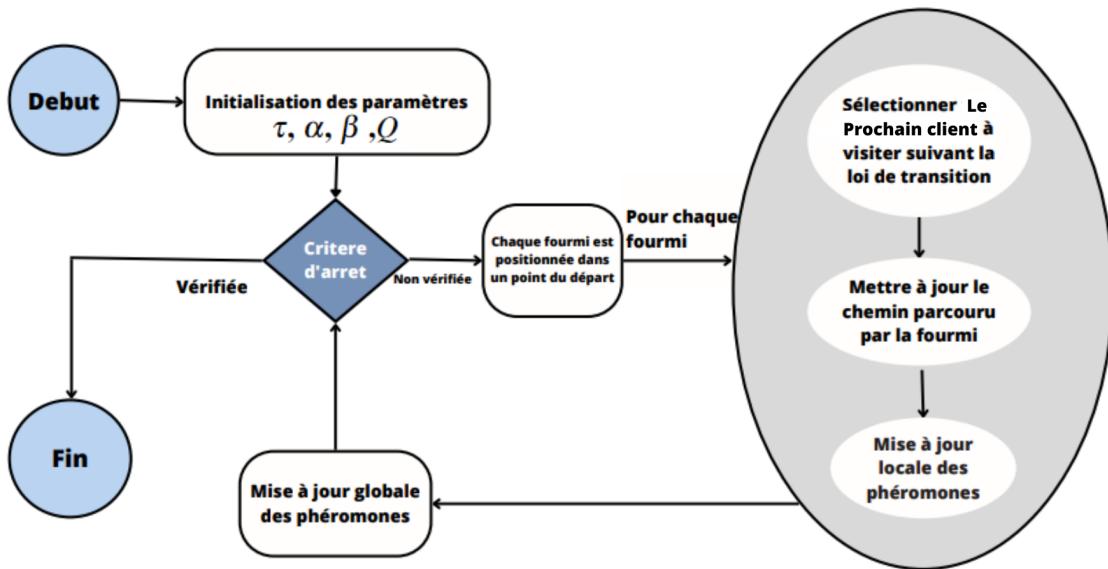


FIGURE 4.6 – Organigramme de la mét-heuristique système de colonie de fourmis

La loi de transition :

La fourmi k chez le client i choisit le client j selon la loi de transition suivante :

$$j = \begin{cases} argmax_{u \in J_k(r)} [\tau(i, u) \eta(i, u)^\beta], & si \quad q \leq q_0 \\ S, & sinon \end{cases}$$

Où $0 \leq q \leq 1$ est un nombre aléatoire, $0 \leq q_0 \leq 1$ est un paramètre fixe, S est la ville choisie en utilisant la probabilité donnée dans ACO.

Mise à jour globale des phéromones :

Une fois que toutes les fourmis ont construit leurs tours, contrairement à l'algorithme ACO seule la meilleure solution obtenue permet de mettre à jour les phéromones :

$$\tau(i, j) = (1 - \rho) * \tau(i, j) + \rho \Delta\tau(i, j)^{best}$$

$$\Delta\tau(i, j)^{best} = \begin{cases} \frac{Q}{L_{best}}, & si (i, j) appartient à la meilleure solution obtenue \\ 0, & sinon \end{cases}$$

Où L_{best} est la longueur de la meilleure tournée obtenue.

Mise à jour locale des phéromones :

En plus de la règle globale de mise à jour des traces de phéromones, dans ACS, les fourmis utilisent une règle locale de mise à jour des phéromones qu'ils appliquent immédiatement après avoir traversé un arc (i, j) pendant la construction de la tournée :

$\tau(i, j) = (1 - \delta)\tau(i, j) + \delta\tau_0$ Où $0 \leq \delta \leq 1$ et τ_0 sont des paramètres fixes. La valeur de τ_0 est la valeur initiale des phéromones. D'après les expériences, la meilleure valeur pour δ est 0.1, et pour τ_0 est $\frac{1}{n*L_{nn}}$, n est le nombre de sommets, L_{nn} est la longueur de la tournée obtenue en utilisant l'algorithme du plus proche voisins.

L'effet de la règle de mise à jour locale est qu'à chaque fois qu'une fourmi utilise un arc (i, j) La quantité du phéromone $\tau_{i,j}$ est réduite, de sorte que l'arc devient moins souhaitable pour la suite. Cela permet d'augmenter

l'exploration des arcs qui n'ont pas encore été visité et, en pratique, à pour effet que l'algorithme ne montre pas de comportement de stagnation [12].

4.3.3 Algorithme de Golden Ball

Introduite par (**E.Osaba et al 2014**), Golden Ball est une métaheuristique inspiré du célèbre jeu de Football est basée sur des populations multiples. En premier dans la phase d'initialisation cette méthode crée une population de solutions réalisables appelés *Joueurs*, cette population sera divisé en plusieurs sub-populations appelées (*équipes*). Une fois que cette phase est terminée, la première saison commence, une saison est divisée en plusieurs semaines dans lesquelles les équipes s'entraînent et s'affrontent en créant une compétition de ligue. Une fois que la saison est finie, la phase des transferts commence dans laquelle les joueurs et les entraîneurs peuvent changer d'équipe, comme illustré dans la figure 4.7. Ce processus est répété jusqu'à ce que le critère d'arrêt soit rencontré. Dans la prochaine section nous présentons en détails les étapes de cette méthode.

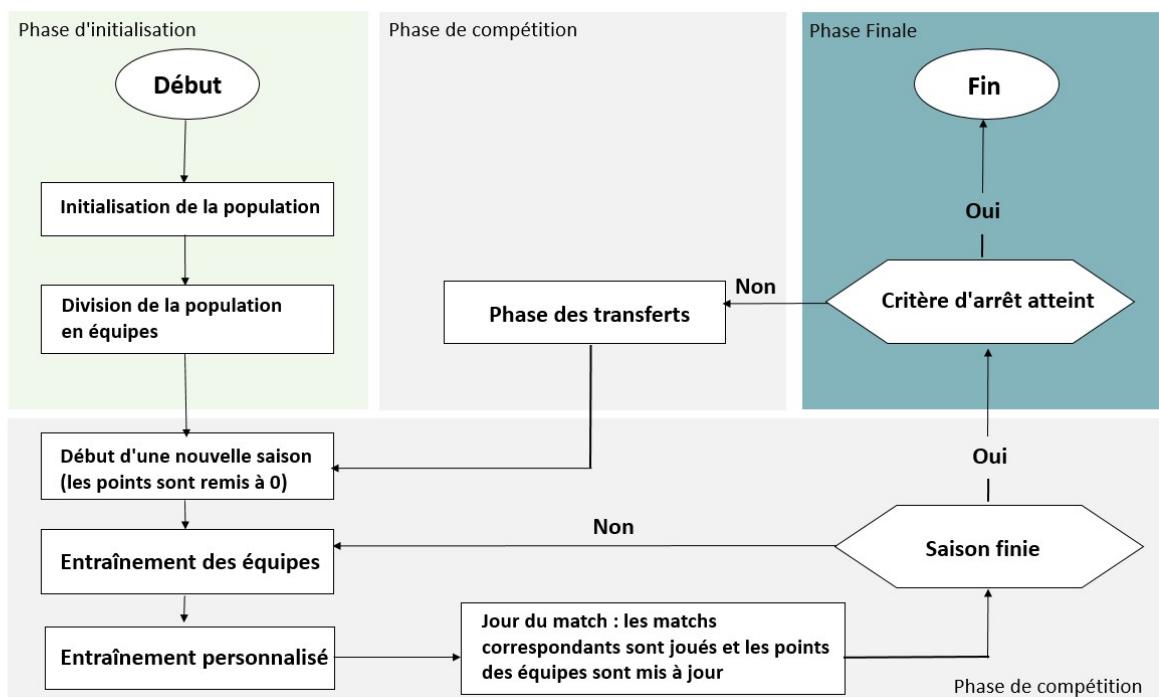


FIGURE 4.7 – Organigramme de la métahéuristique GB

Phase d'initialisation

La première étape de cette phase consiste en la création d'un ensemble de solution réalisable appelé (P). Ces solutions sont soient créées aléatoirement soient créées par une heuristique de construction, où chaque solution est appelée joueur (p_i).

$$P = \{p_1, p_2, p_3, \dots, p_{TN*PT}\}$$

où TN représente le nombre totale d'équipes dans une le système et PT le nombre de joueur dans une équipe.

