



الجمهورية الجزائرية الديمقراطية الشعبية  
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
المدرسة الوطنية العليا للتكنولوجيا والهندسة – عنابة  
ECOLE NATIONALE SUPÉRIEURE DE TECHNOLOGIE ET D'INGÉNIERIE – ANNABA

Département Génie Industriel

## MEMOIRE

En vue d'obtention du diplôme d'INGÉNIEUR D'ETAT

Domaine : Science et Technologie

Filière : Génie Industriel

Spécialité : Maintenance Industriel

Présenté par

**AYA MEHDAOUI**

**HADIL SNANI**

# OPTIMISATION DES TOURNEES DE LIVRAISON VIA L'ALGORITHME DU VRP APPLICATION A LA LAITERIE TELL

Encadré par

**Mme. Lamia AISSAOUI**

ENSTI Annaba

Membres du jury :

Dr. Nour Elislem KARABADJI	Président	ENSTI-Annaba
Dr. Abdallah BOUZITOUNA	Examineur	ENSTI-Annaba
Dr. Imen DRISS	Examineur	ENSTI-Annaba

Année 2025

## Remerciements

Nous remercions Dieu, le Tout-Puissant, de nous avoir accordé le courage et la patience nécessaires pour mener ce travail à son terme.

Nous tenons à exprimer notre profonde gratitude à notre encadrante, **Mme AISSAOUI Lamia**, pour son accompagnement précieux. Sa compétence, sa patience et ses encouragements constants ont grandement contribué à la réussite de ce travail. Son regard critique nous a été d'une grande aide pour structurer notre mémoire et en améliorer la qualité.

Un très grand remerciement et une profonde reconnaissance sont adressés à **l'entreprise TELL** pour nous avoir permis d'effectuer notre stage de fin d'études dans un cadre aussi professionnel qu'enrichissant.

Nous exprimons également notre gratitude à l'ensemble des enseignants du **département de Génie Industriel de l'ENSTI**, pour leur disponibilité constante et leur engagement envers les étudiants, en particulier aux responsables de la spécialité Génie Industriel, pour leur accompagnement tout au long de notre parcours.

Nous remercions sincèrement **les membres du jury** pour l'honneur qu'ils nous font en acceptant de lire et d'évaluer ce travail.

Enfin, nos remerciements vont à **toutes les personnes** qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

## Dédicace

Je dédie ce travail avec une profonde gratitude à ma propre détermination ainsi qu'à tous les efforts que j'ai fournis durant ces cinq années d'études. Je suis également reconnaissante pour les sacrifices consentis et les nombreux obstacles que j'ai su surmonter au fil de ce parcours.

À ma chère mère,

Qui a tout sacrifié pour me permettre d'atteindre cette étape de ma vie, aucune offrande ne saurait exprimer la profondeur de mes sentiments. Qui a été un soutien constant tout au long de ma vie, présente à chaque instant et dans toutes mes décisions. Sa présence et son amour inconditionnel ont été une source de force et d'inspiration pour moi.

À mon père,

Dont les sacrifices et les épreuves tout au long de ces années d'études ont été essentiels pour m'offrir cette opportunité, merci pour votre motivation et votre générosité, et pour votre soutien moral indéfectible.

À mes chers frères, Mohamed et Oussama,

Je vous remercie du fond du cœur pour votre soutien constant et votre présence rassurante à mes côtés. Je n'oublierai jamais les sacrifices que vous avez consentis pour moi.

À ma grande mère (Tama) ,

Pour ton amour inconditionnel, ta sagesse et tes prières silencieuses.

Tu as toujours été une source de force, de tendresse et de lumière dans ma vie.

Ce travail t'est dédié avec tout mon respect et mon affection.

À tous mes amis qui m'ont toujours encouragée, et à qui je souhaite encore plus de succès et de réussite.

AYA

## Dédicace

Je dédie ce travail à tous ceux qui nous sont chers dans ce bagage affectueux qui, tout au long de notre parcours, ont scellé l'histoire de notre vie et du soutien dont nous avons toujours pu bénéficier.

À ma chère mère

Quoi que je fasse ou dise, je ne saurai jamais te remercier comme il se doit.  
Ton amour m'entoure, ta bienveillance me guide, et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

À mon cher père<<kader>>

tu as toujours été à mes côtés pour me soutenir et m'encourager. J'espère que ce travail exprimera ma gratitude et mon amour pour toi.

À mes frères <<med.taki ALLAH>>et <<youcef>>

Merci pour votre gentillesse, votre soutien moral, vos encouragements et moments partagés, Que Dieu vous accorde la santé, le bonheur, le courage et surtout le succès.

À mes véritables amis

Merci pour vos sourires, vos belles paroles réconfortantes, nos bienheureux moments de repos, de jeu et de rire.

À tous celles et tous ceux dont le nom ne nous apparaît pas ici et qui ont contribué de façon marquante à notre parcours,

Voici le témoignage de notre profonde reconnaissance.

Hadil

## Table des matières :

Remerciements.....	I
Dédicace.....	II
Liste des figures.....	VI
Liste des tableaux.....	VII
Résumé.....	VIII
Introduction.....	1
Chapitre I : Présentation de l'entreprise TELL Sétif.....	3
I.1 Présentation de l'entreprise.....	3
I.3 Problématique.....	4
I.4 Objectifs du projet .....	5
Chapitre 2 : État de l'art sur le problème de tournées de véhicules (VRP).....	7
2.1 Introduction générale .....	7
2.2 Définition du VRP.....	8
2.3 Formulation mathématique de base .....	8
2.4 Variantes du VRP .....	10
2.5. Méthode de résolution du problème VRP .....	11
2.5.1. Méthodes exactes.....	12
2.5.2. Méthodes approchées.....	12
2.5.3. Heuristiques.....	12
2.5.4. Métaheuristiques.....	12
2.6 Choix pour ce projet.....	12
2.7 Applications concrètes .....	13
2.8 Synthèse .....	13
Chapitre 3 : Travaux réalisés.....	15
3.1 Cahier des charges fonctionnel .....	15
3.1.1. Ajout et gestion des clients.....	15
3.1.2. Ajout et gestion de départ.....	15
3.1.3. Ajout et gestion des produit.....	15
3.1.4. Ajout et gestion des véhicules .....	15
3.1.5. Lancement de l'optimisation .....	16
3.1.6. Affichage des tournées sur une carte interactive .....	16
3.2 Cahier des charges technique.....	16
3.2.1. Technologies utilisées.....	16
3.2.2. Pourquoi ce choix technologique ?.....	17

3.3.1. Frontend (interface utilisateur).....	17
3.3.2. Backend (API serveur) .....	17
3.3.3 Base de données (MongoDB).....	18
3.5 Implémentation de l'algorithme du VRP .....	22
3.5.1. Saisie des Données (Backend Node.js – server/routes/vrpRoutes.js).....	22
3.5.2. La phase du traitement (Backend Node.js & Script Python).....	23
3.6 Problèmes rencontrés et solutions.....	25
Chapitre 4 : Résultats, discussion, conclusion et perspectives .....	26
4.1 Présentation des résultats .....	26
4.2 Discussion des résultats .....	27
4.3 Conclusion .....	27
4.4 Perspectives .....	28
Bibliographie .....	30

## Liste des figures

<b>Figure 1:</b> Localisation des points de vente de GIPLAIT à Sétif .....	3
<b>Figure 2:</b> Problème de tournées de véhicules .....	7
<b>Figure 3:</b> Modèle Mathématique des tournées de véhicules .....	9
<b>Figure 4:</b> Pseudo-code du VRP .....	10
<b>Figure 5:</b> Les différentes variantes du VRP .....	11
<b>Figure 6:</b> Schéma d'architecture simplifié .....	17
<b>Figure 7:</b> Capture du Schéma de l'architecture des clients .....	18
<b>Figure 8:</b> Capture du Schéma de l'architecture de départ .....	18
<b>Figure 9:</b> page de connexion .....	19
<b>Figure 10:</b> page d'inscription .....	19
<b>Figure 11:</b> Home page .....	19
<b>Figure 12:</b> Ajout de nouveaux clients .....	20
<b>Figure 13:</b> Ajout de nouveaux chauffeurs .....	20
<b>Figure 14:</b> Ajout de nouveau point de départ .....	21
<b>Figure 15:</b> Ajout de nouveaux produits .....	21
<b>Figure 16:</b> Ajout de nouveaux véhicules .....	21
<b>Figure 17:</b> Saisie et Validation des Données d'Entrée VRP (Backend Node.js) .....	22
<b>Figure 18:</b> Acquisition des Données Géospatiales via OSRM (vrpRoutes.js) .....	23
<b>Figure 19:</b> Préparation des Données pour le Solveur .....	23
<b>Figure 20:</b> Post-traitement et Génération des Chemins Détaillés (vrpRoutes.js) .....	24
<b>Figure 21:</b> Renvoi de la Réponse Finale au Client (Backend Node.js) .....	25
<b>Figure 22:</b> Avant l'optimisation .....	26
<b>Figure 23:</b> Après l'optimisation .....	26

## Liste des tableaux

<b>Tableau 1:</b> comparaison avant et après l'optimisation.....	27
--	----

## Résumé :

Dans le cadre de l'amélioration de la performance logistique de la laiterie TELL, ce projet de fin d'études vise à optimiser les tournées quotidiennes de livraison à l'aide de l'algorithme du Vehicle Routing Problem (VRP). La solution développée repose sur une application full-stack dotée d'une interface web, permettant de générer automatiquement des tournées optimisées en intégrant plusieurs paramètres : le poids des camions, la géolocalisation des clients et les volumes de commandes.

Le système s'appuie sur des technologies de dernière génération, à savoir React pour l'interface, Node.js pour le serveur, MongoDB pour la gestion des données, ainsi que l'API OSRM (Open Source Routing Machine) pour le calcul d'itinéraires.

Les résultats obtenus montrent une réduction significative des distances parcourues, un meilleur équilibrage des charges entre les véhicules, et une planification plus fluide. Ce travail met en évidence la valeur ajoutée d'intégrer des outils numériques intelligents dans les chaînes logistiques locales, en contribuant à une gestion plus efficace et durable des livraisons.

**Mots-clés :** Optimisation, Tournée de livraison, Vehicle Routing Problem (VRP), Logistique, Application full-stack, Laiterie TELL, OSRM, Heuristique, Réduction des coûts.

## Abstract:

As part of improving the logistical performance of the TELL dairy company, this final-year project aims to optimize daily delivery routes using the Vehicle Routing Problem (VRP) algorithm. The developed method involves creating a full-stack application that automatically generates optimized delivery routes by taking into account truck load capacities, customer geolocation, and order volumes. The system leverages modern technologies, including React, Node.js, MongoDB, and the Open-Source Routing Machine (OSRM) for route calculation. The results show a significant reduction in distances traveled, better load balancing among vehicles, and easier planning. This work highlights the value of introducing intelligent digital tools into local supply chains.

