Optimisation du dernier kilomètre de livraison d'Amazon

Soutenance de projet – PRRE

Adèle Berger, Manon Chartrin, Colin Coërchon

ENSIIE

24 mai 2024

Informations clefs

- Professeur référents : **Massinissa Merabet**
- Période de travail : du **19 avril** 2024 au **24 mai** 2024

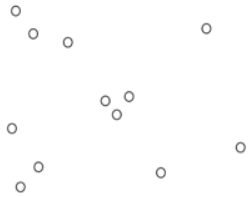
Sommaire

Contexte

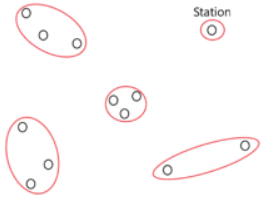
Figure – Issue du Challenge Amazon

Les objectifs du projet

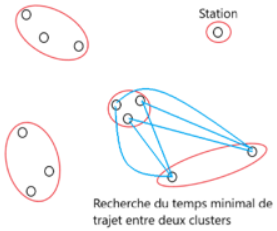
1. Mettre en forme les données brutes fournies.



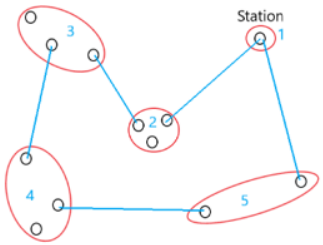
Représentation des stops d'une route



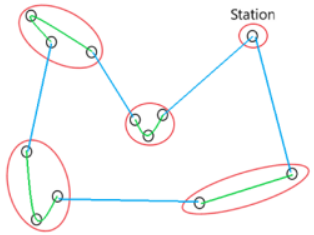
Formation des clusters



Recherche du temps minimal de trajet entre chaque clusters deux à deux



Obtention des vo et vf de chaque cluster

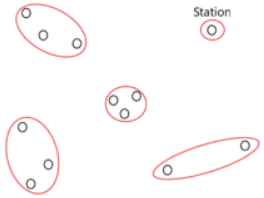


PLNE dans chaque cluster pour obtenir la route finale

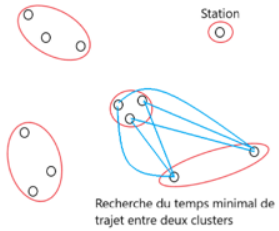
Les objectifs du projet

1. Mettre en forme les données brutes fournies.
2. Diviser l'ensemble des arrêts en différents clusters.

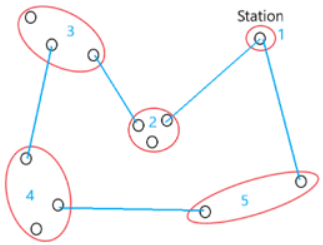
Représentation des stops d'une route



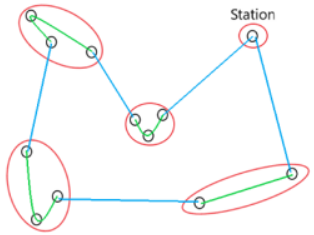
Formation des clusters



Recherche du temps minimal de trajet entre chaque clusters deux à deux



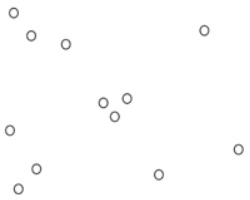
Obtention des vo et vf de chaque cluster



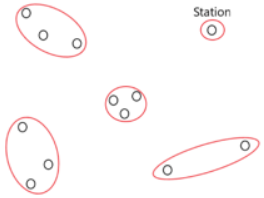
PLNE dans chaque cluster pour obtenir la route finale

Les objectifs du projet

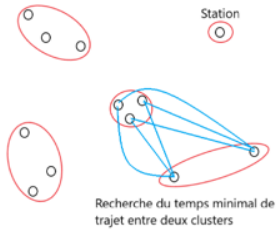
1. Mettre en forme les données brutes fournies.
2. Diviser l'ensemble des arrêts en différents clusters.
3. Déterminer l'ordre de la route entre les différents clusters.



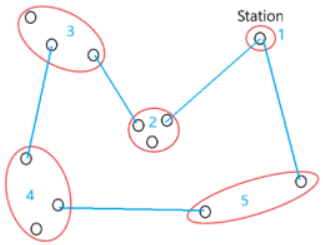
Représentation des stops d'une route



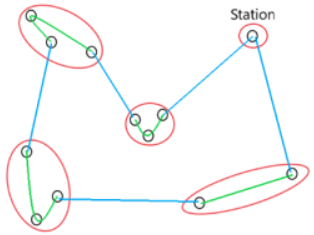
Formation des clusters



Recherche du temps minimal de trajet entre chaque clusters deux à deux



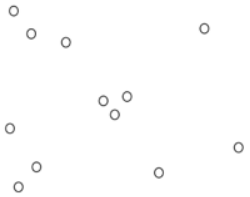
Obtention des vo et vf de chaque cluster



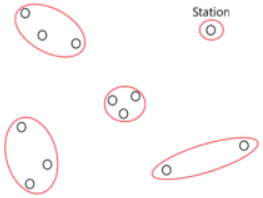
PLNE dans chaque cluster pour obtenir la route finale

Les objectifs du projet

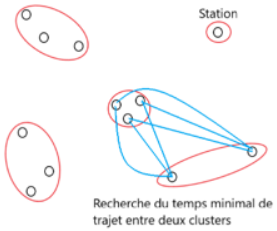
1. Mettre en forme les données brutes fournies.
2. Diviser l'ensemble des arrêts en différents clusters.
3. Déterminer l'ordre de la route entre les différents clusters.
4. Déterminer le premier et le dernier arrêt au sein de chaque cluster.