Une fois que les solutions sont générées, Les $TN * PT$ joueurs sont divisés aléatoirement en TN équipes. Une fois que cette opération est accomplit, un joueur sera donc indiqué par p_{ij} , le joueur j appartenant à l'équipe i .

$$T = \{t_1, t_2, t_3, \dots, t_{TN}\}$$

$$t_1 = \{p_{1,1}, p_{1,2}, p_{1,3}, \dots, p_{1,PT}\}$$

$$t_2 = \{p_{2,1}, p_{2,2}, p_{2,3}, \dots, p_{2,PT}\}$$

.....

$$t_{TN} = \{p_{TN,1}, p_{TN,2}, p_{TN,3}, \dots, p_{TN,PT}\}$$

Chaque joueur $p_{i,j}$ a son propre *fitness* qui sera représenté par la variable $q_{i,j}$, représentant un nombre réel calculé par une fonction objectif $f(p_{i,j})$ dans le cas du **TSP** cette fonction est équivalente à la somme des distances entre les villes de la solution. De plus chaque équipe t_i possède un capitaine représenté par $p_{i,\text{cap}}$ dont le *fitness* est le meilleure de son équipe.

$$p_{i,\text{cap}} = p_{ik} \in t_i \Leftrightarrow \forall j \in \{1, \dots, PT\} : q_{ik} \geq q_{ij}.$$

En outre, Chaque équipe a une valeur de sa force associée par TQ_i qui est égale à la moyenne des *fitness* de tous ses joueurs.

$$TQ_i = \frac{\sum_{j=1}^{PT} q_{ij}}{PT}$$

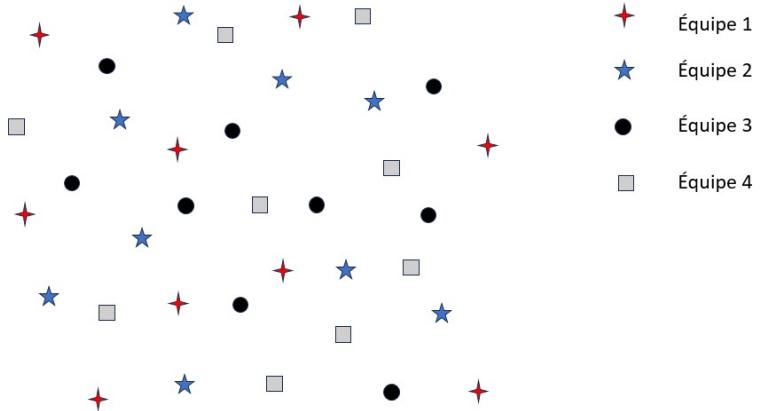


FIGURE 4.8 – Illustration de l'espace des solutions dans le cas de $NT = 4$

Phase de compétition

Cette phase est composée de saisons où les équipes s'entraînent d'une manière indépendante et coopérative pour améliorer leurs TQ_i afin de s'affronter en créant une compétition de ligue qui aide à décider des transferts de joueurs entre différentes équipes. Chaque saison est divisée en deux périodes de durée identique afin que toutes les équipes s'affrontent exactement deux fois, chaque équipe aura autant d'entraînement que de matchs. Le processus d'une saison est montré dans l'algorithme 3.

Méthodes d'entraînement

Dans cette phase chaque équipe sera sujette à un entraînement spécifique afin d'améliorer les performances de ses joueurs et ainsi gagner plus de matchs. De cette façon la qualité de l'entraînement influe sur le classement des équipes en fin de saison. Une méthode d'entraînement est une fonction qui travaille sur une certaine structure du

Algorithm 3 Pseudocode de la procédure d'une saison

```

0: Les points de chaque équipe  $t_i$  sont remis à 0;
0: for  $j = 1, 2$  {Chaque saison est divisée en deux période égale comme dans la vraie vie } do
0:   for chaque journée de match do
0:     for chaque équipe  $t_i$  dans le système système do
0:       Entrainement pour  $t_i$ ;
0:       Entrainement personnalisé pour  $t_i$ ;
0:       Calcule du  $TQ_i$  de  $t_i$ ;
0:     end for
0:     Journée de match;
0:   end for
0:   Période de transferts des joueurs;
0:   Période de transferts des méthodes d'entraînement;
0: end for=0
```

voisinage, dans le cas d'un problème de tourné cette fonction peut se présenter sous forme d'une des célèbres recherches locales, **2-opt**, **3-opt**, **swapping** etc. Cette Méthode peut être appliquée un certain nombre de fois, jusqu'à ce qu'un critère d'arrêt est atteint pour entraînement p_{ij} , p_{ij}' sera retenu après l'entraînement seulement si p_{ij}' est meilleur que p_{ij} . Ainsi chaque équipe examine d'une manière différente le voisinage des solutions qu'elle possède, les joueurs vont de cette manière évoluer tout en étant dépendant de l'équipe à laquelle ils appartiennent, ce qui va donc favoriser l'exploration et l'exploitation de l'espace des solutions, qui vont par la suite s'améliorer par les transferts des deux, joueurs et méthodes d'entraînement. L'algorithme 4 montre le processus d'un entraînement. Il convient de noter que l'augmentation du nombre d'itérations d'entraînement entraîne une augmentation du temps d'exécution. Par conséquent, il peut ne pas être avantageux de se concentrer sur l'augmentation du nombre d'itérations, car cela peut conduire à un minimum local.

D'autre part, l'entraînement personnalisé est différent, ce dernier s'applique lorsque une solution n'a pas évolué dans la phase d'entraînement précédente. Cette situation peut survenir dans la vie réelle lorsqu'un joueur continue à présenter de faibles performances, même après avoir été soumis à des entraînements de qualité supérieure, en raison peut-être d'une fatigue physique ou d'autres facteurs. En optimisation combinatoire ce phénomène peut être traduit par le blocage de la solution p_{ij} dans un optimum local. Afin d'échapper la solution de cette situation, la méthode de *GB* propose des méthodes d'entraînement personnalisées qui vont se dérouler entre deux solutions, le joueur p_{ij} et le capitaine d'équipe $picap$. (cette méthode peut se présenter comme le célèbre opérateur de *croisement* dans l'algorithme génétique). Ainsi ce type d'entraînement aide la méthode au phénomène de l'exploration et donc diversifier les solutions qu'elle possède.

Matchs

Comme dans la vraie vie, un match s'opère entre deux équipes. Plus précisément chaque équipe va devoir trier ses joueurs selon un ordre décroissant de leurs *fitness*, ainsi chaque joueur va affronter un autre joueur de l'équipe adverse, et le joueur avec le meilleur *fitness* marque un But. De ce fait l'équipe qui marque le plus de buts gagne le match. De plus, l'équipe qui gagne le match obtient 3 points et l'autre obtient 0 points; en cas d'égalité de buts les deux équipes obtiennent chacune 1 points. L'algorithme 5 montre le processus d'un match.

Transferts

Algorithm 4 Pseudo-code d'un processus d'entraînement

```

0: Compteur ← 0
0: while Compteur < critred'arrêt do
0:   Créer un nouveau joueur ( $p_{ij}$ ) à partir de  $p_{ij}$  en utilisant la fonction d'entraînement
0:   if  $q_{ij} > q(p_{ij})$  then
0:      $p_{ij} \leftarrow p'_{ij}$  (remplace  $p_{ij}$  par  $p'_{ij}$ )
0:     Compteur ← 0
0:   else
0:     Compteur ← Compteur + 1
0:   end if
0: end while=0

```

Algorithm 5 Procédure d'un match

```

0: ButsEquipe1 ← 0
0: ButsEquipe2 ← 0
0: for chaque joueur d'une équipe  $PT$  do
0:   if  $q_{1i} > q_{2i}$  (Le joueur de la première équipe est meilleur) then
0:     ButsEquipe1 ← ButsEquipe1 + 1
0:   else if  $q_{1i} < q_{2i}$  (Le joueur de la deuxième équipe est meilleur) then
0:     ButsEquipe2 ← ButsEquipe2 + 1
0:   end if
0: end for=0

```

Au cours de cette étape de l'algorithme, les équipes échangent des joueurs à chaque période de la saison. Comme on peut s'y attendre, les meilleures équipes attirent les meilleurs joueurs, tandis que les équipes moins performantes attirent des joueurs moins bons. Tout d'abord, les équipes sont triées par ordre décroissant des points obtenus lors des matchs joués. Ensuite, le joueur ayant le moins bon *fitness* de la meilleure équipe est échangé avec le joueur ayant le meilleur *fitness* de la dernière équipe. De même, le pire joueur de la deuxième équipe est échangé avec le meilleur joueur de l'équipe avant-dernière, et ainsi de suite jusqu'à ce que toutes les équipes soient parcourues. Si le nombre NT est impair, l'équipe située au milieu ne subira aucun changement. Cette phase, telle que décrite par les auteurs [42], est très importante pour l'exploration de l'espace de recherche.