**Keywords:** Optimization, Delivery routing, Vehicle Routing Problem (VRP), Logistics, full-stack application, TELL Dairy, OSRM Heuristic, Cost reduction.

## الملخص:

في إطار تحسين الأداء اللوجستي لمصنع الألبان TELL، يهدف هذا مشروع التخرج إلى تحسين مسارات التوزيع اليومية باستخدام خوارزمية "مشكلة توجيه المركبات" (VRP). تعتمد الطريقة المطورة على إنشاء تطبيق "فول ستاك" مزود بواجهة ويب يقوم تلقائياً بإنشاء مسارات توزيع محسنة، مع الأخذ بعين الاعتبار حمولة الشاحنات، الموقع الجغرافي للعملاء، وحجم الطلبات. يعتمد النظام على تقنيات حديثة مثل React و Node.js و MongoDB، بالإضافة إلى واجهة برمجة التطبيقات الخاصة بحساب المسارات المتقدمة من Routing Machine (OSRM) Open Source. أظهرت النتائج انخفاضاً كبيراً في المسافات المقطوعة، وتحسناً في توازن الحمولة بين المركبات، مما يسهل عملية التخطيط. يبرز هذا العمل أهمية إدخال الأدوات الرقمية الذكية في سلاسل التوريد المحلية.

الكلمات المفتاحية : التحسين، مسارات التوصيل، مشكلة توجيه المركبات (VRP)، الوجيه، تطبيق ويب، ملبنه نيل، OSRM، النكاه الاصطناعي، الخوارزميات التقريبية، تخفيض التكاليف

## **Introduction :**

Dans le contexte économique actuel, la performance logistique est devenue un levier stratégique essentiel pour les entreprises, en particulier celles opérant dans la distribution de produits frais, comme le secteur laitier. Ces entreprises sont confrontées à des exigences élevées en termes de rapidité, d'efficacité, de coût, de qualité et de conditions de conservation, où la pression est constante.

La laiterie TELL, acteur local bien implanté, assure quotidiennement la distribution de ses produits à un vaste réseau de clients répartis sur plusieurs zones géographiques. À ce jour, cette distribution repose sur une organisation manuelle et empirique des tournées, généralement fondée sur l'expérience des chauffeurs ou des responsables logistiques de chaque site. Cette approche montre rapidement ses limites : trajets sous-optimisés, surcoût en carburant par rapport aux prévisions, déséquilibre de la charge de travail entre chauffeurs, et difficultés à respecter les délais de livraison.

C'est dans cette perspective qu'intervient le projet de fin d'études présenté dans ce mémoire, qui vise à automatiser l'organisation des tournées de livraison à l'aide d'un algorithme d'optimisation bien connu : le Vehicle Routing Problem (VRP). Issu du domaine de la recherche opérationnelle, ce problème permet de déterminer les itinéraires optimaux à suivre pour desservir un ensemble de clients à partir d'un ou plusieurs dépôts, tout en respectant diverses contraintes (capacité des véhicules, plages horaires de livraison, etc.).

L'objectif principal de ce travail est donc de concevoir et de développer une application full-stack dotée d'une interface web capable de générer automatiquement des tournées de livraison adaptées aux besoins spécifiques de la laiterie TELL. Ce projet mobilise des compétences en développement logiciel, en modélisation mathématique et en traitement de données géographiques.

Au fil de sa réalisation, nous souhaitons démontrer qu'un outil numérique simple mais rigoureux, basé sur un modèle mathématique solide, peut jouer un rôle fondamental dans l'optimisation des processus logistiques d'une entreprise de proximité.



## Chapitre I : Présentation de l'entreprise TELL Sétif

### I.1 Présentation de l'entreprise

La laiterie TELL, filiale du Groupe GIPLAIT, est une entreprise publique spécialisée dans la production et la commercialisation de produits laitiers sur le marché national. Elle vise à satisfaire les besoins de ses clients par l'amélioration continue de la qualité de ses produits.

Implantée dans la région de Sétif, l'entreprise dessert un réseau de plus de 150 clients, répartis sur plusieurs zones urbaines et rurales [1]. Les livraisons sont planifiées à partir d'un dépôt central situé à proximité de l'usine de production, à l'aide d'une flotte de dix camions effectuant plusieurs tournées par jour pour répondre à la demande quotidienne.

Cependant, les commandes varient considérablement en fonction du type de point de vente et du jour de la semaine [2], ce qui complexifie davantage la planification logistique. Aujourd'hui, cette dernière est réalisée manuellement, en s'appuyant principalement sur l'expérience des chauffeurs et des responsables logistiques. Aucun système informatique n'est utilisé pour optimiser les parcours ou regrouper efficacement les livraisons en tenant compte des contraintes de capacité ou de distance.

Prenons l'exemple d'une tournée quotidienne : un camion quitte le dépôt vers 6h du matin pour desservir plusieurs points de vente — El Hidhab, Abid Ali, Bir Ennas, etc.

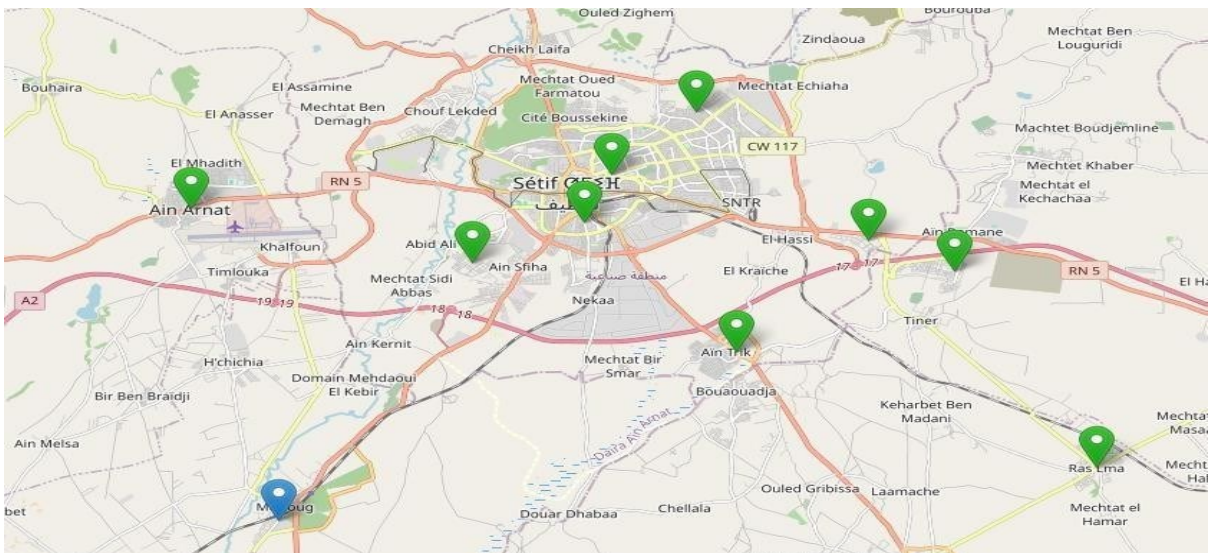


Figure 1: Localisation des points de vente de GIPLAIT à Sétif

situés dans la ville de Sétif, avant de revenir au dépôt vers midi. Les itinéraires sont souvent définis à l'avance, sans réelle prise en compte des conditions de circulation ou d'une optimisation kilométrique, ce qui peut entraîner des trajets plus longs que nécessaire.

## **I.2 Motivation du projet**

Bien que la laiterie TELL dispose d'un système de gestion de tournées, les performances logistiques et les processus métiers restent perfectibles. Le mode de fonctionnement actuel repose sur des pratiques traditionnelles et empiriques, dépendant largement de l'intuition des chauffeurs. Cela engendre plusieurs inefficiences : surcoût en carburant, usure accélérée des véhicules, déséquilibre des charges de travail et fatigue accrue pour les conducteurs.

La variabilité quotidienne des commandes complique davantage la tâche : certains camions se retrouvent surchargés, d'autres partiellement remplis. Cette répartition inégale nuit à l'efficacité de la distribution, augmente les coûts logistiques et engendre parfois des retards, notamment en période de forte demande.

Par ailleurs, des contraintes opérationnelles importantes — telles que les capacités de chargement des véhicules, les horaires spécifiques de livraison des clients ou encore la densité du trafic urbain — sont peu, voire pas du tout, prises en compte. De plus, l'absence de système de suivi ou de visualisation empêche toute analyse de performance ou identification d'axes d'amélioration.

Face à ces constats, la motivation première de ce projet est de proposer une solution automatisée d'optimisation des tournées de livraison. Grâce à l'intégration d'algorithmes d'optimisation, il devient possible de :

- Réduire les distances totales parcourues chaque jour ;
- Diminuer la consommation de carburant et l'usure des véhicules ;
- Améliorer le respect des délais de livraison ;
- Rééquilibrer la charge de travail entre les chauffeurs.

Ce projet s'inscrit également dans une démarche de transformation numérique, en dotant la laiterie TELL d'un outil moderne, évolutif et adapté à ses besoins. L'implémentation d'une application web de gestion automatisée des tournées constitue ainsi une réponse concrète aux enjeux de performance et de compétitivité de l'entreprise.

## **I.3 Problématique**

Dans un contexte où les entreprises cherchent à réduire les coûts tout en améliorant leur efficacité opérationnelle, l'optimisation des tournées de livraison devient un enjeu crucial, notamment pour les distributeurs de produits frais comme la laiterie TELL. Cette dernière est

confrontée à plusieurs défis logistiques liés à la nature périssable de ses produits, à la fréquence élevée des livraisons et à la diversité des points de distribution.

Actuellement, les tournées sont planifiées sans aucun outil d'aide à la décision. Cette approche montre rapidement ses limites à mesure que les contraintes augmentent : horaires de réception des marchandises, capacités de chargement, zones éloignées ou difficiles d'accès, etc. Les fluctuations journalières des volumes de commande rendent d'autant plus difficile l'ajustement manuel des itinéraires.

**La problématique centrale du projet peut donc être formulée comme suit :**

**Comment optimiser les tournées de livraison afin d'améliorer la rentabilité et la performance logistique de la laiterie TELL, tout en respectant les contraintes opérationnelles telles que les capacités des véhicules, les distances, les délais de livraison et les coûts ?**

Pour y répondre, il est nécessaire d'adopter une approche mathématique et informatique capable d'intégrer l'ensemble de ces paramètres. C'est dans ce cadre qu'intervient le Vehicle Routing Problem (VRP), ou problème de tournées de véhicules. Il s'agit d'un problème d'optimisation combinatoire visant à déterminer les itinéraires les plus efficaces pour livrer un ensemble de clients à partir d'un ou plusieurs dépôts, tout en minimisant un ou plusieurs critères (distance, coût, temps), et en respectant les contraintes spécifiques.

