Rapport de recherche opérationnelle

emeric.du-gardin jocelyn.grannec theo.ripoll
arnaud.notter pierre-louis.garneronne

Juin 2022

1 Le parcours du drone

1.1 Formalisation du problème

Nous avons commencé par envisager le problème de façon formelle. À première vue, il s'agit d'un problème représenté par un graphe, dont les nœuds sont les intersections de rue et le poids positif des arêtes la longueur de la rue. Pour le drone, nous pouvons supposer le graphe non-orienté.
L'objectif étant de parcourir toutes les arrêtes du graphe en faisant la plus petite distance, on peut relier cela a chercher un cycle eulérien dans un graphe.
Finalement, on ne peut supposer que le graphe d'une ville soit eulérien par chance. Il faudra donc trouver une méthode pour rendre le graphe eulérien, soit un graphe fortement connexe dont le dégrée de tous les sommets est pair.
On retiendra la complexité temporelle comme unité de mesure pour cette partie.

1.2 Solution basique

Une première solution est de relier les nœuds de degré impairs à leur voisin le plus proche en termes de poids. Pour cela, on cherche tous les nœuds de degré impair en une list Odd. Pour chaque nœud A de cette liste, on effectue Bellman-Ford avec comme source A. On cherche alors le nœud de degré impair le plus proche, et on rajoute au graphe l'arête entre les deux, ainsi que le poids représentant la plus courte distance entre les deux. On supprime les nœuds utilisés à chaque itération.
Bien que cette solution soit d'une complexité raisonnable, O((P/2)NE) avec P le nombre de sommets impair, N le nombre de sommets et E le nombre d'arêtes, on n'a ici qu'une solution viable, qui n'est pas le point d'arrêt de notre recherche. Cependant, on notera que la complexité modérée de cette solution permet le passage à des échelles de grandeurs supérieur, avec des graphes de dizaine de milliers de nœuds.

1.3 Solution optimale

L'approche de la solution optimale la plus simple débute de la même manière. Après avoir listé les sommets de degré pair (Odd), on effectue Floyd-Warshall pour récupérer la matrice de successeur et celle de la plus courte distance entre deux sommets. On génère ensuite l'ensemble des combinaisons à deux éléments parmi la liste Odd. On calcule la taille totale des liaisons crée pour chaque combinaison, et on choisi celle avec la plus petite distance. On fini par ajouter au graphe les chemins empruntés par la combinaison choisie.
Bien qu'optimale, cette approche repose sur une génération de l'ensemble des permutations. On se retrouve alors une complexité en O(N!) avec N le nombre de nœuds impair. Cette complexité contraint donc la solution en termes de sacalabilité

1.4 Conclusion et optimisations

En l'état actuel de notre avancée, il est inenvisageable d'utiliser notre solution optimisée pour résoudre le problème de drone. Cependant, une solution d'optimisation serait d'utiliser l'algorithme hongrois pour baisser la complexité de la recherche des paires. On aurait alors une solution en $O(N^4)$.

2 Le parcour des dammeuses

2.1 Formalisatin du problème

Pour ce problème nous pouvons aussi représenter les rues et les intersections avec les nœuds et les arcs d'un graph. Cette fois ci le graph est orienté car les routes ne peuvent pas être emprunter dans le sens inverse de la circulation si nous déneigeons pendant que celle-ci est active. Chaque arc est accompagné d'un poids qui représente la distance entre les deux intersections. Si une route est en double sens, elle est représentée par deux arêtes respectivement orienté dans un sens et dans l'autres. Nous considérons que les dameuses doivent revenir à leur lieu de départ après leur parcours. Pour faciliter ce processus chaque trajet sera représenté par un cycle eulérien. Pour trouver ce type de cycle nous devons avoir un graphe eulérien. Un graphe orienté est considéré eulérien si et seulement s'il est fortement connexe et aucun nœud est déséquilibré. Un nœud est déséquilibré quand le nombre d'arrête entrante est différent du nombre d'arrêtes sortantes. Ce type de problèmes est plus communément appelé le problème du postier chinois orientés. nous choisirons pour cette partie la distance parcourue par une deneigeuse comme metric pour l'optimisation.

