

PROSIT : Petit détour

FODIL Nel | RICCO Mathéo | GOUADFEL Rayan
27 mai 2024

Table des matières

I.	Introduction	2
II.	Analyse du contexte	2
III.	Objectifs	3
IV.	Problématique	3
V.	Plan d'Action.....	4
VI.	Définitions.....	6
	Graphes et leurs Représentations	6
	Algorithme de Dijkstra	7
	Algorithme A*	7
	Algorithme de Bellman-Ford	8
	Système d'Information Géographique (SIG)	8
	QGIS.....	9
	ArcGIS	9
	Python	9
	NetworkX	10
	Matplotlib.....	10
	Complexité algorithmique	11
	Conclusion	12
	Webographie	Erreur ! Signet non défini.

I. Introduction

Dans le cadre de la transformation des villes en espaces intelligents, notre projet, soutenu par CesiCDP et financé par la région Grand Est, vise à optimiser les itinéraires des équipes techniques modernisant l'éclairage public. Ce rapport explore les moyens d'améliorer l'efficacité énergétique et opérationnelle en développant des parcours optimisés pour l'installation de dispositifs connectés sur les lampadaires, contribuant ainsi à une gestion énergétique urbaine plus durable.

II. Analyse du contexte

Le projet s'inscrit dans une initiative plus large de la région Grand Est pour promouvoir les villes intelligentes. Cette démarche répond à un besoin croissant des collectivités territoriales de minimiser les impacts environnementaux de l'infrastructure urbaine tout en améliorant l'efficacité des services publics. L'éclairage public, souvent perçu comme une simple commodité, représente en réalité un poste significatif de consommation énergétique et un enjeu logistique important pour les municipalités.

Actuellement, les équipes techniques responsables de l'installation des dispositifs intelligents sur les lampadaires planifient leurs itinéraires de manière empirique, ce qui peut conduire à des inefficacités notables en termes de temps et de consommation de carburant. La transition vers un système d'éclairage public intelligent nécessite donc non seulement l'installation de technologies avancées mais aussi une révision des méthodes de travail pour optimiser les déplacements et les interventions.

Cette nécessité d'optimisation se manifeste dans un contexte où les technologies de l'information et la gestion des données urbaines prennent une place centrale dans la planification des villes. Le projet que nous menons vise à intégrer ces technologies dans les opérations quotidiennes des services techniques, transformant ainsi les pratiques traditionnelles en procédés plus efficaces et respectueux de l'environnement.

III. Objectifs

Amélioration de l'efficacité opérationnelle : Répondre à la nécessité d'accroître l'efficacité des équipes techniques en optimisant les itinéraires de maintenance de l'éclairage public. Cela permettra de réduire le temps de parcours et la consommation de carburant, alignant ainsi les opérations avec les principes de durabilité urbaine.

Modernisation de la gestion des données géographiques : Intégrer une approche moderne de gestion des données pour transformer les pratiques de planification des itinéraires, qui sont actuellement basées sur des méthodes empiriques, en une stratégie fondée sur l'analyse précise et la modélisation avancée.

Optimisation continue des trajets : Assurer que les parcours proposés restent les plus pertinents et efficaces malgré les changements dans les configurations urbaines et les exigences opérationnelles, nécessitant une adaptation et une amélioration continues.

Rationalisation de la prise de décision : Fournir aux décideurs des outils de visualisation et d'évaluation qui facilitent l'interprétation des données et soutiennent la prise de décision basée sur des preuves concernant les opérations de terrain.

Mise en place d'une solution durable et évolutif : Déployer un système qui non seulement répond aux besoins actuels mais est également capable de s'adapter aux futures exigences de la gestion urbaine et des technologies émergentes.

IV. Problématique

Comment trouver un chemin optimisant un déplacement au sein d'un espace non restreint tel une ville ?

V. Plan d'Action

1. Collecte et analyse des données géographiques : Nous commencerons par acquérir des données géographiques précises des rues et des emplacements des lampadaires à l'aide de systèmes d'information géographique (SIG) tels que QGIS ou ArcGIS. Cette étape est cruciale pour construire une base de données spatiale qui servira de fondement à notre modèle de graphe. Les données seront nettoyées et préparées pour garantir leur intégrité et leur applicabilité dans les étapes de modélisation.