La méthode comprend également un autre type de transfert appelé "*l'échange spécial*". Dans ce cas, un joueur peut changer d'équipe de manière aléatoire à n'importe quelle période de la saison si ses performances ne s'améliorent pas malgré l'application de l'entraînement conventionnel et l'entraînement personnalisé.

Critère d'arrêt

L'achèvement de l'algorithme dépend de l'accomplissement de trois facteurs. Ce critère doit permettre à l'algorithme d'explorer une grande partie de l'espace de solution. C'est pourquoi le critère développé est constitué de trois clauses d'arrêt :

$$\sum_{i=1}^{TN} q'_{icap} \leq \sum_{i=1}^{TN} q_{icap}$$

$$\sum_{i=1}^{TN} TQ'_{i'} \leq \sum_{i=1}^{TN} TQ_i$$

$$BestSol' \leq BestSol$$

4.4 Conclusion

Dans ce chapitre, nous avons adopté la méthode "Cluster first-route second" pour résoudre le problème de tournées de véhicules. Étant donné que ces deux problèmes sont NP-difficiles, nous avons utilisé des méthodes d'approximation.

Pour le problème de partitionnement, nous avons appliqué la méthode des k-means avec la contrainte de capacité. Cette approche nous a permis de regrouper les clients en clusters tout en respectant les limitations de capacité. Le deuxième problème est un problème de tournées de véhicules avec des clusters. Dans ce cas, chaque véhicule visite exactement un seul cluster et une seule fois. En d'autres termes, nous résolvons le problème du voyageur de commerce au sein de chaque cluster. D'après une revue de la littérature, nous avons constaté que les deux métaheuristiques **Golden ball** et **Système de colonie de fourmis** ont donné de meilleures solutions par rapport aux autres métaheuristiques étudiées, donc nous avons choisi ces deux méthodes pour résoudre **Le problème du voyageur de commerce** dans chaque cluster. Notre objectif était d'utiliser ces approches pour trouver des solutions efficaces pour le problème de partitionnement et le problème de tournées de véhicules avec des clusters.

Chapitre 5

Test et analyse des résultats

5.1 Introduction

Dans ce chapitre, nous effectuerons une étude comparative succincte de deux métaheuristiques : la méthode "golden ball" et l'ACS (Ant Colony System). Ensuite, nous résoudrons le problème en mettant en œuvre ces méthodes dans le langage Python. Afin de faciliter la compréhension de la solution, nous développerons également une interface graphique qui rendra les résultats plus clairs et plus visibles.

5.2 Comparaison des performances des deux métaheuristiques

Dans cette section, une étude comparative des deux algorithmes ACS et GB a été menée sur le problème du **TSP**. Nous avons conduit 10 tests sur chacune des 10 instances obtenus de **TSPLIB** pour observer le comportement des deux méthodes. Le tableau 5.1 présente la valeur moyenne, minimum, maximum et le temps d'exécution moyen de chacun des algorithmes. Sous les configurations expérimentales présenté dans la sous-section 5.2.1

5.2.1 Configuration expérimentale

5.2.1.1 Présentation de python

Python est un langage de programmation de haut niveau, interprété et polyvalent. Il a été créé dans les années 1990 par Guido van Rossum et est aujourd'hui l'un des langages les plus populaires au monde. Python se distingue par sa syntaxe simple et lisible, ce qui facilite l'apprentissage et la compréhension du code.



FIGURE 5.1 – Logo du langage python

Instances		GB				ACS			
Instance	Optimum [50]	Moyenne	Min	Max	Temps (s)	Moyenne	Min	Max	Temps (s)
burma14	3323	3323	3323	3323	0.9	3841	3841	3841	0.12
br17	39	39	39	39	0.21	87	87	87	0.23
ftv33	1286	1501	1480	1558	1.06	1552	1466	1597	3.16
ftv35	1473	1690	1690	1768	1.1	1644	1644	1708	3.8
ftv44	1613	1844	1769	2014	2.2	1875	1862	2014	9.4
ftv55	1608	1971	1961	2005	4.13	2012	2012	2012	22.4
ftv64	1839	2308	2308	2411	6.5	2307	2206	2411	40.8
ftv70	1950	2321	2300	2324	8.6	2484	2408	2484	56.8
Rat99	1211	1252	1230	1327	13.9	1465	1447	1465	299
eil101	629	675	668	681	14.20	725	740	740	254.1

TABLE 5.1 – Résultats pour GB et ACS pour le problème du TSP

5.2.1.2 Matériel utilisé

Nous présentons le matériel utilisé pour l'implémentation de l'algorithme

- **Processeur :** Intel(R) Core(TM) i5-8365U CPU @ 1.60GHz 1.90 GHz
- **RAM :** 8.00 Go

5.2.1.3 Paramètres de ACS

Dans l'algorithme du système de colonie de fourmis nous avons utilisé les paramètres suivants :

$\alpha = 0,1$, $\beta = 2$, $q_0 = 0,9$, $\rho = 0,1$ et $Q = 1$ nous fixons également le nombre de fourmis à $k=3$, et nous définissons le critère d'arrêt comme nombre d'itérations à $N=2$.

5.2.1.4 Paramètres de GB

Dans ce qui suit nous présentons les paramètres utilisés dans l'implémentation de l'algorithme :

Dans le cas du TSP les tournées sont codées par des joueurs.

Création de population : La population a été créée par l'heuristique de construction du plus proche voisin.

Taille de la population : 48

Nombre de joueurs dans une équipe : 12

Nombre d'équipes : 4

Fonctions de recherche locale :

- *2-opt* : Définie par Lin en 1956 le principe de la fonction est de supprimer deux arêtes de la solution actuelle et les remplacer par d'autres.
- *Insertion de sommet (variante 1)* : La fonction consiste à sélectionner un sommet d'une position aléatoire et de le placer dans une position aléatoire d'une autre solution choisie aléatoirement.

- *Swapping* : La fonction consiste à prendre deux sommets au hasard de la solution et échanger leurs positions.
- *Insertion de sommet (variante 2)* : La fonction consiste à sélectionner un sommet d'une position aléatoire et de le placer dans une position aléatoire d'une autre solution choisie aléatoirement. Le nombre d'itération effectué dans cette variante est la seule différence avec la première variante.

Fonction d'entraînement personnalisé :

Dans cette fonction deux solutions prennent partie, le capitaine d'une équipe et un autre joueur appartenant p_{ij} dont les fonctions précédentes n'ont pas contribuer à l'amélioration de son q_{ij} . Nous avons mis en place une fonction de croisement qui sélectionne aléatoirement deux points de coupure dans la solution p_1 et p_2 . La nouvelle solution héritera des sommets situés entre les deux points de coupure de p_{cap} , tandis que le reste du vecteur sera rempli par les sommets compris entre les positions 0 et p_1 d'une part, et entre les positions p_2 et n (où n est la longueur de la solution) d'autre part, provenant du joueur p_{ij} . Il est important de noter que les sommets déjà présents dans la solution ne seront pas inclus à nouveau.

5.2.2 Analyse des résultats

La figure 5.2 représente un graphe comparant les distances moyennes obtenus par les deux algorithmes. Nous remarquons que les deux algorithmes ne présentent pas une différence significative par rapport aux valeurs objectif moyennes. Cependant nous observons que dans la majorité des instances testées, GB a donné des solutions moyennes légèrement meilleures que celles obtenues par ACS.

La figure 5.3 présente une comparaison des meilleures solutions obtenues par les deux algorithmes. Nous observons que GB démontre légèrement une supériorité dans sa capacité à trouver de bonnes solutions.

En ce qui concerne le temps d'exécution, comme illustré dans la figure 5.4, nous constatons que le temps d'exécution augmente de manière quasi exponentielle avec la taille des instances pour ACS, tandis que GB présente une augmentation plus légère.

