
Livrable 1 - P2I2

PPII2 - Rapport

DIONISIO, FREY, BILLARD, MEZUREUX

1^{ère} année



23 mai 2023

Table des matières

1	Introduction	3
I	Contexte et objectifs du projet	4
II	Objectifs du rapport	4
III	Présentation du plan	4
2	Gestion de projet	5
I	Définition du projet	6
II	Approche et méthodologie de gestion de projet	10
III	Suivi et évaluation du projet	12
3	Première étape	13
I	Analyse	14
II	Conception	16
III	Développement	18
IV	Tests et performances	20
4	Seconde étape	22
I	Analyse	23
II	Conception et développement	23
III	Tests et performances	23
5	Conclusion	24
I	Complétion des objectifs	25
II	Avantages et limites du projet	25
III	Améliorations possibles	25
IV	Clôture	25
6	Annexes	26

◁ Chapitre 1 ▷

Introduction

I	Contexte et objectifs du projet	4
II	Objectifs du rapport	4
III	Présentation du plan	4

I ▷ Contexte et objectifs du projet

Ce projet s'inscrit dans le cadre pédagogique de notre première année d'étude en école d'ingénieur au sein de l'établissement TELECOM Nancy. L'objectif y est de mettre en pratique les compétences scientifiques et techniques acquises tout au long de cette première année en se rapprochant d'une étude de cas concrète.

Il se divise en deux étapes distinctes. Il nous a été demandé dans un premier temps de construire en langage «C» un programme qui permet de planifier un parcours de charge pour un véhicule électrique et pour un trajet donné et d'enrichir cette application en y incorporant les ajouts de notre choix. Dans un second temps, il nous a fallu réaliser, toujours en langage «C», un module de simulation qui pour un ensemble d'utilisateurs, calcule le taux de charge des bornes du territoire.

Notons également que tout au long du projet nous avons dû suivre une gestion de projets rigoureuse, ce qui nous a permis d'assurer une planification efficace et un suivi approprié des activités. Nous étions obligé d'adopter une approche méthodologique solide, afin d'assurer que notre projet soit réalisé de manière efficace, dans les délais impartis.

Toute l'équipe a pris plaisir à travailler sur le projet et nous tenons à remercier tous ceux qui nous ont aidés.

II ▷ Objectifs du rapport

Ce rapport synthétise le travail réalisé par l'équipe de PoinCarWash pour répondre à la problématique posée. À savoir travailler à l'élaboration d'outils utiles au développement d'une mobilité plus écologique.

Le présent rapport fait état de la conception et de l'implémentation de notre travail mais également des performances et des tests sur ce dernier, ainsi que d'une présentation de notre gestion de projet.

III ▷ Présentation du plan

Au cours de ce rapport, nous aborderons dans un premier temps les éléments de gestion de projet communs aux deux étapes. Cette première partie nous permettra ainsi de justifier nos choix d'approches méthodologiques pour la réalisation de ce projet.

Nous procéderons ensuite de manière similaire pour les deux étapes du projet. En premier lieu, nous décrirons les éléments de gestion de projet spécifiques à chacune d'entre elles. Nous détaillerons ensuite les phases de conception et de développement avant d'enfin présenter les performances et les tests de nos applications.

Pour conclure nous récapitulerons les résultats obtenus dans chaque partie du projet en les mettant en parallèle avec les objectifs fixés au départ, nous analyserons les avantages et les limites de notre travail, et formulerons des recommandations pour de futures améliorations et développements avant de conclure définitivement le projet.

◁ Chapitre 2 ▷

Gestion de projet

I	Définition du projet	6
I.A	Contexte et justification du projet	6
I.B	Portée du projet	6
I.C	Parties prenantes	7
I.D	Contraintes et risques	8
II	Approche et méthodologie de gestion de projet	10
II.A	Méthode de gestion de projet	10
II.B	Forces et faiblesses	10
II.C	Communication et collaboration	11
III	Suivi et évaluation du projet	12
III.A	Méthodes de suivi	12
III.B	Évaluation des résultats	12
III.C	Gestion des modifications	12
III.D	Évaluation de la performance	12

I ▷ Définition du projet

I.A Contexte et justification du projet

Avec la prise de conscience écologique grandissante dans la population, les citoyens et les Etats se tournent progressivement vers des modes de consommation et des politiques plus responsables et écologiques. La Commission européenne a dans cette optique récemment pris la décision d'interdire la vente de véhicules thermiques à l'horizon 2035. Le développement et l'utilisation des véhicules électriques va donc connaître une croissance élevée dans les années à venir. En effet, selon les premières projections¹, le marché des véhicules électrifiés pourrait représenter 24% des parts de marché d'ici 2025 et 90% d'ici 2040.

