EROhic
Les héros de la conduite sans alcool !

Jeremy Croiset, Mexane Delcroix,
Antoine Ducros, Néphélie Lambrinidis,
Cécile Philippo

Mai 2021

1 Introduction

EROhic, le groupe qui conduit des déneigeuses comme personne, en toute sécurité, non-alcoolisé ! Les héros du déneigement sont heureux de vous présenter leur projet ! Pour répondre à la problématique de Montréal, nous avons passé du temps sur la partie réflexion de ce projet. La question de l'utilisation d'un cycle eulérien a été abordée mais nous avons préféré nous intéresser à l'*ant optimization*.

2 *Ant Colony Optimization* appliquée au *Travelling Salesman Problem* : solution abandonnée

2.1 Fonctionnement

Le principe de l'*Ant Colony Optimization* est simple. C'est une variante du *Traveling Salesman Problem* (Problème du voyageur de commerce), un problème d'optimisation qui a pour but de déterminer le plus court chemin entre plusieures villes. L'*Ant Colony Optimization* est comme son nom l'indique, basé sur le comportement des fourmis. On programme une série de fourmis qui laissent un certain type de phéromones à l'aller jusqu'à trouver une source de nourriture, ou dans notre utilisation, l'un des sommets du graphes. Ensuite, la "fourmi" a pour rôle de suivre le chemin exact en sens inverse en utilisant un autre type de phéromones. Ce sont ces traces de phéromones qui déterminent les directions de départ des prochaines fourmis. Au fur et à mesure du temps qui passe, les traces de phéromones s'estompent, encourageant les fourmis à chercher d'autres chemins. Plus une trace est forte, plus la probabilité que la fourmi suive cette trace sera grande. De ce fait, si la trace disparaît, les fourmis ne se dirigeront plus vers l'ancienne source de nourriture avec la même intensité !

2.2 Solution abandonnée

Comme donné dans l'intitulé de l'exercice, le problème du voyageur de commerce doit, à terme, donner le chemin le plus court entre plusieurs sommets. Quel problème rencontre-t-on alors ? Eh bien, nous regardons les chemins les plus courts entre les sommets, et non pas un moyen de passer par un maximum de ces chemins. Pourtant, le problème que l'on nous demande de résoudre nécessite que l'on s'intéresse au fait de passer un maximum de routes pour les déneiger en un minimum de temps.

3 Solution finale : *Chinese Postman Problem* et *Minimum weight bipartite matching*

Pour la solution finale nous avons employé deux algorithmes, le *Chinese Postman Problem* pour le cas théorique et le *Minimum weight bipartite matching* pour le cas pratique.
Ces deux algorithmes ont pour même objectif de rendre les graphes Eulérien. Le but étant de trouver un cycle Eulérien, c'est-à-dire un chemin passant par toutes les arêtes une seule et unique fois.
Un tel chemin n'existant que si tous les sommets sont de degré pair dans le cas non-dirigé ou ayant un même nombre d'arêtes entrantes que sortantes dans le cas dirigé, il nécessite quelques adaptations pour notre cas, dont la possibilité de passer plusieurs fois sur certaines arêtes (routes).
Pour ce faire, nous dupliquons certaines routes de manière à rendre le graphe de la ville Eulérien et pouvoir y appliquer l'algorithme de recherche de cycle Eulérien.

3.1 Cas théorique (Drones)

Dans le cas théorique, les contraintes de temps ne sont guère importantes : cela rend notre algorithme indépendant de sa complexité temporelle, et donc plus simple à implémenter.
Dans le cas des drones, le graphe peut-être considéré comme non dirigé puisque ceux-ci n'ont besoin de parcourir les routes qu'une seule fois afin d'en extraire des informations.
Pour résoudre le *Chinese Postman Problem*, nous avons utilisé l'algorithme de *Blossom*, cependant celui-ci a ses limitations à grande échelle. En effet, sur un graphe tel que celui de la ville de Montréal possédant 11816 *nodes* impaires, cela implique 69 803 020 combinaisons, et ce pour une complexité de n^3 (donc 3.4011253e+23) dans le cas de *Blossom*, ce qui rend le temps d'exécution irréalisable.

3.2 Cas pratique (Déneigeuses)

Le but du cas pratique est de mettre à l'échelle notre solution en prenant en compte une contrainte de temps, et d'argent.
Dans le cas pratique, donc les déneigeuses, celles-ci ont besoin de parcourir les routes dans les deux sens et de respecter le code de la route, avec notamment le problème des routes à sens unique, qui transforme notre graphe précédemment non-dirigé en un graphe dirigé. Pour cela nous avons utilisé la méthode *Minimum weight bipartite matching* un algorithme lié au problème de flot de coût minimum. L'idée est la même, obtenir un graphe Eulérien afin de trouver un circuit Eulérien.

La résolution du problème du flot de coût minimal se déroule comme suit :

- trouver les noeuds qui ont un surplus d'arêtes entrantes/sortantes
- créer un graphe biparti reliant chaque noeud ayant un surplus d'arêtes entrantes avec chaque noeud ayant un surplus d'arêtes sortantes
- calculer le couplage minimal pondéré sur le graphe précédent
- construire un graphe eulérien en dupliquant toutes les arêtes le long des chemins entre les meilleures paires du couplage
- obtenir le cycle eulérien

Cet algorithme permet donc de convertir notre problème d'origine en un problème de flux, la complexité décroît grandement, ce qui rend cette solution bien plus applicable au cas pratique que la solution que nous avions trouvé pour le cas théorique !

4 Conclusion

À l'aube d'un monde où les saisons changent et où la météo devient de plus en plus imprévisible, il devient important de prévoir des solutions à toutes épreuves afin de pallier à d'éventuels problème. En l'occurrence, EROhic s'intéresse aux solutions de déneigement rapides et efficaces pour éviter de paralyser une ville entière, telle que Montréal.