Janvier 2023

# PROJET DE MAOA

## Autour du tracé d'un métro circulaire

---

**ZHUANG Christian**
**XIA Alexandre**

# 1 Idées et méthodes de résolution

On va présenter les méthodes qu'on a utilisé pour le problème.

## 1.1 Heuristique et métaheuristique

La première méthode pour résoudre le problème est l'heuristique et métaheuristique. L'idée est de passer dans un premier temps le nuage de points dans un algorithme de clustering, Kmeans dans notre cas. Donc on a comme hyperparamètre $p$ le nombre de groupes à former, qui est également le nombre de stations.

Une fois les groupes formés, on assigne pour chaque groupe une station, qui est le point le plus proche du centroïde du groupe. Puis, on performe un algorithme de résolution du TSP (Travelling Salesman Problem) sur les stations ce qui nous donnera une première solution. Il faut noter que l'algorithme de clustering K-means donnant un résultat stochastique, il peut être nécessaire de le relancer plusieurs fois et de résoudre le TSP pour chaque résultat du clustering et choisir le résultat qui nous plaira le mieux.

On cherche ensuite à améliorer la solution donnée par l'heuristique. Pour ce faire, on a décidé d'utiliser la métaheuristique suivante :

- On effectue toutes les échanges 1-1 sur la solution. L'échange se fait en remplaçant une station par un point au sein du même groupe.
- On garde dans la liste des solutions que les solutions non dominées (au sens de Pareto)

On se limite à n'effectuer qu'une seule fois cette opération pour éviter d'avoir trop de calcul à faire. Et on choisit une solution parmi la liste des solutions qui nous semble la meilleure.

## 1.2 PLNE compact et PLNE non-compact : Le problème de l'anneau étoile

Pour la formulation compacte et non-compacte, nous allons utiliser la formulation du problème de l'anneau étoile donnée dans le sujet mais qu'on va rappeler :

**- formulation compacte** :

$$Min \sum_{ij \in E} d_{ij} x_{ij} + \sum_{(i,j) \in V \times V} d_{ij} y_{ij}$$

$$\sum_{i \in V} y_{ii} = p \tag{1}$$

$$\sum_{j \in V} y_{ij} = 1 \qquad \forall i \in V \tag{2}$$

$$y_{ij} \leq y_{jj} \qquad \forall (i,j) \in V \backslash \{j\} \times V \tag{3}$$

$$\sum_{ij \in \delta(i)} x_{ij} = 2y_{ii} \qquad \forall i \in V \tag{4}$$

$$\sum_{j \in V \backslash \{1\}} z_{1j} = p - 1 \tag{5}$$

$$\sum_{j \in V \backslash \{i\}} z_{ji} = \sum_{j \in V \backslash \{1,i\}} z_{ij} + y_{ii} \qquad \forall i \in V \backslash \{1\} \tag{6}$$

$$z_{ij} + zji \leq (p-1)x_{ij} \qquad \forall i \in V, j \in V \backslash \{1, i\} \tag{7}$$

$$x_{ij} \in \{0,1\} \qquad \forall ij \in E$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in V \times V$$

$$z_{ij} \in [0, p-1] \qquad \forall (i,j) \in V \times V \backslash \{1, i\}$$

$$y_{11} = 1$$

$$y_{1j} = 0 \qquad \forall j \in V \backslash \{1\}$$

**- formulation non-compacte** :

$$Min \sum_{ij \in E} d_{ij} x_{ij} + \sum_{(i,j) \in V \times V} d_{ij} y_{ij}$$

$$\sum_{i \in V} y_{ii} = p \tag{1}$$

$$\sum_{j \in V} y_{ij} = 1 \qquad \forall i \in V \tag{2}$$

$$y_{ij} \leq y_{jj} \qquad \forall (i,j) \in V \backslash \{j\} \times V \tag{3}$$

$$\sum_{ij \in \delta(i)} x_{ij} = 2y_{ii} \qquad \forall i \in V \tag{4}$$

$$\sum_{e \in \delta(S)} x(e) \geq 2y_{ii} \qquad \forall S \subset V, 1 \notin S, i \in S \tag{5}$$

$$x_{ij} \in \{0,1\} \qquad \forall ij \in E$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in V \times V$$

$$y_{11} = 1$$

$$y_{1j} = 0 \qquad \forall j \in V \backslash \{1\}$$

# 2 Expériences et résultats

Le langage de programmation utilisé est principalement Julia pour la programmation linéaire, et Python pour Kmeans (avec scikit-learn). Pour les expériences, le nombre de stations est fixé à 6 (valeur arbitraire). On a également choisi arbitrairement un délai de dépassement de 10 minutes (600 secondes).