Pourtant, voyager dans ce type de véhicules peut encore parfois s'avérer difficile, notamment en ce qui concerne leur autonomie toujours relativement limitée et la répartition des bornes de recharge sur le territoire, cette dernière étant encore très inégale sur le sol français. Il est donc nécessaire pour faciliter leur adoption d'en faciliter et d'en optimiser l'utilisation.

La première étape de notre projet consiste alors en la conception d'un algorithme permettant de calculer pour un utilisateur le trajet le plus court entre deux endroits situés en France, en optimisant ses passages aux bornes de recharges. Cet outil pourra aider les conducteurs à naviguer plus efficacement et à ainsi surmonter les inquiétudes liées à l'autonomie limitée des voitures électriques.

La seconde étape du projet est basée sur l'étude du remplissage des stations en fonction d'un ensemble de trajets donnés. L'acquisition de ces statistiques pourrait par exemple permettre de planifier plus efficacement l'expansion future du réseau de recharge.

Ainsi nous espérons au travers de ce projet promouvoir une mobilité plus durable en participant à l'élaboration de solutions pratiques pour surmonter les obstacles liés à l'autonomie et à la disponibilité de stations de recharge pour véhicules électriques.

I.B Portée du projet

Objectifs du projet :

- ▷ Conception d'un algorithme de calcul d'itinéraire optimisé pour véhicule électrique
- ▷ Conception d'une application d'analyse du remplissage des stations de recharge

Ressources :

- ▷ Groupe de projet (4 personnes) : temps de travail variable pendant 9 semaines
- ▷ Equipe pédagogique de TELECOM Nancy : aide ponctuelle
- ▷ Bases de données relatives aux stations de recharge et aux véhicules électriques

1. Selon l'étude [BloombergNEF](#)

Livrables :

- ▷ Code source des deux applications, 24 Mai
- ▷ Guide utilisateur décrivant l'installation du programme et les modalités de son utilisation, 24 Mai
- ▷ Rapport du projet incluant les éléments relatifs à la gestion de projet et décrivant les choix motivés des structures de données et des algorithmes (sur la base d'un état de l'art détaillé), les fonctions réalisées et leur analyse (complexité mémoire, temps), les fonctions réalisées et les tests mis en œuvre, 24 Mai

Éléments non prévus dans le projet :

- ▷ Adaptation du calcul des itinéraires en fonction du taux de remplissage des stations de recharges.

I.C Parties prenantes

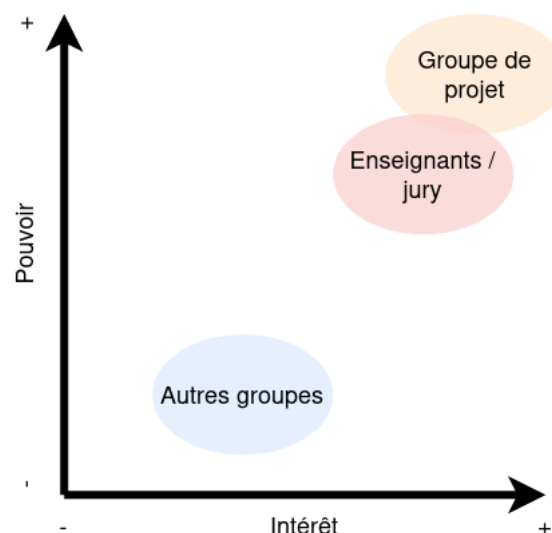
Identification :

- ▷ Interne :
 - Groupe de projet : Les participants directs au projet, responsables de la conception et du développement des outils demandés.
- ▷ Externe :
 - Enseignants et jury : Ils guident et évaluent les étudiants dans le cadre du projet.
 - Autres groupes : D'autres groupes de projet travaillant sur des sujets similaires, avec lesquels des échanges et des discussions peuvent être initiés pour partager des idées et des connaissances.