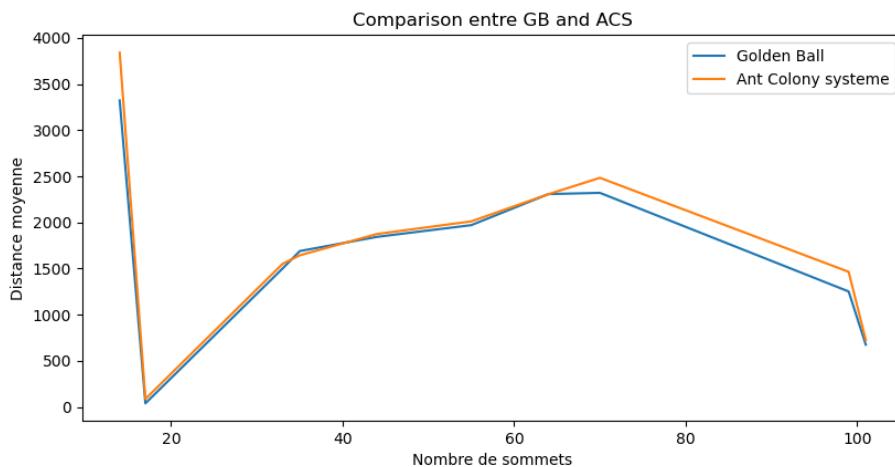


FIGURE 5.2 – Comparaison des distances moyennes obtenues

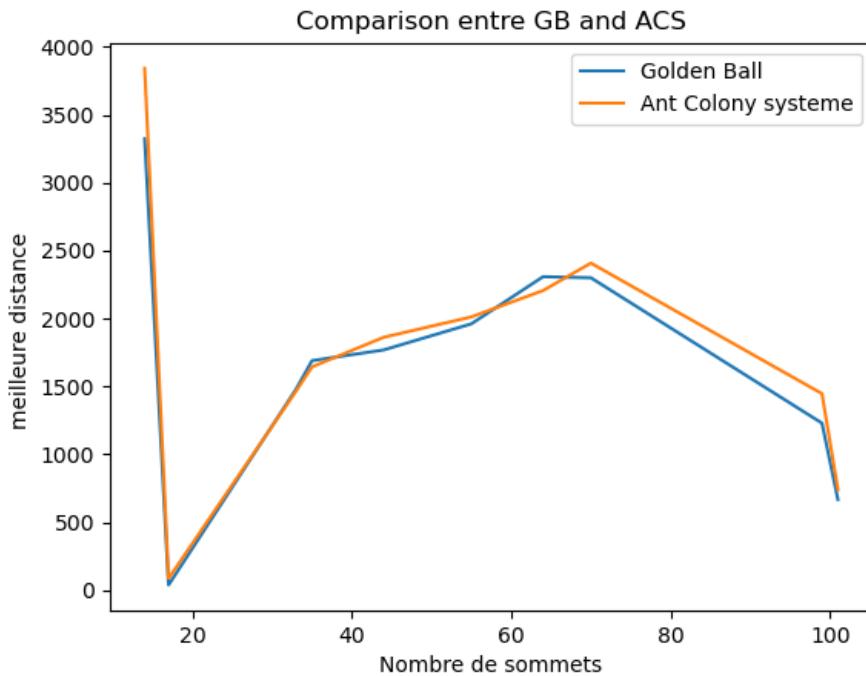


FIGURE 5.3 – Comparaison des meilleures distances trouvées

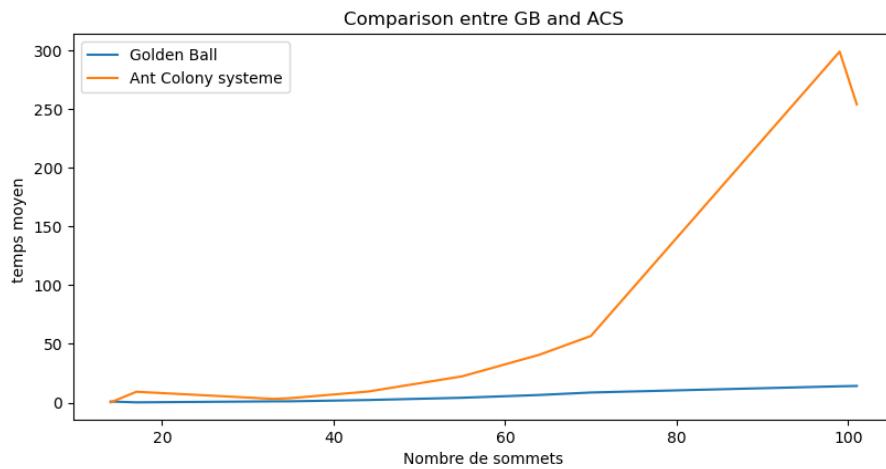


FIGURE 5.4 – Comparaisons du temps d'exécution moyen

5.3 Analyse des résultats du problème traité

5.3.1 Instance du problème

Nous avons rencontré des obstacles lors de nos recherches pour trouver des API ((Application Programming Interface) gratuites qui permettent de calculer des distances réelles pour une instance donnée. Malheureusement, la plupart des API disponibles sont payantes et très chères, ce qui peut ne pas toujours fournir des résultats pertinents, surtout lorsque les sommets de l'instance sont éloignés les uns des autres. De manière avantageuse, la proximité des sommets de l'instance donnée réduit l'influence de la nature des distances sur la solution obtenue. Dans ce qui

suit des distances géographique ont été utilisées pour générer la matrice des distances de l’instance qui suit.

Nous avons testé notre algorithme sur une instance de 229 sommets (clients). La figure 5.5 visualise l’emplacement des clients sur une carte géographique.



FIGURE 5.5 – Emplacement des clients

Partitionnement des clients en clusters

Nous avons utilisé l’algorithme de K-means avec capacité où K est le nombre de livreurs pour résoudre le problème de regroupement des clients en clusters, où les positions des clients sont utilisées comme points de référence et chaque client possède un seul colis. La limite du nombre de colis à affecter à chaque livreur qui est fixé préalablement à 30 colis a engendré 8 clusters. Comme le montre la figure 5.6, La station de livraison est représentée par un marqueur noir sur la carte.

Tournée de chaque cluster

Une fois que les clients ont été partitionnés en clusters, l’algorithme passe à la résolution du problème de tournées spécifique à chaque cluster. Dans cette démarche, le livreur débute son parcours à la station de livraison, traverse le cluster associé, puis revient à la station de livraison. La figure 5.7 montre la visualisation des tournées dans chaque cluster. Nous avons utilisé ACS pour résoudre les tournées de cette instance, car pour des tournées comportant en moyenne 30 sommets, cet algorithme a montré qu’il était capable de fournir de meilleures solutions que GB.

5.3.2 Retour d’information de l’entreprise sur les résultats

Rappelons que l’objectif principal de ce projet était l’amélioration des performances clés liées aux ressources et aux tournées, à savoir la réduction du temps des opérations et des ressources humaines, ainsi que l’amélioration de la qualité du partitionnement et des tournées.