Définition des critères

- le coût correspond à la longueur du tour de métro. Dans l'énoncé, il est écrit qu'on devrait sommer cela au temps de déplacement vers la station la plus proche pour les points qui ne sont pas station (fonction objective de la formulation PLNE de l'anneau étoile). Cependant, cela signifierait à cause du facteur 10 qu'il sera préférable d'avoir un grand nombre de stations.

- Le temps moyen des trajets est le temps moyen de l'ensemble des trajets possibles (d'un point vers un autre). Le trajet prise en compte est le plus court entre le chemin direct à la marche, le chemin vers la station la plus proche plus le plus chemin entre une des stations et l'objectif.

- Le ratio moyen de temps de marche correspond au pourcentage moyen de temps passé à marcher sur tous les trajets possibles.

### 2.1 Évaluation du temps de résolution

Pour les expérimentations sur le temps d'exécution et de calcul, on obtient les résultats suivants:

| Instance | Nombre de points | Heuristique (TSP+Kmeans) | PLNE compact | PLNE non compact |
|---|---|---|---|---|
| burma14 | 14 | 0.982 | 1.24 | 0.236 |
| berlin52 | 52 | 1.2 | 4.6 | 3.27 |
| eil101 | 101 | 1.91 | 208 | 210 |
| a280 | 280 | 27.1 | – (gap 1.79 %) | – (gap 1.95 %) |
| att532 | 532 | 162.11 | – (gap 1.6 %) | – (gap 1.6 %) |

Table 1: Durée d'exécution des méthodes, avec nombre de stations = 6 et timout de 600 secondes (Si le delai de dépassement est atteint, on affiche seulement le gap).

Les temps de calcul (CPU) et d'exécution étant très proches, on ne garde qu'une valeur. Il est clair ici que pour les temps, la méthode heuristique et méta-heuristique est largement plus rapide pour les paramètres choisis (6 stations). Les PLNE compact et non-compact semblent donner des temps assez proches pour les instances plus grandes que berlin52. Cependant, les deux n'arrivent pas à trouver de solutions pour des instances de type a280 et att532 dans le délai de 10 minutes. On évalue également le temps moyen de la méthode heuristique et méta-heuristique.

| Instance | Taille | Kmeans | TSP |
|---|---|---|---|
| burma14 | 14 points | 0.08 | 0.9 |
| berlin52 | 52 points | 0.09 | 1.1 |
| eil101 | 101 points | 0.1 | 1.8 |
| a280 | 280 points | 0.11 | 26.8 |
| att532 | 532 points | 0.11 | 162 |

Table 2: Durée d'exécution moyen (en secondes) en utilisant une heuristique et une métaheuristique (nombre de stations = 6, 3 exécutions).

### 2.2 Comparaison des 2 méthodes

On compare les solutions données par les méthodes utilisées, on note que le PLNE compact et non compact donne les mêmes résultats.

| Méthode | Coût | Temps moyen | Ratio Marche |
|---|---|---|---|
| PLNE | 114.12 | 220.32 | 86.4 % |
| TSP+Kmeans | 125.6 | 217.96 | 84.3 % |

Table 3: Valeurs des critères selon la méthode (PLNE ou Kmeans+TSP) sur l'instance eil101 (p=6)

Les deux solutions diffèrent. La méthode heuristique et métaheuristique propose une solution qui coûterait un peu plus cher, mais avec un ratio de marche et temps moyen de voyage plus petit. À noter que chaque clustering Kmeans donnera un résultat potentiellement différent. Les solutions sont visibles en annexe (Figure 1)

### 2.3 Impact de $p$ sur les critères

| Methode | Instance | Nombre de Stations | Coût | Temps moyen | Ratio Marche |
|---|---|---|---|---|---|
| PLNE | berlin52 | 6 | 2216.28 | 3650.03 | 83.5 % |
| PLNE | berlin52 | 10 | 3061.13 | 2983.41 | 72.2 % |
| PLNE | berlin52 | 15 | 3895.91 | 2572.37 | 62.1 % |
| PLNE | berlin52 | 20 | 4049.7 | 2355.88 | 54.4 % |
| TSP+Kmeans | eil101 | 6 | 125.6 | 217.96 | 86 % |
| TSP+Kmeans | eil101 | 10 | 195.66 | 199.36 | 77.5 % |
| TSP+Kmeans | eil101 | 15 | 241.92 | 181.97 | 69 % |
| TSP+Kmeans | eil101 | 20 | 286.94 | 180.09 | 64.2 % |

Table 4: Valeurs des critères selon le nombre de stations

On observe sur l'instance donnée clairement ici que quand le nombre de stations augmente, le pourcentage ratio marche/metro diminue, ainsi que le temps moyen de trajet diminue. Cependant, le coût de construction augmente également. Un compromis assez correct pour berlin52 pourrait être 20 stations, qui rapproche le ratio de marche et de métro à 50 % en rappelant que le temps de marche est la distance facteur 10.