Évaluation de l'influence et de l'importance :

- ▷ Groupe de projet : Influence élevée et importance élevée, car ils sont directement responsables de la réalisation du projet et de l'atteinte des objectifs fixés.
- ▷ Enseignants/jury : Influence élevée et importance élevée, car ils évalueront le projet et fourniront des conseils et des orientations pour assurer la qualité et la réussite du projet.
- ▷ Autres groupes de projet : Influence faible à moyenne et importance faible à moyenne, car ils peuvent partager des connaissances et des expériences similaires, mais leur impact direct sur le projet est limité.

On peut synthétiser cette analyse au travers de ce diagramme :



Plan d'action :

- ▷ Groupe de projet :
 - Communiquer de façon régulière et transparente au sein du groupe.
 - Organiser des réunions pour discuter des progrès réalisés, des difficultés rencontrées et des décisions à prendre.
- ▷ Enseignants/jury :
 - Communiquer de façon ponctuelle pour s'assurer que nous travaillons toujours dans la bonne direction.
 - Solliciter leur aide en cas de trop grosse difficulté.
- ▷ Autres groupes de projet :
 - Echanger de façon ponctuelle pour recueillir des idées, des connaissances ou des expériences supplémentaires.

I.D Contraintes et risques

Contraintes :

- ▷ Temporelle :
 - Le projet doit être réalisé en 9 semaines.
- ▷ Techniques :
 - Langage de programmation : le langage utilisé doit être le langage «C».

Risques :

- ▷ Techniques :
 - Incompatibilités matérielles : nous ne travaillons pas tous sous le même OS.
 - Problèmes de performance : les outils demandés peuvent demander des traitements lourds, nos machines pourraient être dépassées.
- ▷ Gestion de projet :
 - Gestion du temps et planification : nous ne travaillons pas à temps plein sur le projet et nos autres activités peuvent nous empêcher de consacrer suffisamment de temps au projet.
- ▷ Liés aux données :
 - Les données utilisées pour le projet peuvent être incomplètes, inexactes ou indisponibles.

Evaluation :

- ▷ Contrainte temporelle :
 - Impact : le temps disponible pour la conception, le développement et les tests est limité.
 - Gravité : élevée, un manque de temps peut affecter la qualité globale du projet.
- ▷ Contrainte technique :
 - Impact : le langage «C» est maîtrisé par l'équipe, la contrainte n'a donc que peu d'impact.
 - Gravité : Négligeable, cette contraintes peut être gérée en planifiant un temps de formation mais cela demande du temps.

▷ Risques techniques :

- Impact : les incompatibilités matérielles peuvent entraîner des problèmes d'intégration et de compatibilité lors de la mise en œuvre. Les problèmes de performance peuvent affecter la réactivité de l'application et la satisfaction de l'utilisateur.
- Gravité : Modérée, ces risques peuvent être atténués en identifiant les incompatibilités dès le début du projet et en optimisant le code pour améliorer les performances.

▷ Risque de gestion de projet :

- Impact : Une mauvaise gestion du temps et de la planification peut entraîner des retards dans l'achèvement des tâches et la réalisation à terme des objectifs du projet.
- Gravité : Modérée, une bonne communication et une planification efficace peuvent minimiser les impacts de ce risque.

▷ Risque lié aux données :

- Impact : Des difficultés pour obtenir les données peuvent allonger le temps nécessaire à l'élaboration des applications, de plus des données de mauvaise qualité peuvent affecter la précision et la validité des résultats obtenus.
- Gravité : Modérée, des efforts de collecte et de validation des données seront nécessaires pour minimiser l'impact de ce risque.

