***Licence d’excellence en intelligence artificiel***

***Rapport de Projet de Fin du module***

***l'Optimisation des Tournées de Livraison : Résolution d'un Scénario TSP avec Heuristiques Simples (NN, ACO)***

Réalisé par :

* **HANANE AYAR**
* **AYMAN CHABBAKI**
* **BADREDDINE AFYF**

Encadré par :

* **Prof. El Habib BEN LAHMAR**
* **Dr. Zakaria EL FAKIR**

Année universitaire : 2024/2025

# Remerciements

Nous tenons à exprimer notre profonde gratitude à notre professeur M. EL HABIB Ben Lahmar pour l’attention qu’il a portée à notre projet, pour ses conseils avisés, sa pédagogie exemplaire ainsi que pour sa patience et sa bienveillance tout au long de ce travail. Grâce à son expertise, ses encouragements constants et sa capacité à susciter en nous la rigueur scientifique, nous avons pu progresser avec confiance et méthode. Sa disponibilité et son implication ont été des éléments clés dans l’aboutissement de ce projet.

Nos remerciements les plus sincères s’adressent également à M. Zakaria El fakir, notre encadrant, pour son accompagnement précieux, ses orientations pertinentes et ses remarques constructives qui ont considérablement enrichi notre démarche. Son implication constante, son esprit d’écoute et sa capacité à nous guider avec justesse ont été déterminants pour la réussite de ce travail. Son soutien technique et moral a joué un rôle essentiel tout au long de cette expérience.

À travers ces lignes, nous souhaitons leur témoigner toute notre reconnaissance pour leur engagement et la qualité de leur encadrement, qui ont fait de ce projet une véritable opportunité d’apprentissage et d’épanouissement personnel et professionnel.

# Dédicace

Ce projet est tout particulièrement dédié à nos chers parents, piliers inébranlables de notre vie, dont l’amour inconditionnel, les innombrables sacrifices et le soutien indéfectible ont toujours été pour nous une source de motivation, de courage et de persévérance.

À nos mères, dont les prières silencieuses, la tendresse infinie et la foi en nous ont illuminé nos chemins même dans les moments les plus difficiles. À nos pères, exemples de sagesse, de travail acharné et de dévouement, qui nous ont appris la valeur de l’effort et la dignité face aux épreuves.

Chaque réussite, chaque étape franchie dans notre parcours académique est le fruit de leur engagement discret mais constant, de leurs encouragements quotidiens et de leur confiance profonde en notre potentiel.

Ce travail leur est dédié avec tout notre amour, notre respect et notre gratitude éternelle. Puisse-t-il être le reflet, même modeste, de tout ce qu’ils ont semé en nous.

# Presentation of the faculty of sciences Ben M’sik

La Faculté des Sciences Ben M’Sick (FSBM) (2), relevant de l'Université Hassan II de Casablanca, a été créée en 1984. Depuis sa fondation, elle s’est imposée comme un pilier de l’enseignement supérieur scientifique au Maroc, alliant formation académique de qualité et développement actif de la recherche scientifique.

Durant ses premières années, et jusqu’en 2003, la faculté proposait des formations diversifiées allant du premier cycle (DEUG) au second cycle (licence sciences Bac+4), couvrant plusieurs disciplines scientifiques. Dès 1989, elle s’engage dans la formation de troisième cycle (CEA et DES), rendue possible grâce à la mise en place de plusieurs équipes et laboratoires de recherche dynamiques

L’année 1997 marque un tournant important avec la création des Unités de Formation et de Recherche (UFR). Cette restructuration vise à renforcer la synergie entre formation et recherche, en fédérant les chercheurs autour de thématiques émergentes et porteuses d'innovation.

En 2003, dans le cadre de la réforme de l’enseignement supérieur et l’adoption du système LMD (Licence – Master – Doctorat), la faculté réorganise ses programmes autour d’une formation modulaire et semestrielle, favorisant ainsi une meilleure flexibilité pédagogique et une harmonisation avec les standards internationaux.

La faculté franchit une nouvelle étape en 2008 avec la création du Centre d’Études Doctorales (CED) intitulé « Sciences et Applications ». Ce centre regroupe l’ensemble des structures de recherche accréditées et constitue un levier stratégique pour la formation doctorale et la production scientifique de haut niveau.

Enfin, depuis décembre 2015, dans le cadre d’une initiative de structuration renforcée lancée par l’Université Hassan II, la FSBM a procédé à une réorganisation complète de ses équipes et laboratoires. Cette démarche a abouti à la mise en place de 19 laboratoires de recherche, témoignant de la richesse, de la diversité et de la vitalité scientifique de l’établissement.

Grâce à ses infrastructures, à la qualité de son encadrement, et à son engagement constant en faveur de la recherche et de l’innovation, la Faculté des Sciences Ben M’Sick constitue aujourd’hui un acteur majeur du paysage scientifique national et régional.