Bien que largement étudié dans la littérature scientifique, le VRP demeure un problème complexe, notamment lorsqu'il est enrichi de variantes pratiques. Dans le cas de la laiterie TELL, l'implémentation d'un tel modèle permettrait de structurer les tournées de façon optimale, tout en offrant une flexibilité face à l'évolution quotidienne de la demande. Ce projet dépasse ainsi la simple dimension logistique : il touche à des enjeux stratégiques de compétitivité, de satisfaction client et de maîtrise des coûts.

#### **I.4 Objectifs du projet**

Ce projet vise à développer une application web dédiée à la génération automatique de tournées de livraison optimisées pour la laiterie TELL, en prenant en compte les spécificités opérationnelles de l'entreprise : capacité des véhicules, distances entre points de livraison, quantités commandées, etc.

En considérant chaque point de livraison comme un client distinct, l'application permettra de :

- Déterminer le nombre optimal de camions à mobiliser ;
- Choisir les véhicules les plus adaptés ;
- Réduire les distances parcourues et les temps de trajet.

L'objectif final est d'améliorer la rentabilité et l'efficacité de la chaîne logistique grâce à une organisation intelligente et automatisée des itinéraires.

Parmi les objectifs spécifiques, on retrouve :

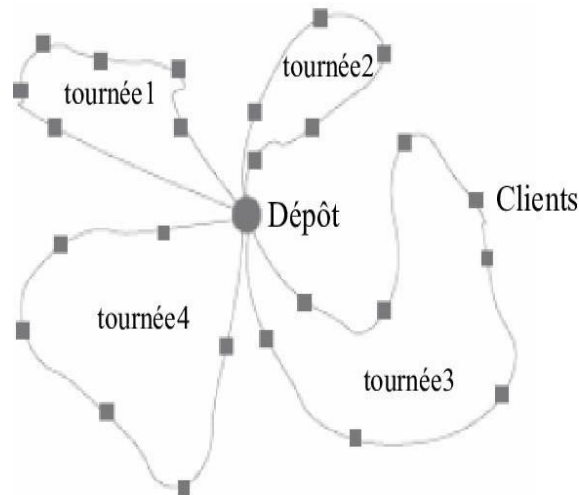
- Le lancement automatique de l'optimisation des tournées à partir des données saisies ;
- La visualisation des itinéraires sur une carte interactive pour une meilleure compréhension ;
- La prise en compte des capacités de chargement pour équilibrer les livraisons ;
- L'intégration de fonctionnalités de gestion des clients (ajout, modification, etc.).

Ce projet revêt donc une double dimension : technique, à travers le développement logiciel, et opérationnelle, par l'amélioration concrète des processus métier de la laiterie TELL. Chapitre I: Présentation de l'entreprise TELL Sétif

## Chapitre 2 : État de l'art sur le problème de tournées de véhicules (VRP)

### 2.1 Introduction générale

Le transport s'est imposé comme une fonction logistique clé, surtout dans un contexte où acheminer des produits périssables engendre des coûts élevés et nécessite une organisation rigoureuse. Ne cessant d'augmenter, une planification efficace des tournées de livraison s'impose comme un levier stratégique pour réduire les dépenses et améliorer la performance. C'est dans ce contexte qu'a émergé le Problème de Tournées de Véhicules (VRP – Vehicle Routing Problem), désormais reconnu comme une composante centrale de l'optimisation des réseaux logistiques.



**Figure 2:** Problème de tournées de véhicules

Le VRP est une version élargie du célèbre Problème du Voyageur de Commerce (TSP), qui consiste à minimiser la distance parcourue par un seul vendeur visitant plusieurs clients. Dans le cas du VRP, plusieurs véhicules partent d'un ou plusieurs dépôts pour desservir les clients, tout en respectant des contraintes comme la capacité des véhicules, les horaires de livraison ou les coûts de transport [3].

Ce problème appartient au domaine de la recherche opérationnelle, qui se concentre sur l'optimisation des processus décisionnels, qui fournit les outils mathématiques nécessaires à la modélisation et à l'optimisation des réseaux logistiques.

Au fil des décennies, le transport est devenu un axe stratégique de la gestion logistique, en particulier dans un environnement où les coûts de distribution, notamment pour les produits à durée de vie limitée, ne cessent d'augmenter. Face à cette réalité, les entreprises ont pris conscience de la nécessité de planifier efficacement les tournées de livraison afin de réduire les

dépenses opérationnelles. C'est dans ce cadre qu'a émergé le Problème de Tournées de Véhicules (VRP), considéré aujourd'hui comme un défi central dans la logistique du transport. Le VRP peut être vu comme une extension du Problème du Voyageur de Commerce (TSP : Traveling Salesman Problem), dans lequel un seul agent doit visiter un ensemble de clients en parcourant la plus courte distance possible. À la différence du TSP, le VRP mobilise plusieurs véhicules au départ d'un ou plusieurs dépôts, et vise à organiser des tournées optimales en tenant compte de contraintes telles que la capacité de chargement, les fenêtres temporelles de livraison ou encore les coûts logistiques.[3].

## 2.2 Définition du VRP

Introduit pour la première fois en 1959 par Dantzig et Ramser [4], le Problème de Tournées de Véhicules (Vehicle Routing Problem – VRP) vise à déterminer les parcours les plus efficaces pour une flotte de véhicules homogènes, tous partant d'un même dépôt. L'objectif principal est de minimiser la distance totale parcourue, tout en respectant des contraintes opérationnelles, notamment la capacité de chargement propre à chaque véhicule.

### Objectifs du VRP :

L'objectif fondamental du VRP est de répondre efficacement à la demande de l'ensemble des clients tout en optimisant l'utilisation des ressources logistiques disponibles. Sa résolution implique de prendre en compte plusieurs contraintes couramment rencontrées, parmi lesquelles :

- la limite de capacité des véhicules, que ce soit en termes de poids ou de volume,
- le nombre restreint de véhicules mobilisables,
- les distances ou durées de déplacement entre les différents points de livraison,
- et, dans certains cas, le respect strict de créneaux horaires imposés pour les livraisons.

## 2.3 Formulation mathématique de base

Le VRP peut être modélisé comme un problème d'optimisation combinatoire. Voici une version simplifiée de cette formulation :

### Paramètres :

- $N$  : ensemble des clients (le dépôt est désigné par le client 0),
- $K$  : ensemble des véhicules,
- $C_{ij}$  : coût ou distance entre les clients  $i$  et  $j$ ,
- $d_i$  : demande du client  $i$ ,
- $Q$  : capacité maximale d'un véhicule,
- $[a_i, b_i]$  : fenêtre de temps autorisée pour le client  $i$ .

## La Fonction objectif :

En s'appuyant sur les hypothèses et notations précédemment définies, le modèle mathématique du Problème de Tournées de Véhicules peut être formulé de la manière suivante :

- Objective Function:

$$\text{Minimize } \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} C_{ij} x_{ijk} \quad (1)$$

- Decision Variables:

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle travels directly from customer } i \text{ to customer } j \\ 0, & \text{otherwise} \end{cases}$$

- Constraints:

$$\sum_{k \in K} \sum_{i \in N} \sum_{j \in N} x_{ijk} = 1, \forall i \in N \quad (2)$$

$$\sum_{i \in N} x_{i,n+1,k} = 1, \forall k \in K \quad (3)$$

$$\sum_{j \in N} x_{0jk} = 1, \forall k \in K \quad (4)$$

$$\sum_{i \in N} x_{ijk} - \sum_{j \in N} x_{ijk} = 0, \forall k \in K \quad (5)$$

$$\sum_{i \in N} d_i \sum_{j \in N} x_{ijk} \leq Q, \forall k \in K \quad (6)$$

$$x_{ijk} \geq 0, \forall k \in K \quad (7)$$

**Figure 3:**Modèle Mathématique des tournées de véhicules

1. **Fonction objective** : Minimiser la distance totale parcourue (ou le coût global) par l'ensemble des véhicules engagés dans les tournées.
2. **Contrainte de couverture** : Chaque client doit être visité une et une seule fois par l'un des véhicules.
3. **Retour au dépôt** : Toute tournée se termine obligatoirement au dépôt de destination noté  $n+1$ .
4. **Départ du dépôt** : Chaque tournée débute à partir du dépôt initial noté 000.
5. **Continuité des tournées** : Un véhicule qui dessert un client doit poursuivre sa tournée en se rendant vers un autre client ou retourner au dépôt.
6. **Capacité des véhicules** : La quantité totale livrée par un véhicule ne doit pas excéder sa capacité maximale.
7. **Contrainte de non-négativité** : Les variables de décision associées aux trajets

et aux quantités transportées doivent être positives ou nulles.

### Contraintes principales :

1. **Unicité de desserte** : chaque client doit être visité une seule fois par un unique véhicule.
2. **Structure des tournées** : chaque tournée doit impérativement commencer et se terminer au dépôt.
3. **Respect des capacités** : la quantité de marchandises transportée par chaque véhicule ne doit en aucun cas dépasser sa capacité maximale.
4. **Optimisation des coûts** : l'objectif global consiste à minimiser le coût total des tournées, généralement exprimé en distance parcourue ou en temps de trajet cumulé.

```

Input:    $I_n$ : The number of iterations;  $P_n$ : The number of solutions in population;
            $T_n$ : The number of trials;  $T_{start}$ : The start temperature;  $T_{end}$ : The end temperature;
            $IN$ : The instance;
Output:  $S_{best}$ : The best solution;  $f(S_{best})$ : The best fitness value
1. Begin
2.  $D \leftarrow \text{Read\_Instance}(IN)$ 
3.  $T \leftarrow T_{start}$ 
4.  $\text{Frac} \leftarrow (T_{end}/T_{start})^{(1.0/(I_n-1.0))}$ 
5.  $x \leftarrow \text{Create\_Initial\_Population}(D, P_n)$ 
6.  $f \leftarrow \text{Compute\_Fitness\_Value}(D, x)$ 
7.  $S_{best} \leftarrow x(1)$ 
8.  $f(S_{best}) \leftarrow f(1)$ 
9. for  $i \leftarrow 1$  to  $I_n$ 
10.  foreach  $s$  in  $x$ 
11.    for  $j \leftarrow 1$  to  $T_n$ 
12.       $S_{new} \leftarrow \text{Develop\_New\_Route}(s)$ 
13.      if  $\text{isInvalid}(S_{new})$ 
14.         $\text{Death\_Penalty}(S_{new})$ 
15.      end if
16.       $f(S_{new}) \leftarrow \text{Compute\_Fitness\_Value}(S_{new})$ 
17.      if  $f(S_{new}) < f(s)$ 
18.         $s \leftarrow S_{new}$ 
19.         $f(s) \leftarrow f(S_{new})$ 
20.      else
21.         $w = e^{-(f(S_{new})-f(s))/T}$ 
22.        if  $\text{random}(0, 1) < w$ 
23.           $s \leftarrow S_{new}$ 
24.           $f(s) \leftarrow f(S_{new})$ 
25.        end if
26.      end if
27.    end for
28.  if  $f(s) < f(S_{best})$ 
29.     $S_{best} \leftarrow s$ 
30.     $f(S_{best}) \leftarrow f(s)$ 
31.  end if
32. end foreach
33.  $T \leftarrow \text{Frac} * T$ 
34. end for
35. return( $S_{best}, f(S_{best})$ )
36. End

```

Figure 4:Pseudo-code du VRP

## 2.4 Variantes du VRP

De nombreuses variantes du VRP ont été développées afin de mieux modéliser les situations réelles

2.4.1 Capacitated VRP (CVRP) :Le véhicule ne peut dépasser une capacité maximale ; très pertinent dans les cas de livraison de produits lourds ou volumineux [5]

2.4.2 VRP avec fenêtres de temps (VRPTW) : Chaque client doit être livré dans un créneau horaire précis. Cette contrainte est fréquente dans les distributions alimentaires ou pharmaceutiques [6].

