EPITA LYON
GROUPE N°17

Rapport de projet ERO1

Élèves :
Thomas FLORION
Chloé VIDAL
Geoffroy JOUANIN
Lisa BOURLIER
Matéo BACHELIER

Enseignants :
Loïc BLET
Laurence DENNEULIN

07 Juin 2024

Table des matières

1 Introduction

La municipalité de Montréal souhaite optimiser ses opérations de déneigement de la ville, qui s'élèvent actuellement à un total de 165M$. Les opérations de déblaiement ont lieu dans le cas où la ville fait face à des chutes de neige de 2,5cm à 15cm. Il s'agit de déblayer la neige des réseaux pédestre et routier de la ville. Notre objectif principal étant de proposer un plan de déneigement à la mairie de Montréal permettant de minimiser les coûts tout en gardant une efficacité optimale. Nous disposons d'un drone pour effectuer une première reconnaissance de la ville. Il permettra par la suite de définir les quartiers à déneiger en priorité. De plus, 2000 déneigeuses sont mise à disposition, dont les parcours restent à définir. Seuls les quartiers détectés par le drone comme enneigés doivent être traité.

2 Présentation des données

Pour optimiser les opérations de déneigement à Montréal, nous avons accès aux données suivantes, fournies par la municipalité.

Coûts / Type de véhicule	Drone	Déneigeuse Type I	Déneigeuse Type II
Coût fixe (par jour)	100	500	800
Coût kilométrique	0.01	1.1	1.3
Coût horaire pour les 8 premières heures	/	1.1	1.3
Coût horaire au-delà des 8 premières heures	/	1.3	1.5
Vitesse moyenne (km/h)	36	10	20

Données additionnelles :
— Superficie et étendue des opérations : Les opérations de déneigement couvrent environ 6400 km de routes.
— Nombre total de déneigeuses disponibles : 2000 appareils.

Objectifs et contraintes :
— Minimiser le coût des opérations de déneigement tout en assurant un déneigement complet.
— Proposer un trajet optimal des véhicules en respectant le code de la route.
— Utiliser un drone pour une reconnaissance initiale des niveaux de neige, permettant de cibler les zones nécessitant un déblaiement prioritaire.

3 Résolution du problème : Partie Drone

Dans un premier temps, dans le cadre de ce projet, nous avons dû organiser le parcours d'un drone à travers Montréal pour qu'il puisse analyser les niveaux de neiges. Celui-ci doit effectuer un examen du réseau routier pour pouvoir apporter une analyse suffisamment fine. Nos différentes variables dans ce problème sont :
— Temps mis par le drone (celui-ci doit être relativement bas dans un contexte de déneigement d'urgence)
— Coût de l'opération en €
— Temps d'exécution du programme de notre solution

3.1 Première itération

Pour notre première analyse de l'utilisation du drone, nous avons choisi de lui faire parcourir la totalité de Montréal afin de détecter quelle zone déneiger. La ville est représentée sous forme de graphe non-orienté car le drone n'a pas besoin de respecter le code de la route. Le but du drone est de passer par tous les sommets du graphe, représentant les intersections, afin de collecter un maximum d'information et de savoir si un quartier doit être déneigé ou non. Pour réduire le temps et la distance parcourue, on part du principe qu'au niveau d'une intersection il peut estimer le niveau de neige de chacune des arrêtes adjacentes à lui, cela nous permet de ne pas avoir à passer sur toutes les arrêtes tout en parcourant tout le graphe.

Pour cela, on a assimilé notre problème à celui du postier chinois. Ce problème consiste à trouver un plus court chemin dans un graphe connexe non orienté qui passe au moins une fois par chaque arête et revient à son point de départ. Dans ce problème, le cas optimal revient à parcourir tout notre graphe en ne passant qu'une fois par chaque arête, ce qui correspond à trouver un cycle eulérien. Un tel chemin existe si et seulement si chaque sommet du graphe est de degré pair. Dans notre cas, notre graphe n'est pas eulérien. D'un point de vue algorithmique, on duplique les arêtes par lesquelles on passe deux fois, en minimisant la longueur totale des arêtes ajoutées. On obtient une solution du problème initial en cherchant un circuit eulérien dans le graphe complété.

