Optimisation des opérations de déneigement à Montréal

Projet de Recherche Opérationnelle – EPITA
Groupe : Norig Ciserani, Nathan Delmarche, Lowen Delhome, Killian Heritier, Julien Fleury

1. Contexte et périmètre d'étude

La ville de Montréal est confrontée chaque année à des épisodes neigeux réguliers entre octobre et avril, engendrant des contraintes logistiques importantes sur la mobilité urbaine. Afin de limiter les impacts économiques et sociaux, des équipes de déneigement interviennent massivement avec plus de 3000 employés et 2000 véhicules spécialisés. L'ensemble des opérations couvre environ 10 000 km de routes et mobilise un budget annuel de 165 millions de dollars.

Le projet qui nous a été confié vise à optimiser ces opérations de déblaiement, en se concentrant sur la phase de reconnaissance préalable (réalisée par drones) ainsi que sur la planification des itinéraires de déneigeuses dans des secteurs localisés. Plus précisément, l'objectif consiste à :

- 1. Déterminer un chemin de vol eulérien pour un drone capable de survoler l'ensemble du réseau routier montréalais.
- 2. Simuler la détection des routes enneigées à partir de ce survol.

- 3. Définir des stratégies de déneigement locales, en tenant compte des contraintes de coût, d'efficacité, de type de véhicule, et de comportement d'agent.

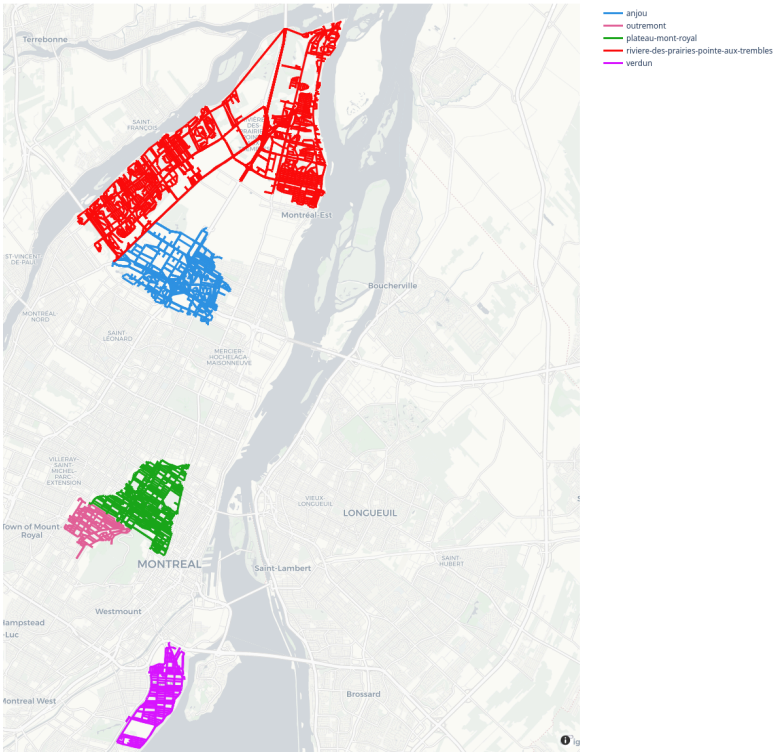
Le périmètre d'étude couvre à la fois l'intégralité de la ville (pour le drone) et cinq arrondissements spécifiques (Outremont, Verdun, Anjou, Rivière-des-Prairies–Pointe-aux-Trembles et le Plateau-Mont-Royal) pour les phases de déneigement.

2. Modélisation du problème

2.1. Génération du graphe routier

Nous avons utilisé la bibliothèque osmnx pour extraire et convertir les données OpenStreetMap en graphes routiers exploitables. Le réseau routier est modélisé comme un graphe non orienté, dont les sommets représentent des intersections et les arêtes des segments routiers. Le poids des arêtes est proportionnel à leur longueur réelle (en mètres), afin de refléter au mieux les distances à parcourir.

Pour la reconnaissance du drone, nous avons fusionné les graphes des différents quartiers afin d'obtenir un graphe couvrant la ville entière. Ce graphe est ensuite traité pour déterminer un chemin eulérien.



2.2. Eulérisation du graphe

Un circuit eulérien est un chemin dans un graphe non orienté qui passe une seule fois par chaque arête et revient à son point de départ. Pour que ce chemin existe, il est nécessaire que tous les sommets du graphe soient de degré pair. Or, dans la structure urbaine réelle, de nombreux sommets sont de degré impair.

Nous avons donc utilisé une approche classique d'eulérisation : identifier tous les sommets de degré impair, calculer les distances minimales entre toutes les paires de ces sommets (en utilisant networkx.shortest_path_length), puis déterminer un couplage parfait de poids minimal (minimum weight perfect matching) afin d'ajouter les arêtes nécessaires. Cette opération est coûteuse en temps, notamment pour un graphe de la taille de Montréal (plus de 12 000 nœuds impairs). Pour optimiser les performances, nous avons parallélisé le calcul des distances à l'aide de ThreadPoolExecutor et avons utilisé tqdm pour afficher une barre de progression complète.

