



Synthèse de recherche : déneigement de Montréal

Boisgontier Owen, Raitiere-delsupexhe Jules, Even Mathieu, Lelong Axel
ING1 Epita 2026

Juin 2024

Sommaire

1	Introduction	2
2	Recherche et modélisation	3
2.1	Résumé des données utilisées, et du périmètre considéré	3
2.2	Hypothèses et choix de modélisation	3
2.3	Solutions retenues, indicateurs, comparaison des scénarios	3
2.4	Les limites du modèle	4
3	Conclusion	5

1 Introduction

Nous sommes une équipe de 4 personnes chargée de donner des simulations de déneigement des quartiers de Montréal pour l'hiver prochain. Ce document indique nos pistes de réflexion, les idées retenues, les solutions mises en places ainsi que les éventuels problèmes rencontrés et résolus (ou non).

2 Recherche et modélisation

2.1 Résumé des données utilisées, et du périmètre considéré

Dans le cadre de ce projet, où nous devons proposer une solution pour déneiger Montréal efficacement, nous nous concentrons particulièrement sur 5 quartiers : Outremont, Verdun, Anjou, Rivière-des-prairies-pointe-aux-trembles, et Le Plateau-Mont-Royal. En effet, ils sont représentatifs en terme de superficie. Nous disposons de données concernant les moyens utilisés par la ville pour réaliser cette tâche :

- Super Drone et ses coûts associés pour analyser le niveau de neige tombé :
 - Coût fixe : 100 €/jour
 - Coût kilométrique : 0.01 €/km
- Les différents types de déneigeuses, les spécificités et leurs coûts associés :
 - Coût fixe : type I : 500€/jour, type II : 800 €/jour
 - Coût kilométrique : type I : 1.1 €/km, type II : 1.3 €/km
 - Coût horaire les 8 premières heures : type I : 1.1 €/h, type II : 1.3 €/h
 - Coût horaire au delà des 8 premières heures : type I : 1.3 €/h, type II : 1.5 €/h
 - Vitesse moyenne : type I : 10 km/h, type II : 20 km/h

2.2 Hypothèses et choix de modélisation

Nous avons choisis d'utiliser Python pour notre modèle. En effet, grâce à la bibliothèque 'OSMNX', nous pouvons extraire les données géographiques de chaque quartier.

Ces extractions nous donnent des graphes orientés avec lesquels nous pouvons travailler. Ainsi, la longueur et la direction (double sens ou sens unique) des routes sont disponibles dans les attributs des arêtes. A partir de cela, nous pouvons commencer à construire notre modèle.

Pour le drone, nous n'avons pas besoin de prendre en compte le sens de la circulation. Ainsi, nous utilisons un graphe non-orienté par quartier. De plus, celui-ci doit être eulérien afin de minimiser le coût kilométrique, tout en analysant toutes les routes.

Pour les déneigeuses, nous gardons les graphes orientés. Nous ajoutons le niveau de neige tombé récolté par le drone.

2.3 Solutions retenues, indicateurs, comparaison des scenarios

Pour rendre un graphe non-orienté eulérien, nous analysons tous les noeuds de degrés impairs pour ensuite les relier en ajoutant des arêtes sur une route déjà existante. Pour optimiser ce processus, nous prenons les couples de noeuds les plus proches en utilisant l'algorithme de Dijkstra.

Ainsi, nous pouvons calculer le coût final du drone sur chaque quartier et le temps d'analyse. En effet, à la fin de son parcours, on connaît la distance totale qu'il a parcouru, et on peut en déduire le coût final.

Pour les véhicules, nous séparons le graphe en différents clusters. Certains sont plus larges que d'autres et différentes dispositions sont testées. La plus optimale est gardée.

Les différentes tailles de clusters sont utiles pour que les deux types de machines aient la même charge de travail. Ainsi, le temps de déneigement est optimale pour un coût qui ne sera que légèrement plus cher. La machine de type II allant deux fois plus vite que l'autre, elle parcourra deux fois plus de distance pour un même temps donné.

2.4 Les limites du modèle

Les premières limites que nous avons rencontrées touchent le temps de calcul des graphes. En effet, notre première itération de transformation de graphe eulérien était rapide mais ajoutée beaucoup trop de temps de trajet pour le drone. En voulant optimiser ce léger souci, nous testons tous les couples de noeuds problématiques pour lier les plus proches et ainsi réduire le trajet. Cependant, ces tests de couples prennent désormais trop de temps. (jusqu'à 45 minutes pour le plus grand quartier).

Nous avons réussi à réduire le temps de calcul en partant sur un algorithme dit 'glouton', faisant moins de comparaisons tout en gardant un résultat proche du cas idéal.

Une seconde limite est le clustering, parce que plusieurs dispositions sont testées, celui-ci prend du temps (une dizaine de minutes pour le plus grand). Mais la disposition est idéal pour garder un coût bas et une efficacité élevée.

Cependant, lors de la visualisation du déneigement, certaines arêtes restent couvertes de neiges, le trajet d'un véhicule vers son cluster associé ne déneige pas la route (même si il est passé par la dite route)

3 Conclusion

Ce projet fut un challenge, surtout en terme de temps.

Nous avons du réfléchir vite et se répartir les tâches efficacement. Pendant qu'une partie traitait le cas du drone, l'autre s'est penchée sur le parcours des véhicules et la division de graphe.

Les multiples problèmes et questions que nous nous sommes posées durant la conception du projet nous ont considérablement freiné. Parfois pour optimiser les coût, le temps, ou les calculs.

Cependant, nous avons réussi à donner une simulation capable de traiter les cinq quartiers avec la contrainte de temps et d'argent.