Nous avons modélisé cet algorithme dans snowPlowV1.py. La première étape a été de récupérer la modélisation de Montréal à l'aide de la bibliothèque osmnx, puis de rendre le graphe non-orienté et eulérien. Nous avons recréé une fonction qui permet de trouver et d'ajouter les arêtes les plus courtes entre deux nœuds de degré impaire. Nous avons aussi utilisé la bibliothèque networkx qui nous permet de faire des opérations de traitement sur notre graphe. Ensuite, on applique l'algorithme du postier chinois en partant du centre de la ville. Il va alors chercher à parcourir le premier circuit eulérien qu'il pourra trouver, et recommencer jusqu'à parcourir tout le graphe (voir image).

Nous l'avons implémenté et observé les retours de notre modélisation : pour un parcours complet de la ville, notre drone parcourt 4898 km, soit près de 3 jours de trajet réel en supposant sa vitesse constante à 60km/h. De plus, pour optimiser au mieux le trajet du drone, certaines routes étaient coupées pour limiter la distance, en passant au-dessus de certains bâtiments, ce qui ne respecte pas les directives de la ville de Montréal. De plus le temps d'exécution de notre solution est de près d'une heure ce qui est trop long pour être viable. Cependant, il est difficile de faire mieux avec cette modélisation étant donné que la distance totale des routes de Montréal s'élève à 6394.17km et que nos contraintes nous imposaient un seul drone pour cette étape.

3.2 Seconde itération

Dans un second temps, nous avons cherché d'autres solutions afin d'améliorer notre première modélisation. Étant donné que le but du drone est de définir quel quartier doit être déneigé, nous avons envisagé plusieurs solutions pour réduire la distance que parcourt le drone tout en gardant une estimation pertinente du niveau de neige de chaque quartier. De plus, le but de cette deuxième tentative de modélisation est de garder un parcours du drone qui colle le plus à la réalité, mais assez simplifié pour réduire grandement la distance parcourue et éviter de passer par-dessus les bâtiments. Nous avons imaginé de nombreuses solution pour de nouvelles modélisations, et sommes partie sur notre dernière idée, faire passer notre drone au centre de chaque quartier tout en faisant une estimation du niveau d'enneigement. L'avantage de cette dernière solution était surtout utile car il permettait de ne pas avoir à passer au-dessus des bâtiments et de ne pas mettre plus de 3 jours pour tout parcourir. Finalement c'est cette dernière qui fut retenue car nous sommes partis du principe que si l'on traverse un quartier de part en part nous devrions avoir une idée globale de l'enneigement du quartier ce qui serait suffisant pour prendre une décision pour les déneigeuses.

Pour cela, nous avons opté pour une solution se rapprochant du problème du voyageur de commerce. Bien que ce problème soit NP-complet et, par conséquent, difficile à résoudre en un temps raisonnable, nous avons estimé que la complexité algorithmique associée à la modélisation de notre problème ainsi que le nombre de nœuds demeure tout à fait acceptable. Nous nous sommes alors inspirés du mode de raisonnement de l'algorithme de Christofides. Il permet de s'approcher d'une solution optimale pour un tel problème et est proche de ce que nous voulons modéliser.

Pour la modélisation, nous avons placé des points situés au centre géographique des 19 quartiers, et fait passer notre drone sur les intersections les plus proches de ces points. On applique ensuite notre algorithme décrit ci-dessus. L'avantage de se limiter à quelques nœuds est de réduire énormément la distance à parcourir pour obtenir les informations que l'on souhaite. Un dernier point est que nous avons choisi de modéliser notre drone comme allant à 36km/h de moyenne en se basant sur un système déjà existant de drones équipés de capteurs LIDAR permettant de faire de la cartographie.