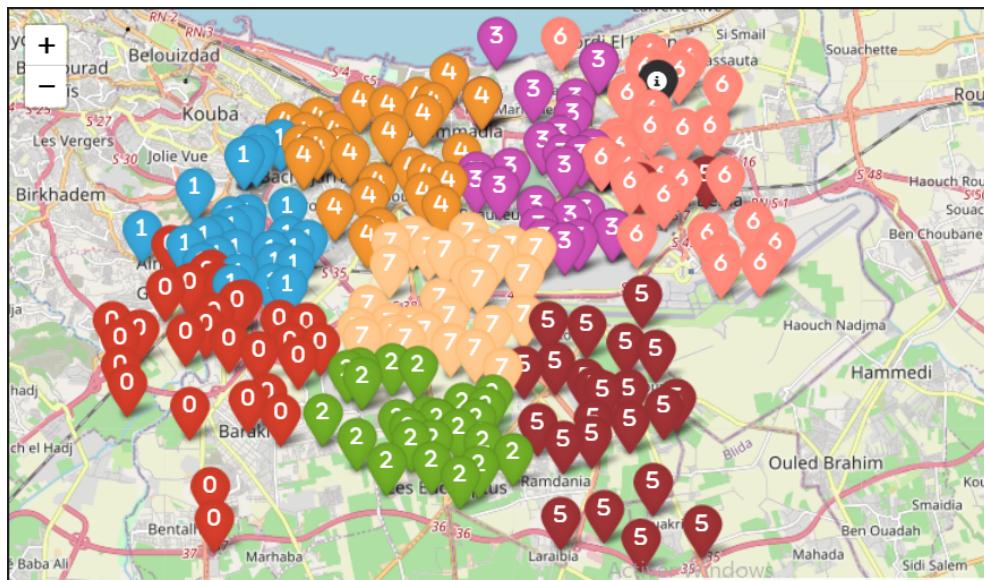


FIGURE 5.6 – Partitionnement des clients en clusters



FIGURE 5.7 – Visualisation des tournées dans chaque cluster

5.3.2.1 Réduction du temps des opérations

La méthode appliquée permettra d'optimiser le temps des opérations de tri (dispatch) des colis au niveau de la station de livraison. Cela est dû à l'automatisation du processus de (dispatch). Notre solution permettra à l'entreprise de réduire considérablement le temps requis pour trier, regrouper et préparer les colis. Auparavant, cette tâche pouvait prendre jusqu'à 2 heures dans une station de livraison avec une charge de travail moyenne. Cependant, grâce à notre nouvelle méthode, le temps nécessaire peut être réduit à seulement 30 à 40 minutes. Cette amélioration contribue à optimiser l'efficacité globale du processus de distribution des colis.

5.3.2.2 Réduction des ressources humaines

En utilisant la solution proposée, l'entreprise pourra réduire le nombre d'opérateurs requis dans une station de livraison de quatre à seulement deux. Cette réduction a été rendue possible grâce à l'automatisation des tâches précédemment effectuées manuellement. Les opérateurs restants sont désormais responsables du remplissage manuel des sacs de livraison pour chaque livreur uniquement (Dans l'annexe A, le lecteur trouvera les détails). Cette réduction des ressources humaines nous permettra d'optimiser les coûts et d'améliorer l'efficacité opérationnelle.

5.3.2.3 Amélioration du partitionnement des clients

La méthode utilisée pour améliorer la qualité des zones de livraison consiste à adopter une approche basée sur des clusters plutôt que sur le regroupement communal. En regroupant les adresses de livraison par proximité géographique, l'entreprise pourra optimiser les itinéraires de livraison et réduire les distances parcourues. Grâce à l'utilisation des données de géo-localisation et des algorithmes d'optimisation, l'entreprise sera en mesure de planifier des tournées de livraison plus efficaces, ce qui se traduit par une réduction significative du temps de livraison, une meilleure satisfaction des clients et un taux de livraison plus élevé.

5.3.2.4 Amélioration des tournées

En appliquant la solution proposée, l'entreprise réussirait à améliorer la qualité des tournées de livraison. L'utilisation des données de géo-localisation et des algorithmes d'optimisation permettra à l'entreprise de planifier les itinéraires plus efficaces, en tenant compte de la densité des livraisons. Cela se traduit par une réduction des retards de livraison et des erreurs de livraison.

5.3.2.5 Amélioration des conditions du travail des livreurs

La solution proposée entraînera une amélioration significative des conditions de travail des livreurs. Grâce à l'optimisation des tournées, les livreurs n'ont plus à se préoccuper de leur prochain client ou de leur destination. Les itinéraires sont désormais planifiés de manière efficace, en tenant compte de la densité des livraisons. Cela permet aux livreurs de mieux organiser leurs tâches, réduisant ainsi le stress et les incertitudes liés à leur travail.

De plus, en utilisant les données de géo-localisation et les algorithmes d'optimisation, les livreurs pourraient éviter de passer de longues heures au volant en empruntant les itinéraires les plus efficaces. Cela réduit la fatigue et le temps passé sur la route, ce qui contribue à améliorer la sécurité des livreurs et à prévenir les accidents.

L'optimisation du service de livraison express améliore les conditions de travail des livreurs. Ils se concentrent mieux sur leurs tâches, améliorent la relation avec les clients grâce à des itinéraires optimisés et des informations bien organisées. Cela entraîne une meilleure qualité de service et fidélise les clients.

5.3.3 Présentation de l'interface graphique

Dans cette sous-section, nous présentons l'interface graphique que nous avons établie. La figure 5.8 présente la page d'accueil dans laquelle l'utilisateur doit sélectionner un fichier Excel contenant les données géographiques des clients, comme le montre la figure 5.10. Une fois cette étape terminée, l'utilisateur doit insérer la capacité maximale des livreurs, puis choisir une méthode de résolution.

Finalement, le nombre de livreurs nécessaires, les tournées et les distances parcourues pour chaque livreur s'afficheront, comme le montre la figure 5.11.

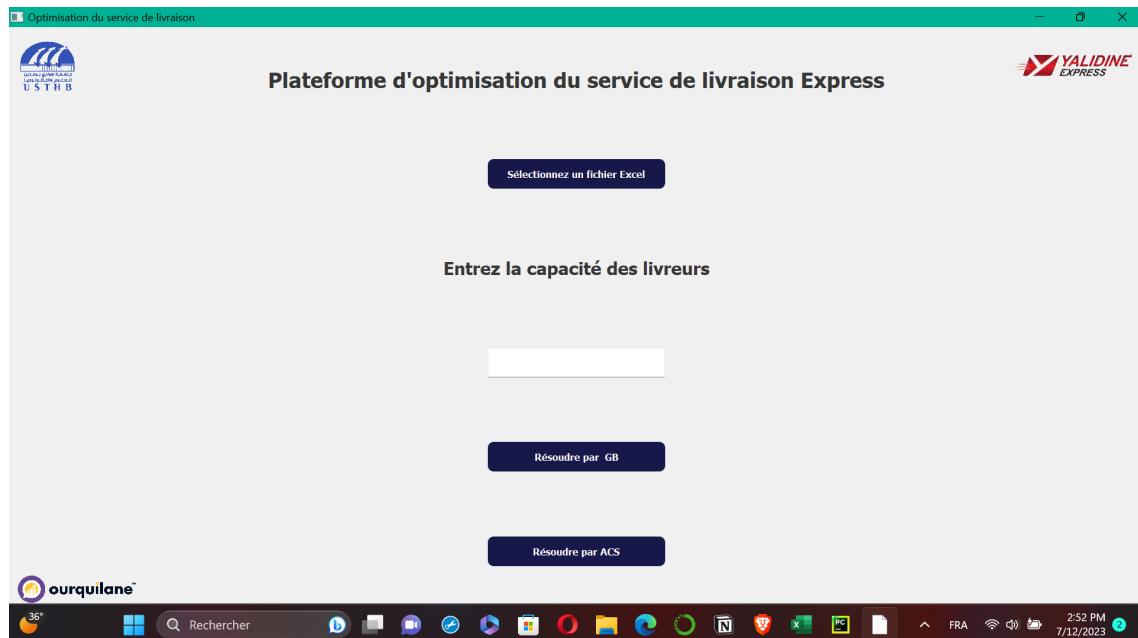


FIGURE 5.8 – Page d'accueil de l'interface graphique

5.4 Conclusion

Après une brève étude comparative entre les deux métaheuristiques, nous avons constaté que ACS et GB ne présentent pas une très grande différence dans la qualité des solutions fournies, mais GB est plus rapide en termes d'exécution. Nous avons partitionné les clients en clusters puis nous les avons visualisé dans une carte géographique, ainsi que l'itinéraire dans chaque cluster, nous avons également développé une interface graphique afin de faciliter l'utilisation des méthodes que nous avons implémenté qui permet d'insérer les coordonnées des clients, ensuite insérer la capacité des livreurs et enfin afficher les tournées de chaque cluster ainsi que les distances parcourues.

Notre solution permettra à l'entreprise de réduire le temps des opérations et les ressources humaines dans une station de livraison. Le partitionnement des clients et les tournées, ainsi que les conditions du travail des livreurs ont été améliorés.