# Abstract

This study investigates the problem of delivery route optimization through the well-known Travelling Salesman Problem (TSP), a classical combinatorial optimization challenge where the objective is to determine the shortest possible route that visits a set of cities exactly once and returns to the origin city. The primary goal is to minimize the total travel cost, which may represent distance, time, or other logistical factors in real-world delivery contexts.

To address this problem, two heuristic methods are explored and compared: the Nearest Neighbor (NN) algorithm and the Ant Colony Optimization (ACO) algorithm. The NN method is a simple and fast greedy approach, which constructs a tour by repeatedly visiting the nearest unvisited city. Although efficient in terms of execution time, NN often yields suboptimal solutions due to its myopic nature. In contrast, ACO is a bio-inspired metaheuristic based on the foraging behavior of ants, which utilizes pheromone trails and probabilistic transitions to explore multiple solution paths and gradually converge towards better routes.

The performance of both algorithms is evaluated on small-scale instances from the TSPLIB dataset, a standard benchmark in TSP research. In order to improve the effectiveness of ACO, automatic parameter tuning is applied using Optuna, a state-of-the-art hyperparameter optimization framework. This tuning process allows the algorithm to automatically discover near-optimal parameter configurations, significantly enhancing the quality of the resulting solutions without notably increasing the computation time.

Experimental results indicate that while NN provides a quick and simple baseline, the optimized version of ACO achieves a better trade-off between solution quality and computational efficiency. These findings highlight the potential of ACO, especially when fine-tuned, as a viable and scalable approach for addressing delivery route optimization in real-world logistics applications.

# Table des matières

[Remerciements 1](#_Toc199711745)

[Dédicace 2](#_Toc199711746)

[Presentation of the faculty of sciences Ben M’sik 3](#_Toc199711747)

[Abstract 4](#_Toc199711748)

[Table des matières 5](#_Toc199711749)

[Liste des figures 7](#_Toc199711750)

[Liste des tableaux 8](#_Toc199711751)

[Introduction générale 9](#_Toc199711752)

[Chapitre 1 : Contexte général du projet 12](#_Toc199711753)

[1. Introduction 12](#_Toc199711754)

[2. Problématique et cahier des charges 13](#_Toc199711755)

[2.1. Problématique et solution 13](#_Toc199711756)

[2.2. Cahier des charges 14](#_Toc199711757)

[3. Planification du projet 15](#_Toc199711758)

[4. Conclusion 16](#_Toc199711759)

[17](#_Toc199711760)

[Chapitre 2 : Analyse et Conception Théorique 18](#_Toc199711761)

[1. Choix méthodologiques et justification 18](#_Toc199711762)

[1-1. Optimisation combinatoire et formulation du TSP 18](#_Toc199711763)

[1-2. Heuristiques gloutonnes : Nearest Neighbor (NN) 18](#_Toc199711764)

[1-3. Métaheuristiques bio-inspirées : Ant Colony Optimization (ACO) 19](#_Toc199711765)

[1-4. Optimisation automatique avec Optuna 19](#_Toc199711766)

[2. Présentation du jeu de données 20](#_Toc199711767)

[Conclusion 21](#_Toc199711768)

[Chapitre 3 : Implémentation et résultats 23](#_Toc199711769)

[1) Introduction : 23](#_Toc199711770)

[2) Optimisation des Hyperparamètres de l’Algorithme ACO: 24](#_Toc199711771)

[3) Évaluation Comparative 26](#_Toc199711772)

[4) Comparaison Visuelle des Coûts Obtenus 26](#_Toc199711773)

[Conclusion Générale 28](#_Toc199711774)

# Liste des figures

[Figure 1:Travelling Salesman Problem TSP 10](#_Toc199671830)

[Figure 2:Diagramme de Gantt 16](#_Toc199671831)

[Figure 3: présentation du dataset 21](#_Toc199671832)

[Figure 7: importance des hyperparamètres 25](#_Toc199671833)

[Figure 8: comparaison des Coûts 27](#_Toc199671834)

# Liste des tableaux

[Table 1: Description des étapes du projet 13](#_Toc199607169)

[Table 2: Spécifications technique 14](#_Toc199607170)

# Introduction générale

Le problème du voyageur de commerce (Travelling Salesman Problem ou TSP) figure parmi les défis les plus emblématiques et les plus étudiés de l’optimisation combinatoire. Il s’agit de déterminer un parcours de longueur minimale permettant à un voyageur de visiter une seule fois chaque ville d’un ensemble donné, puis de revenir à son point de départ. Bien que l’énoncé du problème soit simple et intuitif, sa résolution devient exponentiellement plus complexe à mesure que le nombre de villes augmente. En effet, le TSP appartient à la classe des problèmes NP-difficiles, ce qui signifie qu’aucun algorithme connu ne permet de le résoudre de manière exacte en un temps raisonnable pour des instances de grande taille.