Les retours de notre modélisation sont les suivants : pour un parcours par quartier, notre drone parcourt désormais 102 km en 2 heures et 50 minutes de trajet réel en supposant qu'il vole à 36km/h cette fois-ci, ce qui est bien plus réaliste. Un autre avantage de cette méthode est qu'elle s'exécute en moins de 2 minutes (voir image). Cependant, il reste clair qu'une meilleure méthode pour la prise d'information serait d'augmenter le nombre de drones en circulation cependant les contraintes du sujet nous imposaient de résoudre cette problématique à l'aide d'un seul appareil.

4 Résolution du problème : Partie Déneigeuses

Le but des déneigeuses est de parcourir complètement les 5 quartiers qui ont été détectés par le drone comme enneigés. Les quartiers sont les suivants : Outremont, Verdun, Anjou, Rivière-des-Prairies-Pointe-aux-Trembles, Le Plateau-Mont-Royal. Pour cela, on représente Montréal sous forme de graphe, mais orienté pour respecter le sens des routes. De plus, les déneigeuses n'ont besoin que d'un seul passage pour les voies à double sens. A l'issue de chacun de nos scénarios, il faudra être capable de proposer un parcours pour chaque déneigeuse utilisée ainsi qu'un détail du coût et du temps réel de déblaiement de notre solution.

Pour nous donner un ordre d'idée, nous avons décidé de tracer la courbe de l'évolution du coût total en fonction du temps de déneigement. En effet, nous avons alors remarqué que pour déneiger à moindre coût, le mieux serait de ne passer qu'avec une déneigeuse de type 2, mais le temps total serait alors de 68h, ce qui n'est pas réalisable. Le but ici est de permettre de placer plusieurs déneigeuses afin de trouver un équilibre entre le temps des opérations et les coûts. Nous avons donc codé un petit script qui trouve la combinaison de déneigeuses la plus optimisée pour déneiger l'entièreté de ces 5 quartiers en un temps maximum donné et pour le moins cher possible (voir image).

Avec ces informations, l'idée de déneiger les quartier en 5h était un compromis acceptable entre le temps et le coût des opérations. Avec cette configuration, nous utilisons en théorie 2 déneigeuses de type 1 et 13 déneigeuses de type 2. Ce qui nous donne :

$D1 \times (500 \times jour + 1.1 \times km + 1.1 \times h) + D2 \times (800 \times jour + 1.3 \times km + 1.3 \times h)$
$= 2 \times (500 + 1.1 \times 1357.9 + 1.1 \times 5) + 13 \times (800 \times 1 + 1.3 \times 1357.9 + 1.3 \times 5) = 13295.50$ €

4.1 Première itération

Pour une première implémentation, nous avons cherché avant tout à diviser notre graphe en sous-graphe, afin de répartir au mieux nos déneigeuses. Cependant, il n'existe pas de solution optimisée pour découper un graphe en sous graphe parfaitement réparti. Nous avons alors réutilisé l'algorithme du postier chinois.

Nous avons d'abord défini un nombre de points, qui représentent chacun le centre de la zone à parcourir d'une déneigeuse. À partir de ce point, nous utilisons l'algorithme du postier chinois pour tracer le trajet optimal d'une déneigeuse, jusqu'à atteindre la zone d'une autre déneigeuse. Chaque partie représente alors un sous graphe qui pourra alors être parcouru par un véhicule (voir image).

Une fois cette solution mise en place, nous l'avons testé sur les cinq quartiers à la fois. Nous avons pu constater que ce modèle, même s'il permet de bien découper visuellement nos quartiers, possède beaucoup de limites. Tout d'abord, il implique de faire apparaître nos déneigeuses au centre de leur zone à déneiger, sans prendre en compte de point de départ au niveau d'un local commun, et elles ne reviennent pas non plus à leur point de départ. C'est dans le but de résoudre ce problème de réalisme que nous avons créer la deuxième modélisation.

4.2 Seconde itération

Après les problèmes de réalisme sur la première représentation nous avons cherché s'il était possible de représenter l'aller-retour que doivent faire les déneigeuses entre leur secteur attribué et l'entrepôt duquel elles vont partir. Pour cela il suffit de trouver un moyen de parcourir tout notre graphe en envoyant nos déneigeuses dans des directions différentes.