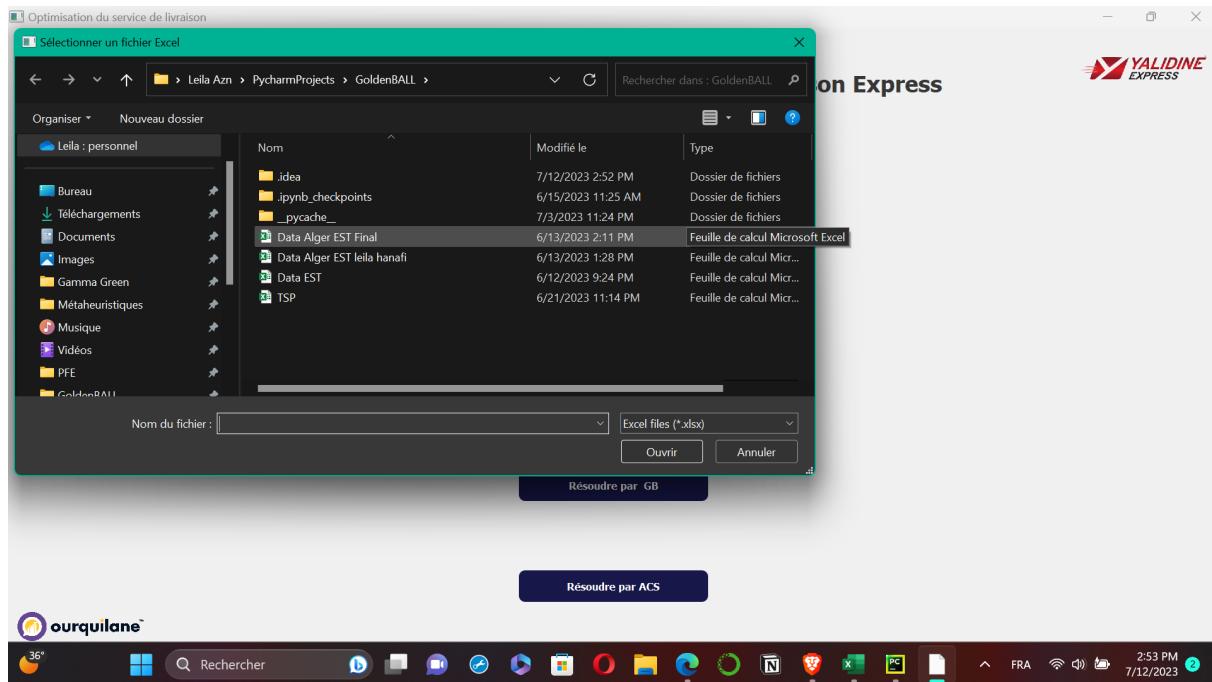


FIGURE 5.9 – Sélection du fichier Excel contenant les données géographiques des clients

Data Alger EST Final - Excel (Produit sans licence)																			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	P	Latitude:	Longitude:	adresse															
2	P-01	36.73252	3.20016	Zone d'Activité Kaidi, Mouhous, Kaidi, Bordj El Kiffan	الكونفديراتي	Dar el-Beida, Alger, 16110, Algeria													
3	P-019	36.68412	3.104716	Ben Ghazi, Baraki, Baraki	براقی، براکی، براکی	Baraki, Alger, 16210, Algeria													
4	P-023	36.69823	3.10263	Gué de Constantine, Djasr Kasentina	خواه کنستانتین، دیسر کاسنتینا	Bir Mourad Rais, Alger, 16048, Algeria													
5	P-026	36.66016	3.124162	جعوى بيرى،	جعوى بيرى،														
6	P-030	36.68363	3.111405	Autoroute Est-Ouest, Ben Ghazi, Baraki, Baraki	طريق السیار شرق-غرب، براکی، براکی، براکی	Baraki, Alger, 16210, Algeria													
7	P-031	36.7069	3.106767	Gué de Constantine, Djasr Kasentina	خواه کنستانتین، دیسر کاسنتینا	Bir Mourad Rais, Alger, 16048, Algeria													
8	P-043	36.63982	3.195979	2e Rocade Sud, Ramdmania, Larabia, Eucalyptus	الدائري الثاني، رامدانيا، لارابيا، إوكالبتوس	Baraki, Alger, 16061, Algeria													
9	P-044	36.68247	3.124157	Raffinerie Sidi Arcine, Branchement A2-Berraki Est	رافيئر سید ارسين، شاخة A2-براكى الشرقي	Baraki, Alger, 16210, Algeria													
10	P-046	36.6804	3.125573	Raffinerie Sidi Arcine, Branchement A2-Berraki Est	رافيئر سید ارسين، شاخة A2-براكى الشرقي	Baraki, Alger, 16210, Algeria													
11	P-047	36.64425	3.17578	Larabia, Eucalyptus	لارابيا، إوكالبتوس	Baraki, Alger, 16057, Algeria													
12	P-048	36.70676	3.118863	SEAAL El Harrach, RN 38, Bourouba, Gué de Constantine	سيال الحراش، الطريق رقم 38، بوروبا، خواه کنستانتین	El Harrach, Bir Mourad Rais, Alger, 16162, Algeria													
13	P-051	36.64255	3.211407	CW 118, Souakria, Meftah	CW 118، سواكريا، مفتاح	Daira Meftah, Blida, Algeria													
14	P-053	36.68677	3.126969	Baraki, Baraki	براكى، براکی	Baraki, Alger, 16210, Algeria													
15	P-054	36.64568	3.186574	NAFTAL Service Station 2, Rocade Sud, Ramdmania, Larabia, Larbâa	نافتل سيرفس ستيشن 2، رامدانيا، لارابيا، لاربا	Daira Larbâa, Blida, 16061, Algeria													
16	P-063	36.70902	3.127273	Boumati, El Harrach	بوماتي، الحراش	El Harrach, Alger, 16162, Algeria													
17	P-064	36.72769	3.132058	Abattoir d'El Harrach, Rue Kettala Benyoucef, Boumati, El Harrach	الحراش، رuelle كتلا بنيوcef، بوماتي، الحراش	El Harrach, Alger, 16122, Algeria													
18	P-067	36.71179	3.129369	Marché de Gros, Impasse Ouled Adda, Boumati, El Harrach	سوق العجمي، حي العجمي، بوماتي، الحراش	El Harrach, Alger, 16200, Algeria													
19	P-070	36.72217	3.132458	El Harrach, Rue Hassen Bey Ali, Boumati, El Harrach	الحراش، رuelle حسن بيك علي، بوماتي، الحراش	El Harrach, Alger, 16200, Algeria													
20	P-072	36.6511	3.199048	Souakria, Meftah	سواكريا، مفتاح	Blida, 16057, Algeria													
21	P-083	36.66314	3.169754	Cherarba, Eucalyptus	چراربا، إوكالبتوس	Baraki, Alger, 16057, Algeria													
22	P-084	36.70096	3.138404	RN 8, El Harrach	الحراش، رقم 8	El Harrach, Alger, 16220, Algeria													
23	P-087	36.68815	3.145285	CW 58, Chaouche	شوش، إوكالبتوس	Baraki, Alger, 16220, Algeria													
24	P-089	36.66083	3.183675	Chanan, Eucalyptus	شنان، إوكالبتوس	Baraki, Alger, 16057, Algeria													

FIGURE 5.10 – Exemple du fichier Excel



FIGURE 5.11 – Affichage des résultats

Chapitre 6

Perspectives et conclusion

Lors de notre projet de fin d'études, nous avons réalisé l'importance cruciale de la recherche opérationnelle dans la gestion de la chaîne logistique mondiale, en particulier dans le domaine de service de livraison. Nous avons développé une solution visant à optimiser ce service, en mettant l'accent sur la définition d'itinéraires que chaque livreur doit suivre. Pour résoudre notre problème, nous avons adapter une variante du problème classique des tournées de véhicules, appelée tournées de véhicule avec *clusters*. Dans cette approche, nous avons regroupé les clients en *clusters* et résolu le problème de la tournée pour chaque cluster individuellement.

Ce projet marque le début du développement de la gestion du dernier kilomètre au sein de l'entreprise Yalidine et il offre potentiellement des opportunités pour des recherches futures plus précises et approfondies. Dans ce contexte, il ouvre la voie à de nouvelles pistes de recherches qui pourraient explorer plus en détails les aspects spécifiques de l'optimisation de la chaîne logistique, en particulier dans les services de livraison. Les perspectives de notre travail sont :

- Utiliser des distances réelles plutôt que des distances à vol d'oiseau permet d'obtenir une optimisation plus précise.
- Tenir en compte les horaires de travail de chaque livreur.
- Disponibilité plus large de données pour prendre en considération d'autres facteurs tel que les heures de pointes, les données liées aux congestions des routes pour une optimisation plus précise.