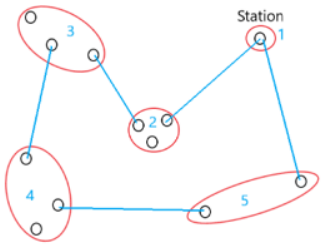
Représentation des stops d'une route



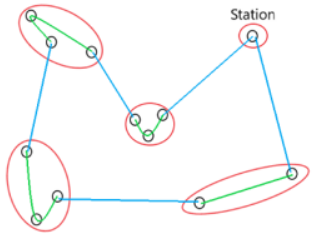
Formation des clusters



Recherche du temps minimal de trajet entre chaque clusters deux à deux



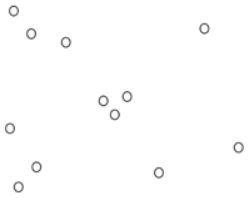
Obtention des vo et vf de chaque cluster



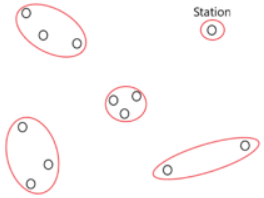
PLNE dans chaque cluster pour obtenir la route finale

Les objectifs du projet

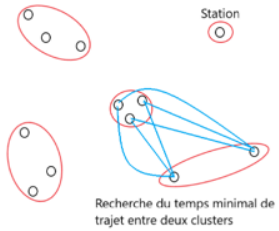
1. Mettre en forme les données brutes fournies.
2. Diviser l'ensemble des arrêts en différents clusters.
3. Déterminer l'ordre de la route entre les différents clusters.
4. Déterminer le premier et le dernier arrêt au sein de chaque cluster.
5. Déterminer la route optimale à l'intérieur de chaque cluster (PLNE).