Dans un monde de plus en plus tourné vers l’efficacité logistique, la résolution pratique du TSP prend tout son sens, notamment dans les domaines de la livraison de colis, de la gestion des flottes de véhicules, ou encore de la planification des tournées commerciales. Optimiser les itinéraires permet non seulement de réduire les coûts (carburant, temps de travail, usure des véhicules), mais aussi d’améliorer la qualité de service, la ponctualité et la durabilité des opérations de transport.

Ce projet s’inscrit dans cette optique et propose une étude comparative de deux approches heuristiques couramment utilisées pour résoudre le TSP. La première est l’algorithme du Plus Proche Voisin (Nearest Neighbor, NN), une méthode gloutonne simple et rapide, qui construit une solution en choisissant à chaque étape la ville la plus proche. Bien que peu coûteuse en temps de calcul, cette méthode peut produire des solutions éloignées de l’optimum global, notamment dans des cas où des décisions locales entraînent des conséquences globales inefficaces.

La seconde approche étudiée est l’Optimisation par Colonie de Fourmis (Ant Colony Optimization, ACO), une métaheuristique bio-inspirée fondée sur le comportement collectif des fourmis lors de la recherche de nourriture. L’ACO exploite des mécanismes de communication indirecte par phéromones pour orienter la construction de solutions, permettant une exploration intelligente de l’espace des solutions. Cette méthode, bien que plus complexe à paramétrer et plus gourmande en temps de calcul que NN, offre généralement de meilleures performances en termes de qualité de solution.

L’objectif principal de ce projet est donc de comparer ces deux heuristiques, à travers une série d’expérimentations sur des instances de petite taille issues de la bibliothèque TSPLIB. Un accent particulier est mis sur l’optimisation automatique des paramètres de l’algorithme ACO à l’aide de la bibliothèque Optuna, spécialisée dans l’ajustement d’hyperparamètres par des méthodes intelligentes (comme l’optimisation bayésienne). Cette étape vise à tirer le meilleur parti du potentiel de l’ACO, en améliorant la qualité des solutions générées sans engendrer un surcoût excessif en temps de calcul.

Ainsi, ce travail met en lumière les forces et les limites de chaque approche, tout en illustrant l’impact significatif qu’une bonne configuration algorithmique peut avoir sur les performances d’une heuristique dans le cadre de problèmes de logistique réels.



Figure 1:Travelling Salesman Problem TSP

**Chapitre 1 : Contexte général du projet**

# Chapitre 1 : Contexte général du projet

## Introduction

Dans un monde de plus en plus connecté et dynamique, la logistique et la gestion des livraisons jouent un rôle crucial dans la performance des entreprises. Qu'il s'agisse de commerce en ligne, de distribution alimentaire, de services de transport ou de logistique industrielle, la capacité à livrer rapidement, efficacement et à moindre coût est devenue un avantage concurrentiel stratégique. Les clients sont de plus en plus exigeants en matière de délais, de transparence et de flexibilité, ce qui impose aux entreprises une optimisation rigoureuse de leurs processus logistiques.

Parmi les nombreux défis rencontrés dans ce domaine, l’optimisation des tournées de livraison figure parmi les plus complexes et les plus déterminants. Elle consiste à planifier des trajets pour une ou plusieurs entités (véhicules, livreurs, commerciaux, etc.) de manière à minimiser les distances parcourues, réduire les coûts, tout en respectant certaines contraintes opérationnelles. Cette problématique est couramment modélisée par le problème du voyageur de commerce (TSP), un classique de l’optimisation combinatoire, reconnu pour sa complexité algorithmique et sa pertinence dans de nombreuses applications concrètes.

Le TSP, bien que formulé de manière simple – trouver le plus court chemin passant par un ensemble de villes une seule fois et revenant à la ville de départ – devient très difficile à résoudre dès que le nombre de villes augmente, du fait de sa nature NP-difficile. Pour contourner l’infaisabilité du calcul exact sur des instances de taille réelle, des approches heuristiques et métaheuristiques ont été développées, permettant de générer rapidement des solutions approximatives de bonne qualité.

Parmi ces méthodes, l’algorithme du Plus Proche Voisin (Nearest Neighbor, NN), de nature gloutonne, et l’Optimisation par Colonie de Fourmis (Ant Colony Optimization, ACO), inspirée des comportements naturels des insectes sociaux, se distinguent par leur simplicité d’implémentation et leur efficacité. Cependant, ces méthodes présentent des performances variables en fonction de la configuration des instances et des paramètres utilisés, notamment dans le cas de l’ACO.

## Problématique et cahier des charges

### Problématique et solution

**Problématique :**

Dans un contexte où l’efficacité logistique devient un facteur clé de compétitivité, les entreprises cherchent en permanence à optimiser leurs tournées de livraison pour réduire les coûts opérationnels, les délais, et l’impact environnemental. Le problème du voyageur de commerce (TSP) représente un modèle mathématique classique de cette problématique. Cependant, sa résolution exacte est rapidement limitée par la croissance exponentielle des possibilités dès que le nombre de villes augmente. Cette complexité justifie le recours à des approches heuristiques et métaheuristiques pour obtenir de bonnes solutions en un temps raisonnable.