2. Modélisation du réseau routier en tant que graphe : À partir des données géographiques, nous utiliserons Python couplé à la bibliothèque NetworkX pour transformer ces informations en un graphe où les intersections et les lampadaires seront les nœuds, et les routes les arêtes. Cette représentation nous permettra d'appliquer des algorithmes de recherche de chemin et d'optimiser les parcours.

3. Implémentation et test des algorithmes de parcours : Les algorithmes de Dijkstra, A*, et Bellman-Ford seront programmés pour déterminer les itinéraires les plus courts entre les points de départ et d'arrivée. Nous effectuerons des tests pour comparer l'efficacité de chaque algorithme en termes de temps de calcul et de pertinence des itinéraires proposés. Ces tests aideront à sélectionner l'algorithme le plus adapté pour notre application spécifique.

4. Optimisation des itinéraires : Après avoir sélectionné l'algorithme le plus efficace, nous optimiserons les itinéraires en ajustant les paramètres de l'algorithme et en réaffinant notre modèle de graphe pour améliorer encore la précision et l'efficacité des parcours. Cette phase inclura l'analyse des résultats des simulations pour identifier les zones de congestion ou les points critiques nécessitant une attention particulière.

5. Visualisation et évaluation des résultats : Utiliser des outils de visualisation tels que Matplotlib pour représenter graphiquement les itinéraires optimisés. Cela non seulement facilitera l'interprétation des résultats mais aussi permettra de présenter clairement les avantages du nouveau système de planification aux parties prenantes et aux équipes techniques.

6. Déploiement et suivi : Le système optimisé sera implémenté dans un environnement de production après une série de tests concluants. Nous mettrons en place des mécanismes de suivi pour évaluer la performance du système en conditions réelles et apporterons des ajustements basés sur les retours d'expérience et les données collectées.

VI. Définitions

Graphes et leurs Représentations

- **Graphe** : Structure composée de nœuds (ou sommets) et d'arêtes connectant ces nœuds. Les graphes sont utilisés pour modéliser des réseaux dans divers contextes.
 - **Liste d'adjacence** : Chaque nœud est associé à une liste contenant les nœuds qui lui sont adjacents. Cela permet une représentation efficace pour les graphes comportant de nombreuses connexions.
 - **Matrice d'adjacence** : Matrice M où l'élément $M[i][j]$ est non nul (typiquement 1) s'il existe une arête entre le sommet i et le sommet j . Utile pour les graphes denses.
 - **Matrice d'incidence** : Matrice qui établit une relation entre les arêtes et les sommets, où chaque colonne représente une arête et chaque ligne un sommet.
- **Types de Graphes**
 - **Graphe biparti** : Les sommets peuvent être divisés en deux ensembles distincts tels que toutes les arêtes relient un sommet de l'un à un sommet de l'autre ensemble. Aucune arête ne relie deux sommets du même ensemble.
 - **Graphe complet** : Chaque paire de sommets dans le graphe est connectée par une arête. Ce type de graphe est intensément connecté.
 - **Graphe cyclique** : Contient au moins un cycle, c'est-à-dire un chemin fermé qui revient à son point de départ sans répéter aucun sommet, à l'exception du premier/dernier.
- **Propriétés des Graphes**
 - Degré d'un sommet : Nombre d'arêtes incidentes à un sommet, indiquant combien de connexions directes le sommet possède.
 - Cycle : Chemin fermé commençant et se terminant par le même sommet, sans répétition des autres sommets.
 - Chemin : Séquence de sommets connectés par des arêtes, où chaque paire de sommets consécutifs est directement reliée par une arête.
 - Connexité : Propriété d'un graphe où chaque paire de sommets est reliée par un chemin. Un graphe est dit connexe s'il est possible de se déplacer de n'importe quel sommet à n'importe quel autre sommet par ces chemins.

- Composante connexe : Sous-ensemble de sommets dans un graphe où il existe un chemin entre chaque paire de sommets du sous-ensemble, et aucun chemin vers des sommets hors de ce sous-ensemble.

