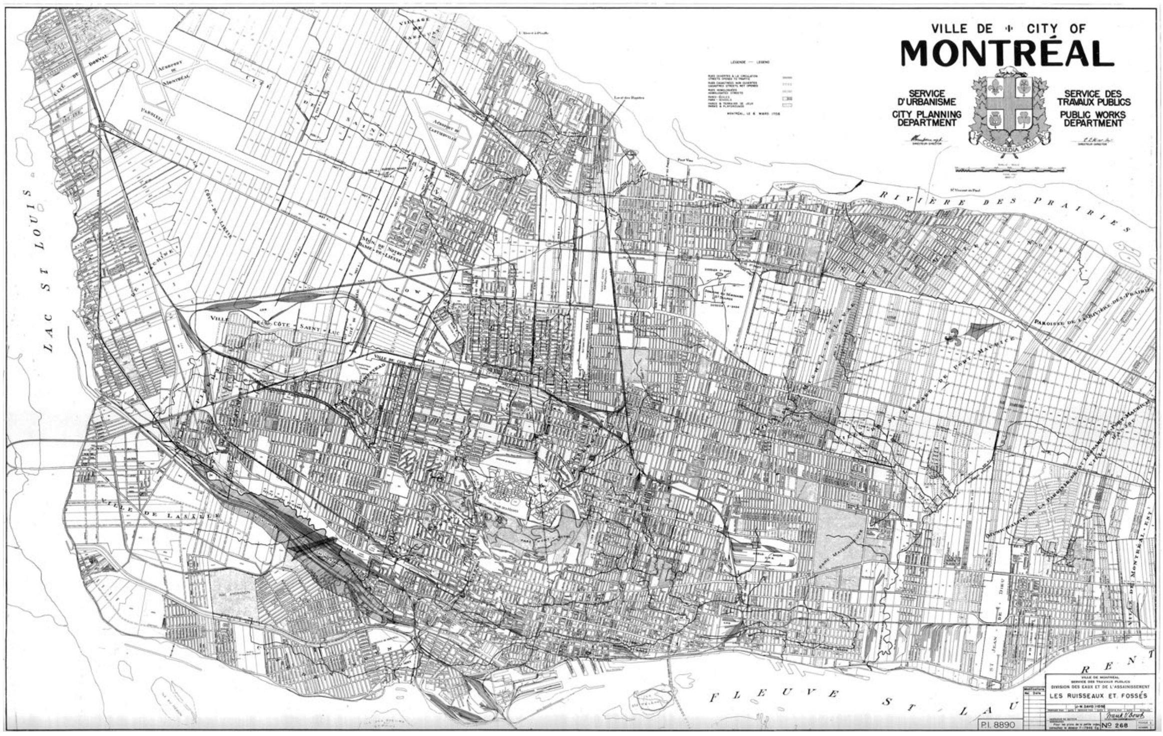
Rapport de projet

Eléments de Recherche Opérationnelle



THEZE Lucas
FOUCHARD Samuel
DUNAND Tom
NAKUSI Romain
SACHA Bellier

1. Cas du drone

- **Problème**

La première étape de la question est de bien formaliser le problème. Nous avons donc choisis de représenter la ville par un graphe, où les rues sont les arêtes et les carrefours sont les sommets du graphe. Nous avons décidé de faire un graphe non-orienté car le drone peut circuler peu importe le sens, du fait qu'il vole, et qu'il n'a besoin de repérer la rue qu'une seule fois pour savoir si elle doit être déneigée.

Maintenant que nous avons notre représentation de la ville, nous nous sommes interrogés sur comment il fallait parcourir ce graphe par rapport aux attentes de la question. Cette première question doit répondre à deux demandes: repérer toutes les rues pour savoir lesquelles doivent être déneigées, ce qui est notre but premier, mais également rester efficace. En effet, pour des questions d'une part de budget, et d'autre part d'utilité (si le drone met tellement de temps à parcourir la ville que le niveau de neige a changé, cela perd son utilité).