On peut synthétiser les résultats de cette analyse dans une matrice des risques :

Matrice des risques						
		Gravité				
		Négligeable	Mineure	Modérée	Majeure	Catastrophique
Probabilité	Très probable					
	Probable			- Incompatibilités entre les OS (risque technique) - Difficultés d'acquisition des données (risque lié au données)	Manque de temps / mauvaise gestion du temps (contrainte temporelle et risque de gestion de projet)	
	Possible			Capacité de calcul trop limitée (risque technique)		
	Peu probable			Données de mauvaise qualité (risque lié au données)		
	Très peu probable			Non maîtrise du langage "C" (contrainte technique)		

II ▷ Approche et méthodologie de gestion de projet

II.A Méthode de gestion de projet

Pour ce projet nous avons décidé de nous orienter vers une gestion de projet classique pour les raisons suivantes :

- ▷ **La précision des spécifications** : le sujet étant clair à ce niveau, nous savions dès le départ tout ce qui était attendu de nous. Ainsi, comme les livrables ne risquaient pas de changer en cours de route et que la participation des parties prenantes était limitée, une gestion agile n'était pas de mise.
- ▷ **La structuration claire et la gestion du temps** : La gestion de projet classique offre une structure claire et bien définie avec des étapes séquentielles. Cela permet de diviser le projet en phases distinctes, ce qui facilite la planification, l'organisation et le suivi des progrès. La gestion de la contrainte temporelle en est également plus aisé.

Nous avons décidé de découper le projet ainsi :

- Semaine 1 : définition, planification et répartition du projet.
 - Semaine 2 à 5 : conception, développement, tests et vérifications de la première étape.
 - Semaine 5 à 8 : conception, développement, tests et vérifications de la seconde étape.
 - Semaine 9 : Evaluation, rédaction du rapport et clôture du projet
- ▷ **La documentation** : La méthode classique encourage la documentation régulière des résultats intermédiaires du projet, cela renforce le suivi des activités, la détection des erreurs et simplifie l'étape de rédaction du rapport.

II.B Forces et faiblesses

Pour travailler efficacement il est important de connaître les forces sur lesquelles on peut s'appuyer, ainsi que les faiblesses auxquelles il faut faire attention.

Nos forces :

- ▷ On a de l'expérience ensemble, on a déjà eu l'opportunité de travailler de concert. On connaît donc les méthodes de travail de chacun et on peut alors mieux adapter la planification, la répartition et l'organisation du travail en fonction de chaque individu. On peut également noter que l'équipe fait preuve d'une forte cohésion.
- ▷ Chaque membre du groupe est doté de bonnes capacités de communication. Cela sera un avantage lors de nos réunions pour se comprendre rapidement mais aussi lors de la soutenance pour communiquer avec le jury.
- ▷ Notre expérience dans le domaine associatif nous a solidement formé dans les domaines relatifs à l'organisation. Ces points forts seront utiles dans tout ce qui est relatif à la gestion de projet.

Nos faiblesses :

- ▷ Nos connaissances en langage «C» sont encore très récentes, ainsi on aura de manière certaine besoin de plus temps pour développer ce type d'outils que des développeurs expérimentés.
- ▷ On est tous très engagés dans le milieu associatif de TELECOM et les cours nous prennent beaucoup de temps, on a donc peu de temps à consacrer au projet. Il faudra être discipliné sur le temps que l'on arrive à dégager.

On peut synthétiser cette analyse sous forme de tableau :

Forces	Faiblesses
Forte expérience en équipe	Novices en langage «C»
Bonne capacité de communication	Très pris par l'associatif et les cours
Expérience solide en organisation et en gestion de projet	

II.C Communication et collaboration

- ▷ **La communication interne** : Pour notre communication interne, nous avons décidé de continuer à utiliser Discord.
- ▷ **Les réunions** : En ce qui concerne les réunions nous avons décider dans la mesure de possible de faire une réunion d'avancement par semaine et des réunions techniques ponctuelles pour échanger sur les détails techniques. Toutes les réunions feront l'objet de compte rendu de réunion.
- ▷ **La répartition du travail** : Pour gérer nos ToDo, nous avons décidé d'utiliser le logiciel Trello.
- ▷ **Partage des ressources** : En ce qui concerne le partage des ressources et de nos avancées, nous utiliserons le serveur GitLab de l'école.

