Présentation Fil rouge

Yishan Sun, Simon Kurney, Pablo Aldana, Cédric Jung, Baptiste Deconihout, Zoé Poupardin
Groupe ICOnique

Sommaire

Gestion Projet

Méthode Tabou

Cédric
JUNG

Yishan
SUN

Méthode Recuit Simulé

Zoé
POUPARDIN

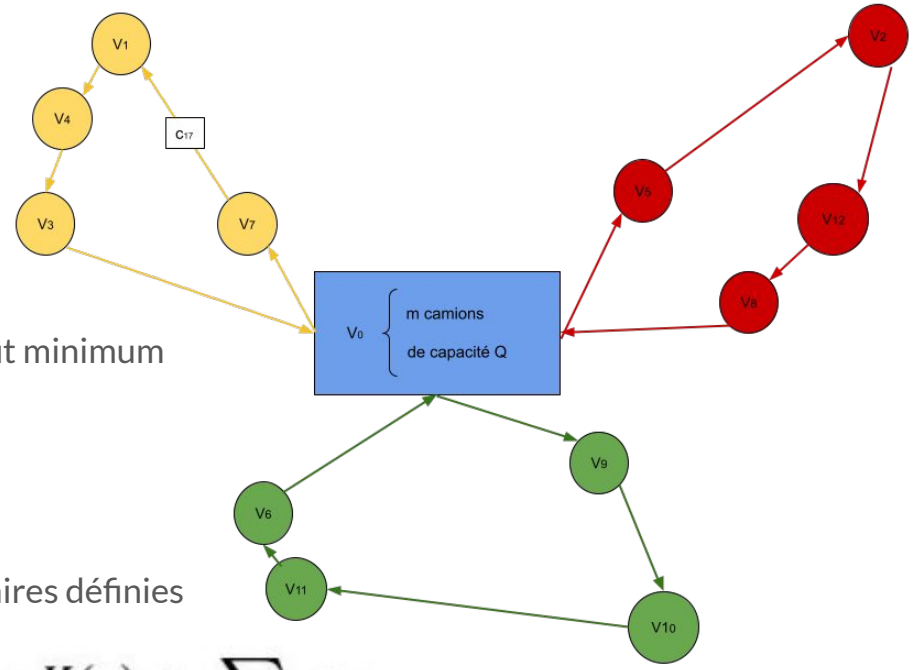
Baptiste
DECONIHOUT

Méthode Algorithme Génétique

Pablo
ALDANA

Simon
KURNEY

Objectif de l'ensemble



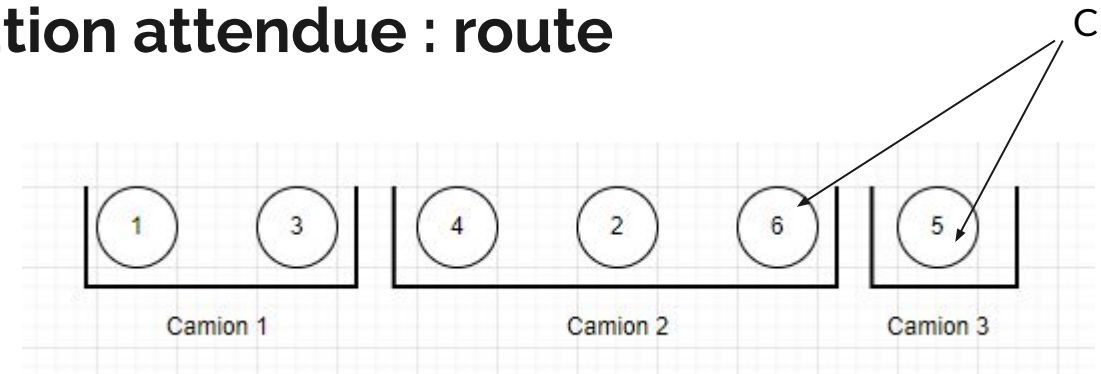
Construire des trajets pour desservir les clients pour un coût minimum

Contraintes :

- Les clients doivent être desservi dans des plages horaires définies
- Le nombre de camion n'est pas fixé