Ce type de problème est connu sous le nom de : Vehicle Routing problem. Après avoir étudié la problématique, nous avons découvert un algorithme de résolution correspondant à nos objectifs : l'algorithme de Clarke-Wright Savings. (CWS - plus d'infos ici). La problématique reste encore très peu étudiée donc il a fallu redoubler d'ingéniosité pour adapter les pseudo-algorithmes existants.

Ainsi nous avons commencé à modéliser notre problème afin de le rendre résoluble par le CWS. Tout d'abord la première étape serait de trouver un point où mettre notre "local à déneigeuses" d'où elles doivent sortir. Ensuite l'idée est de parcourir tout notre graphe a l'aide du CWS en lui spécifiant la distance maximale que chaque déneigeuse peut parcourir afin qu'il assigne les zones de déneigement à chacune d'entre elles. La seule subtilité réside dans le fait que l'algorithme ne fonctionne que sur une composante connexe, il faudrait alors itérer cet algorithme sur chacun des quartiers afin de ne pas avoir de problèmes là dessus.

Les points centraux ont été déterminés grâce à la même méthode que celle utilisée lors de la deuxième itération des drones et sont les points desquels les déneigeuses sont déployées (ainsi nous avons 5 locaux, 1 par quartier à déneiger). Notre algorithme dessine alors les routes que chaque déneigeuse doit effectuer afin de faire une solution la plus optimisée possible, certaines routes sont empruntées par plusieurs déneigeuses pour qu'elles puissent aller jusqu'à leur secteur assigné, puis elles reviennent à leur point de départ et le tout afin que chaque déneigeuse fasse un maximum de kilomètres de déneigement. On obtient avec cette configuration à 15 déneigeuses une distance totale de 1299 km car on a exclu les impasses de notre étude et les voies à double sens qui ne sont parcourues qu'une seule fois (voir image). Le prix total de l'opération s'élève à 13 462 € et le temps d'exécution est d'environ 2 minutes. C'est un modèle très équilibré qui nous permet de modéliser toutes les solutions de vitesse/de coûts possibles. C'est un modèle très proche de l'idéal théorique que nous avons calculé au début de notre étude, ce qui est signe d'une solution adéquate.

4.3 Troisième itération

Bien que la deuxième implémentation soit réaliste et équilibrée, nous avons voulu tenter un dernier scénario optimisé. Celui-ci apporte en plus un pré-traitement du réseau routier en supprimant les impasses du graphe à parcourir. Pour l'algorithme, nous avons décider d'utiliser un BFS modifié afin d'accepter plusieurs entrées.

Pour modéliser notre problème, nous avons choisi de traiter la carte, quartier par quartier, afin d'optimiser le nombre de déneigeuses. En premier lieu, nous déterminons les meilleurs points de départ des déneigeuses avec des mesures de centralité. Ensuite tout le graphe est parcouru, les arêtes sont sauvegardées avec leur déneigeuse et leur longueur d'arc. Grâce à ces données, nous pouvons tracer le chemin des déneigeuses en couleur, et calculer précisément la distance qu'elles ont parcouru (voir image).

Après avoir parcouru tous les quartiers, on obtient le coût final de 13 773€ en utilisant 15 déneigeuses. Ce modèle permet donc d'obtenir un coût réduit comparable au deuxième scénario, seulement il présente plusieurs défauts. Tout d'abord, il ne parcourt pas la totalité des routes vu qu'il évite les impasses, et il présente le même problème que la version 1 en faisant apparaître des déneigeuses au milieu des quartiers.

5 Conclusion

Finalement nous conseillons d'utiliser la solution 2 pour le drone et la solution 2 pour les déneigeuses car ce sont les plus optimisées. De plus elles permettent plus d'adaptation en fonction des besoins. Pour exécuter ces solutions nous vous invitons à faire un ./start.sh à la racine du repository du projet et de suivre les instructions affichées à l'écran. Enfin ce projet nous a aussi appris l'importance d'une formalisation des modèles et qu'il est aussi important d'avoir en tête les objectifs d'un projet et nous sommes satisfaits du résultat.