Algorithme de Dijkstra

Cet algorithme, conçu par l'informaticien néerlandais Edsger W. Dijkstra en 1956, est une méthode de calcul du chemin le plus court qui permet de trouver le trajet le plus court entre un nœud de départ donné et tous les autres nœuds dans un graphe pondéré, où les poids des arêtes représentent, par exemple, des distances ou des coûts. L'algorithme de Dijkstra est particulièrement efficace dans les réseaux routiers où il est nécessaire de minimiser le temps de parcours ou la distance parcourue. Il fonctionne en étendant de manière répétée un ensemble de nœuds, appelé le front d'onde, à partir du nœud initial, en explorant les nœuds adjacents ayant le coût cumulé le plus faible jusqu'à présent, et en mettant à jour ces coûts au fur et à mesure. Cet algorithme est couramment utilisé dans les systèmes de navigation GPS pour suggérer le chemin le plus rapide entre des points géographiques.

Algorithme A*

Développé par Peter Hart, Nils Nilsson et Bertram Raphael en 1968, l'Algorithme A* est une technique de recherche de chemin qui est souvent utilisée pour trouver le chemin le plus court entre un point de départ et un point d'arrivée dans un graphe pondéré. Contrairement à l'algorithme de Dijkstra, A* améliore l'efficacité en utilisant une heuristique pour estimer le coût le plus bas restant pour atteindre l'objectif. Cette heuristique aide l'algorithme à prioriser la recherche vers des chemins qui semblent mener plus directement vers la destination. A* est particulièrement utile dans les applications de planification de trajet et de jeu où il est nécessaire de trouver rapidement des solutions optimales. L'efficacité de l'Algorithme A* dépend grandement de la qualité de la fonction heuristique utilisée, qui doit être adéquate pour éviter l'exploration excessive de chemins non pertinents tout en garantissant que le chemin optimal est trouvé.

Algorithme de Bellman-Ford

Cet algorithme, nommé d'après ses développeurs Richard Bellman et Lester Ford, est utilisé pour calculer les chemins les plus courts depuis un seul nœud source vers tous les autres nœuds dans un graphe. L'une des caractéristiques distinctives de l'Algorithme de Bellman-Ford est sa capacité à gérer les graphes qui contiennent des arêtes avec des poids négatifs, ce qui n'est pas possible avec l'algorithme de Dijkstra. Cela le rend particulièrement utile dans des contextes où les poids des arêtes peuvent représenter des coûts, des risques ou d'autres valeurs susceptibles de varier de manière négative. L'algorithme fonctionne en relaxant progressivement les arêtes, c'est-à-dire en améliorant les estimations de coût minimum pour atteindre chaque nœud du graphe à partir de la source, et ce, à travers plusieurs itérations. Malgré sa flexibilité, l'algorithme est moins efficace que l'Algorithme de Dijkstra en termes de temps d'exécution, particulièrement dans les graphes denses, mais il offre une solution robuste lorsque les poids négatifs sont présents.

Théorème d'Euler

Le théorème d'Euler, formulé par Leonhard Euler en 1736, stipule qu'un graphe connexe possède un cycle eulérien (un cycle qui passe par chaque arête exactement une fois) si et seulement si chaque sommet du graphe a un degré pair. Dans le contexte de notre projet, ce théorème est crucial car il nous permet de déterminer si un itinéraire optimal peut être trouvé en passant par chaque rue exactement une fois, ce qui est essentiel pour l'optimisation des trajets des équipes techniques.

