

# Projet MAOA

## Autour du problème des tournées de techniciens

Lachiheb Sarah

Sorbonne Université Pierre et Marie Curie  
Master 2 Androide

26 Janvier 2019

- 1 Résolution par méthode approchée
  - Résolution par méthode approchée : GRASP
  - Résolution par méthode approchée : ELS
  - Résolution par méthode approchée : Conclusion
  - Résolution par méthode approchée : Expérimentation
- 2 Résolution par méthode exacte
- 3 Idée de résolution avec ajout des techniciens par méthode exacte : trois indices

On utilise une méthaheuristique de type **GRAPS/ELS**, une hybridation entre :

- **GRASP**
- Une recherche locale évolutionnaire de type **ELS**

Le problème du CVRP est extrêmement combinatoire  
⇒ D'où l'utilisation de la recherche locale.

On relaxe la contrainte de la capacité des véhicules.

La méthode **GRASP** est composée de 2 phases :

- Une initialisation randomisée avec **Route first - Cluster seconde**
- Une amélioration avec la **Recherche locale**

**Route first** : Construction d'une solution du VRP basée sur la technique des K plus proches voisins randomisés.

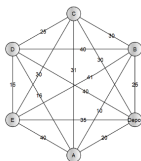


FIGURE 1 – Graphe complet

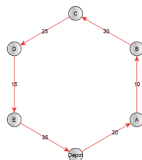
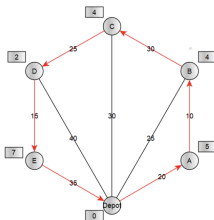


FIGURE 2 – Tour géant

**Cluster second** : On cherche à trouver les différentes tournées pour construire un élément de l'espace des solutions.

↔ Entrepris un découpage du tour géant en tournées respectant les capacités des véhicules via l'algorithme de **Split**

Découpe de la meilleure façon une séquence de clients en tournées de véhicule de capacité limitée.



**FIGURE 3** – Tour géant sur l'ensemble des clients

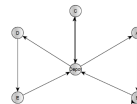
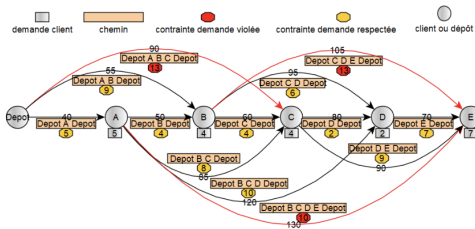


FIGURE 4 – Graphe auxiliaire des tournées possibles

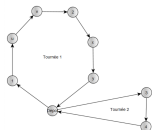
FIGURE 5 – Sous tours optimaux ici

⇒ Pas possible d'avoir un graphe avec circuit. Il permet d'utiliser ainsi l'algorithme de Bellman pour calculer le plus court chemin entre les différentes tournées possibles

**Recherche locale** : Permet d'améliorer la solution candidate de départ en cherchant la meilleure solution par une succession de transformation.

- **2-opt-intra tournée**
- **2-opt-inter tournée**

Dans une première phase :



**FIGURE 6** – Proposition initiale  
T1



**FIGURE 7** – Remplacement dans  
T1



Dans un second temps :



FIGURE 8 – Selection de T1 et T2

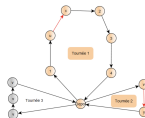


FIGURE 9 – Selection d'un arc de T1 et d'un arc de T2



FIGURE 10 – Remplacement des arcs



FIGURE 11 – Remplacement des sommets

# Résolution par méthode approchée : ELS

La méthode **ELS** :

- A partir d'une solution candidate du CVRP sous la forme d'un ensemble de tournées, l'ELS concatène la suite des tournées pour générer un tour géant
- il y a génération de p-solutions enfants sur lesquels on applique une perturbation sous forme de mutation légère
- Applique alors l'algorithme de Split afin d'obtenir un ensemble de tournée.