$$f(x) = \omega K(x) + \sum_{(i,j)\in E} c_{ij}$$

Choix de la forme de la solution attendue : route

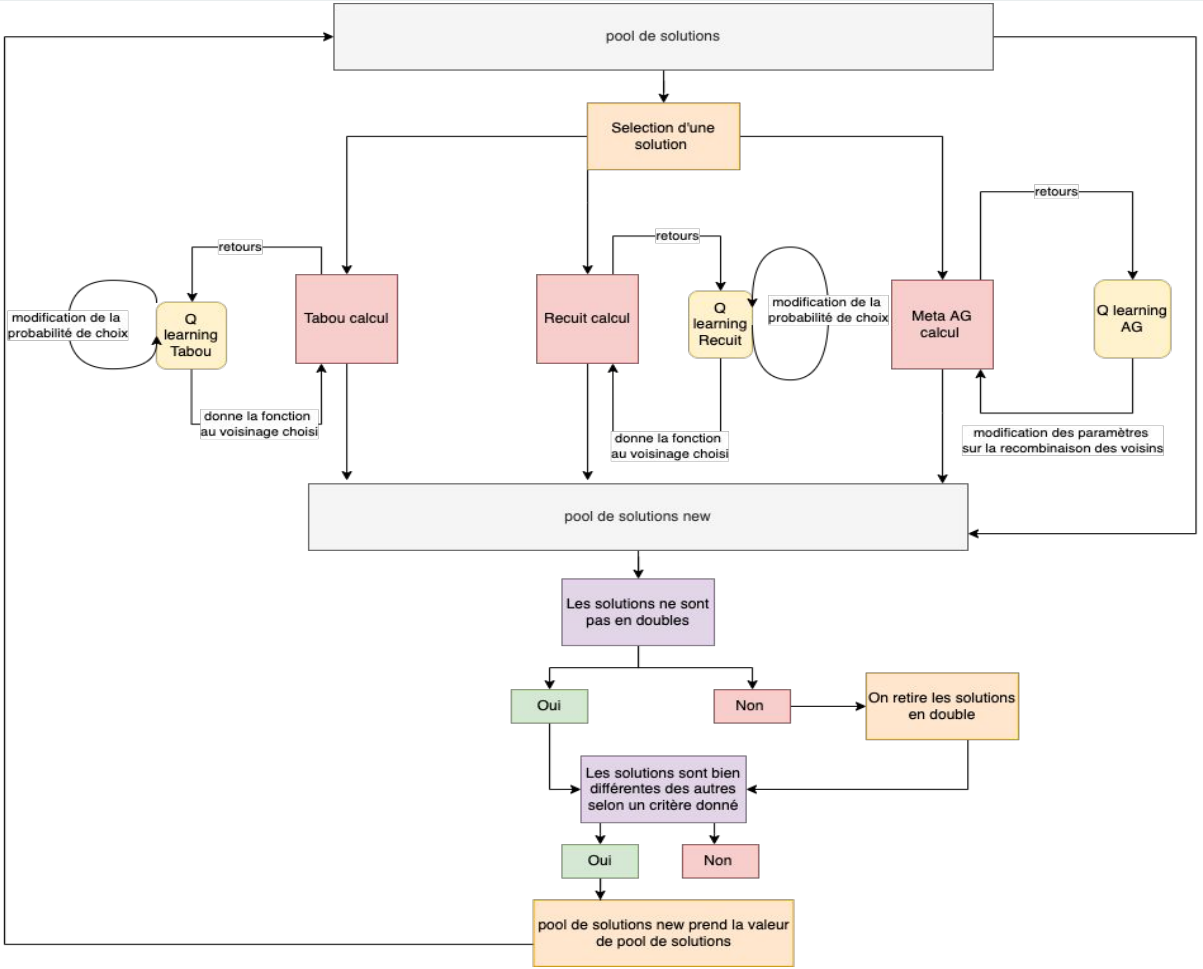



```
sol example = [[1, 3], [4, 2, 6], [5]]
```

Paramètres pour le calcul du coût :

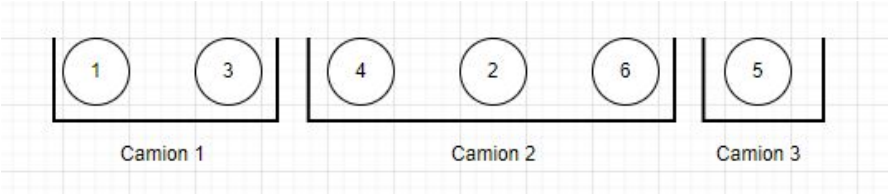
1. Nombres de camions et coût des routes
2. Volumes des camions
3. Temps et fenêtre de livraison

