

Rapport de recherche sur le déneigement de Montréal

Léo Sambrook Mathéo Crespel Briac Guellec Augustin Claude
Inès Metiba

8 juin 2024



Résumé

Dans ce rapport, nous allons expliquer l'étendue de nos recherches sur le déneigement de Montréal. Nous avons étudié le parcours de drones pour surveiller les accumulations de neige et de déneigeuses pour dégager les routes. En modélisant les routes de Montréal comme des graphes, nous avons développé nos propres algorithmes pour optimiser les trajets de déneigement, réduisant ainsi les coûts et améliorant l'efficacité des opérations. Le tout a été réalisé en Python.

Table des matières

1	Introduction	3
1.1	Modélisation	3
1.2	Algorithmes	3
2	Les Données utilisées	3
3	Les Hypothèses et choix de modélisation	4
3.1	Drone	4
3.2	Déneigeuses	4
4	Le Modèle	4
4.1	Les indicateurs	4
4.2	La comparaison des algorithmes	5
4.2.1	Drone	5
4.2.2	Déneigeuses	5
4.3	Les solutions retenues	5
5	Les limites du modèle	5
6	Bibliographie	6

1. Introduction

Pour commencer, nous avons orienté nos recherches sur deux thématiques principales : la modélisation et les algorithmes à implémenter.

1.1. Modélisation

Pour extraire un graphe des routes de la ville de Montréal, nous avons utilisé OpenStreet-Map (avec la bibliothèque OSMnx) combiné à NetworkX. Il nous a vite semblé nécessaire de créer deux graphes distincts : un pour la partie drone, couvrant l'ensemble de la ville, et un autre pour la partie déneigeuse, couvrant les 5 quartiers à déneiger : Outremont, Verdun, Anjou, Rivière-des-Prairies-Pointe-aux-Trembles et Plateau Mont-Royal. Ces deux graphes sont non-orientés.

1.2. Algorithmes

Afin de trouver les algorithmes les plus optimisés pour les transports, nous avons envisagé plusieurs possibilités (parcours en largeur, Dijkstra, A*, etc.) avant de nous intéresser au problème du postier chinois qui semble être la solution la plus optimale pour parcourir le graphe avec le drone ainsi qu'avec les potentielles déneigeuses. Il nous faudrait donc modifier les deux graphes résultants de la modélisation pour qu'ils soient d'abord connexes puis euliériens.

2. Les Données utilisées

Voici les données prises en compte pour les transports :

Appareil	Coût fixe	Coût kilométrique	Coût horaire (<8h)	Coût horaire (>8h)	Vitesse moyenne
Drone	100€	0,01€	/	/	60km/h
Déneigeuse type 1	500€	1,1€	1,1€	1,3€	10km/h
Déneigeuse type 2	800€	1,3€	1,3€	1,5€	20km/h

Concernant les routes de Montréal, nous avons utilisé OpenStreetMap, une base de données géographiques libre et modifiable à l'échelle mondiale. Cela nous a permis d'extraire les routes qui nous intéressaient. Nous avons manipulé ces données avec la librairie OSMnx, qui transforme les données extraites en graphes. Ensuite, nous avons utilisé NetworkX, qui offre une grande variété de fonctions, pour manipuler ces graphes.

3. Les Hypothèses et choix de modélisation

3.1. Drone

Tout d’abord, nous avons établi que le drone avait une vitesse moyenne d’environ 60km/h. Nous avons donc estimé, en visualisant la carte de Montréal, qu’un seul drone serait suffisant pour la reconnaissance de la neige dans toute la ville. Nous avons donc basé la suite de notre recherche sur l’hypothèse que ce drone unique doit parcourir tout le graphe de manière la plus optimisée possible. Afin d’optimiser le temps d’exécution de notre code, nous avons décidé de séparer le graphe de Montréal en ses 19 quartiers et de paralléliser l’exécution de notre algorithme. Grâce à ce choix, qui réduit très légèrement l’optimalité de notre solution, nous pouvons analyser Montréal dans son ensemble en une dizaine de minutes environ, au lieu de plusieurs heures.

3.2. Déneigeuses

Nous pensions dans un premier temps utiliser la programmation linéaire pour répondre au problème d’assignation des déneigeuses en considérant comme fonction objectif la minimisation du coût total avec les contraintes suivantes : vitesse des déneigeuse, leur coûts respectifs ainsi que la distance minimale à parcourir pour déneiger la ville (somme des longueurs des routes enneigées). Il nous a finalement paru plus logique de les assigner par quartier, surtout qu’une déneigeuse par quartier suffit à avoir un résultat dans un temps raisonnable même si le quartier est totalement enneigé.

4. Le Modèle

Dans cette section, nous détaillons les indicateurs utilisés pour évaluer les différentes stratégies de déneigement, comparons les scénarios envisagés, et présentons la solution retenue en fonction des analyses effectuées.

4.1. Les indicateurs

Les indicateurs que nous avons pris en compte pour comparer l’efficacité de plusieurs solutions sont :

- La distance totale parcourue
- Le coût
- Le temps du parcours

Ils sont pris en compte indépendamment pour le drone et pour chaque déneigeuse.