Annexe A

Annexe

A.1 Rapport de visite : Compréhension de la logistique de livraison au sein d'une station de Yalidine

Station visitée : Yalidine station de livraison Bordj el kiffan

Date : 1 Mars 2023

Heure d'arrivée des livreurs à la station : A partir de 8H00, les livreurs arrivent à la station suivant un planning.

Organisation à l'intérieur de la station :

- La station couvre un certain nombre de communes (Dar el-beida, Bab ezzouar, Mohamadia, El-harach, Bordj el kiffan, Ain taya, Ecalyptus, El-Hamiz, Dergana, Bordj el bahri, Oued semar, El marsa) où chaque commune est représentée par un ou plusieurs casiers (selon la demande de la commune) dans lequel on met les colis destinés à cette dernière. Par la suite, les colis de chaque casier seront mis dans un sac après les avoir intégrés un par un dans l'application. Chaque sac sera affecté à un livreur. Le nombre moyen de livreurs par jour est de 27.

Méthodologie d'affectation des colis aux livreurs :

- Chaque livreur prend en charge les colis destinés à la commune à laquelle il est affecté, il peut livrer une partie des colis d'une autre commune en cas d'absence du livreur de cette dernière. L'affectation est faite manuellement par le chef de la station.

Informations générales sur les livreurs :

- Chaque livreur est désigné par sa capacité (en nombre) de colis dont il est capable de livrer dans la journée, cette information est seulement connue et sauvegardée (manuellement) par le chef de station qui s'en sert

pour effectuer l'affectation, et cela dans le but de maximiser le taux de livraison.

- Les livreurs travaillent 5 jours sur 7, les deux jours de repos sont les vendredis et un autre jour de semaine pour chaque livreur.

Informations générales sur les colis :

- Les colis ne sont pas homogènes (Yalidine ne s'occupe pas de l'emballage des colis).
- La majorité des colis sont d'une taille petite.
- Un colis est caractérisé par son poids, la wilaya, la commune et le quartier de sa destination.
- En cas d'indisponibilité du client le jour de livraison, le nombre de colis non livrés seront comptés dans le sac que le livreur prendra le lendemain.

Heure approximatif du départ des livreurs :

- L'heure estimé du départ du dernier livreur est de 11H00.

Remarques générales :

- l'entreprise souhaite homogénéiser le nombre de colis à affecter aux livreurs.
- La donnée qui désigne le quartier du colis n'est pas exploitée.
- L'opération du chargement d'un sac d'un livreur prend en moyenne 5 minutes.

A.2 Les API's et les bibliothèques utilisés

Dans cette section, nous présentons les librairies que nous avons utiliser pour programmer les algorithmes utilisé et visualiser les données et les tournées.

A.2.1 OpenStreetMap

OpenStreetMap est bâti par une communauté de cartographes bénévoles qui contribuent et maintiennent les données des routes, sentiers, cafés, stations ferroviaires et bien plus encore, partout dans le monde.

A.2.2 GraphHopper

GraphHopper est un puissant outil de routage qui offre des performances rapides et une optimisation économique. Il est capable de calculer avec précision les distances et les temps de trajet, en fournissant des itinéraires détaillés entre deux points ou plusieurs points. GraphHopper peut être utilisé dans comme une bibliothèque python ou un serveur web autonome. GraphHopper utilise des données OpenStreetMap.

A.2.3 Folium

Folium est une bibliothèque utilisable dans plusieurs langages tels que Python, JavaScript et Java. Elle offre la possibilité d'accéder à des cartes géographiques et de créer des visualisations interactives.

A.2.4 Geopy

Geopy est une bibliothèque intégrée au langage Python qui permet de localiser les coordonnées des adresses, des villes et des points de repère à travers le monde. En utilisant Geopy, il est également possible de calculer les distances à vol d'oiseau entre différents lieux.

A.2.5 PyQt5

PyQt5 est une bibliothèque Python populaire pour le développement d'interfaces graphiques (GUI). Elle offre une liaison Python pour la bibliothèque Qt, qui est une plate-forme de développement multiplateforme utilisée pour créer des applications graphiques avancées.

A.2.6 Matplotlib

Matplotlib est une bibliothèque Python populaire utilisée pour la visualisation de données. Elle permet de créer une grande variété de graphiques, diagrammes et visualisations à partir de données numériques.

Références

- [1] Babiche AERTS, Trijntje CORNELISSENS et Kenneth SØRENSEN. “The joint order batching and picker routing problem : Modelled and solved as a clustered vehicle routing problem”. In : *Computers & Operations Research* 129 (2021), p. 105168.
- [2] Arun Prakash AGRAWAL et Arvinder KAUR. “An empirical evaluation of three popular meta-heuristics for solving travelling salesman problem”. In : *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*. IEEE. 2016, p. 16-21.
- [3] PETER BAKER ALAN RUSHTON PHIL CROUCHER. *the handbook of logistics and distribution management 5th edition*.
- [4] John E BEASLEY. “Route first—cluster second methods for vehicle routing”. In : *Omega* 11.4 (1983), p. 403-408.
- [5] Christian BLUM et Andrea ROLI. “Metaheuristics in combinatorial optimization : Overview and conceptual comparison”. In : *ACM computing surveys (CSUR)* 35.3 (2003), p. 268-308.
- [6] O. CARTON et D. PERRIN. *Langages formels, calculabilité et complexité*. Vuibert, 2008. ISBN : 9782711720774.
- [7] Vincent E CASTILLO et al. “Hybrid last mile delivery fleets with crowdsourcing : A systems view of managing the cost-service trade-off”. In : *Journal of Business Logistics* 43.1 (2022), p. 36-61.
- [8] Hongrui CHU et al. “Data-driven optimization for last-mile delivery”. In : *Complex & Intelligent Systems* (2021), p. 1-14.
- [9] Geoff CLARKE et John W WRIGHT. “Scheduling of vehicles from a central depot to a number of delivery points”. In : *Operations research* 12.4 (1964), p. 568-581.
- [10] George DANTZIG, Ray FULKERSON et Selmer JOHNSON. “Solution of a large-scale traveling-salesman problem”. In : *Journal of the operations research society of America* 2.4 (1954), p. 393-410.
- [11] George B DANTZIG. “Maximization of a linear function of variables subject to linear inequalities”. In : *Activity analysis of production and allocation* 13 (1951), p. 339-347.
- [12] Marco DORIGO, Mauro BIRATTARI et Thomas STUTZLE. “Ant colony optimization”. In : *IEEE computational intelligence magazine* 1.4 (2006), p. 28-39.
- [13] Petr DOSTÁL et Pavel POKORNÝ. “Cluster Analysis and Genetic Algorithms”. In : () .
- [14] Michael R. GAREY et David S. JOHNSON. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
- [15] S GEETHA, G POONTHALIR et PT VANATHI. “Improved k-means algorithm for capacitated clustering problem”. In : *INFOCOMP Journal of Computer Science* 8.4 (2009), p. 52-59.