2.2 Transformer le graphe en un graphe eulérien si nécessaire

Dans le cadre de ce problème nous devions être capable de pouvoir trouver un trajet pour la déneigeuse même si le graphe produit par les routes n'est pas eulérien. Pour résoudre ce souci nous devions alors modifier ce graphe pour le rendre eulérien. Une solution initiale et basique serait d'ajouter des arcs entre les sommets qui sont déséquilibrés. En effet, une condition pour que le graphe possède un circuit eulérien est que tous ses sommets soit de degré équilibré. Pour cela, nous listons tous les sommets déséquilibrés négativement et les sommets déséquilibrés positivement, pour chaque combinaisons possibles, on calcule la distance entre les paires puis on choisis celui qui a la plus petite distance. On ajoute alors toutes les paires de cette combinaison au graphe. Cette solution fonctionne mais n'est pas optimale sur le coût des arrêtes. Nous pouvons alors modéliser notre problème sous la forme d'un problème de transports afin de le résoudre plus efficacement.L'offre représenté par les nœuds ayant besoin d'arrêtes sortantes, la demande comme les nœuds ayant besoin d'arrêtes entrantes et le coût est représenté par l'addition des coûts du chemin entre deux nœuds. Pour résoudre ce problème nous avons utilisé l'algorithme de Ballas-Hammer qui permet de trouver une solution optimale pour les problèmes de transports.

2.3 Trouver le circuit eulérien

Une fois que nous possédons un graphe eulérien, nous devons trouver un cycle eulérien. Pour ce faire, nous avons utilisé l'algorithme de Hierholzer qui à une complexité temporelle O(E) où E est le nombre d'arêtes.

3 Partie pratique dans Montréal

Dans cette partie, nous allons mettre en pratique tout ce que nous avons vu précédemment. Nous allons utiliser un graphe à plus grande échelle, celui de la ville de Montréal. Pour cela, nous avons utilisé la librairie osmnx qui nous a permis de récupérer des graphes provenant de carte. Le site https://www.openstreetmap.org/nous a permis de trouver les différents noms a appelé pour load nos différents graphes.

Notre algorithme de résolution prend en paramètre les différents quartiers de Montréal. Depuis cette liste de quartiers, nous récupérons leur graphe avec la fonction 'graph from place' de osmnx. Nous devons récupérer deux graphes : celui pour les zones piétonnes et pour les voies de véhicule. Ensuite, nous transformons ces graphes orientés en graphe non-orientés. Cette action était obligatoire afin de pouvoir eulériser nos graphes avec les fonctions fournis par networkx. Avec notre nouveau graphe, nous pouvons alors le rendre Euler et donc ainsi trouver un circuit eulérien dedans. Ce circuit eulérien va nous permettre de définir le chemin que le drone devra parcourir pour analyser la situation. Nous n'avons pas pu implémenter, mais avec les résultats que le drone donne nous devons changer le graphe en retirant les arêtes sans neige. Avec ce circuit eulérien et les outils à notre disposition, nous pouvons connaître la longueur du circuit eulérien et ainsi prévoir le temps de trajet de notre drone en fonction de sa vitesse. La prochaine étape est de diviser nos engins de déneigement entre les voies piétonnes et les routes. Nous attribuons un nombre d'engins proportionnels au taux de chaque type de voies dans la rue. S'il y a 60% de routes alors il y aura 60% d'engins pour les routes. Cela étant fait, nous faisons la même en fonction de la longueur de routes de nos quartiers. Une fois, cela fait, nous avons pour chaque quartier un nombre de machines proportionnel à la charge de travail à effectuer. À cette étape, nous allons réutiliser le circuit eulérien trouvé par le drone. Nous allons diviser ce circuit eulérien en n chemins de nœuds égaux ; n'est pas le nombre de déneigeuse pour le quartier. Il nous suffit ensuite plus que de relier ce bout du cycle au point de départ et nous avons ainsi tous les itinéraires pour tous nos engins de déneigement. Nous avons chargé une entreprise privée pour avoir les engins et nous payons leur employé 20€/h, nous en avons 2200 employés qui nous coûte 4 708 000€/an + achat des 22 drones (1 par quartier) qui revient à 176 000€. Nous arrivons à une estimation des coûts de 4 884 000€