Structure choisie



Métaheuristiques

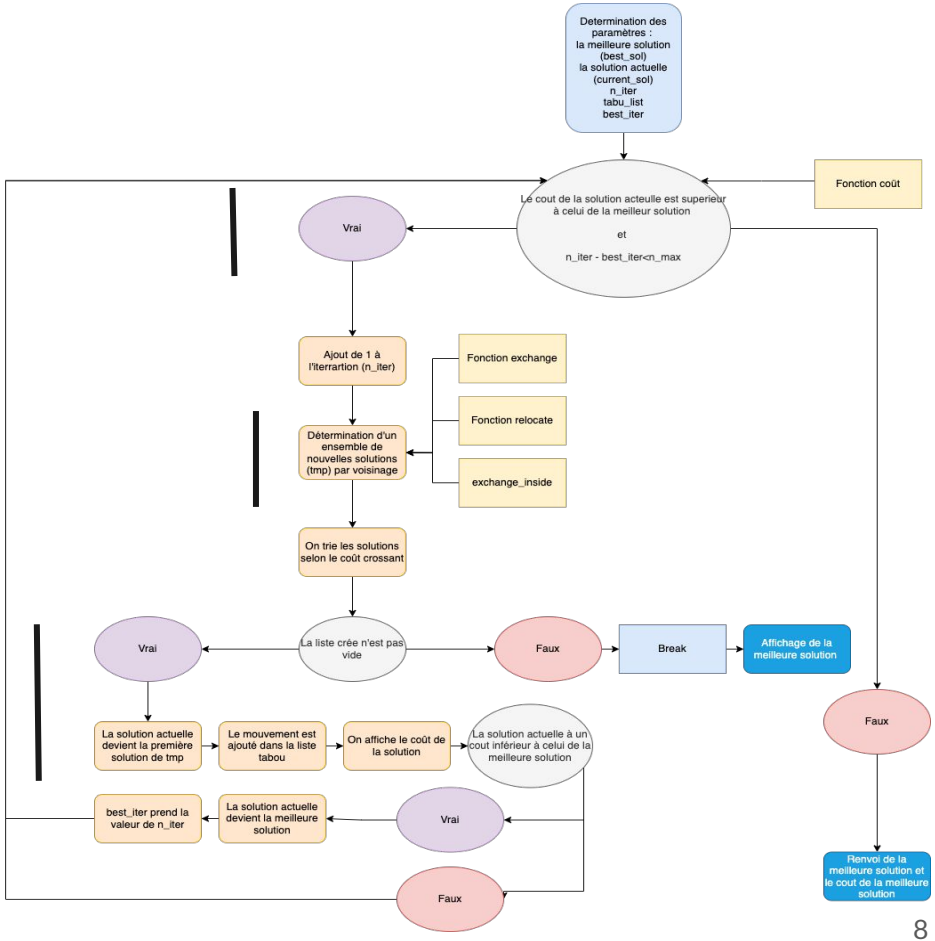
Méthode Tabou



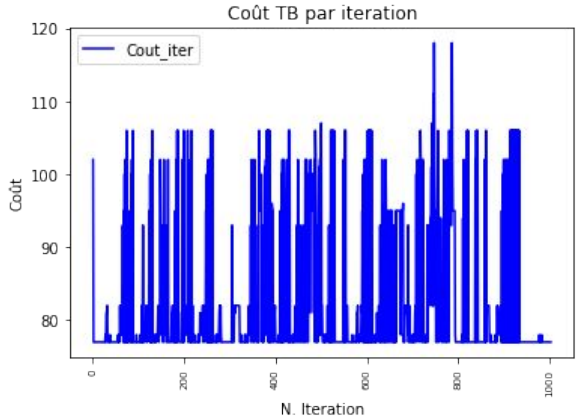
Etape 1 : Création de la nouvelle solution - Fonctions “exchange”, “relocate”, “exchange_inside”;

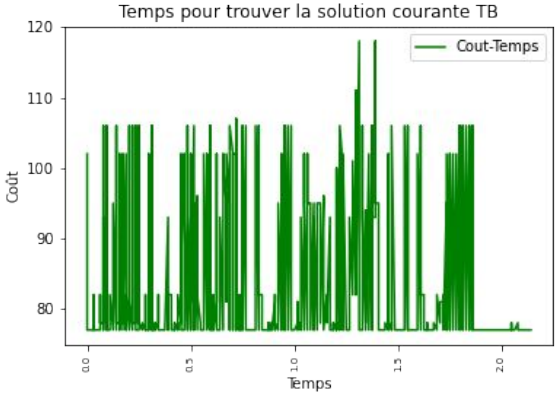
Etape 2 : Les listes trouvées peuvent-elles correspondre à des solutions? - Fonction “acceptable” (vis-à-vis de la capacité des camions, et de la liste tabou)

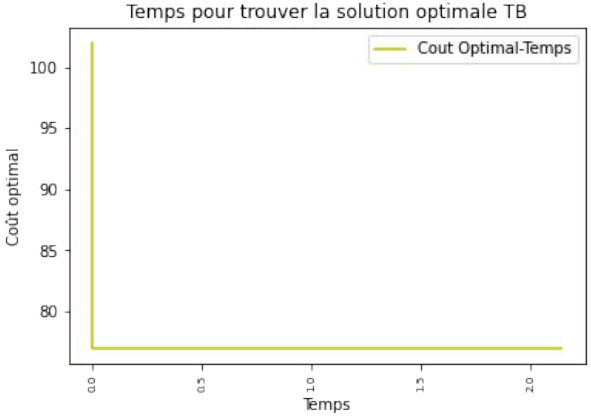
Etape 3 : La solution trouvée est-elle meilleure que la meilleure solution d’après le critère coût? - Fonction “coût”