2.4.3 VRP dynamique (DVRP) : Les données (commandes, clients, horaires) peuvent changer en temps réel pendant l'exécution [7].

2.4.4 VRP multi-compartiments (MCVRP) : Les véhicules doivent transporter plusieurs types de produits incompatibles, chacun dans un compartiment spécifique [8].

2.4.5 VRP à multiples tournées (VRPMT) : Chaque véhicule peut effectuer plusieurs tournées par jour, utile quand la flotte est réduite [9].

2.4.6 VRP multi-dépôts (MDVRP) : Les livraisons partent de plusieurs dépôts au lieu d'un seul [9].

2.4.7 VRP ouvert (OVRP) : Les véhicules ne reviennent pas nécessairement au dépôt après leur tournée [10].

2.4.8 VRP stochastique (SVRP) : Certaines données sont aléatoires (temps de trajet, demandes clients, etc.) [7].

2.4.9 VRP avec livraisons fractionnées (SDVRP) : Un client peut être visité plusieurs fois si sa demande dépasse la capacité d'un seul véhicule [6].

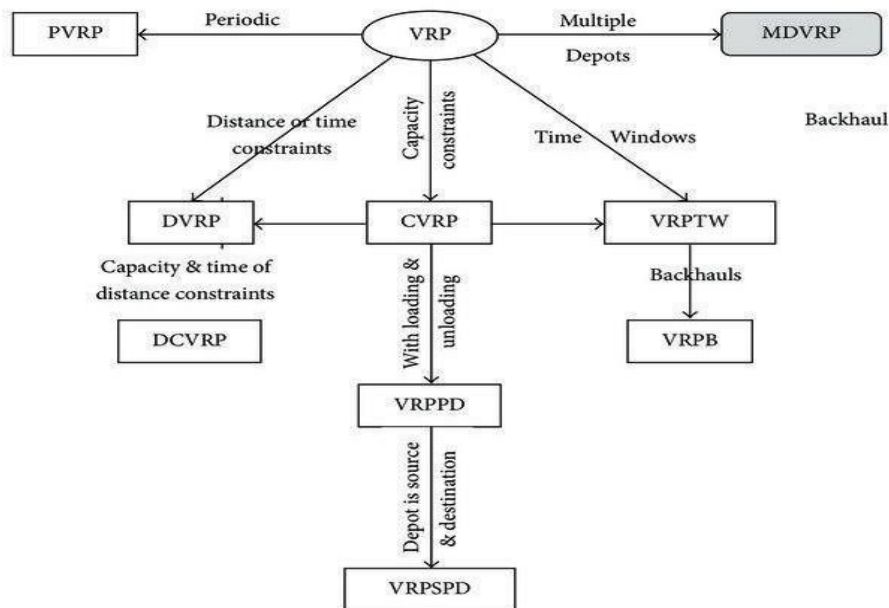


Figure 5: Les différentes variantes du VRP

## 2.5. Méthode de résolution du problème VRP :

Dans la littérature, diverses méthodes ont été développées pour résoudre le VRP de manière efficace, en visant de bonnes solutions avec un temps de calcul réduit. Elles se divisent en deux grandes catégories : les **méthodes exactes** et les **méthodes approchées**. [3]

### 2.5.1. Méthodes exactes :

Les méthodes exactes ont pour objectif de déterminer la solution optimale au problème. Elles utilisent des algorithmes qui énumèrent implicitement toutes les solutions possibles pour garantir l'optimalité. Cependant, ces méthodes sont limitées par leur temps de calcul, qui augmente exponentiellement avec la taille du problème. Parmi les approches exactes les plus courantes figurent les méthodes arborescentes telles que Branch and Bound, ainsi que la programmation dynamique et la programmation linéaire en nombres entiers. [8]

### 2.5.2. Méthodes approchées :

Ce sont des techniques heuristiques utilisées pour résoudre des problèmes d'optimisation complexes. Elles permettent d'obtenir une solution de bonne qualité en un temps raisonnable, surtout lorsque les méthodes exactes sont trop coûteuses en temps de calcul. Ces méthodes sont particulièrement adaptées aux problèmes de grande taille, où l'on cherche une solution satisfaisante proche de l'optimal, plutôt qu'une solution exacte. [11] On distingue deux grandes catégories de ces méthodes :

### 2.5.3. Heuristiques :

Sont conçues pour produire des solutions réalisables et de bonne qualité en un temps raisonnable. [12] Elles ne garantissent pas l'optimalité mais sont souvent plus rapides que les méthodes exactes. Parmi ces méthodes, on trouve :

**L'heuristique de Clarke & Wright** est l'une des approches les plus célèbres pour résoudre le VRP. qui commence par attribuer une tournée à chaque client puis les fusionne pour réduire les coûts.[13]

**Insertion Heuristique** : Commencer avec une solution vide et insérer les clients un par un en minimisant le coût à chaque insertion, On peut trouver diverses méthodes telles que l'insertion la plus proche (Nearest insertion), l'insertion optimale (best insertion), entre autres.[14]

### 2.5.4. Métaheuristiques :

Les métaheuristiques sont des techniques avancées développées pour résoudre des problèmes d'optimisation complexes, souvent caractérisés par des données incomplètes, incertaines ou bruitées, ainsi que par des contraintes de temps de calcul. Leur popularité s'explique par leur large applicabilité : elles peuvent être utilisées pour résoudre tout problème formulé comme une tâche d'optimisation de critères.[15]

Ces méthodes s'inspirent généralement de phénomènes naturels issus de domaines variés tels que la physique (par exemple, le recuit simulé), la biologie (comme les algorithmes génétiques) ou encore l'éthologie (comme les essais particuliers ou les colonies de fourmis).[15]

Parmi les principales métaheuristiques, on distingue :

- **Le recuit simulé** : inspiré du comportement thermodynamique des matériaux, ce procédé assimile la fonction objective à une énergie  $E$  à minimiser, et introduit une température fictive  $T$  qui décroît progressivement pour éviter les minima locaux.[15]

- **La recherche tabou** : cette méthode exploite une mémoire des solutions récemment explorées (la liste taboue) afin d'empêcher le retour vers des configurations déjà visitées. À chaque itération, le voisinage de la solution courante est analysé, en excluant celles qui figurent dans la liste taboue, et l'on sélectionne la meilleure solution admissible.[6].
- **Les algorithmes génétiques** : basés sur les théories de l'évolution de Darwin et de l'hérédité de Mendel, ces algorithmes opèrent sur une population de solutions initialement aléatoires. Grâce aux opérateurs de sélection, de croisement et de mutation, de nouvelles générations sont produites, permettant une amélioration progressive des solutions, sans toutefois garantir l'optimalité globale. Le processus implique deux populations : les parents et les enfants.[16]

## 2.6 Choix pour ce projet :

Dans le cadre de notre projet, nous avons opté pour une approche heuristique simple : en l'occurrence l'algorithme **Clarke & Wright**, qui constitue un bon compromis entre efficacité, rapidité de calcul et facilité de compréhension. Cette méthode est parfaitement adaptée à une PME locale comme la laiterie TELL, et permet une intégration rapide dans une application web.

## 2.7 Applications concrètes :

Le VRP est largement appliqué dans de nombreux secteurs :

- Industrie laitière : livraisons quotidiennes en fonction de la durée de conservation des produits,
- Distribution alimentaire : entreprises comme Danone ou Nestlé utilisent des systèmes d'optimisation VRP pour approvisionner leurs points de vente,
- Livraison de colis : acteurs comme Amazon ou Chronopost exploitent des variantes avancées du VRP incluant des fenêtres de temps, des contraintes de circulation, etc.

## Outils open source :

- Google OR-Tools : bibliothèque puissante et gratuite permettant de résoudre de nombreuses variantes du VRP.
- OptaPlanner : outil Java intégré dans les ERP pour la planification automatisée.

## 2.8 Synthèse :

Le **Vehicle Routing Problem** est au cœur de l'optimisation logistique. Bien que de nombreuses solutions existent, leur complexité rend difficile leur adoption par les PME. La laiterie TELL, comme beaucoup d'entreprises de taille intermédiaire, continue de gérer ses tournées de manière empirique, ce qui génère des inefficiences importantes.

Notre projet propose donc une solution simple, contextualisée et automatisée, reposant sur des outils open source et des algorithmes accessibles. En intégrant cette approche à une application web, nous visons à offrir à la laiterie TELL une meilleure organisation de ses tournées, une réduction des coûts logistiques, et un gain en flexibilité et en performance.

## **Chapitre 3 : Travaux réalisés**

### **3.1 Cahier des charges fonctionnel**

Afin de répondre efficacement aux besoins de la laiterie TELL, nous avons commencé par établir un cahier des charges fonctionnel. Celui-ci définit les fonctionnalités essentielles que l'application web doit offrir pour assurer une gestion optimisée des tournées de livraison.

#### **Fonctionnalités attendues :**

##### **3.1.1. Ajout et gestion des clients**

- Possibilité d'ajouter les clients manuellement.
- Pour chaque client, les informations suivantes doivent être renseignées :
  - Nom ou identifiant unique,
  - Coordonnées géographiques (latitude/longitude ou adresse),
  - Type (point de vente ou laiterie)
- Possibilité de modifier ou supprimer un client existant.