Dans ce cadre, deux méthodes sont particulièrement intéressantes : le Plus Proche Voisin (Nearest Neighbor), une heuristique gloutonne simple et rapide, et l’Optimisation par Colonie de Fourmis (ACO), une métaheuristique bio-inspirée capable de générer des solutions de meilleure qualité. Toutefois, l’efficacité de l’ACO dépend fortement du choix de ses paramètres. D’où l’intérêt d’utiliser une solution comme Optuna pour ajuster ces paramètres automatiquement.

**Solution :**

Pour répondre à la problématique posée, plusieurs approches ont été envisagées. La première consiste à implémenter l’algorithme Nearest Neighbor (NN), une solution rapide mais qui tend à donner des résultats sous-optimaux sur certaines configurations. La seconde approche est l’algorithme Ant Colony Optimization (ACO), plus avancé, mais nécessitant un paramétrage précis. Pour améliorer les performances de l’ACO, une troisième solution est proposée : intégrer un système de tuning automatique des paramètres à l’aide de la bibliothèque Optuna, permettant de tester différentes configurations de manière intelligente et efficace. Ces approches seront comparées en termes de qualité de la solution obtenue et de temps de calcul requis, sur des instances réelles issues de la bibliothèque TSPLIB.

### Cahier des charges

#### Objectifs

L’objectif principal de ce projet est de mettre en œuvre et de comparer des heuristiques pour résoudre le TSP, en vue d’identifier une solution à la fois performante et adaptée à un usage pratique dans un contexte de livraison. Plus précisément, ce projet vise à :

* Comprendre en profondeur la formulation et la complexité du TSP.
* Implémenter l’algorithme Nearest Neighbor et analyser ses résultats.
* Implémenter l’algorithme Ant Colony Optimization (ACO) et évaluer ses performances.
* Intégrer Optuna pour optimiser automatiquement les paramètres de l’ACO.
* Comparer les différentes solutions sur des instances standardisées (TSPLIB).
* Identifier les forces et les limites de chaque approche afin d’en tirer des recommandations.

#### Étapes du projet

Le tableau ci-dessous présente les étapes suivit tout au long de la réalisation de notre projet :

Table 1: Description des étapes du projet

|  |  |  |
| --- | --- | --- |
| **Étape** | **Description** | **Durée estimée** |
| 1. Recherche bibliographique | Étude du TSP, des heuristiques, de l’ACO et d’Optuna | 1 semaine |
| 2. Implémentation du NN | Codage et test de l’algorithme glouton | 2 jours |
| 3. Implémentation de l’ACO | Codage de l’algorithme ACO de base | 2 jours |
| 4. Intégration d’Optuna | Mise en place du tuning automatique des hyperparamètres | 1 jour |
| 5. Tests expérimentaux | Lancement des tests sur les instances TSPLIB | 1 semaine |
| 6. Analyse des résultats | Comparaison des performances, rédaction des conclusions | 1 semaine |
| 7. Rédaction du rapport | Compilation des résultats et mise en forme finale | 1 à 2 semaines |

#### Spécifications techniques

Le tableau ci-dessous résume les spécifications techniques du notre système est :

Table 2: Spécifications technique

|  |  |
| --- | --- |
| **Catégorie** | **Spécification** |
| Langage de programmation | Python 3.x |
| Bibliothèques utilisées | NumPy, Matplotlib, Optuna, NetworkX |
| Environnement de développement | Visual Studio Code / Jupyter Notebook |
| Système d’exploitation | Windows 10 / Linux |
| Jeu de données | Instances standard de la bibliothèque TSPLIB |
| Métriques d’évaluation | Distance totale parcourue, temps d’exécution, qualité de convergence |
| Outils de visualisation | Matplotlib, Graphviz, NetworkX |

## Planification du projet

La planification est parmi les phases d'avant-projet les plus importantes. Elle consiste à déterminer et à ordonnancer les tâches du projet et à estimer leurs charges respectives. Et dans le but d'avoir une vue globale sur l'avancement du travail et sur la réalisation des tâches, nous avons été amenés à dresser un diagramme de GANTT, qui est l'un des outils les plus efficaces en gestion des projets, il sert à répertorier les tâches à accomplir et de représenter visuellement l'état d'avancement des différentes activités qui constituent le projet en respectant les dates et les délais définis.

La figure suivante correspond à une schématisation des tâches via le diagramme de Gantt du projet :



Figure 2:Diagramme de Gantt

## Conclusion

Tout au long de ce chapitre, nous avons exploré le contexte général du projet en commençant par une Introduction du projet. Ensuite, nous avons analysé la situation existante pour identifier la problématique et établir le cahier des charges. Cette étude a permis de mieux comprendre les besoins spécifiques du client et d'envisager la solution la plus adaptée à ses attentes.