Pour acquérir une suite d'optimum locaux, on emploie les trois recherches locales vue et parmi les p enfants, on remplace la solution actuelle par le meilleur enfant en cas d'amélioration

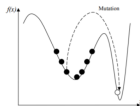


FIGURE 12 – Intérêt mutation

# Résolution par méthode approchée : Conclusion

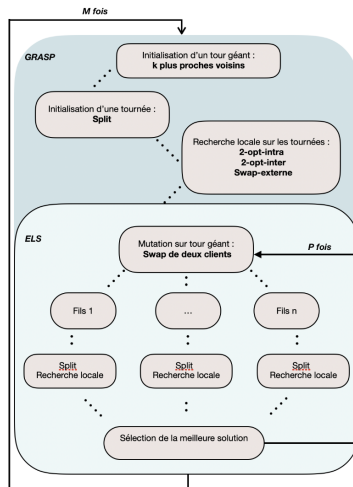


FIGURE 13 – Métaheuristique  
GRASPxELS

# Résolution par méthode approchée : Expérimentation(1)

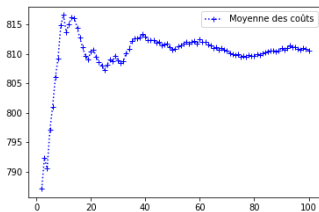


FIGURE 14 – Moyenne des coûts de l'instance A-n32-k8 optimal 784

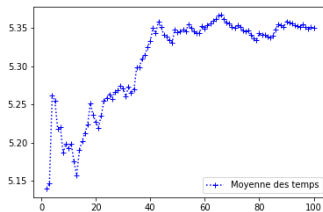


FIGURE 15 – Moyenne des temps de l'instance A-n32-k8

# Résolution par méthode approchée : Expérimentation(2)

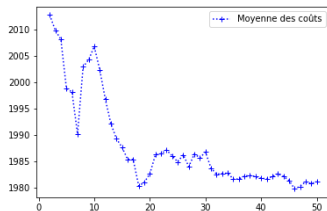


FIGURE 16 – Moyenne des coûts de l'instance A-n80-k10 optimal 1763

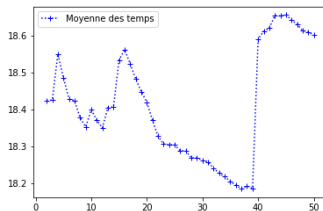


FIGURE 17 – Moyenne des temps de l'instance A-n80-k10

# Résolution par méthode approchée : Expérimentation(3)

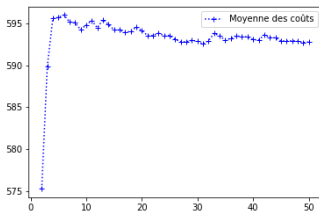


FIGURE 18 – Moyenne des coûts de l'instance P-n55-k7 optimal 568

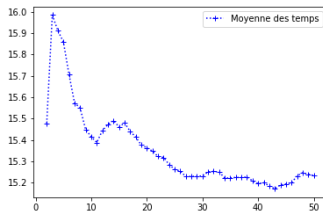


FIGURE 19 – Moyenne des temps de l'instance P-n55-k7

# Résolution par méthode approchée : Expérimentation(4)

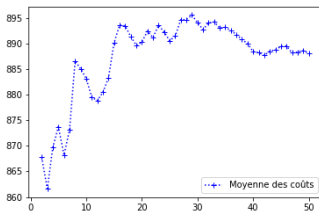


FIGURE 20 – Moyenne des coûts  
de l'instance M-n101-k10 optimal  
820

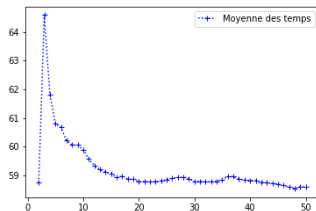


FIGURE 21 – Moyenne des temps  
de l'instance M-n101-k10

# Résolution par méthode approchée : Expérimentation(3)

Instance	Moyenne du temps (s)	Moyenne coût	Meilleur coût	% opti- mal meilleur	% opti- mal moyenne
A-n32-k8	5.35	810.61	787.082	0,40	3,2
A-n54-k7	8.35	1253.42	1190.9	1,97	6,0
A-n80-k10	18.60	1981.10	1894.74	6,9	11
P-n16-k8	1.97	451.94	451.34	0,42	0,29
P-n22-k2	2.70	222.92	217.85	0,82	3
P-n55-k7	15.23	592.82	575.26	1,21	4
P-n101-k4	104.871	737.132	705.006	3.5	8,24
M-n101-k10	58.59	888.07	828.28	0,97	7.66
M-n200-k17	235.345	1554.902	1486.22	16.6	21.9