##### **3.1.2. Ajout et gestion de point de départ**

- Possibilité d'ajouter les clients manuellement.
- Pour chaque client, les informations suivantes doivent être renseignées :
  - Nom ou identifiant unique,
  - Coordonnées géographiques (latitude/longitude),
- Possibilité de modifier ou supprimer un client existant.

##### **3.1.3. Ajout et gestion des produits**

- Possibilité d'ajouter les produits manuellement.
- Pour chaque produit, les informations suivantes doivent être renseignées :
  - Nom, Catégories,
  - L'unité (g ou l) et le prix,
- Possibilité de modifier ou supprimer un produit existant.

##### **3.1.4. Ajout et gestion des véhicules**

- Possibilité d'ajouter les véhicules manuellement.
- Pour chaque client, les informations suivantes doivent être renseignées :
  - Matricule, Type, Tonnage maximum

- Cas véhicule disponible ou non
- Possibilité de modifier ou de supprimer un véhicule qui ne fait plus partie de la flotte de l'entreprise.

### **3.1.5. Lancement de l'optimisation**

- Saisir le point de départ de la tournée en question.
- Choisir les véhicules et les chauffeurs disponibles.
- Une fois les clients saisis,
- Choisir les produits avec les quantités désirées
- Lancement de l'algorithme d'optimisation.
- L'algorithme va :
  - Regrouper les clients par tournée selon le principe du VRP,
  - Générer les itinéraires les plus adaptés pour chaque véhicule.

### **3.1.6. Affichage des tournées sur une carte interactive**

- Visualisation claire et dynamique de chaque tournée sur une carte.
- Utilisation de couleurs différentes pour identifier les trajets de chaque véhicule

## **3.2 Cahier des charges technique**

Le développement de cette application full-stack repose sur une architecture web moderne, modulaire et évolutive. Le choix des technologies a été motivé par leur performance, leur compatibilité avec les API de cartographie, ainsi que par leur facilité de maintenance.

### **3.2.1. Technologies utilisées :**

#### **Frontend**

Framework : Next.js (basé sur React.js)

Next.js permet le développement d'interfaces performantes avec rendu côté serveur (SSR), une gestion de routage avancée, et une excellente intégration avec des services comme Google Maps. Il garantit une expérience utilisateur fluide et réactive.[17].

#### **Backend**

Environnement : Node.js avec Express.js

Node.js est particulièrement adapté pour les applications web en temps réel grâce à sa nature asynchrone. Express.js facilite la création d'API REST robustes tout en restant léger et modulaire.

Cette combinaison offre une communication rapide entre le client et le serveur, avec une structure claire des routes et des contrôleurs.[18] .

## Base de données

### MongoDB

Base de données NoSQL souple, idéale pour stocker des documents clients, les itinéraires calculés, et l'historique des tournées. L'utilisation de Mongo DB permet une interaction fluide avec Node.js.[19] .

#### 3.2.2. Pourquoi ce choix technologique ?

- **Stack JavaScript Full-Stack** : cohérence du langage entre frontend et backend.
- **Technologies open-source** : communautés actives, documentation abondante.
- **Facilité d'intégration avec des API externes** : en particulier OSRM pour le calcul d'itinéraires.
- **Structure évolutive et adaptée aux besoins d'une PME** : évolutivité, performance, et simplicité.
- **3.3 Architecture logicielle** L'application repose sur une architecture web en trois couches : frontend, backend et base de données. Cette séparation des responsabilités facilite la lisibilité, la maintenance, ainsi que l'extension future du système.

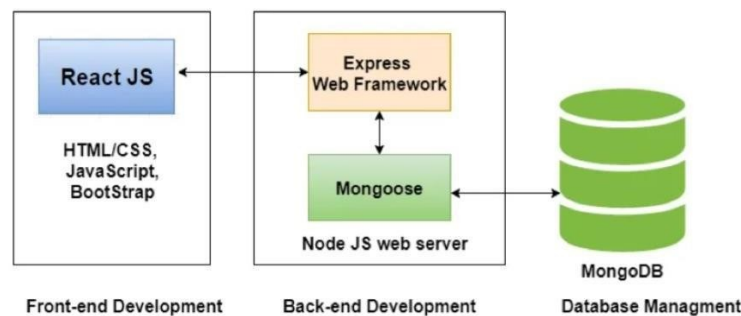


Figure 6:Schéma d'architecture simplifié

#### 3.3.1. Frontend (interface utilisateur)

Développé avec Next.js / React.js, il offre une interface conviviale permettant :

- L'ajout ou la modification de clients via un formulaire assisté par carte interactive,
- Le lancement d'une optimisation avec retour visuel rapide,
- L'affichage des tournées sur carte (intégration de OSRM – Open Source Routing Machine).

#### 3.3.2. Backend (API serveur)

- Conçu avec Node.js et Express.js, le backend assure :

- La gestion des clients (ajout, mise à jour, suppression),
- L'appel et le traitement de l'algorithme d'optimisation VRP,
- le calcul des distances via OSRM,
- La génération et l'enregistrement des tournées.

### 3.3.3 Base de données (MongoDB)

- La base de données permet de stocker :
- Les fiches clients : coordonnées, identifiants, quantité demandée,
- Les résultats des optimisations : véhicules, distances, points desservis,
- L'historique des tournées, utile pour les bilans logistiques ou les analyses futures.

```
// Fonction asynchrone pour récupérer les données des clients
const fetchCustomers = async () => {
  try {
    // Effectue une requête GET à l'URL spécifiée pour obtenir les clients
    const response = await axios.get('http://localhost:3000/api/customers');
    // Affiche les données des clients reçues de l'API dans la console
    console.log('Clients:', response.data);
  } catch (error) {
    // En cas d'erreur pendant la requête, affiche un message d'erreur dans la console
    console.error('Erreur lors de la récupération des clients:', error);
  }
};

fetchCustomers(); // Appelle la fonction pour exécuter la récupération des clients
```

Figure 7: Capture du Schéma de l'architecture des clients

Ce code émet une requête de type GET vers l'API backend afin de récupérer la liste complète des clients

```
// Fonction asynchrone pour récupérer les données des lieux de départ
const fetchDepartures = async () => {
  try {
    // Effectue une requête GET à l'URL spécifiée pour obtenir les lieux de départ
    const response = await axios.get('http://localhost:3000/api/departures');
    // Affiche les données des lieux de départ reçues de l'API dans la console
    console.log('Lieux de départ:', response.data);
  } catch (error) {
    // En cas d'erreur, affiche un message d'erreur
    console.error('Erreur lors de la récupération des lieux de départ:', error);
  }
};

fetchDepartures(); // Appelle la fonction pour exécuter la récupération des lieux de départ
```

Figure 8: Capture du Schéma de l'architecture de départ

Ce code émet une requête de type GET vers l'API backend afin de récupérer la liste complète de tous les points de départ configurés.

### 3.4 Interface Utilisateur :

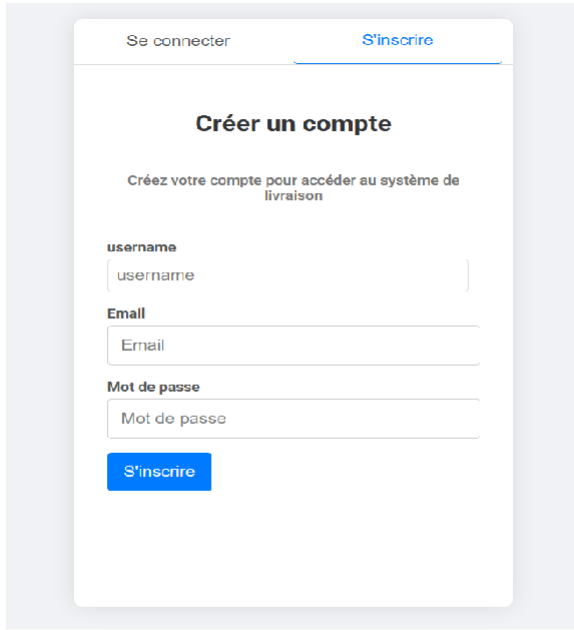


Figure 10: page d'inscription

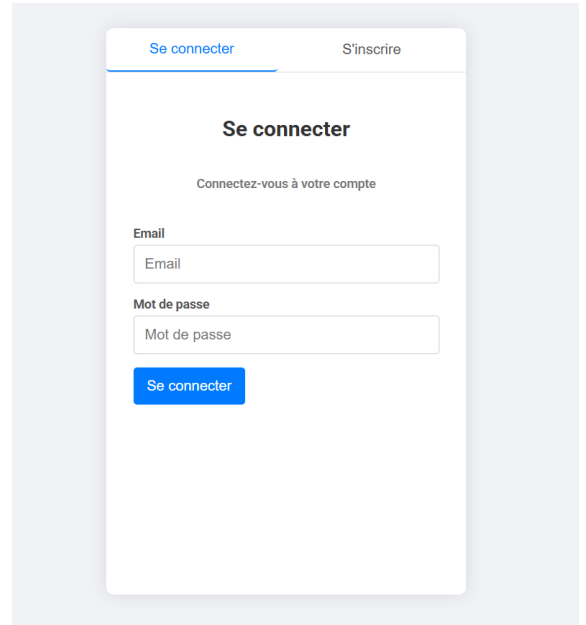


Figure 9: page de connexion

Ces deux images représentent les interfaces d'authentification d'un système permettant aux utilisateurs après inscription de créer un compte ou de se connecter à un compte déjà existant. La première image présente un écran de création de compte, et la seconde un accès au compte déjà inscrit sur le système.