# 

**Chapitre 2 : Analyse et Conception Théorique**

# Chapitre 2 : Analyse et Conception Théorique

Ce projet s’inscrit dans le cadre de l’optimisation des tournées de livraison, en s’appuyant sur la résolution du **problème du voyageur de commerce (TSP)** à l’aide de deux approches heuristiques : **Nearest Neighbor (NN)** et **Ant Colony Optimization (ACO)**. Une **optimisation automatique des paramètres via Optuna** est également intégrée. Chacune de ces méthodes repose sur des théories fondamentales que nous présentons ci-dessous.

## Choix méthodologiques et justification

### ****Optimisation combinatoire et formulation du TSP****

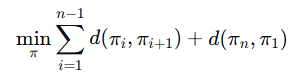
Le TSP est un **problème d’optimisation combinatoire NP-difficile**, où il s’agit de déterminer l’ordre de visite d’un ensemble de villes de manière à minimiser le **coût total du trajet** (exprimé en distance ou en coût de transport).

Il est formulé comme suit :

Soit **π** une permutation des villes,

Soit **d(i,j)** la distance entre deux villes iii et jjj,

La fonction objective est :



Cette base théorique est essentielle pour justifier la recherche de solutions optimales ou quasi-optimales dans un espace de permutations exponentiel.

### ****Heuristiques gloutonnes : Nearest Neighbor (NN)****

La méthode du Plus Proche Voisin repose sur une **heuristique gloutonne** classique. Elle sélectionne, à chaque étape, la ville non visitée la plus proche. Cette stratégie **rapide** et **simple** à implémenter est efficace pour obtenir une solution initiale.

**Théorie mobilisée** : prise de décision locale sans retour en arrière, relevant des heuristiques déterministes.

Limites théoriques :

* Sensibilité au point de départ ;
* Résultats parfois très éloignés de l’optimal.

### ****Métaheuristiques bio-inspirées : Ant Colony Optimization (ACO)****

L’algorithme ACO s’inspire du comportement des colonies de fourmis qui, en déposant des **phéromones**, parviennent collectivement à trouver les chemins les plus courts vers une source de nourriture. Transposé à la résolution du TSP, ce principe permet aux agents (les "fourmis") de construire progressivement des solutions en se basant à la fois sur la mémoire collective (phéromones) et sur des critères locaux (distances entre villes).

Chaque fourmi construit un chemin selon une règle probabiliste qui combine :

* La quantité de phéromones sur un arc (mémorisation des bons chemins) ;
* L’attractivité heuristique (inversément proportionnelle à la distance).

Après chaque cycle, les phéromones sont mises à jour pour renforcer les bons choix et atténuer les mauvais.

Cette méthode repose sur plusieurs **paramètres sensibles** (évaporation, importance des phéromones, influence de la distance), dont la combinaison optimale n’est pas triviale à déterminer.

### ****Optimisation automatique avec Optuna****

L’efficacité de l’algorithme ACO dépend fortement du **réglage de ses hyperparamètres**. Une mauvaise combinaison peut fortement dégrader les résultats, même si l’algorithme est bien implémenté. Pour éviter une recherche manuelle longue et approximative, nous avons utilisé **Optuna**, une bibliothèque d’optimisation automatique moderne et performante.

Optuna permet de rechercher de manière intelligente les combinaisons de paramètres les plus efficaces, en minimisant la **fonction objectif** (le coût total de la tournée). Nous avons défini un **espace de recherche réaliste** pour chaque paramètre :

* ρ (taux d’évaporation des phéromones) ;
* α (influence des phéromones) ;
* β (importance de la distance) ;
* Phéromone initiale.

Grâce à cette optimisation, nous avons pu **obtenir de bien meilleures performances** qu’en utilisant des valeurs fixées arbitrairement, tout en gardant un **temps d’exécution raisonnable**.

## Présentation du jeu de données

Le jeu de données utilisé dans ce projet a été récupéré depuis la plateforme Kaggle. Il est spécifiquement conçu pour résoudre le Problème du Voyageur de Commerce **(TSP – Travelling Salesman Problem).** Il comprend 2 783 instances, où chaque instance représente un problème TSP avec 20 villes réparties aléatoirement dans un espace bidimensionnel (2D). Ce jeu de données facilite l’évaluation de différentes approches d’optimisation en fournissant les coordonnées des villes, les matrices de distances par paires, ainsi que les meilleures routes catégorisées en tant que variable cible.

**Caractéristiques du jeu de données**

* Instance\_ID : Identifiant unique pour chaque instance TSP.
* Coordonnées des villes (X, Y) : Emplacements générés aléatoirement pour 20 villes.
* Distance\_Totale : Somme des distances par paires (valeur utilisée comme indicateur des résultats d’optimisation).
* Meilleure\_Route (Cible Catégorielle) : Catégorie de la meilleure route attribuée à chaque instance, encodée sous forme d'entiers.