III ▷ Suivi et évaluation du projet

PPII2

III.A Méthodes de suivi

III.B Évaluation des résultats

III.C Gestion des modifications

III.D Évaluation de la performance

◁ Chapitre 3 ▷

Première étape

I	Analyse	14
I.A	Etat de l'art	14
I.A.1	Le problème	14
I.A.2	Algorithme	14
I.A.3	Applications existantes	14
I.B	Charte de projet	15
I.C	Planification	15
II	Conception	16
II.A	Manipulation des données	16
II.B	Théorie des graphes	17
II.C	Paramètres d'entrée et de sortie	17
III	Développement	18
III.A	Structures de données	18
III.B	Algorithmes	18
III.B.1	Calcul de plus court chemin : A-star	18
III.B.2	Conversion d'adresse en coordonnées géographiques	18
III.C	Visualisation	19
IV	Tests et performances	20
IV.A	Tests unitaires	20
IV.B	Tests de performance	20
IV.C	Interprétation des résultats	21

I ▷ Analyse

La première consiste à construire en langage «C» un programme qui permet de proposer à tout usager souhaitant se rendre d'un point A à un point B du territoire, un parcours de charge de son véhicule électrique.

I.A Etat de l'art

I.A.1 Le problème

Le problème qui nous a été donné est un problème de recherche de plus court chemin. Il est alors fortement probable que nous soyons amenés à utiliser une structure de type graphe. Le problème du plus court chemin entre un sommet A et un sommet B revient à chercher le chemin de A à B où la somme des poids des arcs qu'il traverse est minimale¹. Il faut donc choisir un algorithme qui trouve des solutions à ce problème.

I.A.2 Algorithme

Les potentiels algorithmes pour ce problème sont :

- ▷ Dijkstra
- ▷ A*²
- ▷ Bellman-Ford
- ▷ Viterbi
- ▷ Floyd-Warshall³
- ▷ Johnson³

I.A.3 Applications existantes

Ce problème ayant déjà été étudié, il existe des applications permettant de répondre aux besoins de notre sujet, notamment :

- ▷ **Chargermap** : Son fonctionnement est très simple. Il faut d'abord lancer l'application sur son smartphone Android ou Apple, puis renseigner ses points de départ et d'arrivée. Vous devez ensuite préciser le modèle de votre véhicule, les niveaux de batterie souhaités au départ comme à l'arrivée et enfin, la vitesse maximale que vous prévoyez au cours du trajet. À partir de ces informations, l'algorithme calcule en une poignée de secondes le chemin le plus adapté. L'itinéraire à suivre et les bornes où s'arrêter sont ainsi affichés sur une carte. Une feuille de route vous indique également la distance à parcourir entre chaque escale et la durée de chaque recharge. Vous verrez qu'il n'est pas systématiquement nécessaire de faire un plein complet, un appoint de quelques dizaines de minutes suffisant la plupart du temps.
- ▷ **A Better Routeplanner** : Le fonctionnement de cet outil est globalement similaire celui de Chargermap. Lieu de départ et d'arrivée, niveaux de batterie, modèle de véhicule : on y renseigne les mêmes informations. A Better Routeplanner (ABRP) se démarque cependant en proposant un mode expert, qui ouvre accès à un panneau extrêmement complet de données à préciser : poids des bagages et passagers, météo, consommation moyenne personnalisée, type de chargeurs, etc. Contrairement à Chargermap, ABRP présente aussi l'avantage d'estimer les prix des recharges

1. Si on ajoute des contraintes à ce problème comme des fenêtres de temps, le problème peut devenir NP-difficile.
2. Cet algorithme semble le plus efficace d'après les recherches.
3. Si l'on veut tous les chemins à partir d'un point

sur sa feuille de route. Les différences de temps de charge et de parcours entre Chargemap et A Better Routeplanner sont souvent le résultat d'estimations différentes de ses concepteurs. Le premier outil peut, par exemple, être plus ou moins optimiste sur la capacité d'un véhicule à recharger rapidement que le second.

PPII2

I.B Charte de projet

I.C Planification

II ▷ Conception

PPII2

II.A Manipulation des données

L'énoncé fournit deux ressources. La première est une base de données du gouvernement répertoriant les stations de recharges pour véhicules électriques du territoire français ainsi que diverses informations sur ces stations (nombre de bornes, puissance, ...). La seconde est une base de données tirée du site <https://ev-database.org/> qui contient une kyrielle de données sur les véhicules électriques du marché (autonomie, vitesse de recharge, ...).