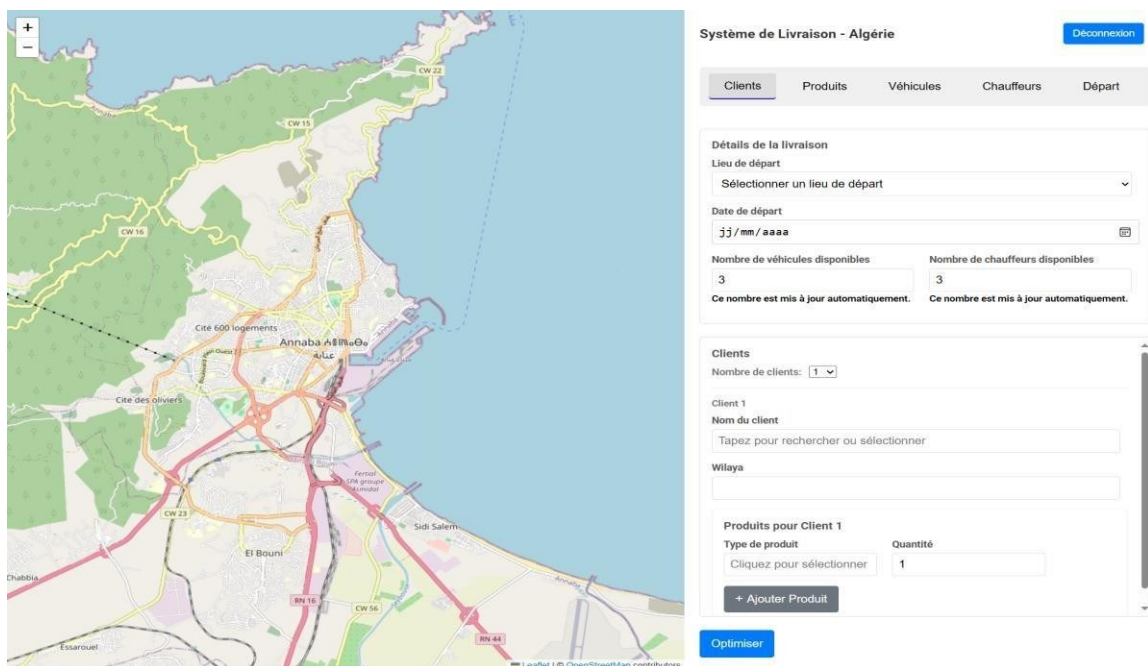


Figure 11: Home page

Cette image montre un système de livraison à l'intérieur du territoire national Algérien, il représente une carte visible à gauche et un tableau permettant d'afficher et de gérer les informations concernant la livraison à droite.

### Ajouter un nouveau client

Name:

Address:

Type:

Coordinates:   
Latitude:  Longitude:

### Liste des clients

Name	Address	Type	Coordinates (Lat, Lng)	Actions
El Hidhab	CITÉ El Hidhab	point de vente	36.21021, 5.43545	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
Abid Ali	CITÉ Abid Ali	point de vente	36.170866, 5.382086	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
Bir Ensa	CITÉ 1500 LOGEMENTS	point de vente	36.17678, 5.47654	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
Ain Trik	CITÉ Ain Trik	point de vente	36.14758, 5.44523	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
BAKHOUCHE	cite yahiaouj_satif	point de vente	36.19373, 5.41523	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
BIZARD	cite bizard	point de vente	36.18135, 5.40859	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE AMIZOUR	JRQX+7PM, W21, Amizour 06300	point de vente	36.638245, 4.849309	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE NUMIDIA	7H6F+WQ El Khroub	point de vente	36.2623716, 6.5742933	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE SIDI-KHALED TIARET	LAITERIE SIDI-KHALED TIARET	laiterie	35.3217291, 1.2990884	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE BIR KHADEM-SPA	P2GX+JJ7, Rue Mustapha Ourari, Birkhadem	laiterie	36.7241848, 3.0442505	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE BOUDOUAOU-SPA	18 Cité Benadjel, Boudouaou. Bourmerdes 35400	laiterie	36.7270801, 3.3940448	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE EL EMIR	Tizi	laiterie	35.3200724, 0.0795236	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
le pôle urbain de Tiner	cité 5 juillet 1962 ,tiner	point de vente	36.16874, 5.49729	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
Ain Arnat	Ain Arnat	point de vente	36.1849367, 5.3144869	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
Ras El Ma	Ras El Ma	point de vente	36.1174542, 5.5314017	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
LAITERIE EDOUGH	Route Elle Hadjar Lalalig BP3087, El Bouni	laiterie	36.84524780840263, 7.7518579496710265	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>

Figure 12: Ajout de nouveaux clients

Cette image représente notre onglet responsable de la gestion des clients, il est divisé en deux zones principales : À gauche, le formulaire intitulé « ajouter un nouveau client » permet à l'utilisateur de saisir les Informations (Nom, Adresse, Type, et coordonnée Latitude/Longitude) permettant la création d'un nouveau client. À droite, la section nommée « Liste des clients » montre un tableau contenant les informations de plusieurs clients avec le listing de leur Nom, Adresse, Type, Coordonnées ainsi que des actions possibles (Modifier, Supprimer) suivant les lignes du tableau.

### Ajouter un Chauffeur

Nom du Chauffeur:

Téléphone:

Statut:

### Liste des Chauffeurs

Nom	Téléphone	Statut	Actions
mohamed	+213*****	disponible	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
hicham	+213*****	disponible	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
lamin	+213*****	disponible	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
khallil	+213*****	disponible	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>

Figure 13: Ajout de nouveaux chauffeurs

Cette image représente notre onglet en charge de la gestion des chauffeurs : À gauche, le formulaire intitulé « ajouter un nouveau chauffeur » en saisissant son nom, numéro de téléphone

et statut (disponible non disponible). Ainsi que des actions possibles (Modifier, Supprimer) suivant les lignes du tableau.

The image shows a web interface for adding a new departure point. On the left, there is a form titled 'Ajouter un Nouveau Lieu de Départ' with input fields for 'Lieu (Wilaya/Ville)', 'Latitude', and 'Longitude', and buttons for 'Enregistrer le client' and 'Annuler'. On the right, there is a table titled 'Liste des Lieux de Départ' with columns for 'Lieu', 'Coordonnées (Lat, Long)', and 'Actions'. The table contains one row for 'Mezloug' with coordinates '36.103514, 5.335616' and 'Modifier' and 'Supprimer' buttons.

**Figure 14:** Ajout d'un nouveau point de départ

Cette image représente notre onglet en charge du point de départ de la tournée : À gauche, le formulaire intitulé « ajouter un nouveau départ » permet à l'utilisateur de saisir les Informations (Nom, coordonnée Latitude/Longitude), ainsi que des actions possibles (Modifier, Supprimer) suivant les lignes du tableau.

The image shows a web interface for adding a new product. On the left, there is a form titled 'Ajouter un nouveau produit' with input fields for 'Name', 'Category', 'Unit', 'Weight', and 'Price', and buttons for 'Enregistrer le produit' and 'Annuler'. On the right, there is a table titled 'Liste des produits' with columns for 'Name', 'Category', 'Unit', 'Weight', 'Price', and 'Actions'. The table contains five rows of products: 'CREME FRAICHE', 'YAOURT BRASSE', 'YAOURT FRUITE', 'RAIB', and 'LPC SACHET 01LT', each with its respective category, unit, weight, price, and 'Modifier' and 'Supprimer' buttons.

**Figure 15:** Ajout de nouveaux produits

Cette image représente notre onglet gestion des produits : À gauche, le formulaire intitulé « ajouter un nouveau produit » permet à l'utilisateur de saisir les Informations (Nom, Catégories, l'unité (g ou l) et le prix) permettant la création d'un nouveau produit. À droite, ainsi que des actions possibles (Modifier, Supprimer) suivant les lignes du tableau

The image shows a web interface for adding a new vehicle. On the left, there is a form titled 'Ajouter un nouveau véhicule' with input fields for 'Matricule', 'Type', 'Tonnage Capacity (tonnes)', and 'Status', and buttons for 'Enregistrer' and 'Annuler'. On the right, there is a table titled 'Liste des véhicules' with columns for 'Matricule', 'Type', 'capacite (tonnes):', 'Status', and 'Actions'. The table contains ten rows of vehicles, all of type 'CAMION HYNDAI FRIGO HD', with various matricule numbers, capacities, and statuses (disponible or non disponible), and 'Modifier' and 'Supprimer' buttons.

**Figure 16:** Ajout de nouveaux véhicules

Cette image représente notre onglet véhicules, il est divisé en deux zones principales : À gauche, le formulaire intitulé « ajouter un nouveau véhicule » permet à l'utilisateur de saisir les Informations (Matricule, Type, Tonnage, Cas disponible ou no disponible) permettant la création d'un nouveau véhicule. À droite, ainsi que des actions possibles (Modifier, Supprimer) suivant les lignes du tableau

### 3.5 Implémentation de l'algorithme du VRP :

L'optimisation des tournées de livraison s'appuie sur le Vehicle Routing Problem (VRP), qui consiste à déterminer le meilleur sous-ensemble de trajets pour une flotte de véhicules devant satisfaire un ensemble de clients, minimisation de la distance parcourue par exemple ou du nombre de véhicules utilisés, en respectant les contraintes, notamment de capacité des camions. repose sur un modèle hybride : d'une part, un backend Node.js gérant la communication et l'intégration avec les services externes et d'autre part, un script Python implémentant l'heuristique de résolution. L'algorithme heuristique choisi est une heuristique de proximité de type Plus Proche Voisin en raison de ses performances d'exécution. Le processus s'effectue en trois étapes principales :

- 1. Saisie des Données :** réception et contrôle de la neutralité des paramètres (localisation du dépôt, des clients, du nombre de véhicules).
- 2. Traitement :** acquisition des matrices de distances/durées via OSRM, briefing de données, lancement de l'algorithme Python, post-traitement des résultats.
- 3. Sortie :** mise en forme et envoi des routes optimisées au client.

Algorithme Employé : Heuristique du Plus Proche Voisin

L'heuristique du Plus Proche Voisin réalise les tournées en ajoutant le client non encore visité le plus proche à la tournée du véhicule, à la suite du dépôt. Les temps de service par client sont intégrés dans le calcul de la durée totale de chaque tournée.

#### 3.5.1. Saisie des Données (Backend Node.js – server/routes/vrpRoutes.js)

Le serveur Node.js réceptionne les requêtes, contrôle les données d'entrée (dépôt, clients, véhicules)

```
// POST /api/vrp/optimize
router.post('/optimize', async (req, res) => {
  try {
    const { departure_location, clients, num_vehicles } = req.body;

    // --- Step 1: Input Validation ---
    console.log('[VrpRoutes] Received VRP request with data:', { departure_location, clientsCount: clients ? clients.length : 0, num_vehicles });

    if (!departure_location || !Array.isArray(departure_location) || departure_location.length !== 2 ||
        typeof departure_location[0] !== 'number' || typeof departure_location[1] !== 'number') {
      console.error('[VrpRoutes] Validation Error: Invalid departure_location format or value.');
```

Figure 17: Saisie et Validation des Données d'Entrée VRP (Backend Node.js)

### 3.5.2. La phase du traitement (Backend Node.js & Script Python)

Cette étape essentielle regroupe des étapes telles que, l'interrogation auprès d'OSRM des données géospatiales, la préparation des données pour le solveur Python, l'exécution proprement dite du solveur, et le post-traitement pour rendre les résultats exploitables et exploitables.