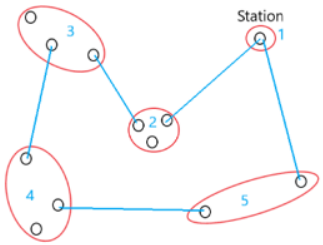
Représentation des stops d'une route



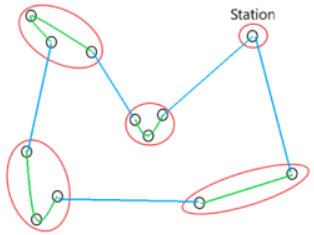
Formation des clusters



Recherche du temps minimal de trajet entre chaque clusters deux à deux

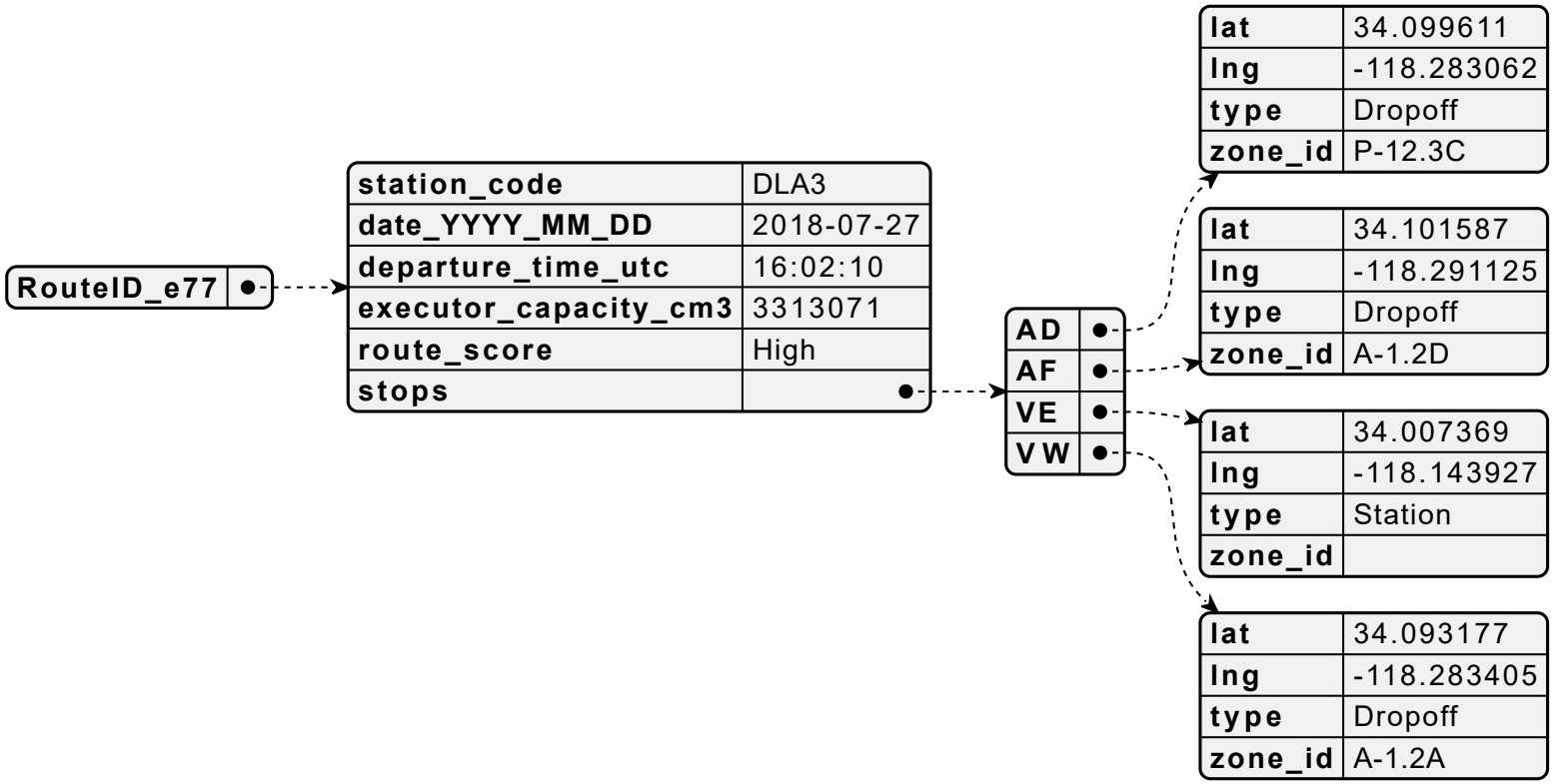


Obtention des vo et vf de chaque cluster

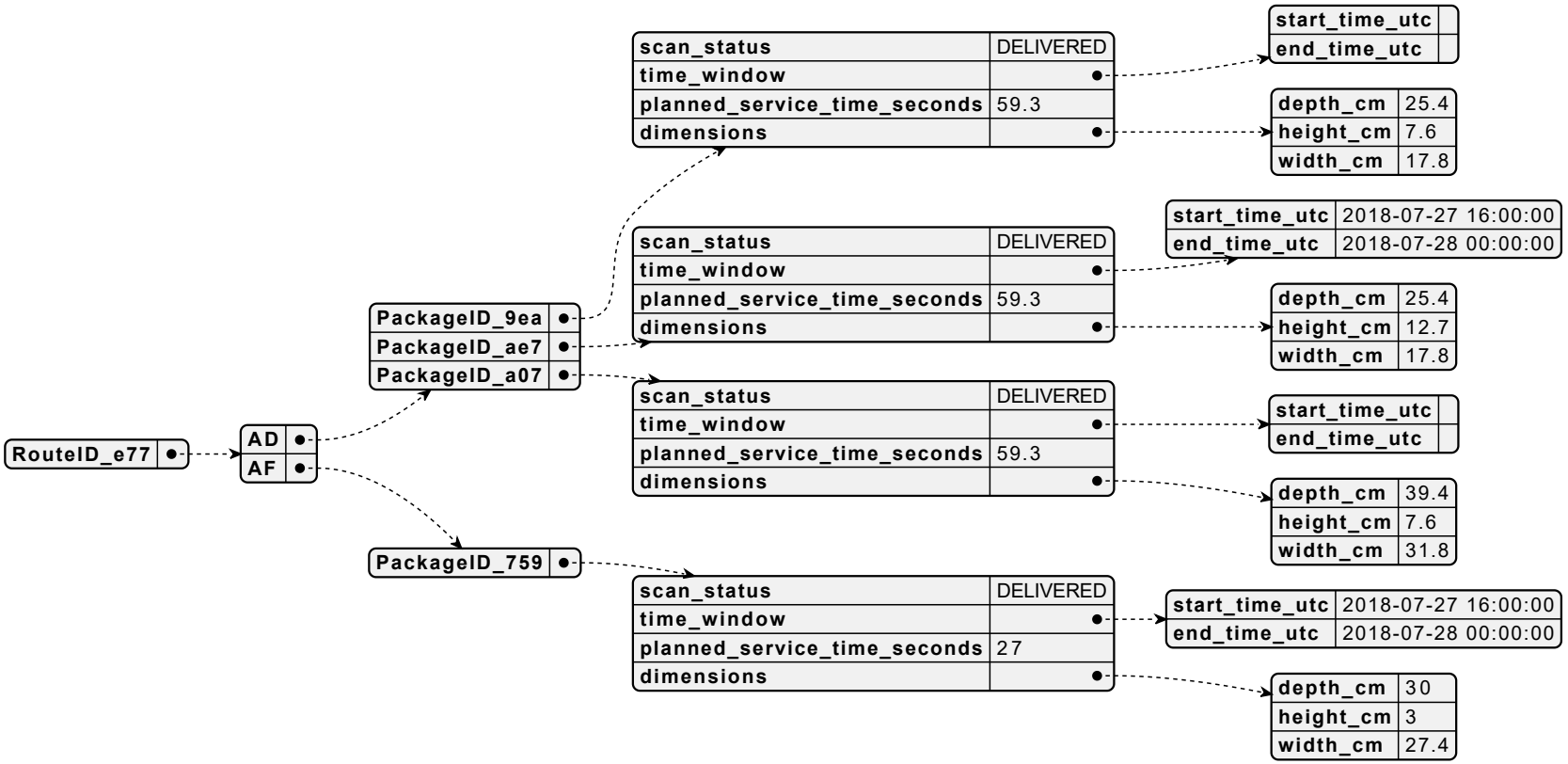


PLNE dans chaque cluster pour obtenir la route finale

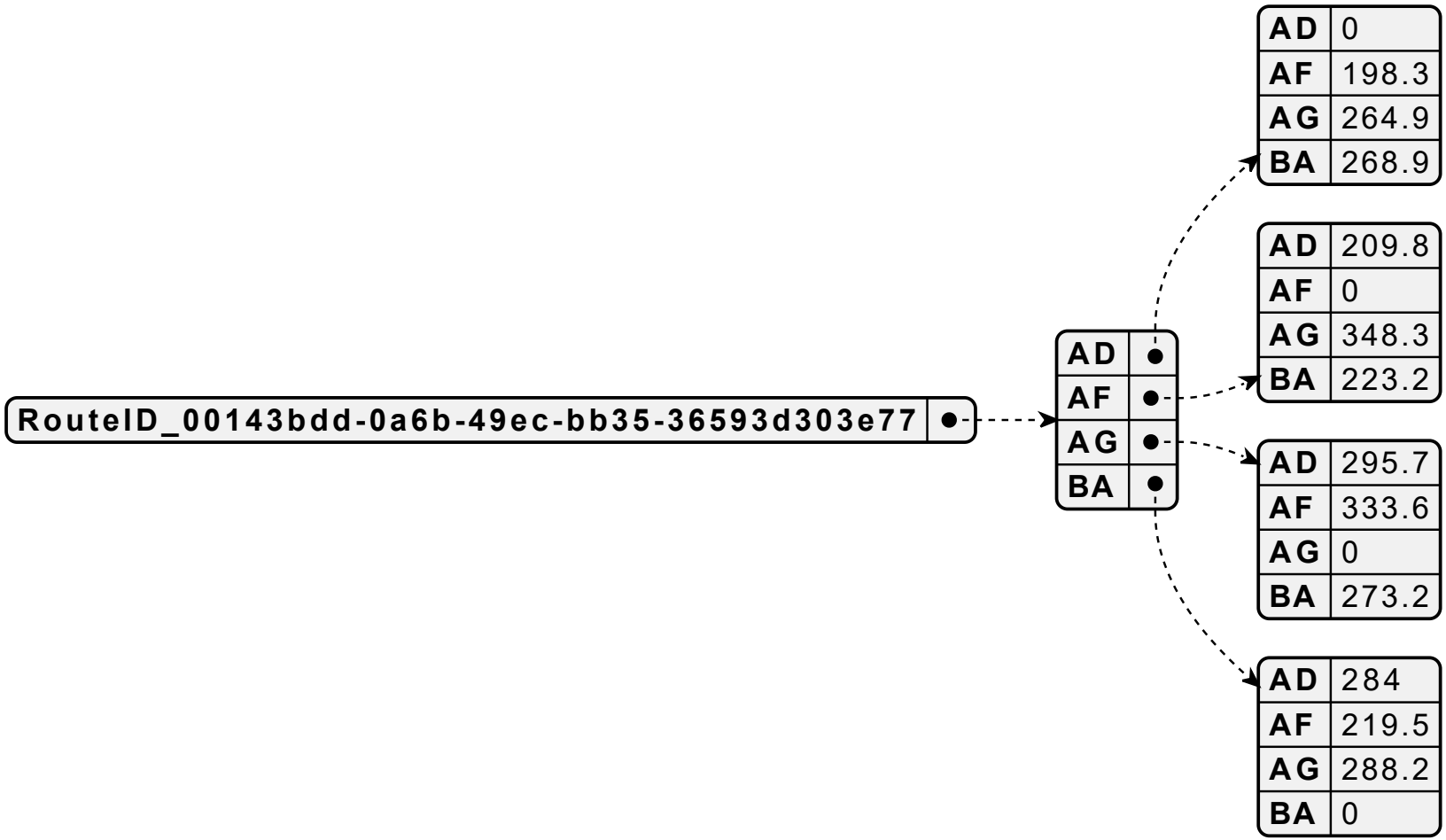
Informations brutes contenues dans `route_data.json`