- **Solution**

Maintenant que le problème a été bien formalisé, il nous reste à trouver une méthode de résolution pour répondre aux deux besoins. Pour celui de parcourir toutes les rues, il suffit simplement de parcourir toutes les arêtes du graphe. Quant au problème d'efficacité, il faut alors éviter de traverser les arêtes plusieurs fois, ou si c'est impossible, trouver un cycle le plus court possible lorsque c'est impossible de faire autrement. Notre problème est en réalité bien connu car il s'agit du Chinese Postman problem.

- **Cas théorique**

Pour le cas théorique nous avons donc pris le cas d'un graphe eulérien afin que le cas théorique ne soit pas trop complexe. Nous avons alors implémenté une fonction qui trouve un chemin eulérien qui permet donc au drone de parcourir toutes les arêtes une unique fois et ainsi de trouver un chemin optimal.

- **Cas réel**

Le cas réel s'est avéré être plus compliqué. En effet, nous avons tout d'abord fait un graphe du réseau routier avec osmnx: où chaque arête est étiquetée avec la longueur de la rue. Cependant, le graphe obtenu était énorme ce qui complique fortement l'analyse de celui-ci. Tout d'abord, une ville de la taille de Montréal ne peut pas être eulérien du fait du nombre colossal de routes et de carrefours, il serait totalement improbable qu'il soit eulérien.
Nous arrivons donc sur la partie compliquée du cas réel, la transformation du graphe en un graphe eulérien. Pour ce faire, nous avons envisagé dans un premier temps d'avoir recours à la fonction “Eulerize” de la bibliothèque networkx. Le graphe étant beaucoup trop important, la puissance de calcul demandée pour rendre ce graphe eulérien était alors impossible à réaliser pour nos ordinateurs. Nous avons alors dégagé trois stratégies pour pallier cet inconvénient.

La première était l'utilisation d'un super calculateur, qui serait réaliste du fait du budget de Montréal et qu'il faudrait calculer qu'une seule fois ce parcours puis le réaliser tous les jours en cas de besoin. Après réflexion, nous avons abandonné cette méthode car celle-ci demanderait de réeffectuer ce calcul plus régulièrement que prévu car un réseau routier peut souvent être modifié.

La deuxième était la découpe de Montréal en plusieurs quartiers pour ensuite les transformer en graphe eulérien. Bien que cette méthode nous semblait convenable nous avons décidé de nous orienter vers la troisième car la deuxième méthode semblait plus adaptée en cas d'utilisation de plusieurs drones ce qui augmenterait les dépenses de la ville. Enfin la troisième méthode, qui est celle sélectionnée, a été de simplifier le graphe en rassemblant certains sommets trop proches. En effet, nous supposons qu'un drone possède un champ de vision suffisamment élevé pour ne pas forcément devoir passer par toutes les rues ou carrefours. Par exemple le cas d'une impasse d'une dizaine de mètres de long le drone a-t-il vraiment besoin d'aller au bout puis de revenir ? Via cette méthode, nous avons donc consolidé les sommets à 50m de distance car un drone peut voir à 50m s'il faut déneiger ou pas, on réduit notre graphe de 19 000 sommets et 49 000 arêtes à un graphe à 3 000 sommets et 9 000 arêtes. Maintenant que le graphe a été simplifié, on peut aisément le rendre eulérien puis, nous n'avons plus qu'à chercher un chemin eulérien si on veut dégager un chemin précis ou bien calculer la somme des poids des arêtes pour connaître la distance.

2. Cas des déneigeuses

- **Problème**

Nous cherchons à présent à calculer le trajet minimal d'une déneigeuse. Cette question est très similaire à la précédente à la différence que contrairement au drone, la déneigeuse doit nettoyer les routes sur les deux voies lorsque l'on se trouve sur une route à double sens de circulation. Cette différence qui semble pourtant minime se traduit en réalité l'orientation du graphe. Nous devons donc procéder à la même opération mais cette fois-ci avec un graphe orienté. La bibliothèque networkx ne comportant pas de fonction pour rendre eurélien un graphe orienté nous avons implémenté la fonction.