La première étape de cette partie consiste à extraire les données mises à notre disposition de manière à pouvoir être exploitées facilement en C par la suite. Pour les stations de recharge, le site du gouvernement ([serveur Etalab](#)) propose nativement un export au format CSV qui est un format qui nous être le plus simple pour arriver à nos fins. Cependant, le site qui fournit les données sur les véhicules, lui, ne propose aucune option d'export. Nous avons donc commencé par écrire un script de *parsing* en Python qui va récupérer les données sur le site et les enregistrer dans un fichier CSV. Nous n'avons récupéré que les données qui nous semblaient utiles parmi toutes celles proposées sur le site internet. C'est-à-dire le nom du véhicule, son autonomie ainsi que sa vitesse de recharge

CODE 3.1 – Script de récupération des données sur les véhicules électriques

```
1 import requests
2 from bs4 import BeautifulSoup
3 import json
4
5 url = """https://ev-database.org/#sort:path-type~order=.erange_real~number~desc|
6     range-slider-range:prev~next=0~1200|range-slider-acceleration:prev~next=2~23|
7     range-slider-topspeed:prev~next=110~350|range-slider-battery:prev~next=10~200|
8     range-slider-towweight:prev~next=0~2500|range-slider-fastcharge:prev~next=0~1500|
9     paging:currentPage=0|paging:number=all"""
10
11 response = requests.get(url)
12 assert (response.ok)
13
14 soup = BeautifulSoup(response.text, 'html.parser')
15
16 vehicle_info = []
17 vehicles = soup.find_all('div', class_='list-item')[1:] # skip header row
18 for vehicle in vehicles:
19     vehicle_tmp = {}
20     vehicle_tmp['name'] = vehicle.find('h2').text
21     specs = vehicle.find('div', class_='specs').find_all('p')
22     vehicle_tmp['range'] = specs[2].find_all('span')[1].text[:-3]
23     vehicle_tmp['fast_charge'] = specs[4].find_all('span')[2].text
24
25     vehicle_info.append(vehicle_tmp)
26
27 with open('vehicle_info.json', 'w') as f:
28     json.dump(vehicle_info, f, indent=4)
```


II.B Théorie des graphes

PPII2

Le problème posé dans cette première partie nous a tout de suite fait penser à la théorie des graphes et plus particulièrement à un calcul de plus court chemin. L'algorithme de DIJKSTRA³ ayant été étudié en MSED, nous avons été sceptiques dès le début quant à son utilisation. En effet, cet algorithme nécessite de parcourir au moins une fois chaque borne de recharge pour calculer le chemin le plus court, ce qui ne semble pas très efficace compte tenu des ~ 50000 stations. Nous nous sommes donc tournés vers un autre algorithme, A^* ⁴ qui considère l'éloignement au point d'arrivée et donc ne parcourt pas toutes les stations.

Une fois l'algorithme choisi, une zone d'ombre subsiste. Comment prendre en compte l'autonomie de la voiture dans notre calcul de plus court chemin ? La solution est venue assez naturellement, il suffit à chaque itération de ne pas considérer les voisins dont la distance est supérieure à l'autonomie du véhicule considéré.

II.C Paramètres d'entrée et de sortie

En plus des deux fichiers CSV contenant les données sur les stations de recharge et les véhicules électriques contenus dans le répertoire `/data/raw/`, l'exécutable prendra en paramètres directement sur la ligne de commandes les arguments suivants :

Entrée :

- ▷ Adresse de départ ou coordonnées géographiques
- ▷ Véhicule utilisé
- ▷ *(optionnel)* Pourcentage de batterie minimum à ne pas dépasser
- ▷ *(optionnel)* Temps de recharge maximum

Sortie :

- ▷ Liste des stations à visiter
- ▷ Pourcentage de batterie et distance restante à chaque étape
- ▷ Lien de visualisation sur Google Maps

3. https://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra

4. https://fr.wikipedia.org/wiki/Algorithme_A*

III ► Développement

III.A Structures de données

La première structure de données à choisir et sans aucun doute la plus importante est celle qui va contenir les informations sur les stations importées depuis le fichier CSV. Étant données que ces informations vont être consultées plusieurs fois à chaque étape du calcul de plus court chemin, il est important de choisir une structure de données qui profère un accès rapide. Ainsi, nous nous sommes tournés vers une table de hachage appelée `Table_t` qui sera chargée une seule fois à partir du fichier CSV puis qui permettra de consulter les informations avec une complexité temporelle en $\mathcal{O}(1)$.