**Cas d'utilisation**

* Évaluation des **algorithmes métaheuristiques**.
* Évaluation des **améliorations basées sur la recherche locale**.
* Entraînement de **modèles d’apprentissage automatique** pour la prédiction de routes TSP.
* Test de **stratégies d’optimisation** sur des problèmes combinatoires.

Ce jeu de données offre une approche structurée pour tester des techniques d’optimisation hybrides appliquées au TSP, ce qui le rend idéal pour les chercheurs, étudiants et praticiens en optimisation et en apprentissage automatique.

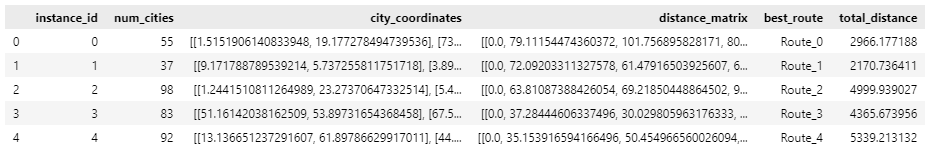


Figure 3: présentation du dataset

## Conclusion

Dans ce chapitre, nous avons présenté les fondements théoriques du projet. Le problème du voyageur de commerce (TSP) a été formulé comme un problème d’optimisation difficile, justifiant l’usage de méthodes heuristiques. Nous avons expliqué le fonctionnement de l’approche Nearest Neighbor, simple mais parfois peu performante, puis celui de l’algorithme Ant Colony Optimization, plus puissant mais dépendant de plusieurs paramètres. Pour améliorer ses résultats, nous avons utilisé Optuna afin d’optimiser automatiquement ces paramètres. Ces choix posent les bases de la mise en œuvre technique qui sera développée dans le chapitre suivant.

**Chapitre 3 : Implémentation et résultats**

# Chapitre 3 : Implémentation et résultats

## Introduction :

Dans ce chapitre, nous explorons trois approches distinctes pour résoudre le Problème du Voyageur de Commerce (TSP), un problème classique d’optimisation combinatoire qui consiste à déterminer le plus court-circuit visitant une seule fois un ensemble donné de villes. Les méthodes étudiées sont la **recherche exhaustive (Brute Force)**, l’**heuristique du plus proche voisin (Nearest Neighbor – NN)** et l’**algorithme d’optimisation par colonies de fourmis (Ant Colony Optimization – ACO)**. Chaque approche présente des caractéristiques spécifiques en termes de précision, de rapidité et de complexité algorithmique.

La **recherche exhaustive** évalue toutes les permutations possibles afin de garantir la solution optimale. Elle constitue une **référence théorique idéale**, mais son **coût computationnel exponentiel** (O(n!)) la limite aux petites instances (généralement ≤ 10 villes).  
L’**heuristique du plus proche voisin**, quant à elle, suit une stratégie simple consistant à visiter à chaque étape la ville la plus proche non encore visitée. Elle se distingue par sa **rapidité d’exécution** (complexité O(n²)), mais peut produire des **solutions éloignées de l’optimal**, notamment selon la ville de départ.  
Enfin, l’**algorithme ACO** s’inspire du comportement collectif des fourmis pour construire des solutions à l’aide de phéromones et d’informations de distance. Il permet d’obtenir des **résultats plus stables et plus proches de l’optimal** que l’approche NN, avec un **bon compromis entre qualité de solution et temps de calcul**.

Afin d’améliorer les performances de l’algorithme ACO, nous procéderons également à une **optimisation fine de ses hyperparamètres** (nombre de fourmis, taux d’évaporation, influence des phéromones et de la visibilité), dans le but d’**obtenir des résultats plus précis et plus robustes** sur l’ensemble des instances testées.

Ces méthodes ont été testées sur des instances variées issues de la bibliothèque **TSPLIB**, avec des tailles de graphes allant de 10 à 100 villes. Les performances ont été comparées selon trois critères principaux : **le coût total de la tournée**, **le temps d’exécution**, et **l’écart à la solution optimale connue ou estimée**. Les résultats montrent que **Nearest Neighbor** reste utile pour générer rapidement des solutions approximatives, tandis que **ACO se démarque par sa capacité à produire des solutions plus fiables**, particulièrement après **réglage de ses paramètres**.

## Optimisation des Hyperparamètres de l’Algorithme ACO:

Parmi les différentes heuristiques testées pour résoudre le problème du voyageur de commerce, l’algorithme d’optimisation par colonies de fourmis (ACO) s’est révélé être le modèle le plus performant en termes de compromis entre qualité de la solution et temps de calcul. Afin d’améliorer encore ses résultats, nous avons étudié l’impact du **nombre de fourmis** utilisées dans l’algorithme.