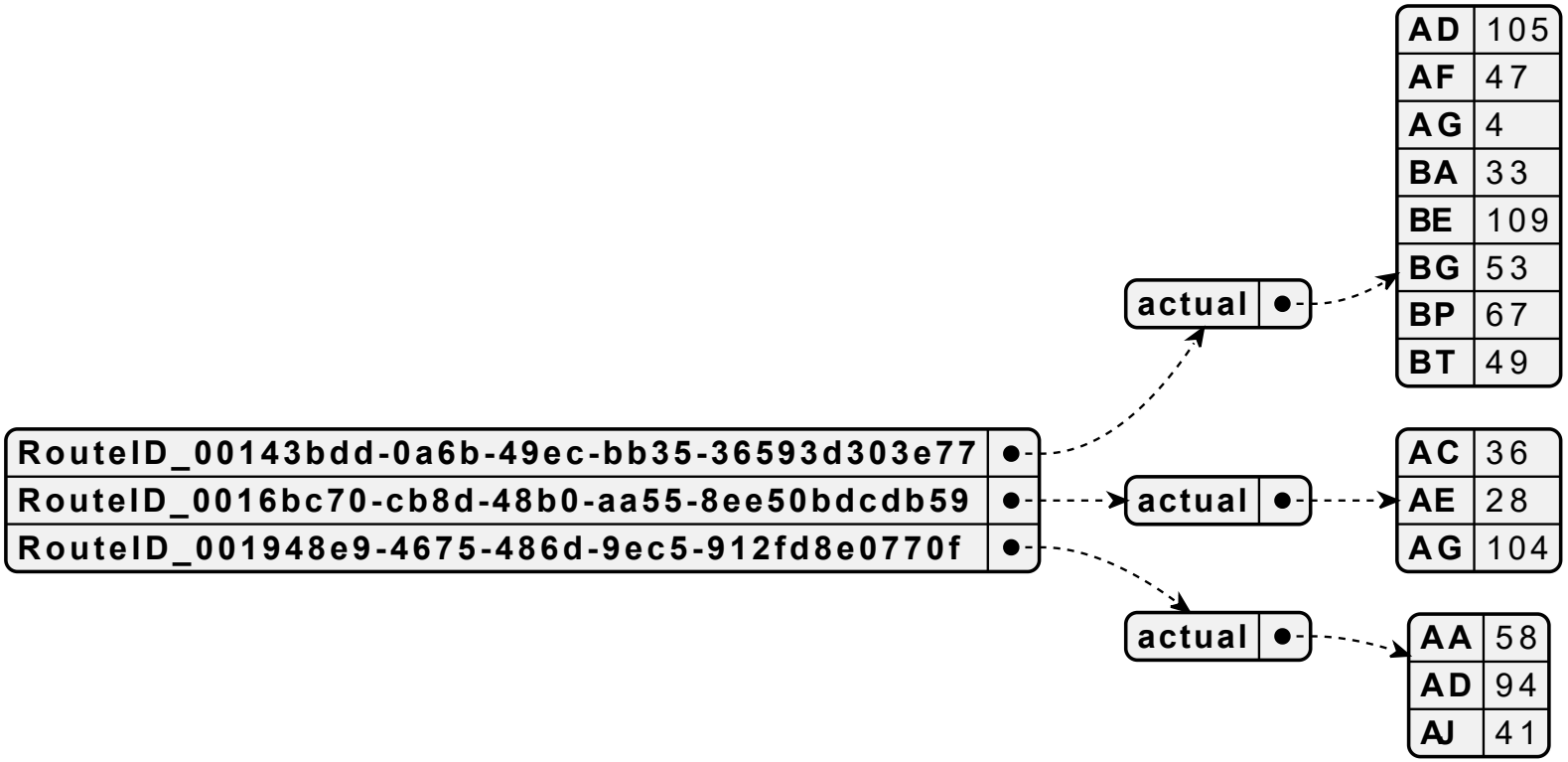
Informations brutes contenues dans `package_data.json`