6 Images annexes

Figure 1 – Première solution du drone

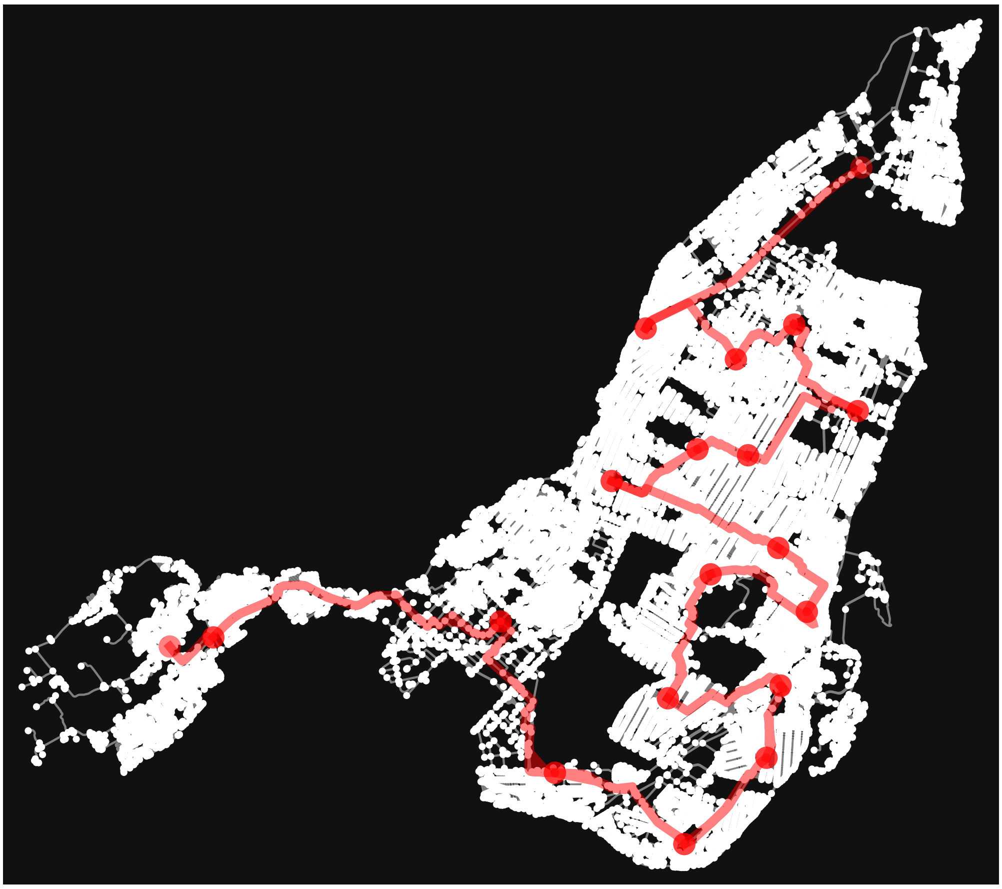

FIGURE 2 – Deuxième solution du drone

```
[1:27pm]-:~/ERO1-Lyon17-
> python3 drone/droneV2.py
Downloading the street network for Montreal...
Downloaded the street network for Montreal.
Total Distance Traveled (kilometers): 102.112634
Price of this operation (in euros): 101.02112634
Execution Time: 00hrs 02mins 01secs
Plotting the Montreal map with the shortest path between each pair of nodes in red...
```