TABLE 1 – Valeurs expérimentaux



- 1 Résolution par méthode approchée
- 2 Résolution par méthode exacte
  - Résolution par méthode exacte : deux indices
  - Résolution par méthode exacte : trois indices
- 3 Idée de résolution avec ajout des techniciens par méthode exacte : trois indices

On utilise différents programmes linéaires :

- Un à **deux indices**, dans un graphe orienté
- Un à **trois indices**, dans un graphe non orienté

# Résolution par méthode exacte : deux indices

La méthode utilisée est celle décrite dans le projet de MAOA :

- Formulation MTZ avec renforcement  
ILOUSERCUTCALLBACK et ILOLAZYCONSTRAINTCALLBACK  
avec la coupe minimum où l'on applique la coupe donnée dans  
l'énoncé.

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \quad |S| \geq 2$$

- Formulation par les inégalités de coupes  
ILOLAZYCONSTRAINTCALLBACK : Détection de cycle sans  
dépôt nommé  $W$  avec rajout de la contrainte :

$$\sum_{i \in W} \sum_{j \in \{0,1,\dots,n\} \setminus W} x_{ij} \geq \left\lceil \frac{\sum_{i \in W} d_i}{Q} \right\rceil$$

et détecte aussi les cycles avec dépôt ne respectant pas la capacité du véhicules, on rajoute alors la même contrainte.

# Résolution par méthode exacte : trois indices

$$\min \quad \sum_{e \in E} c_e \sum_{k=1}^K x_{ek}$$

$$\text{s.c} \quad \sum_{k=1}^K y_{ik} = 1 \quad \forall i \in V / \{0\}$$

$$\sum_{k=1}^K y_{0k} = K$$

$$\sum_{e \in \delta(i)} x_{ek} = 2y_{ik} \quad \forall i \in V, k = 1, \dots, K$$

$$\sum_{i \in V} d_i y_{ik} \leq C \quad \forall k = 1, \dots, K$$

$$\sum_{e \in \delta(S)} x_{ek} \geq 2y_{hk} \quad \forall S \subseteq V / \{0\}, h \in S, k = 1, \dots, K$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in V, k = 1, \dots, K$$

$$x_{ek} \in \{0, 1\} \quad \forall e \notin \delta(0), k = 1, \dots, K$$

$$x_{ek} \in \{0, 1, 2\} \quad \forall e \in \delta(0), k = 1, \dots, K$$

Pour les variables de décisions  $x_{ek}$  entières par  
ILOLAZYCONSTRAINTCALLBACK

Dans le cadre de ce programme linéaire, la contrainte (5) impose que la route d'un véhicule soit connexe.

Mais elle peut être remplacée par une contrainte d'élimination de sous tours (SECs) :

$$\sum_{i \in W} \sum_{j \in \{0,1,\dots,n\} \setminus W} x_{ek} \geq 2 \lceil \frac{\sum_{i \in W} d_i}{Q} \rceil \forall k \in 1, \dots, K$$

⇒ Impose que pour chaque véhicule  $k$  au moins 1 arc quitte chaque ensemble de sommets  $S$  visité par  $k$  et ne contenant pas le dépôt.

Pour les variables de décisions  $x_{ek}$  fractionnaire par  
ILOUSERCUTCALLBACK

On créer le graphe Lemon avec le coût des arcs fractionnaires dans le but d'utiliser l'algorithme de coupe minimale de Nagamochi-Ibaraki. Une fois cette coupe  $S$  récupérée on applique la contraint :

$$\sum_{e \in W} \sum_{e \notin W} x_{ek} \geq 2$$

Seulement si le nombre de client dans la coupe est supérieur ou égale à 2.

- 1 Résolution par méthode approchée
- 2 Résolution par méthode exacte
- 3 Idée de résolution avec ajout des techniciens par méthode exacte :  
trois indices

# Idée de Résolution avec ajout des techniciens par méthode exacte : trois indices

Pour une heuristique avec les capacités des techniciens nous pourrions lancer notre métaheuristique vue précédemment mais en plusieurs étapes :



FIGURE 22 – Schéma tiré de l'article[11] donné pour ce sujet.



Merci de votre attention