4.2. La comparaison des algorithmes

4.2.1. Drone

Avec la mise en place du parcours du drone, nous avons commencé par implémenter le parcours du postier chinois mais nous avons constaté que celui-ci mettait trop de temps et que cela pourrait être inconfortable pour la mairie de Montréal ainsi qu'improductif. C'est pourquoi, nous avons décidé d'apporter quelques modifications à l'algorithme de base pour qu'il soit plus rapide.

4.2.2. Déneigeuses

Pour les déneigeuses, nous avons envisagé plusieurs algorithmes de parcours de graphes :

- A*/Dijkstra (pour le chemin le plus court)
- Le problème du voyageur de commerce (pour un parcours optimisé passant par tous les sommets une unique fois)
- Le problème du postier chinois (pour trouver le chemin le plus court passant par toutes les arêtes)

4.3. Les solutions retenues

Pour le drone, le parcours du graphe se fait à l'aide d'une version modifiée du postier chinois, avec un parcours de quartier en quartier. Nous avons pensé à deux manières de rendre efficace le parcours des déneigeuse. La première, assez intuitive, consiste à relier chaque route enneigée à celle la plus proche et enfin de relier le point d'arrivée au point de départ avec un algorithme de plus court chemin créant ainsi un circuit que la déneigeuse peut parcourir. La deuxième, comme pour l'algorithme du postier chinois, vise à rendre le graphe connexe puis eulérien avant d'effectuer un algorithme calculant un circuit eulérien sur le graphe nouvellement obtenu. . Malheureusement toutes les routes enneigées ne sont pas liées les unes aux autres rendant ainsi l'utilisation de l'algorithme du postier chinois impossible tel quel. Pour répondre à ce problème nous avons lié les noeuds isolés en insérant des arêtes fictives que nous remplaçons à la fin de l'algorithme par un chemin bien réel constitué de routes non enneigées. Nous utilisons l'algorithme de Dijkstra pour trouver le plus court chemin mentionné précédemment dans le graphe de la ville entière.

5. Les limites du modèle

La génération de la vidéo du parcours est principalement compatible avec Linux. La fonction 'eulerize', utilisée dans notre solution pour générer des circuits eulériens, produit parfois des graphes semi-eulériens, ce qui signifie qu'un chemin eulérien existe, mais il n'est pas nécessairement un circuit fermé. Dans le contexte du déneigement de Montréal, l'utilisation de graphes semi-eulériens peut poser des défis supplémentaires, car ils exigent que les véhicules de déneigement commencent et finissent leur trajet à des points différents, ce qui

complique la planification et peut augmenter la distance totale parcourue. Une autre limitation de notre modèle réside dans le découpage des quartiers à déneiger et le choix du niveau de ces dernières. Notre modèle actuel assigne une déneigeuse à chaque quartier qu’importe la configuration et calcule lequel des deux niveaux est le plus optimisé en termes de coût. Par ailleurs, Osmnx, pour éviter la redondance des informations entre tous les arcs des graphes, ne représente pas les routes à double sens par 2 arcs opposés, mais uniquement un seul. Cela nous a donc empêché à la fois d’avoir une représentation correcte des routes que devaient prendre les déneigeuses mais aussi de trouver à coup sûr des chemins entre deux noeuds. Une solution était d’utiliser des graphes non orientés mais, pour être le plus proche du réel, nous avons décidé de rendre eulérien les sous graphes orientés enneigés puis, lorsque l’on calcule l’itinéraire des déneigeuses, on considère les routes de Montréal en non orienté. Les déneigeuses, selon notre modèle, ne respectent donc pas le code de la route.

6. Bibliographie

Références

- AUTHIER, Pierre (1975). *Détermination d’un parcours optimum pour le débleaement de la neige*. URL : https://savoirs.usherbrooke.ca/bitstream/handle/11143/13182/Authier_Pierre_MScA_1975.pdf?sequence=1&isAllowed=y.
- BOEING, Geoff (2016-2024). *Documentation de la librairie OSMnx*. URL : <https://osmnx.readthedocs.io/en/stable/>.
- BROOKS, Andrew (2017). *Graph Optimization with NetworkX in Python*. URL : <https://www.datacamp.com/tutorial/networkx-python-graph-tutorial>.
- MONTRÉAL (2023). *Tout savoir sur le déneigement à Montréal*. URL : <https://montreal.ca/articles/tout-savoir-sur-le-deneigement-montreal-22600#:~:text=En%20moyenne%2C%205%20op%C3%A9rations%20de,ramasser%2015%20cm%20de%20neige..>
- NETWORKX, Développeurs (2004-2024). *Documentation de la librairie NetworkX*. URL : <https://networkx.org/documentation/stable/reference/index.html>.
- QUIRION-BLAIS, Olivier (2017). *Optimisation de TOURNÉES de véhicules en viabilité hivernale*. URL : <https://core.ac.uk/download/pdf/213621086.pdf#page67>.

MONTRÉAL 2023 BOEING 2016-2024 NETWORKX 2004-2024 AUTHIER 1975 BROOKS 2017 QUIRION-BLAIS 2017