FIGURE 3 – Affichage textuel de la deuxième solution du drone

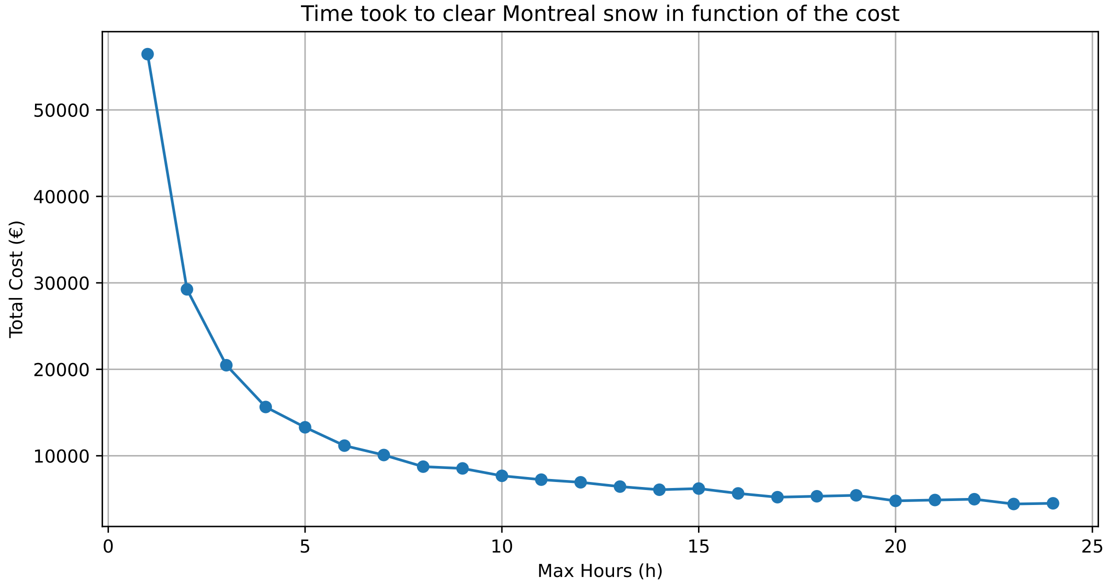

FIGURE 4 – Courbe des coûts minimum théoriques

```
Max hours: 3
Combinaison optimale de déneigeuses : Type I = 2, Type II = 22
Total cost : 20474.40 €
Max hours: 4
Combinaison optimale de déneigeuses : Type I = 2, Type II = 16
Total cost : 15644.00 €
Max hours: 5
Combinaison optimale de déneigeuses : Type I = 2, Type II = 13
Total cost : 13295.50 €
Max hours: 6
Combinaison optimale de déneigeuses : Type I = 1, Type II = 11
Total cost : 11174.40 €
Max hours: 7
Combinaison optimale de déneigeuses : Type I = 2, Type II = 9
Total cost : 10089.30 €
Max hours: 8
Combinaison optimale de déneigeuses : Type I = 1, Type II = 8
Total cost : 8744.00 €
Max hours: 9
Combinaison optimale de déneigeuses : Type I = 2, Type II = 7
Total cost : 8539.50 €
Max hours: 10
Combinaison optimale de déneigeuses : Type I = 2, Type II = 6
Total cost : 7683.20 €
Max hours: 11
Combinaison optimale de déneigeuses : Type I = 1, Type II = 6
Total cost : 7239.10 €
Max hours: 12
Combinaison optimale de déneigeuses : Type I = 2, Type II = 5
Total cost : 6934.00 €
Max hours: 13
Combinaison optimale de déneigeuses : Type I = 1, Type II = 5
Total cost : 6437.80 €
Max hours: 14
Combinaison optimale de déneigeuses : Type I = 2, Type II = 4
Total cost : 6074.80 €
Max hours: 15
Combinaison optimale de déneigeuses : Type I = 2, Type II = 4
Total cost : 6209.40 €
Max hours: 16
Combinaison optimale de déneigeuses : Type I = 1, Type II = 4
Total cost : 5648.80 €
Max hours: 17
Combinaison optimale de déneigeuses : Type I = 2, Type II = 3
Total cost : 5212.70 €
Max hours: 18
Combinaison optimale de déneigeuses : Type I = 2, Type II = 3
Total cost : 5319.80 €
Max hours: 19
Combinaison optimale de déneigeuses : Type I = 2, Type II = 3
Total cost : 5426.90 €
Max hours: 20
Combinaison optimale de déneigeuses : Type I = 1, Type II = 3
Total cost : 4789.60 €
Execution Time: 00hrs 00mins 06secs
```