#### a. L'Acquisition de la Matrice de Distances / Durées (OSRM)

Le service (distanceMatrixService.js) interroge OSRM pour se faire communiquer les temps de trajet et distances entre tous les points

```
1 // server/routes/vrpRoutes.js (suite)
2
3 // Combine le dépôt et les clients pour créer une liste complète de localisations
4 const allLocations = [
5   { lat: departure_location[0], lng: departure_location[1] },
6   ...clients.map(c => ({ lat: c.location[0], lng: c.location[1] }))
7 ];
8 // Appel du service OSRM pour obtenir les matrices de distances et de durées
9 const { distances, durations } = await getDistanceMatrix(allLocations);
10
11 // Vérification de la validité des matrices OSRM
12 if (!distances || !durations || distances.length !== allLocations.length) {
13   return res.status(500).json({ message: 'Erreur lors de la récupération des données OSRM.' });
14 }
15
16 // server/services/distanceMatrixService.js (extrait pertinent)
17 import axios from 'axios';
18
19 // Fonction pour récupérer la matrice des distances et durées via OSRM
20 export async function getDistanceMatrix(locations) {
21   const osrmCoords = locations.map(loc => `${loc.lng},${loc.lat}`).join(';');
22   const osrmUrl = `http://router.project-osrm.org/table/v1/driving/${osrmCoords}?annotations=duration,distance`;
23   try {
24     const response = await axios.get(osrmUrl);
25     return { distances: response.data.distances, durations: response.data.durations };
26   } catch (error) {
27     throw new Error("Impossible d'obtenir la matrice de distances/durées de la part du service OSRM.");
28   }
29 }
```

Figure 18: Acquisition des Données Géospatiales via OSRM (vrpRoutes.js)

#### b. Le traitement des données pour le solveur Python

Les données sont ensuite structurées en tant qu'objet Python de type JSON qui sera passé en entrée au script Python. Un temps de service fixe par client est également établi.

```
// --- Step 3: Prepare data for Python VRP Solver ---
// Définir la durée de service fixe par client à 45 minutes
const serviceTimePerClientMinutes = 45;

const vrpDataForPython = {
  departure_location: departure_location,
  clients: clients,
  num_vehicles: parsedNumVehicles,
  distance_matrix: distances,
  duration_matrix: durations,
  // Passer la durée de service totale par client au script Python
  service_time_per_client_sec: serviceTimePerClientMinutes * 60,
};
```

Figure 19: Préparation des Données pour le Solveur

### c. Le Post-Traitement et la Génération de Chemins Détaillés

Les indices fournis sont convertis en coordonnées réelles, avant de recevoir la demande des polygones détaillées à OSRM

```

1  [vrpRoutes] OSRM returned empty or incomplete distance/duration matrices.");
2  return res.status(500).json({ message: "Failed to get complete routing data from OSRM service. Check OSRM server status or input coordinates." });
3  }
4
5  // --- Step 3: Prepare data for Python VPP solver ---
6  // Définir la durée de service fixe par client à 45 minutes
7  const serviceTimePerClientMinutes = 45;
8
9  const vrpDataForPython = {
10   departure_location: departure_location,
11   clients: clients,
12   num_vehicles: numVehicles,
13   distance_matrix: distances,
14   duration_matrix: durations,
15   // Passer la durée de service totale par client au script Python
16   service_time_per_client_min: serviceTimePerClientMinutes * 60,
17 };
18
19 console.log("vrpRoutes) Calling Python VPP solver...");
20 const pythonSolverResult = await solveVPP(vrpDataForPython);
21 console.log("vrpRoutes) Python VPP solver raw results: ", JSON.stringify(pythonSolverResult, null, 2));
22
23 let optimized_routes_with_paths = [];
24 let unvisitedClientsCount = pythonSolverResult.unvisited_clients_count || 0;
25
26
27 if (pythonSolverResult.okArray.isArray(pythonSolverResult.optimized_routes)) {
28   // --- Step 4: Process VPP solver output and get detailed paths ---
29   optimized_routes_with_paths = await Promise.all(
30     pythonSolverResult.optimized_routes.map(async (route) => {
31       if (route.stops_indices || Array.isArray(route.stops_indices) || route.stops_indices.length > 0) {
32         console.warn("vrpRoutes) Route for vehicle {route.vehicle_id} has invalid 'stops_indices': skipping.");
33         return { route, path: [], distance: 0, duration: 0, travel_duration: 0, service_duration: 0 };
34       }
35
36       const routeCoordinates = route.stops_indices.map(index => {
37         if (index <= 0 || index > allocations.length) {
38           const loc = allocations[index];
39           return { lat: loc.lat, lng: loc.lng };
40         } else {
41           console.warn("vrpRoutes) Invalid index {index} found in route for vehicle {route.vehicle_id}: This index might not correspond to a valid location in 'allocations'.");
42           return null;
43         }
44       });
45       // Filter (used == coord loc null);
46
47       let detailedPathInfo = { path: [], distance: 0, duration: 0 };
48       if (routeCoordinates.length > 0) {
49         try {
50           detailedPathInfo = await getDetailedRoutePath(routeCoordinates);
51         } catch (pathError) {
52           console.warn("vrpRoutes) Could not get detailed path for vehicle {route.vehicle_id} from OSRM: {pathError.message}. Using direct coordinates as path and solver's distance/duration.");
53           detailedPathInfo.path = routeCoordinates;
54           detailedPathInfo.distance = route.distance || 0;
55           // Ne jamais utiliser la durée totale de solver Python, car elle inclut les temps de service
56           detailedPathInfo.duration = route.duration || 0;
57         }
58       } else if (routeCoordinates.length === 1) {
59         detailedPathInfo.path = routeCoordinates;
60         detailedPathInfo.distance = 0;
61         detailedPathInfo.duration = 0;
62       }
63
64       const clientOrderByRoute = route.stops_indices
65         .filter(id => id > 0)
66         .map(id => {
67           // (id - 1) * 60 min - 1 + clients.length()
68           return clients[id - 1].id;
69         })
70         .filter(id => id != null);
71
72       return {
73         vehicle_id: route.vehicle_id,
74         stops: routeCoordinates,
75         path: detailedPathInfo.path,
76         distance: (detailedPathInfo.distance + 0 || detailedPathInfo.path.length > 0) ? detailedPathInfo.distance : (route.distance || 0),
77         // Toujours utiliser la durée totale de solver Python, car elle inclut les temps de service
78         duration: route.duration,
79         travel_duration: route.travel_duration, // Durée de voyage seule renvoyée par Python
80         service_duration: route.service_duration, // Durée de service seule renvoyée par Python
81         clients_served: clientOrderByRoute
82       };
83     })
84   );
85 }

```

Figure 20: Post-traitement et Génération des Chemins Détaillés (vrpRoutes.js)

### 3.5.3. La Sortie (Backend Node.js - server/routes/vrpRoutes.js)

En sortie on obtient une réponse structurée en JSON, envoyée au client.

```
1 res.json({ message: 'Optimization successful', optimized_routes: optimized_routes_with_paths, unvisited_clients_count: unvisitedClientsCount });
2
3 } catch (error) {
4   console.error('[VrpRoutes] Error in VRP optimization endpoint:', error);
5   res.status(500).json({
6     message: 'Internal server error during route optimization.',
7     error: error.message,
8     stack: process.env.NODE_ENV === 'development' ? error.stack : undefined
9   });
10 }
11 });
12
13 export default router;
```

**Figure 21:** Renvoi de la Réponse Finale au Client (Backend Node.js)

### 3.6 Problèmes rencontrés et solutions

L'élaboration de l'application d'optimisation des tournées de livraison a constitué un ensemble de défis techniques et pratiques. En résumé, les principaux problèmes rencontrés, accompagnés de leurs solutions mises en œuvre.

1. **Affichage cartographique dynamique** : Difficulté à représenter les trajets optimisés sur la carte de manière lisible, surtout avec plusieurs véhicules, dans ce cas on a utilisé la bibliothèque Leaflet.js avec des polygones colorés, une pour chaque tournée. Des marqueurs interactifs permettent de visualiser les informations détaillées sur chaque client (nom, adresse, quantité)
  2. **Gestion des coordonnées géographiques** : Certaines adresses saisies par les utilisateurs n'étaient pas converties correctement en coordonnées GPS, ce qui bloquait le calcul des distances, dans ce cas on a Ajout d'un géocodeur automatique avec gestion des erreurs, et possibilité de placer manuellement un point sur la carte si le géocodage échoue.
  3. **Limitations en temps de traitement** : Lorsque le nombre de clients dépasse 100, le temps de calcul augmente considérablement, on a introduction d'une limite temporelle du nombre de clients à optimiser,
  4. **Calcul des distances entre clients** : L'utilisation initiale de l'API Google Maps pour générer une matrice de distances s'est révélée limitée : nombre de requêtes restreint, lenteur dans le traitement, et coût élevé pour des volumes importants pour cette raison ont utilisé **OSRM (Open Source Routing Machine)**, installé en local sur un serveur. OSRM a permis de :
- Générer rapidement les matrices de distances,
  - Éviter les limites de quota,
  - Conserver les données sensibles au sein de l'entreprise.

Ces défis ont permis de renforcer la robustesse de l'application, tout en conservant une expérience utilisateur fluide. L'adaptation aux contraintes terrain a guidé les choix technologiques et algorithmiques.

## Chapitre 4 : Résultats, discussion, conclusion et perspectives

### 4.1 Présentation des résultats :

Après l'intégration complète de l'algorithme VRP et le déploiement de l'application, plusieurs scénarios de test ont été réalisés afin d'évaluer la performance de la solution proposée. Ces tests ont été effectués à partir de données réelles de la laiterie TELL, incluant des listes de clients, leurs emplacements géographiques et les quantités de produits demandés.