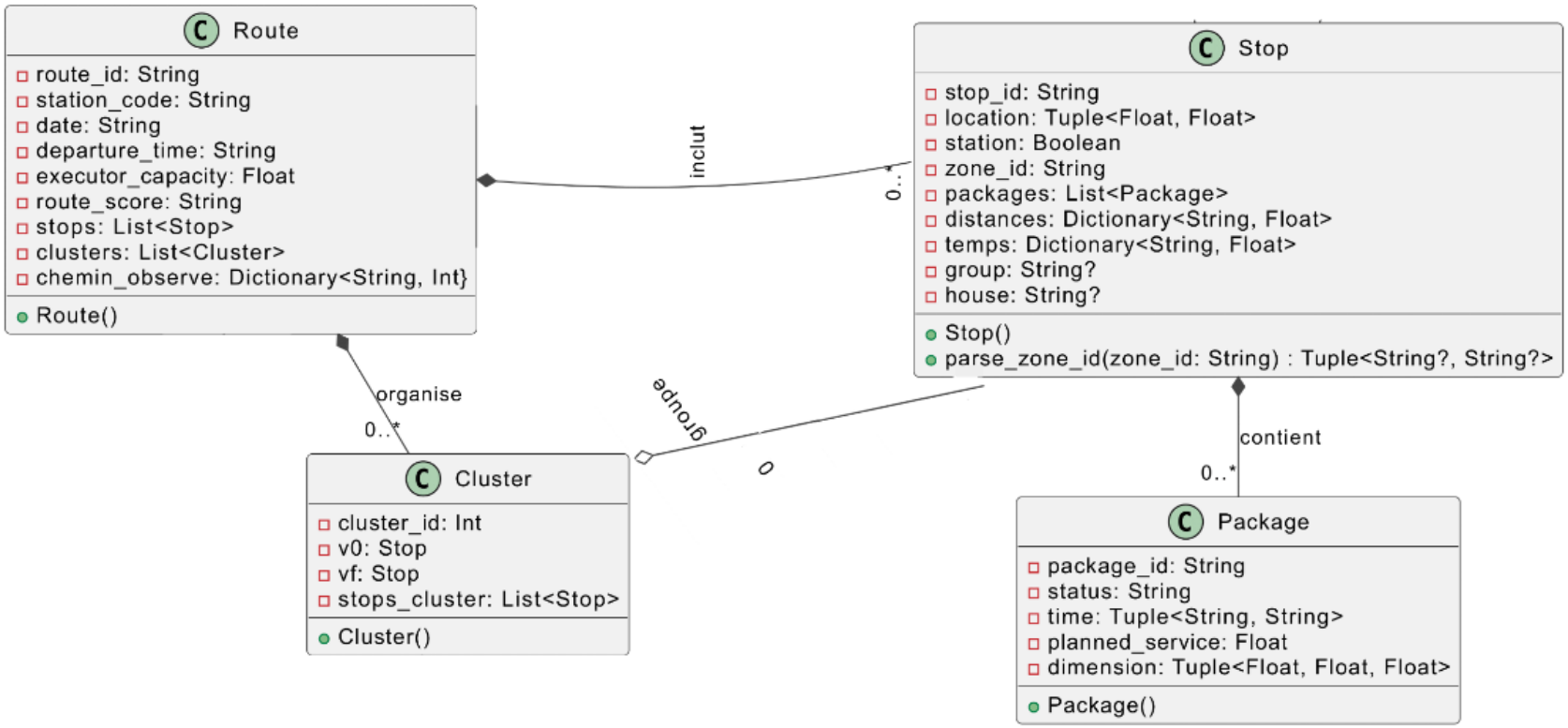


Informations brutes contenues dans `travel_times.json`



Informations brutes contenues dans `actual_sequences.json`




Route
route_id: String
station_code: String
date: String
departure_time: String
executor_capacity: Float
route_score: String
stops: List<Stop>
clusters: List<Cluster>
chemin_observe: Dictionary<String, Int}
Route()
inclut
0..*
Stop
stop_id: String
location: Tuple<Float, Float>
station: Boolean
zone_id: String
packages: List<Package>
distances: Dictionary<String, Float>
temps: Dictionary<String, Float>
group: String?
house: String?
Stop()
parse_zone_id(zone_id: String) : Tuple<String?, String?>
organise
0..*
groupe
0
contient
0..*
Cluster
cluster_id: Int
v0: Stop
vf: Stop
stops_cluster: List<Stop>
Cluster()
Package
package_id: String
status: String
time: Tuple<String, String>
planned_service: Float
dimension: Tuple<Float, Float, Float>
Package()

Remplacement des données manquantes

Les données manquantes

Attributs package.time

Remplacement des données manquantes

Les données manquantes

Attributs package.time

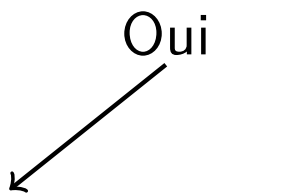
- Présence d'autres colis à livrer au même arrêt ?

Remplacement des données manquantes

Les données manquantes

Attributs package.time

- Présence d'autres colis à livrer au même arrêt ?

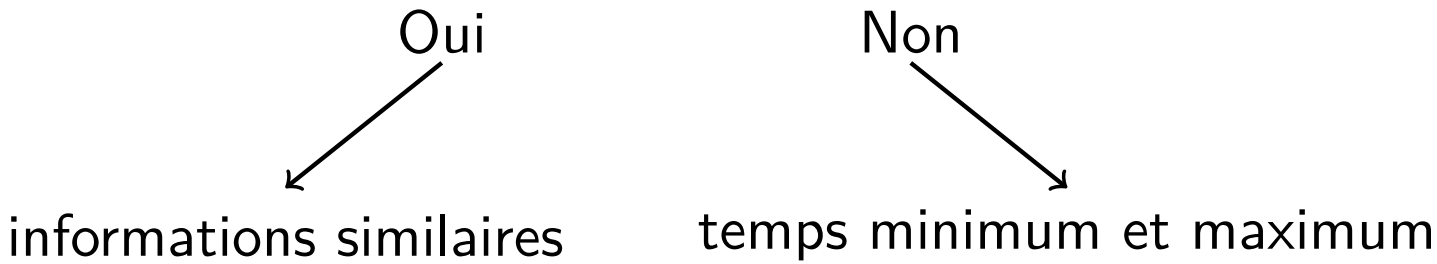


Remplacement des données manquantes

Les données manquantes

Attributs package.time

- Présence d'autres colis à livrer au même arrêt ?



- **Récupération et Transformation des données** :
  - `read_data.py` : Chargement et lecture des données initiales.
  - `transformationData.py` : Transformation et préparation des données pour le traitement. Contient aussi l'initialisation des instances de la classe `Cluster` une fois ceux-ci déterminés.
- **Utilitaires et Classes** :
  - `utils.py` : Fonctions utilitaires générales utilisées à travers le projet.
  - `classes.py` : Définition des 4 classes utilisées pour structurer les données récupérées

- **Clustering et Optimisation** :
  - `ClusteringData.py` : Il y a dans ce fichier trois fonctions extrêmement importantes :
    - Implémentation de l'heuristique de clustering avec la fonction `cluster_zone_id`.
    - Implémentation de l'heuristique utilisée pour réaliser un problème du voyageur de commerce entre les différents clusters construits. Cela se fait à l'aide de la fonction `find_optimal_path`
    - Choix des points de départ et d'arrivée dans chaque cluster.
  - `fonction_plne.py` : Exécution de l'algorithme TSPTW avec CPLEX pour chaque cluster.
- **Affichage et Interface** :
  - `affichage.py` : Regroupe toutes les fonctions d'affichage pour les fichiers main.

- **Exécution Principale** :
  - `main.py` : Exécution principale pour une route unique (la première route).
  - `main_routes.py` : Gère l'exécution sur plusieurs routes.
  - `main_sans_cluster.py` : Test de l'algorithme TSPTW sur l'ensemble des stops d'une route sans pré-cluster.
- **Données de travail et Logs** :
  - Dossiers `Data_test` `Data_test2` et `100routes` : Contiennent des données pour tester l'exécution des fichiers main. `100routes` contient les données des 100 premières routes d'Amazon.
  - `clone_directory` : Utilisé pour récupérer les logs de CPLEX lors de l'exécution de la programmation linéaire.

Formation des clusters

Formation des clusters

- Station assignée seule au **cluster 0**.

Formation des clusters

Formation des clusters

- Station assignée seule au **cluster 0**.
- Première approche : groupes par `zone_id`.

Formation des clusters

Formation des clusters

- Station assignée seule au **cluster 0**.
- Première approche : groupes par `zone_id`.
- Heuristique finale : détermination d'une limite minimale (3) et maximale (15) de stops par cluster.