FIGURE 5 – Affichage textuel de la courbe des coûts

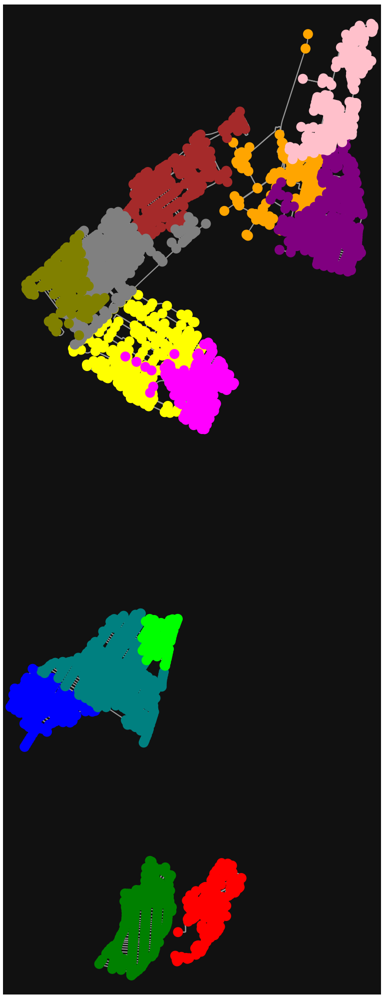

Figure 6 – Première solution des déneigeuses

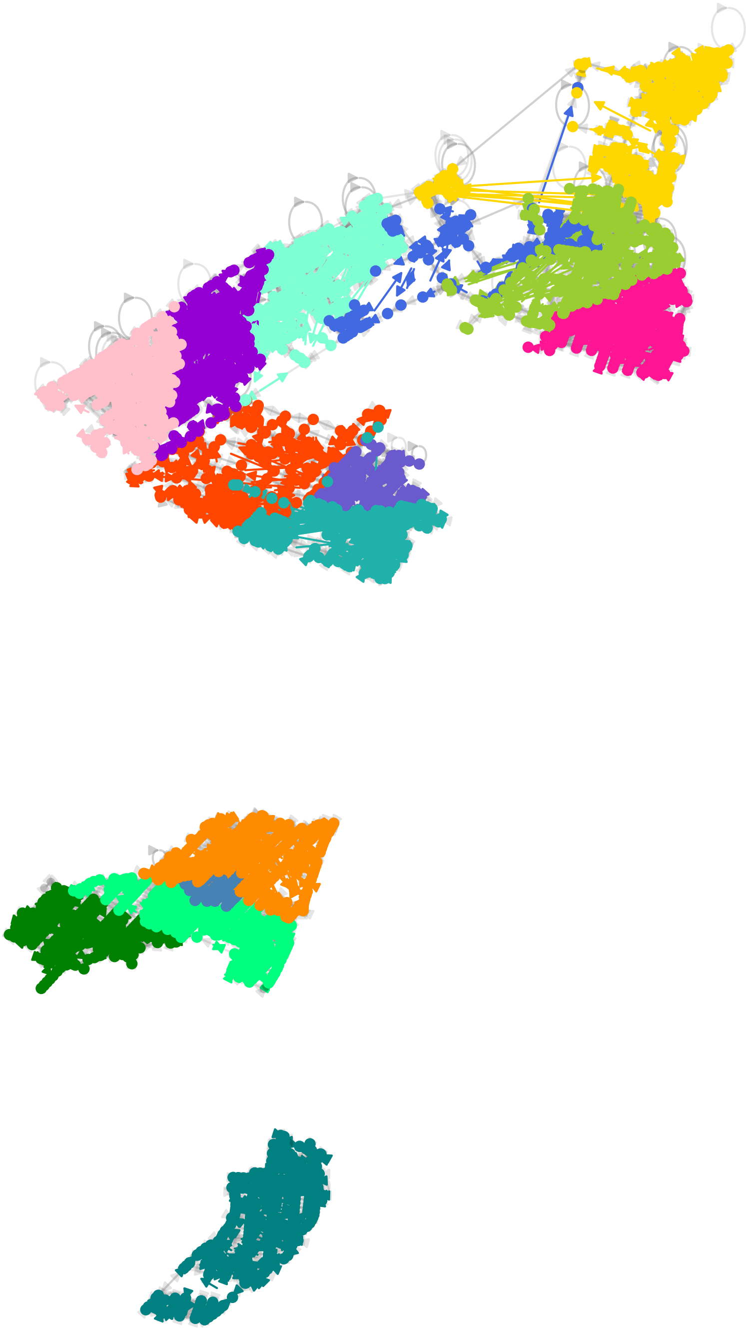

Figure 7 – Deuxième solution des déneigeuses

```
[8:29pm]-:~/ERO1-Lyon17-
> python3 initialDeneigeuse/snowPlowCWSAlgo.py
Processing Outremont, Montreal, Canada...
Sector distance 74.60794500000006 km
Total cost of the operation in this sector 590.2756134500002 €
Completed Outremont, Montreal, Canada.

Processing Verdun, Montreal, Canada...
Sector distance 101.59089299999992 km
Total cost of the operation in this sector 938.6715689449999 €
Completed Verdun, Montreal, Canada.

Processing Anjou, Montreal, Canada...
Sector distance 235.37074899999953 km
Total cost of the operation in this sector 2721.2810723849993 €