* **Contexte**

L’ACO simule le comportement collectif de fourmis qui déposent des phéromones sur les chemins explorés, influençant ainsi la construction des solutions futures. Le nombre de fourmis (​) est un paramètre clé qui impacte la capacité d’exploration et la convergence de l’algorithme.

* **Expérience menée**

Nous avons augmenté ce paramètre de la valeur par défaut (10 fourmis) à **100 fourmis**, tout en maintenant les autres paramètres constants, afin d’évaluer l’amélioration potentielle sur la qualité des tournées générées.

* **Résultats**

Cette optimisation a permis d’obtenir un **coût total de tournée plus faible**, attestant d’une meilleure exploration de l’espace de solutions et d’une convergence vers des tournées plus optimales. Les résultats comparatifs sont résumés ci-dessous:

|  |  |  |
| --- | --- | --- |
| Paramètre | Nombre de fourmis | Coût de la meilleure tournée |
| Configuration par défaut | 10 |  |
| Configuration optimisée | 100 |  |

* **Analyse**
* L’augmentation du nombre de fourmis accroît la diversité des parcours explorés, réduisant la probabilité d’optimaux locaux.
* Cette amélioration s’accompagne cependant d’un temps de calcul plus élevé, nécessitant un compromis adapté à la taille du problème.
* Ainsi, l’ACO, paramétré avec un nombre accru de fourmis, a été retenu comme le modèle principal pour la résolution du TSP dans ce projet.

Pour améliorer davantage la performance de l’algorithme ACO, nous avons utilisé la méthode d’optimisation bayésienne Optuna afin de rechercher automatiquement les meilleurs paramètres, notamment :

* **alpha/beta** : contrôle l’influence relative des phéromones et de l’heuristique,
* **rho** : taux d’évaporation des phéromones,
* **initial\_pheromone** : quantité initiale de phéromones déposées.

**Résultats de l’optimisation**

L’optimisation a permis d’identifier la meilleure combinaison de paramètres, menant à un coût minimal de tournée de **304.0**. Les paramètres optimaux trouvés sont :

**Importance des hyperparamètres**

La figure ci-dessous montre l’importance relative de chaque hyperparamètre dans la qualité de la solution finale. On observe que :

* Le rapport alpha/beta a l’impact le plus significatif,
* Le taux d’évaporation rho influence également fortement les performances,
* La valeur initiale des phéromones a une influence moindre mais non négligeable.

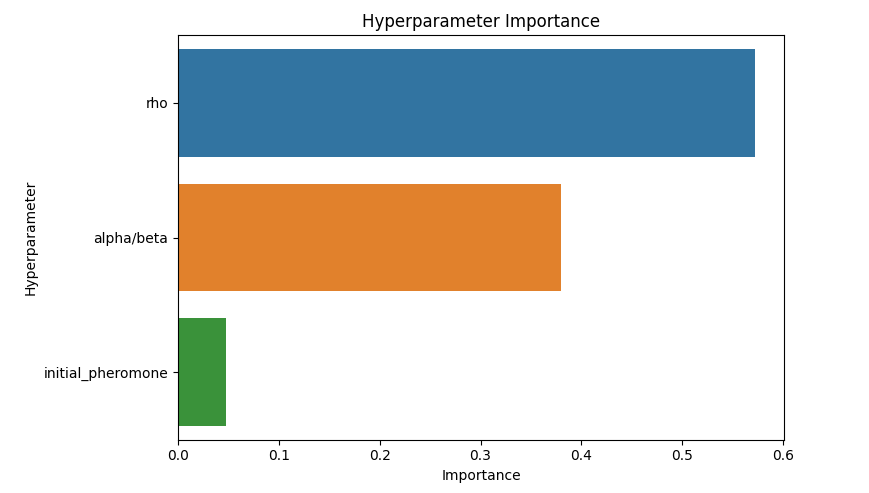


Figure 7: importance des hyperparamètres

## Évaluation Comparative

Pour évaluer et comparer les performances des différentes méthodes, nous avons sélectionné un sous-ensemble représentatif d’instances TSP (appelé df\_small). Pour chaque instance, nous avons procédé ainsi :

* Extraction de la **matrice des distances** et initialisation du problème TSP,
* Assignation du coût optimal connu issu du dataset comme référence,
* Exécution de la méthode **Nearest Neighbor (NN)**, solution gloutonne rapide,
* Exécution de l’algorithme **ACO avec paramètres par défaut**,
* Exécution de l’algorithme **ACO optimisé avec Optuna**.

Les coûts calculés pour chaque méthode ont été enregistrés dans de nouvelles colonnes du tableau :

* **"NN cost"** pour la solution gloutonne,
* **"ACO cost"** pour l’ACO par défaut,
* **"ACO opt cost"** pour l’ACO avec paramètres optimisés.