- [16] Nadia GIUFFRIDA et al. "Optimization and Machine Learning Applied to Last-Mile Logistics : A Review". In : *Sustainability* 14 (avr. 2022), p. 5329. DOI : [10.3390/su14095329](https://doi.org/10.3390/su14095329).
- [17] Fred GLOVER. "Tabu search : A tutorial". In : *Interfaces* 20.4 (1990), p. 74-94.
- [18] Fred GLOVER et Manuel LAGUNA. *Tabu search*. Springer, 1998.
- [19] Jack E. GRAVER. "On the foundations of linear and integer linear programming I". In : *Mathematical Programming* 1975-dec vol. 9 iss. 1 9 (1 1975). DOI : [10.1007/bf01681344](https://doi.org/10.1007/bf01681344). URL : libgen.li/file.php?md5=c74aca32349389029817eb335b751fd0.
- [20] Angappa GUNASEKARAN, Nachiappan SUBRAMANIAN et Shams RAHMAN. *Green supply chain collaboration and incentives : Current trends and future directions*. 2015.
- [21] Xiaotong GUO, Baichuan MO et Qingyi WANG. "Amazon Last-Mile Delivery Trajectory Prediction Using Hierarchical TSP with Customized Cost Matrix". In : *arXiv preprint arXiv:2302.02102* (2023).
- [22] Divya GUPTA. "Solving tsp using various meta-heuristic algorithms". In : *International Journal of Recent Contributions from Engineering, Science & IT (iJES)* 1.2 (2013), p. 22-26.
- [23] Sofiane HADDAD. "Le E-Commerce en Algérie : Enjeux et défis à surmonter E-Commerce in Algeria : Issues and challenges to overcome". In : Université d'Alger 03, 2020.
- [24] Pierre HANSEN et al. *Variable neighborhood search*. Springer, 2019.
- [25] John H HOLLAND. "Genetic algorithms". In : *Scientific american* 267.1 (1992), p. 66-73.
- [26] Md Anisul ISLAM, Yuvraj GAJPAL et Tarek Y ELMEKKAWY. "Hybrid particle swarm optimization algorithm for solving the clustered vehicle routing problem". In : *Applied Soft Computing* 110 (2021), p. 107655.
- [27] Li JIANG et al. "A variable neighborhood search for the last-mile delivery problem during major infectious disease outbreak". In : *Optimization Letters* (2022), p. 1-21.
- [28] Li JIANG et al. "A variable neighborhood search for the last-mile delivery problem during major infectious disease outbreak". In : *Optimization Letters* (2022), p. 1-21.
- [29] Leonard KAUFMAN. "Partitioning around medoids (program pam)". In : *Finding groups in data* 344 (1990), p. 68-125.
- [30] James KENNEDY et Russell EBERHART. "Particle swarm optimization". In : *Proceedings of ICNN'95-international conference on neural networks*. T. 4. IEEE. 1995, p. 1942-1948.
- [31] Patrick KLEIN et Bastian POPP. "Last-Mile Delivery Methods in E-Commerce : Does Perceived Sustainability Matter for Consumer Acceptance and Usage ?" In : *Sustainability* 14.24 (2022), p. 16437.
- [32] Alain LACROUX. "Le salarié, grand oublié de la « logistique du dernier kilomètre » ?" In : *Revue Française de Gestion* 44 (nov. 2018), p. 155-158. DOI : [10.3166/rfg.2019.00300](https://doi.org/10.3166/rfg.2019.00300).
- [33] Jan Karel LENSTRA et AHG Rinnooy KAN. "Complexity of vehicle routing and scheduling problems". In : *Networks* 11.2 (1981), p. 221-227.
- [34] S LIM, Xin JIN et J SRAI. "Last-mile logistics structures : a literature review and design guideline". In : *Institute for Manufacturing, Cambridge University* (2016).
- [35] Yaoyao LIU et al. "HA-CCP : A Hybrid Algorithm for Solving Capacitated Clustering Problem". In : (2022).
- [36] J MACQUEEN. "Classification and analysis of multivariate observations". In : *5th Berkeley Symp. Math. Statist. Probability*. University of California Los Angeles LA USA. 1967, p. 281-297.

- [37] Thomas MAXNER, Giacomo DALLA CHIARA et Anne GOODCHILD. "Identifying the Challenges to Sustainable Urban Last-Mile Deliveries : Perspectives from Public and Private Stakeholders". In : *Sustainability* 14.8 (2022), p. 4701.
- [38] Pietro MENGA et al. "Promotion of freight mobility in Milan : Environmental, energy and economical aspects". In : *World Electric Vehicle Journal* 6.4 (2013), p. 1014-1020.
- [39] David MESTER et Olli BRÄYSY. "Active guided evolution strategies for large-scale vehicle routing problems with time windows". In : *Computers Operations Research* 32.6 (2005), p. 1593-1614. ISSN : 0305-0548. DOI : <https://doi.org/10.1016/j.cor.2003.11.017>. URL : <https://www.sciencedirect.com/science/article/pii/S0305054803003605>.
- [40] Ondřej MÍČA. "Comparison metaheuristic methods by solving travelling salesman problem". In : *INPRO-FORUM 2015*. 2016.
- [41] Pablo MOSCATO et al. "On evolution, search, optimization, genetic algorithms and martial arts : Towards memetic algorithms". In : *Caltech concurrent computation program, C3P Report* 826.1989 (1989), p. 37.
- [42] Eneko OSABA, Fernando DIAZ et Enrique ONIEVA. "Golden ball : a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts". In : *Applied Intelligence* 41 (2014), p. 145-166.
- [43] Eneko OSABA, Fernando DIAZ et Enrique ONIEVA. "Golden ball : a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts". In : *Applied Intelligence* 41 (2014), p. 145-166.
- [44] Jérôme Leboeuf PASQUIER et al. "A comparative study of three metaheuristics applied to the traveling salesman problem". In : *2007 Sixth Mexican International Conference on Artificial Intelligence, Special Session (MICAI)*. IEEE. 2007, p. 243-254.
- [45] S. PERIFEL. *Complexité algorithmique*. Références sciences. Ellipses, 2014. ISBN : 9782729886929.
- [46] Elham POURRAHMANI et Miguel JALLER. "Crowdshipping in last mile deliveries : Operational challenges and research opportunities". In : *Socio-Economic Planning Sciences* 78 (2021), p. 101063.
- [47] Karina Aginta PUTRI et al. "Genetic algorithm with cluster-first route-second to solve the capacitated vehicle routing problem with time windows : A case study". In : *Jurnal Teknik Industri* 23.1 (2021), p. 75-82.
- [48] Luigi RANIERI et al. "A review of last mile logistics innovations in an externalities cost reduction vision". In : *Sustainability* 10.3 (2018), p. 782.
- [49] MR RAO. "Cluster analysis and mathematical programming". In : *Journal of the American statistical association* 66.335 (1971), p. 622-626.
- [50] Gerhard REINELT. "Tsplib95". In : *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg* 338 (1995), p. 1-16.
- [51] Eric Touboul RODOLPHE LE RICHE Stéphane Mottelet. "Optimisation locale et globale". In : *Ecole des Mines de Saint-Etienne Université de Technologie de Compiègne CNRS* (2010).
- [52] M. SAKAROVITCH. *Optimisation combinatoire : Graphes et programmation linéaire*. Collection Enseignement des sciences. Hermann, 1984. ISBN : 9782705659745. URL : <https://books.google.dz/books?id=wUfvAAAAMAAJ>.
- [53] Dustin SCHÖDER et al. "The impact of e-commerce development on urban logistics sustainability". In : *Open Journal of Social Sciences* 4.03 (2016), p. 1.

- [54] Marc SEVAUX, Kenneth SÖRENSEN et al. “Hamiltonian paths in large clustered routing problems”. In : *Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME*. T. 8. 2008, p. 411-417.
- [55] Qian SHAO et Shih-Fen CHENG. “Preference-Aware Delivery Planning for Last-Mile Logistics”. In : *arXiv preprint arXiv :2303.04333* (2023).
- [56] Krishna Veer TIWARI et Satyendra Kumar SHARMA. “An optimization model for vehicle routing problem in last-mile delivery”. In : *Expert Systems with Applications* 222 (2023), p. 119789. ISSN : 0957-4174. DOI : <https://doi.org/10.1016/j.eswa.2023.119789>. URL : <https://www.sciencedirect.com/science/article/pii/S0957417423002907>.
- [57] Paolo TOTH et Daniele VIGO. *The vehicle routing problem*. SIAM, 2002.
- [58] Peter JM VAN LAARHOVEN et al. *Simulated annealing*. Springer, 1987.
- [59] Karl VON FRISCH. “Decoding the Language of the Bee”. In : *Readings in Zoosemiotics* 8 (2011), p. 141.
- [60] Darrell WHITLEY et Timothy STARKWEATHER. “Genitor II : A distributed genetic algorithm”. In : *Journal of Experimental & Theoretical Artificial Intelligence* 2.3 (1990), p. 189-214.
- [61] Zefeng WU. “A Comparative Study of solving Traveling Salesman Problem with Genetic Algorithm, Ant Colony Algorithm, and Particle Swarm Optimization”. In : *Proceedings of the 2020 2nd International Conference on Robotics Systems and Vehicle Technology*. 2020, p. 95-99.
- [62] Xin-She YANG. “Firefly algorithms for multimodal optimization”. In : *Stochastic Algorithms : Foundations and Applications : 5th International Symposium, SAGA 2009, Sapporo, Japan, October 26-28, 2009. Proceedings* 5. Springer. 2009, p. 169-178.
- [63] Qing ZHOU et al. “Heuristic search to the capacitated clustering problem”. In : *European Journal of Operational Research* 273.2 (2019), p. 464-487. DOI : [10.1016/j.ejor.2018.08.043](https://doi.org/10.1016/j.ejor.2018.08.043). URL : <https://hal.science/hal-02310000>.