Completed Anjou, Montreal, Canada.

Processing Rivière-des-prairies-pointe-aux-trembles, Montreal, Canada...
Sector distance 718.3041329999982 km
Total cost of the operation in this sector 6580.4851415449975 €
Completed Rivière-des-prairies-pointe-aux-trembles, Montreal, Canada.

Processing Le Plateau-Mont-Royal, Montreal, Canada...
Sector distance 169.30479300000027 km
Total cost of the operation in this sector 2631.1010424450005 €
Completed Le Plateau-Mont-Royal, Montreal, Canada.

Total distance 1299.178512999998 km
Total cost of the operation 13461.814438769998 €
Execution Time: 00hrs 02mins 40secs
Nb colors : 15
```

Figure 8 – Affichage de la deuxième solution des déneigeuses

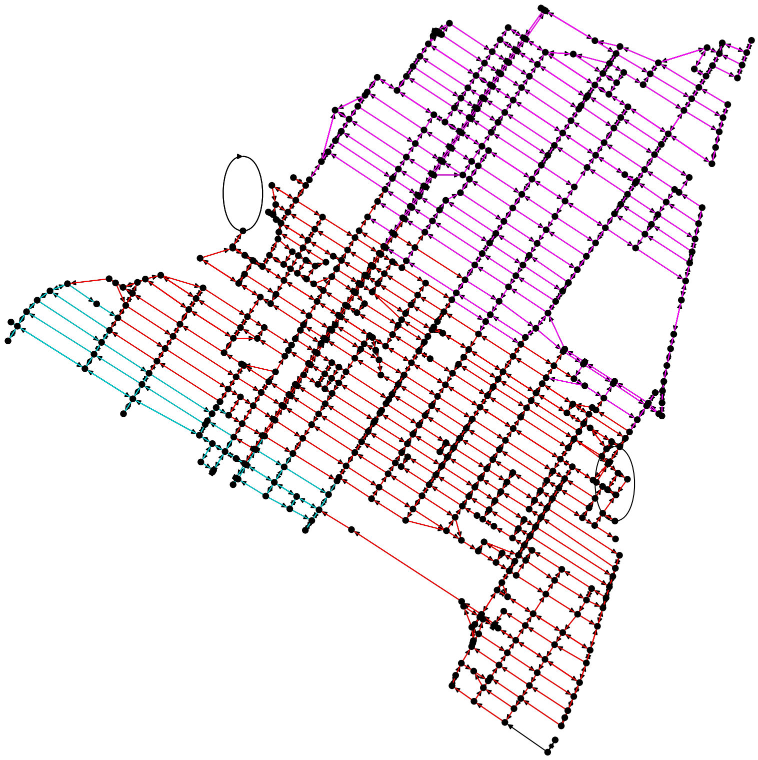

FIGURE 9 – Troisième solution des déneigeuses