Système d'Information Géographique (SIG)

Un Système d'Information Géographique (SIG) est un outil technologique conçu pour capturer, stocker, manipuler, analyser, gérer et présenter toutes sortes de données géographiques. Le SIG permet aux utilisateurs de visualiser, de comprendre, d'interpréter et de visualiser des données géospatiales sous des formes variées telles que des cartes, des graphiques et des rapports. Ces systèmes sont essentiels pour intégrer des informations liées à la localisation, ce qui les rend inestimables dans une multitude de domaines incluant l'urbanisme, la gestion des ressources naturelles, les transports, l'ingénierie, la planification, et plus encore. Dans le contexte de votre projet, un SIG serait utilisé pour recueillir des données précises sur les routes, les emplacements des lampadaires, et d'autres infrastructures urbaines. Cela faciliterait la modélisation précise du réseau routier et des itinéraires, en fournissant une base solide pour l'optimisation des parcours et la planification stratégique. Les exemples de systèmes SIG populaires incluent QGIS et ArcGIS, qui offrent des fonctionnalités

avancées pour la manipulation de données spatiales et la création de visualisations détaillées.

QGIS

QGIS est un système d'information géographique (SIG) open source qui permet aux utilisateurs de créer, modifier, visualiser, analyser et publier des informations géospatiales sur différentes plateformes. Il est largement reconnu pour sa facilité d'utilisation, sa flexibilité et ses fonctionnalités étendues, qui le rendent accessible à la fois aux professionnels et aux amateurs de géographie. QGIS supporte une variété de formats de données vectorielles, raster, et bases de données, et offre des outils puissants pour l'analyse spatiale, la gestion des données et l'intégration avec d'autres applications SIG. Dans le contexte de votre projet, QGIS peut être utilisé pour analyser le réseau routier et planifier les itinéraires optimisés pour l'installation de dispositifs sur les lampadaires, facilitant ainsi la collecte de données précises et la visualisation des résultats des analyses pour une meilleure prise de décision et planification stratégique.

ArcGIS

ArcGIS est une plateforme de système d'information géographique (SIG) développée par Esri, qui offre des outils complets pour la collecte, l'analyse, la visualisation et le partage de données géospatiales. Cette suite logicielle est largement utilisée par les professionnels dans des domaines tels que l'urbanisme, l'environnement, la gestion des ressources, et plus encore, pour créer des cartes interactives et des visualisations complexes qui facilitent la prise de décisions basée sur des données géographiques précises. ArcGIS se distingue par sa capacité à intégrer des données provenant de diverses sources, son soutien pour l'analyse spatiale avancée et ses fonctionnalités de modélisation 3D. Pour votre projet, ArcGIS pourrait être utilisé pour cartographier et analyser le réseau routier de la ville, permettant de déterminer les itinéraires les plus efficaces pour les équipes techniques et d'optimiser la gestion des ressources dans le cadre de l'installation des dispositifs d'éclairage intelligent.

Python

Python est un langage de programmation de haut niveau, interprété et polyvalent, qui est largement utilisé pour le développement de logiciels, l'analyse de données, l'intelligence artificielle, le calcul scientifique, et bien d'autres domaines. Il est particulièrement apprécié pour sa syntaxe claire et lisible qui facilite la rédaction de code propre et maintenable. Python supporte plusieurs paradigmes de programmation,

y compris la programmation impérative, orientée objet et fonctionnelle. Il est doté d'une vaste bibliothèque standard et d'un écosystème riche en bibliothèques tierces, offrant des outils puissants pour manipuler une variété de données, y compris des données géospatiales et des graphes. Dans le cadre de notre projet, Python peut être utilisé pour implémenter des algorithmes de parcours de graphe, comme l'algorithme de Dijkstra, l'algorithme A*, et l'algorithme de Bellman-Ford, ainsi que pour la modélisation, la simulation, et la visualisation des données de réseau routier et des itinéraires optimisés à l'aide de bibliothèques telles que NetworkX et Matplotlib.

NetworkX

NetworkX est une bibliothèque Python conçue pour la création, la manipulation et l'étude de la structure, de la dynamique et des fonctions de graphes complexes. Elle permet de modéliser des graphes sous forme de structures de données et fournit des algorithmes avancés pour l'analyse de graphes, y compris les parcours de graphe, la détection de communautés, et le calcul de mesures de centralité. NetworkX supporte différents types de graphes (orientés, non orientés, pondérés, non pondérés) et peut gérer des graphes de taille variable, des petits réseaux aux grands réseaux complexes. Dans le cadre de notre projet, NetworkX sera utilisé pour modéliser le réseau routier de la ville, implémenter et tester des algorithmes de recherche de chemin comme Dijkstra, A* et Bellman-Ford, et analyser les itinéraires optimisés pour les équipes techniques chargées de l'installation des dispositifs sur les lampadaires.