En ce qui concerne les informations sur les véhicules électriques, nul besoin d'une table de hachage, lors de l'appel de la fonction, le fichier CSV sera parcouru une seule fois de manière linéaire et les informations associées au véhicule considérées seront placées dans une structure de données appelée `Vehicle_t`.

Le résultat de l'algorithme *i.e.* la liste des stations à visiter, sera stocké dans une liste contiguë (déjà implantée pour la table de hachage) appelée `List_t`.

III.B Algorithmes

III.B.1 Calcul de plus court chemin : A-star

III.B.2 Conversion d'adresse en coordonnées géographiques

Afin de faciliter l'utilisation de l'application, et d'être plus productifs lors des tests de nos programmes, nous avons implémenté l'utilisation d'une API de nominatims pour rechercher les coordonnées géographiques d'un endroit à partir d'une description variable de son adresse. Par exemple : «TELECOM Nancy», «193 avenue paul muller», «LORRAINE», «Place Stanislas», etc.

Pour ce faire, nous avons utilisé la librairie `libcurl`⁵ du projet *open source* cURL. Cette librairie permet de faire des requêtes HTTP et HTTPS, et donc de communiquer avec des API. Il nous a donc fallu créer un certain nombre de structures de données, car la requête à l'API génère un *buffer* en temps direct, et gérer tous les éventuels cas d'erreurs : réponse nulle, connexion impossible, etc.

L'API utilisée est celle d'**OpenStreetMap**⁶. Elle permet de faire des recherches d'adresses, de lieux, de villes, etc. à partir d'une description textuelle. Elle renvoie un fichier JSON contenant les informations demandées, dont les coordonnées géographiques.

Notre programme exécute donc une requête HTTPS à cette API et récupère la réponse sous forme de JSON, qu'une autre fonction parse pour en extraire les informations utiles.

`libcurl`⁷

5. <https://curl.se/libcurl/>

6. <https://nominatim.org/release-docs/develop/api/Search/>

7.

III.C Visualisation

PPII2

L'utilisation de bibliothèques graphiques telles que GTK⁸ ou d'un framework web tel que Flask⁹ aurait été une solution envisageable pour la visualisation des résultats. Cependant, nous avons choisi d'utiliser Google Maps¹⁰ pour plusieurs raisons :

- ▷ Beaucoup plus simple à mettre en place car il suffit de générer un lien avec les coordonnées géographiques des stations à visiter et les points de départ et d'arriver pour obtenir une carte interactive
- ▷ Le lien peut être partagé facilement
- ▷ Le temps économisé sur cette partie nous a permis d'ajouter des fonctionnalités supplémentaires