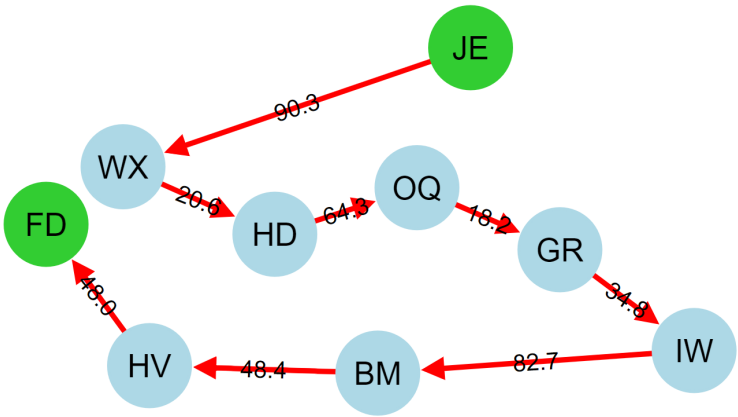
Formation des clusters

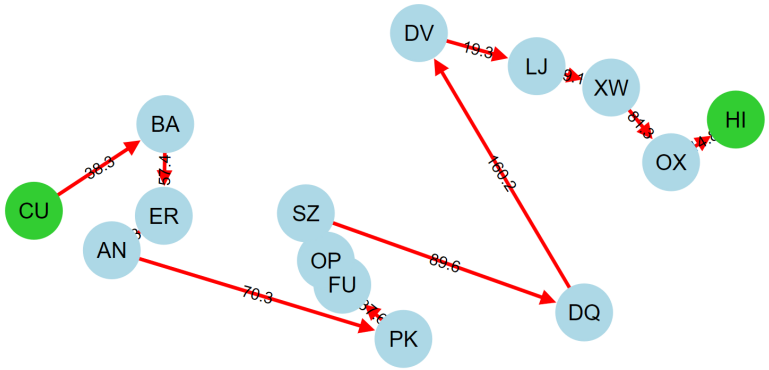
Formation des clusters

- Station assignée seule au **cluster 0**.
- Première approche : groupes par `zone_id`.
- Heuristique finale : détermination d'une limite minimale (3) et maximale (15) de stops par cluster.
- Clustering obtenu pour la première route :

```
Nombre de clusters: 13
Temps pour clustering: 0.00 secondes
Cluster 0: ['VE']
Cluster 1: ['AD', 'CM', 'EC', 'EH', 'EX', 'HB', 'IP', 'QO', 'SI', 'SQ', 'TY', 'UN', 'XD', 'YJ', 'ZP']
Cluster 2: ['HW', 'JM', 'TQ']
Cluster 3: ['BA', 'CP', 'DN', 'DQ', 'HN', 'YH']
Cluster 4: ['CG', 'DJ', 'LG', 'NL', 'UR', 'VA', 'ZE']
Cluster 5: ['AF', 'BT', 'CA', 'GN', 'KA', 'KJ', 'LD', 'MW', 'NE', 'TC', 'UJ']
Cluster 6: ['GU', 'IJ', 'IW', 'KN', 'KP', 'MR', 'NR', 'PT', 'SC', 'TH', 'UW', 'XB']
Cluster 7: ['BY', 'CW', 'DL', 'FH', 'JH', 'LB', 'SF', 'TK', 'VW']
Cluster 8: ['AG', 'FY', 'GB', 'GP', 'HO', 'HT', 'NM', 'QM', 'RY', 'TG', 'WS', 'YR']
Cluster 9: ['BZ', 'EO', 'GS', 'IA', 'NU', 'QX', 'RG', 'VC', 'YE', 'ZB', 'ZU']
Cluster 10: ['BE', 'BG', 'CO', 'HR', 'KG', 'KM', 'PB', 'PX', 'SD', 'UI', 'UU', 'WJ', 'YY']
Cluster 11: ['BP', 'CK', 'EY', 'FF', 'HG', 'IM', 'KU', 'LK', 'MA', 'MO', 'MQ', 'QE', 'RA', 'YN']
Cluster 12: ['GW', 'LY', 'PJ', 'PS', 'US']
```

Résultats sur 2 clusters





Justification du clustering

Pourquoi faire un clustering ?

- PLNE sur tous les stops : pas de résultat au bout de 1 heure.
- Très grand jeu de données : clustering puis application du PLNE sur chaque groupe de données.
- Importance de créer des clusters de petite taille (avec une limite de 15 stops par clusters).

Algorithme de parcours des clusters

Recherche d'un chemin optimal entre les clusters

1. Recherche des temps de trajets minimaux entre chaque paire de clusters.

Algorithme de parcours des clusters

Recherche d'un chemin optimal entre les clusters

1. Recherche des temps de trajets minimaux entre chaque paire de clusters.
2. Création de la matrice des temps de trajet minimaux entre les clusters.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.0	1719.0	1676.5	1696.2	1557.8	1722.7	1614.1	1597.4	1509.7	1676.2	1755.6	1624.6	1612.3
1	1869.3	0.0	127.3	217.6	298.9	134.5	107.7	126.6	192.9	17.0	38.4	17.7	109.7
2	1866.8	42.4	0.0	287.4	314.0	211.6	126.2	124.1	172.6	114.8	211.5	42.3	76.6
3	1908.4	242.8	294.4	0.0	95.2	61.5	64.5	227.6	282.6	153.2	179.5	203.9	238.1
4	1783.8	306.2	303.6	91.1	0.0	136.9	71.7	156.3	192.7	211.6	238.4	213.1	247.3
5	1922.1	148.9	236.5	55.7	137.0	0.0	100.7	178.5	222.2	57.3	12.6	146.0	162.2
6	1724.3	127.8	121.2	53.1	79.7	95.4	0.0	45.8	72.2	32.7	84.4	30.7	64.9
7	1732.8	137.9	119.2	215.8	171.8	168.5	43.8	0.0	72.1	89.0	157.0	22.3	61.1
8	1651.4	186.3	167.6	268.4	165.0	203.5	56.5	43.5	0.0	124.0	203.4	70.7	6.9
9	1827.3	9.3	131.8	157.5	215.1	41.7	27.3	83.7	127.4	0.0	41.6	39.1	67.4
10	1872.9	38.4	248.6	147.1	228.4	12.9	85.9	169.4	216.9	49.4	0.0	140.7	160.3
11	1764.8	27.5	49.6	197.7	223.7	121.9	36.5	22.1	88.4	25.1	121.8	0.0	33.2
12	1759.8	101.9	75.5	237.1	243.8	143.3	75.9	68.6	6.3	63.8	143.2	33.7	0.0

Algorithme de parcours des clusters (suite)

Recherche d'un chemin optimal entre les clusters

3. Détermination du chemin optimal entre les clusters.
   - Idée initiale : comparer toutes les solutions entre elles
     ↪ longue et donc impraticable.
   - Heuristiques du plus proche voisin et de l'insertion du plus proche voisin.