Matplotlib

Matplotlib est une bibliothèque Python largement utilisée pour la création de visualisations statiques, animées et interactives de données. Elle permet de générer divers types de graphiques, tels que des lignes, des barres, des histogrammes, des scatter plots, des heatmaps, et bien d'autres. Matplotlib est appréciée pour sa flexibilité et sa capacité à produire des visualisations de haute qualité, intégrables dans des applications web et des rapports. Dans le cadre de notre projet, Matplotlib sera utilisé pour visualiser les résultats des simulations d'itinéraires optimisés, permettant ainsi une évaluation claire et une présentation efficace des solutions proposées aux parties prenantes et aux équipes techniques.

Complexité algorithmique

La complexité algorithmique mesure l'efficacité d'un algorithme en termes de temps d'exécution (complexité temporelle) ou de l'espace mémoire requis (complexité spatiale) en fonction de la taille de l'entrée. Cette mesure est cruciale pour évaluer la performance des algorithmes, surtout lorsque les données à traiter sont volumineuses. Dans le cadre de notre projet, la complexité algorithmique est particulièrement pertinente pour choisir et optimiser les algorithmes de recherche de chemin (Dijkstra, A*, Bellman-Ford). Étant donné que nous travaillons avec un réseau routier pouvant comporter un grand nombre de nœuds (intersections) et d'arêtes (routes), comprendre et gérer la complexité nous permet de garantir que nos solutions seront performantes même à grande échelle.

Notation	Nom de la complexité	Exemple d'algorithme
$O(1)$	Complexité constante	Accès à un élément d'un tableau
$O(\log n)$	Complexité logarithmique	Recherche binaire
$O(n)$	Complexité linéaire	Parcours d'un tableau
$O(n \log n)$	Complexité linéarithmique	Tri par fusion, tri rapide
$O(n^2)$	Complexité quadratique	Tri à bulles, tri par insertion
$O(n^3)$	Complexité cubique	Multiplication de matrices par la méthode naïve
$O(2^n)$	Complexité exponentielle	Algorithme de la tour de Hanoï
$O(n!)$	Complexité factorielle	Résolution d'un problème du voyageur de commerce (TSP)

Conclusion

En conclusion, ce rapport présente une approche méthodologique pour optimiser les itinéraires des équipes techniques chargées de l'installation des dispositifs connectés sur les lampadaires, dans le cadre de l'initiative de ville intelligente financée par la région Grand Est. Nous avons proposé une solution basée sur la collecte et l'analyse des données géospatiales, permettant de modéliser le réseau routier sous forme de graphe à l'aide de la bibliothèque NetworkX.

Nous prévoyons d'implémenter et de tester plusieurs algorithmes de recherche de chemin, notamment Dijkstra, A*, et Bellman-Ford, afin d'identifier les itinéraires les plus courts et les plus efficaces. De plus, le théorème d'Euler jouera un rôle clé dans notre projet, en nous permettant de déterminer si un cycle eulérien (un chemin passant une seule fois par chaque arête) peut être trouvé dans notre réseau. Cela est crucial pour optimiser les trajets de manière à ce que chaque rue soit couverte de manière efficace et systématique.

La visualisation des résultats avec Matplotlib sera essentielle pour l'évaluation et la communication des solutions proposées, facilitant la compréhension et l'acceptation par les parties prenantes. Le déploiement du système optimisé, associé à un suivi continu de la performance, garantira une solution adaptable et évolutive, capable de répondre aux futures exigences des collectivités territoriales.

En somme, notre projet vise à poser les bases d'une gestion urbaine plus intelligente et éco-responsable, avec l'ambition de servir de référence pour d'autres initiatives similaires. Nous sommes convaincus que cette approche méthodologique, combinant l'utilisation du théorème d'Euler et des algorithmes de parcours de graphe, contribuera significativement à l'amélioration de l'efficacité des services publics dans le cadre des villes intelligentes.