- **Solution**

Pour ce faire, notre fonction va observer chaque sommet afin d'évaluer son degré entrant et sortant pour en déduire son degré. Si celui-ci est négatif (plus d'entrée que de sortie), on le met dans une liste de possibilités de sommets, s'il est positif, on le met dans une liste d'arêtes manquantes. Si son degré est nul, il n'y a rien à faire. On va parcourir nos sommets qui ont besoin d'une sortie et ajouter des chemins de ce sommet vers le sommet le plus proche à qui il manque des entrées tant que son degré n'est pas nul. Dès qu'un sommet à son degré nul, on le retire de la liste dans laquelle il se situe. On crée alors une nouvelle arête entre les deux sommets, qui possède pour poids le poids total du chemin les reliant. Pour définir le chemin le plus court, on utilise la fonction Dijkstra. Une fois que tous nos sommets ont un degré nul, et que l'on a vérifié que le graphe est fortement connexe, nous avons donc notre graphe eulérien, auquel il nous reste plus qu'à déterminer un circuit eulérien. Pour le trouver, nous appliquons l'algorithme de Hierholzer.

Le circuit garantit donc de passer au moins une fois par toutes les routes dans chacun des sens.

3. Coût du déneigement

Pour cette question, nous avons cherché à réduire les coûts de déneigement de la ville de Montréal au maximum possible. Dans ce but, nous avons pensé qu'il était préférable de réduire au maximum le nombre de déneigeuses utilisées. En effet l'entretien de celles-ci ainsi que remplacer celles qui deviennent obsolètes représente un budget non négligeable. Cependant, il faut que les habitants de Montréal ne soient pas impactés par une vitesse de déneigement trop lente. Nous avons alors élaboré une stratégie. Tout d'abord nous avons pensé que le moment qui peut être le plus dérangeant pour les habitants est le matin, quand après une nuit où il a neigé peuvent avoir des difficultés à circuler. Il faut donc que les déneigeuses passent le matin avant l'heure de pointe sur les routes. Toujours dans un but d'éviter les dépenses excessives nous décidons de commencer le travail des fonctionnaires à 6h30 pour éviter les heures de nuit. Nous avons donc prévu un créneau d' une heure entre 6h30 et 7h30 afin que les routes soient dégagées à pour l'heure de pointe. Sachant que notre chemin est d'une longueur totale de 7900 km, il faut que l'on trouve le nombre de déneigeuses permettant de remplir ce critère. Une déneigeuse pouvant circuler jusqu'à plus de 40km/h, nous avons décidé, une vitesse arbitraire de 20km/h pour différentes raisons, à savoir le fait qu'elles ne peuvent pas forcément circuler à leur vitesse maximale en ville, ou en fonction de la neige sur les routes. Enfin au prix actuel du carburant, limiter la vitesse n'est pas plus mal. Le temps étant de 1h, chaque déneigeuse devra balayer pour 20km en moyenne, ce qui nous donne donc un nombre optimal de déneigeuse de 395. Afin de pallier les potentiels retards de circulation, nous avons donc décidé que le mieux était d'envoyer 400 déneigeuses pour déblayer la ville.

4. Amélioration possible

Pour aller plus loin, nous pouvons remettre en question l'intérêt de l'utilisation d'un drone et surtout la manière dont il est utilisé actuellement. En effet, un drone peut voler largement au-dessus des maisons et il serait plus judicieux qu'il aille plus en hauteur et prenne des vues plus globales. D'autant plus, il est possible d'en avoir plusieurs en même temps et ainsi qu'ils n'aient même pas à se déplacer. L'utilisation de drones peut cependant être dangereuse en cas de panne, notamment si l'un d'entre eux tombe directement sur la population. On pourrait alors conseiller l'utilisation de satellites qui sont aujourd'hui largement assez précis pour permettre une analyse et savoir en globalité les endroits à déneiger en priorité. Cela ferait gagner pas mal de temps et des coûts en moins, les satellites étant déjà sur place et d'après nos recherches, il y en a suffisamment pour récupérer une photo chaque jour.