Stop de départ et d'arrivée des clusters

Algorithme qui détermine v0 et vf pour chaque cluster

1. **Initialisation** : Le `v0` du premier cluster du chemin optimal est déjà connu.

Stop de départ et d'arrivée des clusters

Algorithme qui détermine v0 et vf pour chaque cluster

1. **Initialisation** : Le `v0` du premier cluster du chemin optimal est déjà connu.
2. **Itération sur les clusters** : Pour chaque paire de clusters consécutifs : identification de `vf` du cluster courant et de `v0` du cluster suivant

Stop de départ et d'arrivée des clusters

Algorithme qui détermine v0 et vf pour chaque cluster

1. **Initialisation** : Le `v0` du premier cluster du chemin optimal est déjà connu.
2. **Itération sur les clusters** : Pour chaque paire de clusters consécutifs : identification de `vf` du cluster courant et de `v0` du cluster suivant
3. **Enregistrement des résultats** : Les identifiants des arrêts `v0` et `vf` sont enregistrés dans une liste..

Le PLNE

Le problème revient à minimiser la fonction objectif

$$\sum_{(i,j)\in E} x_{i,j} * p_{i,j} \tag{B.1}$$

soumis à

$$\begin{cases}
\sum_{(i,j)\in E} x_{i,j} = 1 & \forall i \in V - \{v_f\} \quad \text{(B.2a)} \\
\sum_{(i,j)\in E} x_{i,j} = 1 & \forall j \in V - \{v_0\} \quad \text{(B.2b)} \\
\sum_{(i,j)\in E} x_{i,j} = \sum_{(i,j)\in E} x_{j,i} & \forall i \in V - \{v_0, v_f\} \quad \text{(B.2c)} \\
\sum_{(i,j)\in E} x_{i,j} - \sum_{(i,j)\in E} x_{j,i} = 1 & i = v_0 \quad \text{(B.2d)} \\
\sum_{(i,j)\in E} x_{i,j} - \sum_{(i,j)\in E} x_{j,i} = -1 & i = v_f \quad \text{(B.2e)} \\
x_{i,i} = 0 & \forall i \in V \quad \text{(B.2f)} \\
t_i + p_{i,j} - t_j \leq M * (1 - x_{i,j}) & \forall (i,j) \in E \quad \text{(B.2g)} \\
a_i \leq t_i \leq b_i & \forall (i,j) \in E \quad \text{(B.2h)} \\
x_{i,j} \in \{0,1\} & \forall (i,j) \in E \quad \text{(B.2i)}
\end{cases}$$

Implémentation du PLNE

- **Les variables**

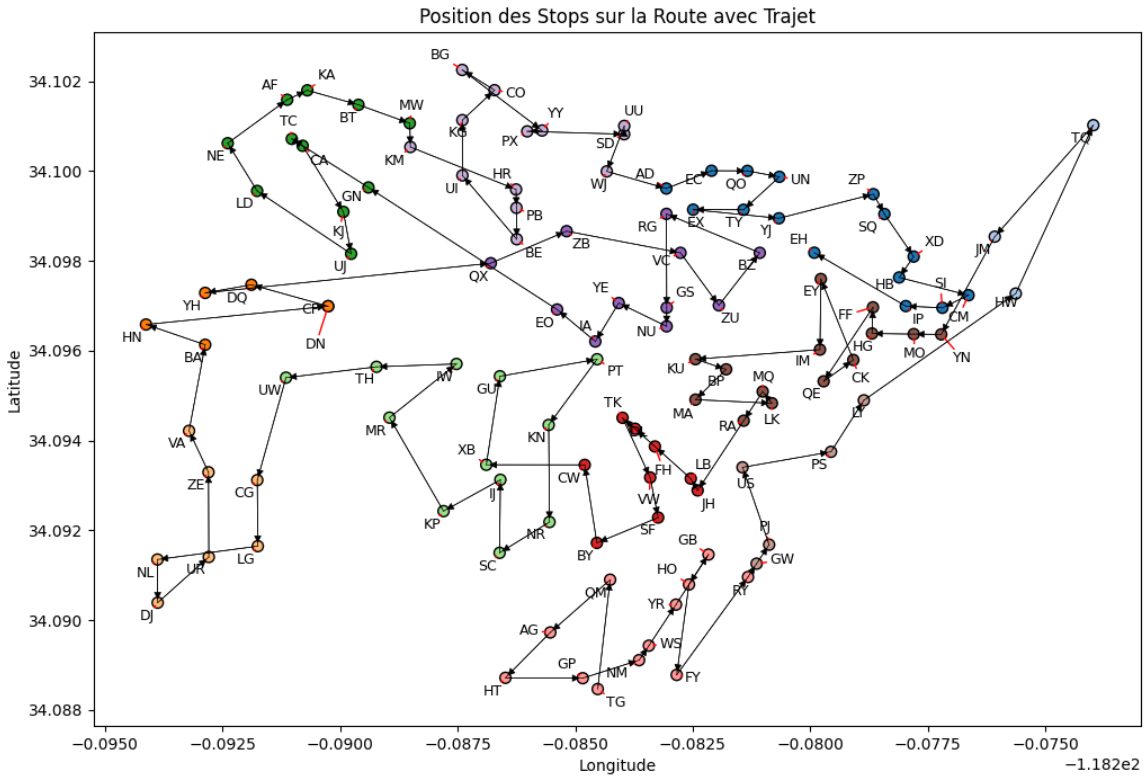
```
x = pulp.LpVariable.dicts("x", E, cat=pulp.LpBinary)
t = pulp.LpVariable.dicts("t", V, lowBound=0,
↪  cat=pulp.LpContinuous)
```

- **La fonction objectif**

```
prob += pulp.lpSum(x[i, j] * p[i, j] for i, j in E),
↪  "Minimiser le temps de parcours total"
```