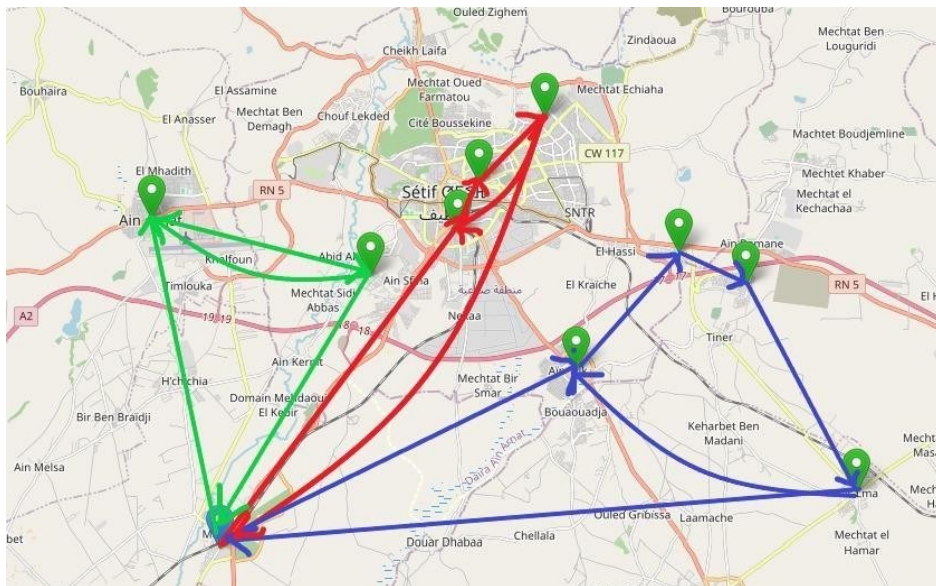


Figure 22: Avant Optimisation

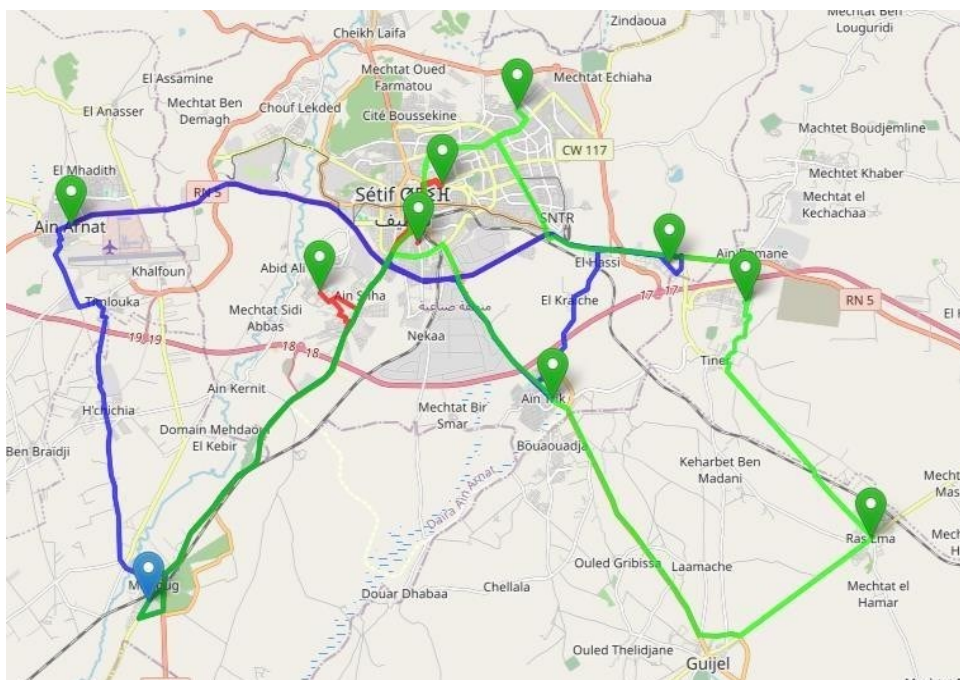


Figure 23: Après Optimisation

	Avant l'optimisation	Après l'optimisation
Nombre de clients	2	1
Durée max des tournée	5h	3h
Distance moyen(km)	170	85
Coût carburant estimés	Élevé	Réduit

**Tableau 1:** Comparaison avant et après l'optimisation

## 4.2 Discussion des résultats

Les améliorations apportées à la gestion logistique quotidienne grâce à l'intégration de l'algorithme VRP se sont révélées nombreuses et significatives.

1. Réduction du temps de tournée  
Avant l'optimisation, les tournées pouvaient durer jusqu'à 5 ou 6 heures, ce qui avait un impact direct sur la fatigue des chauffeurs et la productivité globale. Grâce à la réorganisation intelligente des trajets, la durée de chaque tournée a été réduite à environ 3 heures. Cela permet non seulement de respecter les normes de travail, mais aussi d'envisager, si nécessaire, un double circuit journalier pour certains chauffeurs.
2. Réduction du kilométrage total  
L'algorithme a permis de supprimer les trajets redondants, notamment les allers-retours inutiles pour récupérer les box de lait vides, en combinant livraison et ramassage en un seul passage. Cette stratégie a pour effet direct de limiter l'usure des véhicules et les frais d'entretien associés.
3. Économie de carburant et réduction de l'empreinte carbone  
Une baisse d'environ 30 % des trajets liés à une mauvaise organisation permet de réduire considérablement la consommation de carburant. Cette optimisation a un double impact : un gain économique immédiat pour l'entreprise et une contribution positive à la réduction de son empreinte carbone.

## 4.3 Conclusion

Ce travail de fin d'études avait pour objectif principal de concevoir une solution numérique capable d'automatiser et d'optimiser les tournées de livraison de la laiterie TELL, en s'appuyant sur l'algorithme bien connu du Vehicle Routing Problem (VRP).

À travers une architecture moderne (Node.js, React, MongoDB, OSRM), nous avons développé une application web fonctionnelle, intuitive, et adaptée aux contraintes spécifiques du transport de produits frais. Les résultats obtenus ont démontré que cette approche permet de :

- Réduire sensiblement les distances parcourues,

- Diminuer les coûts liés au carburant et au temps de travail,
- Équilibrer la charge entre les chauffeurs,
- Améliorer la traçabilité et la gestion des tournées.

La problématique posée initialement – "**Comment optimiser les tournées de livraison pour améliorer la rentabilité et la performance logistique tout en respectant les contraintes opérationnelles ?**" a trouvé une réponse concrète à travers ce projet. Il marque le passage d'un mode de gestion manuel à un système data-Driven, efficace et évolutif.

Cependant, cette solution reste une première étape vers un système plus complet, capable de prendre en charge des contraintes supplémentaires et d'évoluer vers une digitalisation plus poussée.

#### 4.4 Perspectives

Même si l'application développée a déjà apporté une nette amélioration au fonctionnement logistique de la laiterie TELL, plusieurs axes d'évolution peuvent être envisagés à court ou moyen terme :

1. **Intégration du trafic en temps réel**  
L'ajout de données de circulation (via des API comme Google Maps ou Here Maps) permettrait d'ajuster dynamiquement les tournées en cas d'embouteillage, d'accident ou de travaux, offrant ainsi une meilleure réactivité et précision dans les délais de livraison.
2. **Prise en compte des créneaux horaires (VRPTW)**  
Certains points de vente imposent des heures spécifiques de livraison. Intégrer ces contraintes temporelles renforcerait l'efficacité de l'algorithme en s'alignant encore mieux sur les besoins réels des clients.
3. **Développement d'une application mobile pour les livreurs**  
Une version mobile permettrait aux chauffeurs de :
  - Visualiser leur itinéraire en temps réel,
  - Valider les livraisons à chaque étape,
  - Suivre leur performance quotidienne.
4. **Amélioration de l'optimisation via l'intelligence artificielle**  
À moyen terme, le recours à des métaheuristiques avancées (algorithmes génétiques, colonies de fourmis...) permettrait de gérer des cas plus complexes (multi-dépôts, produits différenciés, contraintes multiples) avec une meilleure précision.
5. **Intégration dans un système d'information global (ERP)**  
Coupler cette application avec d'autres modules (gestion des stocks, des commandes,

de la facturation) permettrait une automatisation complète de la chaîne logistique et commerciale.

En conclusion, ce projet représente une base solide et évolutive pour la transformation digitale de la gestion logistique de l'entreprise. Il constitue un levier stratégique en faveur de l'amélioration de la rentabilité, du renforcement de la durabilité et de l'accroissement de la compétitivité. Il ouvre ainsi la voie à une modernisation progressive et cohérente des processus logistiques.

## Bibliographie

- [1] : Documents internes de l'entreprise TELL.
- [2] : <http://www.laiterietell.dz/>
- [3] : Laporte Gilbert [et al.] Classical and modern heuristics for the vehicle routing problem [Revue]. - [s.l.] : Intl. Trans. in Op. Res, 2000. - 285-300 : Vol. 7.
- [4] : Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. Management science, 6(1), 80-91.
- [5] : Alba, E., & Dorronsoro, B. (2006). Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm. Information processing letters, 98(6), 225-230.
- [6] : AKLI Meriem Problème de tournées de véhicule avec contraintes de fenêtre de temps [Rapport] : Thèse de Magister / UNIVERSITÉ MOULOUD MAMMERI. - TIZI OUZOU : [s.n.], 2013.
- [7] : ILLOUL, Y. (2020). Optimisation des coûts de collecte du lait pour l'entreprise Danone (Doctoral dissertation, M. BRAHAMI Mustapha Anwar) .
- [8] : BENANTAR Abdelaziz Optimisation pour des problèmes industriels de tournées de véhicules : Vers une transition énergétique [Rapport] : thèse de doctorat / service informatique ; l'université du Havre-Normandie. - 2017.
- [9] : Mahmud, N., & Haque, M. M. (2019, February). Solving multiple depot vehicle routing problem (MDVRP) using genetic algorithm. In 2019 International conference on electrical, computer and communication engineering (ECCE) (pp. 1-6). IEEE.
- [10] : Eglese R, Li LYOL et Fu Z A new tabu search heuristic for the open vehicle routing problem [Revue]. - China : Journal of the Operational Research Society, 2004. - 56 : Vol. 3.
- [11] : Laporte, G., & Osman, I. H. (1995). Routing problems: A bibliography. Annals of operations research, 61, 227-262.
- [12] : Gardeux, V. (2011). Conception d'heuristiques d'optimisation pour les problèmes de grande dimension : application à l'analyse de données de puces à ADN (Doctoral dissertation, Paris Est).
- [13] : Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations research, 12(4), 568-581.
- [14] : Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2), 254-265.
- [15] : Benaichouche, A. N. (2014). Conception de métaheuristiques d'optimisation pour la segmentation d'images : application aux images IRM du cerveau et aux images de tomographie par émission de positons (Doctoral dissertation, Université Paris-Est).
- [16] : Merdjaoui, B. (2006). Optimisation multi-objectif par algorithmes génétiques et approche pareto des paramètres d'usinage sous contraintes des limitations de production (Doctoral dissertation, Boumerdes, Université M'hamed Bouguerra. Faculté des Sciences de L'ingénieur).

[17] : Ballamudi, V. K. R., Lal, K., Desamsetti, H., & Dekkati, S. (2021). Getting Started Modern Web Development with Next. js: An Indispensable React Framework. Digitalization & Sustainability Review, 1(1), 1-11.

[18] : Peters, C. (2017). Building rich internet applications with node. js and express. js. Rich Internet Applications w/HTML and Javascript, 15.

[19] : Phaltankar, A., Ahsan, J., Harrison, M., & Nedov, L. (2020). MongoDB Fundamentals: A hands-on guide to using MongoDB and Atlas in the real world. Packt Publishing Ltd.