8. <https://www.gtk.org/>

9. <https://flask.palletsprojects.com/en/1.1.x/>

10. <https://www.google.com/maps>

IV ▷ Tests et performances

PPII2

IV.A Tests unitaires

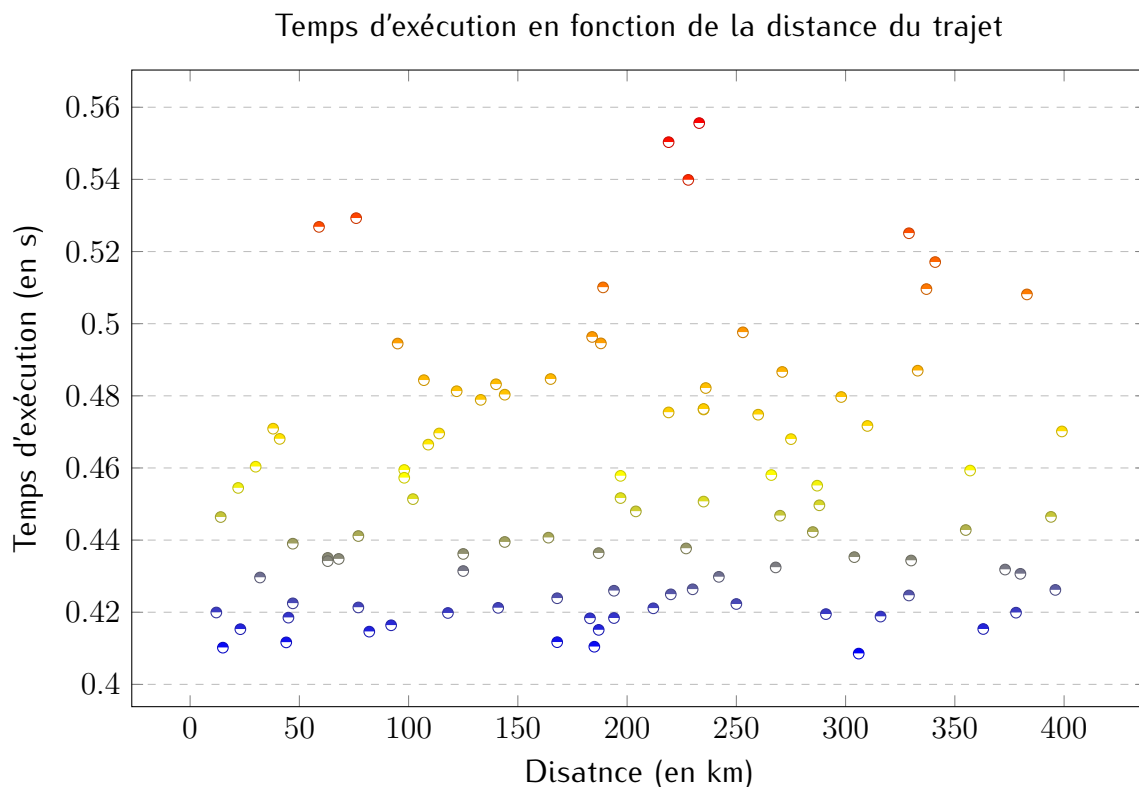
Dès le début du projet, nous avons choisi d'utiliser la bibliothèque `Snow`¹¹ pour les tests unitaires. En effet, cette bibliothèque est très simple d'utilisation et permet de tester les fonctions de manière isolée. De plus, l'utilisation de `Snow` a été couplée à celle du serveur d'intégration continue de `GitLab`¹². Cela nous a permis de vérifier à chaque *commit* que le code poussé sur le serveur n'entravait pas les fonctions déjà poussées mais également de ne pas faire fusionner malencontreusement une branche dont les tests unitaires ne passaient pas.

IV.B Tests de performance

L'implantation d'un module de teste de performances nous semblé être important compte tenu de la taille des données manipulées et du fait de leur impact direct sur les performances de la partie suivante. Ainsi, le répertoire `/benchmark/` contient un programme de test pouvant être exécuté avec les arguments suivants :

- ▷ `-f` pour spécifier le nom du fichier CSV contenant les résultats des tests (fichier placé dans le répertoire `/data/benchmark/`)
- ▷ `-d` pour utiliser les paramètres par défaut, à savoir
 - 100 trajets
 - trajets aléatoires
 - distances entre 1 et 400km (le point de référence étant le milieu de la France)

L'exécution sur le serveur de l'école <https://gitlab.telecomnancy.univ-lorraine.fr/> avec les paramètres par défaut a permis de générer les graphiques suivants :



11. <https://github.com/mortie/snow>

12. <https://docs.gitlab.com/ee/ci/>

IV.C Interprétation des résultats

PPII2

Remarquons que l'impact de la distance ne se ressent quasiment pas sur le temps d'exécution. Ce qui montre que l'utilisation d'une table de hash est pertinent. De plus, les temps d'exécutions comprennent le temps de remplissage de la table de hachage et de lecture du fichier CSV. Lors de la seconde partie, plusieurs appels au calcul de plus court chemin ne demanderont qu'un seul chargement de la table de hachage.

◁

Chapitre 4

▷

Seconde étape

I	Analyse	23
II	Conception et développement	23
III	Tests et performances	23

I ▷ Analyse

II ▷ Conception et développement

PPII2

III ▷ Tests et performances

◁

Chapitre 5

▷

Conclusion

I	Complétion des objectifs	25
II	Avantages et limites du projet	25
III	Améliorations possibles	25
IV	Clôture	25

I ▷ Complétion des objectifs

II ▷ Avantages et limites du projet

III ▷ Améliorations possibles

PPII2

IV ▷ Clôture

◁ Chapitre 6 ▷

Annexes