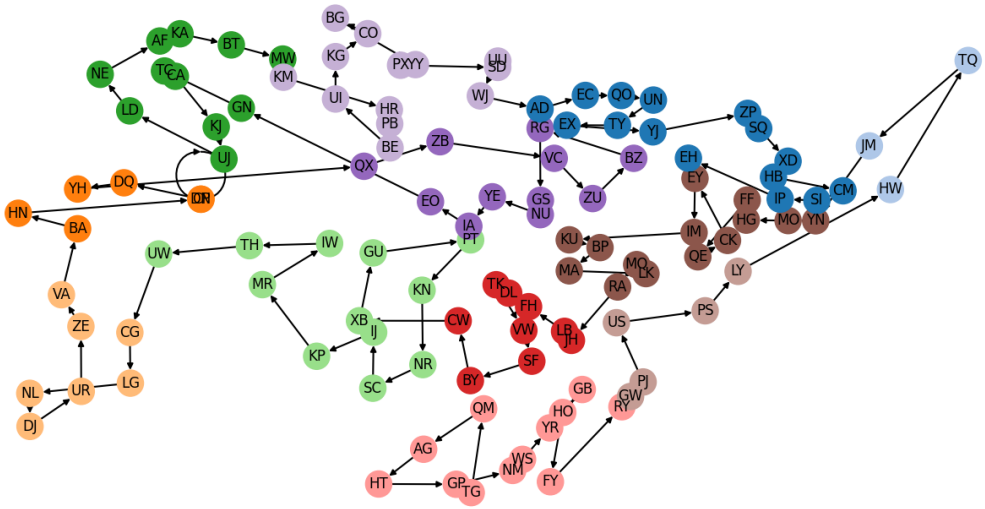
Pour une route - Matplotlib

- Les stops sont coloriés selon le cluster auquel ils appartiennent.
- Le trajet total est représenté par des flèches.



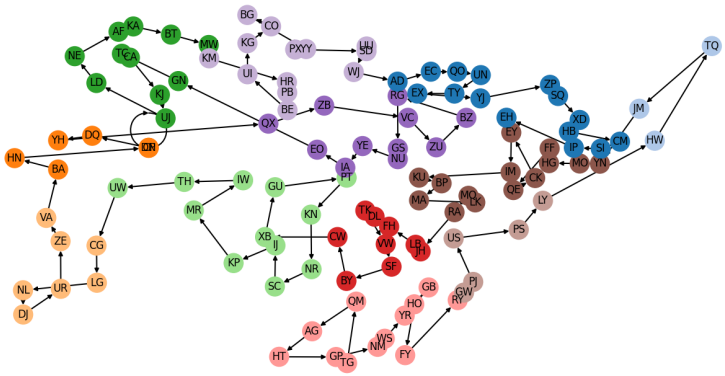
Pour une route - Networks

- Les stops sont coloriés selon le cluster auquel ils appartiennent.
- Le trajet total est représenté par des flèches.

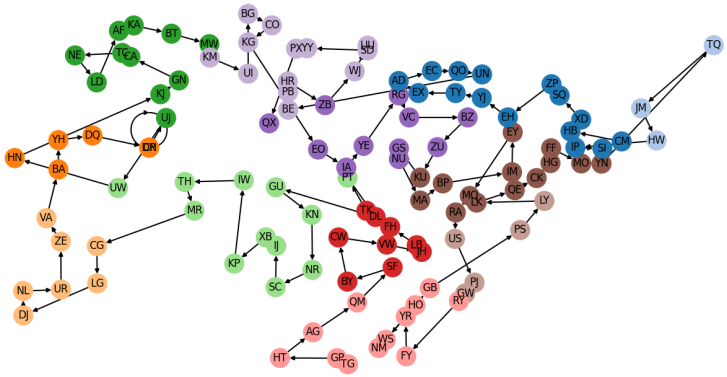


Temps de trajet total: 9078.0 secondes

Comparaison pour une route



Temps de trajet total: 9078.0 secondes



Temps de trajet total: 9618.8 secondes

- 540.8 secondes de moins soit environ 9 minutes

Pour 3 routes

- Route 1, 2 et 3
- Résultats cohérents avec les résultats attendus
- On gagne exactement 9 minutes, 4 min 17, et 1 min 46 par rapport au chemin observé des livreurs

Route	Nombres de Stops	Heure de Début	Heure de Fin	Temps de Calcul	Temps Gagné
Route 1	119	16 : 02 : 10	18 : 33 : 28	4.73 s.	09 : 00
Route 2	106	15 : 44 : 41	18 : 08 : 21	5.51 s.	04 : 17
Route 3	128	15 : 32 : 04	18 : 57 : 29	6.34 s.	01 : 46

Pour 100 routes

Description	Temps Moyen (secondes)
Moyenne des temps pour tous les cluster de chaque route	0.01
Moyenne des temps pour trouver le chemin optimal	0.00
Moyenne des temps pour trouver tous les `v0` et `vf`	0.00
Moyenne des temps totaux pour les TSP avec fenêtres de temps	7.08
Moyenne des temps de calcul	7.09
Temps moyen gagné	00 : 10 : 21

Ainsi, en moyenne, sur des tranches de travail allant de 1h30 à 3h30 :

Notre travail pourrait permettre de faire gagner
10 minutes de travail au livreur durant son travail.

Limites et Difficultés

Limites

- Notre heuristique de clustering a été choisie pour sa **simplicité** et son **efficacité**, bien qu'il en existe sûrement beaucoup d'autres, peut être meilleurs
- Les informations sur les **capacités des camions**, la **taille des colis** ainsi que la **qualité de la route** n'ont pas été prises en compte dans notre analyse

Limites et Difficultés

Difficultés

- Le traitement d'un très grand jeu de données
- Au début du projet `start_time_utc` n'était pas fonctionnel
- Le traitement des valeurs manquantes (notamment dans les données `start_time_utc` et `end_time_utc`).
- L'implémentation de tous les fichiers `clusteringData.py` et `fonction_plne.py` ont été difficiles.

Limites et Difficultés

Difficultés

- Le traitement d'un très grand jeu de données
- Au début du projet `start_time_utc` n'était pas fonctionnel
- Le traitement des valeurs manquantes (notamment dans les données `start_time_utc` et `end_time_utc`).
- L'implémentation de tous les fichiers `clusteringData.py` et `fonction_plne.py` ont été difficiles.

Surprise de dernière minute

D'autres valeurs manquantes : quelques routes possédaient des stops sans `zone_id`. Nous avons manqué de temps pour corriger ceci.

Conclusion

- Explorer et de mettre en œuvre des **approches innovantes**
- Démarche structurée en **plusieurs étapes clés**
- Résultats **intéressants** et **prometteurs**, puisqu'ils sont (au moins) meilleurs que ceux observés