On a le même genre de résultat attendu avec TSP+Kmeans que pour le PLNE. Pour l'instance eil101, on aurait donc un compromis d'avoir 15 stations (car le temps moyen ne change pas beaucoup entre 15 et 20) pour réduire le temps moyen de déplacement même si le ratio de temps de marche reste assez important. Cela peut aussi être dû à la nature stochastique de Kmeans. Quelques solutions sont visuellement affichées en annexe (Figure 2 et Figure 3)

### 2.4 Impact du clustering sur les critères

Nous expérimentons la méthode heuristique et métaheuristique qui donne des résultats différents selon le clustering.

| Clustering | Coût | Temps moyen | Ratio Marche |
|---|---|---|---|
| 1 | 13058.18 | 17595.81 | 84.3 % |
| 2 | 12454.69 | 17691.92 | 86.2 % |
| 3 | 15331.21 | 16982.61 | 84.5 % |

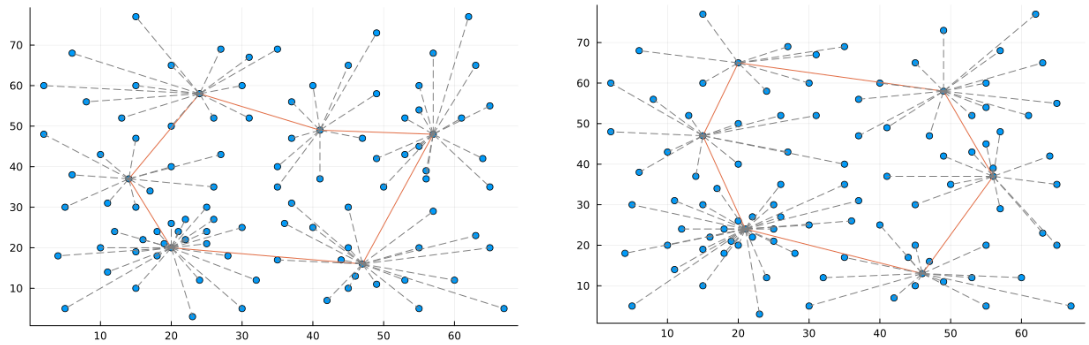
Table 5: Valeurs des critères pour 3 différents clusterings donné par Kmeans sur l'instance att532 et pour p = 6

Les résultats obtenus ont tous un avantage et un inconvénient. La solution du clustering 3 coûte plus cher, mais réduit le temps moyen ainsi que le ratio de temps de marche. La solution 2 est le moins cher, mais le ratio de temps de marche et le temps moyen de déplacement est le plus élevé des 3. La solution 1 est un peu plus entre les 2, mais semble un peu moins intéressante que la solution 2. Les solutions 1 et 2 sont visibles en annexe (Figure 4)