Cette phase permet de produire un graphe eulérisé, à partir duquel le chemin eulérien est calculé (grâce à networkx.eulerian_circuit) au format pkl, puis visualisé et sauvegardé dans différents formats (JSON, HTML).

2.3. Simulation des conditions météorologiques

La reconnaissance aérienne a pour objectif d'identifier les segments routiers effectivement enneigés. En l'absence d'un modèle météorologique réel, nous avons simulé l'état de la route en utilisant des algorithmes de simplex noise pour générer du pseudo aléatoire. Cette information est enregistrée dans un fichier CSV (snow_map.csv) qui concerne toute la ville et permet de visualiser les segments enneigés de manière distincte.

La superposition des couches (chemin eulérien, segments enneigés, routes dégagées) est rendue via Plotly dans des cartes interactives, facilitant ainsi l'analyse.

3. Simulation des véhicules de déneigement

3.1. Modèle d’agent

Le véhicule de déneigement est modélisé comme un agent autonome parcourant un graphe orienté. À chaque itération, l’agent observe son voisinage immédiat (les sommets accessibles), et prend une décision en fonction de plusieurs critères : état d’enneigement, historique de passage, priorité stratégique, comportement glouton ou aléatoire.

Nous avons intégré un système de mémoire locale, simulant la capacité du véhicule à se rappeler des derniers sommets visités. Cela permet de comparer des stratégies à court terme (sans mémoire) et à plus long terme (avec historique des trajets). Plusieurs variantes d’agents ont été implémentées, simulant des types de véhicules différents (ex. : déneigeuse rapide mais coûteuse, lente mais robuste, etc.).

3.2. Paramètres et configuration

Lors de l'exécution de la simulation, l'utilisateur sélectionne un quartier à déneiger, puis choisit une stratégie :

- économie de temps,
- économie de carburant,
- ou coût minimal (en utilisant une seule déneigeuse de type 1).\

Le programme affiche les données principales à l'écran (`print`) et enregistre les informations détaillées dans le fichier `reports/all_runs.json`.

4. Comparaison des types de déneigement

Nous avons donc développé deux programmes permettant de comparer les avantages et les inconvénients de deux types de déneigement. Grâce à un tableau comparatif et avec un histogramme, nous avons analysé plusieurs indicateurs clés tels que le temps nécessaire, la consommation de carburant, la distance parcourue (en kilomètres) ainsi que le coût global de l'opération.

Pour faciliter la lecture et l'interprétation des résultats, nous avons calculé des moyennes pour chaque type de déneigement, puis nous avons normalisé ces données sur la base de 500 arêtes à déneiger.

Les résultats montrent que le second type de déneigement est nettement plus rapide, mais il présente un coût plus élevé et une consommation de carburant bien plus importante. À l’inverse, le premier type est plus économique en carburant et en coût, mais il nécessite davantage de temps pour accomplir la tâche.

5. Limites du modèle et perspectives

5.1. Limites actuelles

Le modèle actuel repose sur des hypothèses simplificatrices : neige distribuée aléatoirement, réseau routier sans contraintes de circulation fines (ex. : sens interdits complexes, interdictions horaires), agents isolés (pas de coordination explicite).

La phase d'eulérisation, bien que efficace avec notre version multithreadée, reste coûteuse sur des graphes massifs.

5.2. Perspectives

Plusieurs pistes pourraient être explorées :

- Utiliser un historique météo réel pour déterminer les zones les plus à risque, voire souscrire à une API.
- Introduire une coordination entre agents (partage de mémoire, stratégies collectives).
- Ajouter des contraintes temporelles (fenêtres horaires, début/fin de service).
- Comparer les performances des stratégies selon un critère économique global (optimisation multi-objectif).
- Introduire une utilisation de la mémoire: étant donné que la complexité d'accès à la mémoire est en $\Theta(1)$, tandis que celle du parcours du graphe est en $O(d \log(d))$ avec pour chaque recherche de prochain noeud avec d le nombre moyen de noeuds dans un graphe, il est plus optimal de vérifier dès l'exploration si un nœud potentiel est présent en mémoire, afin d'éliminer immédiatement les possibilités inutiles, plutôt que de les vérifier systématiquement dans le graphe.

6. Conclusion

Ce projet de recherche opérationnelle nous a permis d'aborder de manière concrète la modélisation et l'optimisation d'un système logistique réel à travers l'usage d'algorithmes de graphes, de simulation d'agents, et d'analyse de données.

Nous avons conçu un pipeline complet, de la génération du graphe urbain à l'analyse détaillée de stratégies de déneigement, en intégrant des éléments réalistes (coûts, efficacité, contraintes physiques) et en développant des outils de visualisation et de mesure.

Les résultats obtenus montrent que des choix algorithmiques simples, lorsqu'ils sont bien adaptés aux contraintes du terrain, peuvent offrir des solutions robustes et comparables à des stratégies plus complexes.