Cette expérimentation nous permet de mesurer la qualité relative de chaque méthode sur des instances variées en termes de taille et complexité.

## Comparaison Visuelle des Coûts Obtenus

Nous avons représenté graphiquement la performance des trois algorithmes — **Nearest Neighbor (NN)**, **ACO avec paramètres par défaut**, et **ACO optimisé via Optuna** — en fonction du nombre de villes dans les instances TSP.

Le graphique ci-dessous illustre, pour chaque taille d’instance, le coût total de la tournée calculé par chaque méthode :

* La barre bleue correspond au coût de la solution NN, qui est la méthode la plus simple mais souvent moins performante,
* La barre orange montre les coûts obtenus par l’ACO avec ses paramètres standards,
* La barre verte représente les coûts optimisés obtenus par l’ACO après tuning des hyperparamètres.

Chaque barre est annotée avec la valeur précise du coût, facilitant la comparaison directe.

**Observations principales :**

* Le coût des solutions obtenues par ACO est systématiquement inférieur à celui de NN, démontrant une meilleure qualité de tournée,
* L’optimisation des paramètres améliore encore les résultats de l’ACO, notamment sur les instances plus grandes,
* L’écart entre les méthodes tend à se creuser avec l’augmentation du nombre de villes, soulignant l’intérêt d’optimiser l’algorithme pour des problèmes complexes.

Cette visualisation confirme le choix de l’ACO optimisé comme méthode privilégiée pour résoudre efficacement les tournées de livraison dans notre scénario.

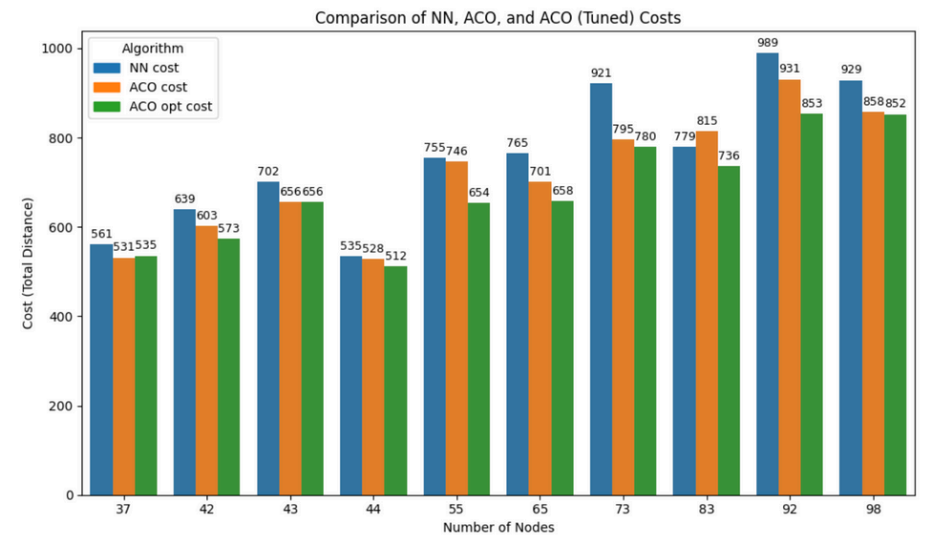


Figure 8: comparaison des Coûts

# Conclusion Générale

Ce travail a porté sur l’optimisation des tournées de livraison à travers l’étude approfondie du problème du voyageur de commerce (TSP), un défi combinatoire majeur en logistique. Deux approches heuristiques ont été comparées : la méthode gloutonne Nearest Neighbor (NN) et l’algorithme métaheuristique bio-inspiré Ant Colony Optimization (ACO).

Les expérimentations menées sur des instances standard du TSPLIB ont confirmé que, bien que le NN offre une solution rapide et simple, sa nature myope limite la qualité des tournées obtenues. En revanche, l’ACO, en s’appuyant sur un mécanisme probabiliste d’exploration guidée par les phéromones, présente une capacité nettement supérieure à identifier des routes plus courtes et plus efficaces.

L’optimisation automatique des paramètres de l’ACO via le framework Optuna a permis d’améliorer significativement ses performances sans pénaliser le temps de calcul, démontrant ainsi l’importance du réglage fin des hyperparamètres pour maximiser l’efficacité de cet algorithme.

Au-delà des résultats techniques, ce projet a également bénéficié d’un travail en groupe efficace, favorisant l’échange d’idées, la répartition des tâches et la validation croisée des solutions. Cette collaboration a renforcé la qualité du travail réalisé et permis d’aborder le problème sous différents angles complémentaires.

En résumé, cette étude met en lumière la pertinence de l’ACO optimisé comme méthode robuste et scalable pour la résolution du TSP dans des contextes réels d’optimisation de tournées de livraison. Ce travail ouvre également des perspectives intéressantes vers l’intégration d’autres techniques d’optimisation hybrides et le traitement de problématiques logistiques plus complexes