## 3 Conclusion et critique

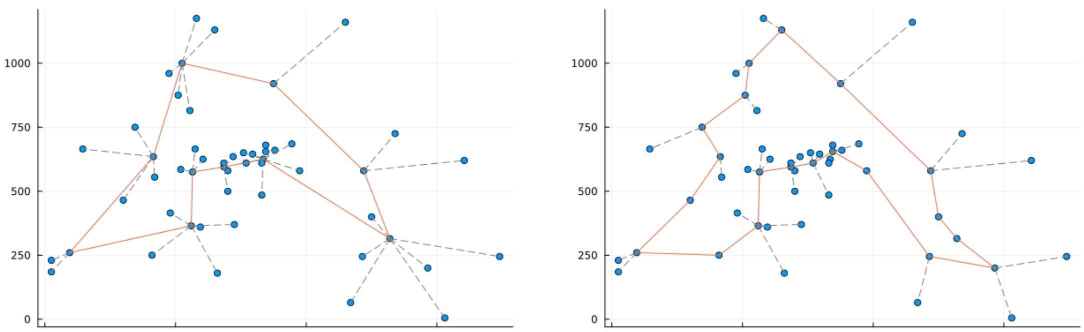
Pour conclure, il est clair que dans les expériences effectuées, la méthode heuristique et métaheuristique donnent des résultats dans des délais très rapides. Cependant, le facteur aléatoire de Kmeans signifie qu'avec plusieurs essais, nous pouvons avoir des solutions distinctes qui peuvent être moins bien ou mieux et donc sans garantie. De plus, la méthode de voisinage fait que plus le nombre de points est grand, plus le temps nécessaire pour parcourir toutes les solutions est long (surtout qu'on fait un nouveau TSP à chaque changement de station). Les PLNE ont une faiblesse au niveau du temps mis pour résoudre le problème, mais donne une solution unique ce qui peut être aussi un avantage par rapport à la première méthode. Le manque de décideur ne nous permet pas vraiment de vraiment savoir quelle solution est mieux en dehors de décisions que nous pourrions prendre personnellement et les seuls objectifs qu'on peut comparer ici sont : le temps d'exécution (et de calcul) et la garantie de solution (unique). Donc, si on souhaite trouver une solution rapidement la méthode heuristique et métaheuristique proposée serait la plus adéquate, mais si on préfère une solution unique, la formulation PLNE de l'anneau étoile est celle préférée.

## 4 Annexe : Quelques exemples d'affichage des solutions



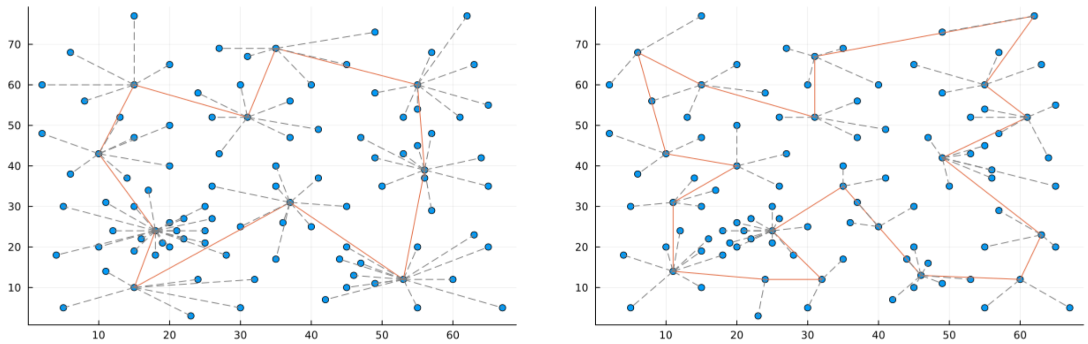

PLNE anneau étoile    Heuristique et métaheuristique

Figure 1: Comparaison de solutions pour les méthodes implémentées sur l'instance eil101 et $p = 6$ stations



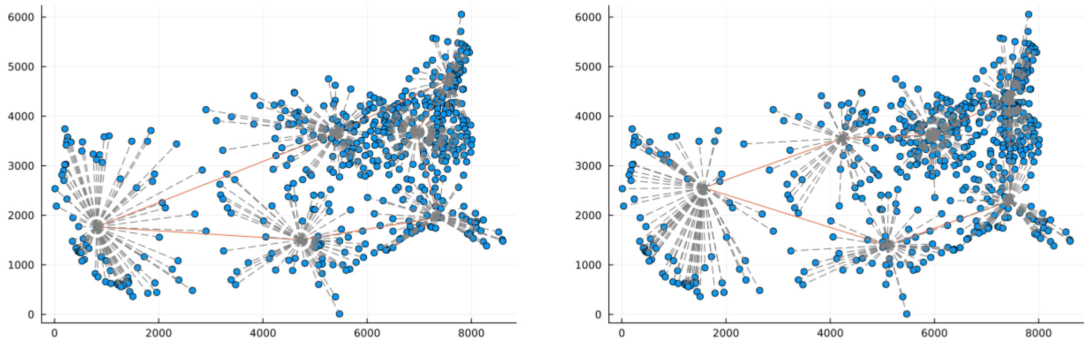

$p = 10$ stations    $p = 20$ stations

Figure 2: Solutions PLNE pour l'instance berlin52




$p = 10$ stations $\qquad$ $p = 20$ stations

Figure 3: Solutions heuristiques et métaheuristiques pour l'instance eil101




Solution 1 $\qquad$ Solution 2

Figure 4: 2 solutions avec un clustering différent pour la méthode heuristique et métaheuristique