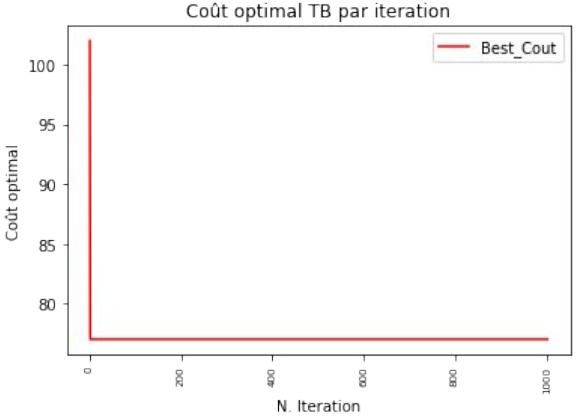


Courbes et Tableaux sans capacité







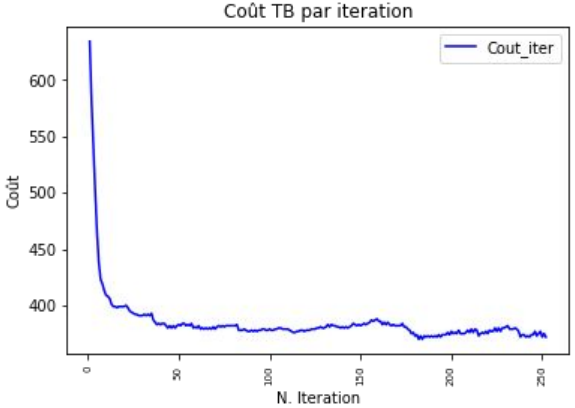


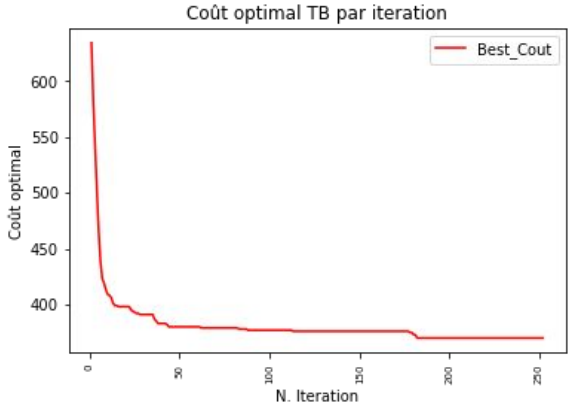
Solution initiale : [[1, 2, 5], [4, 3]],

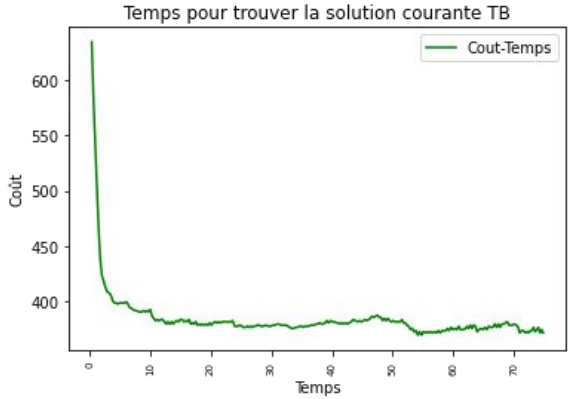
Solution : [[1, 5], [4], [2, 3], []]

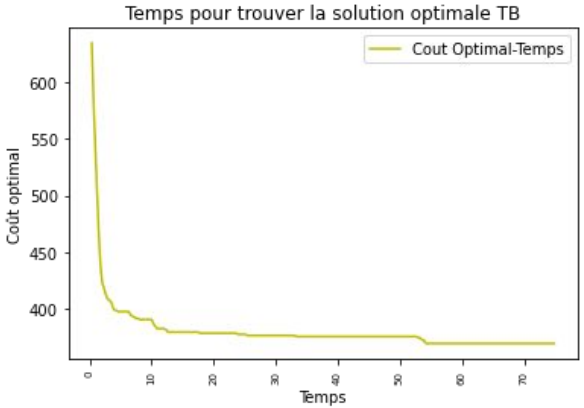
Coût : 77

Courbes et Tableaux avec capacité









Fonction **filter_on_capacities :**

Ne garde que les solutions qui répondent à une capacité maximale de chaque camion pour chaque route.

Exemple :

Nombre de clients : 51

Poids : ω=5 pour le calcul du coût

Nombre d'itération maximum : 100

Facteur d'aspiration : 3

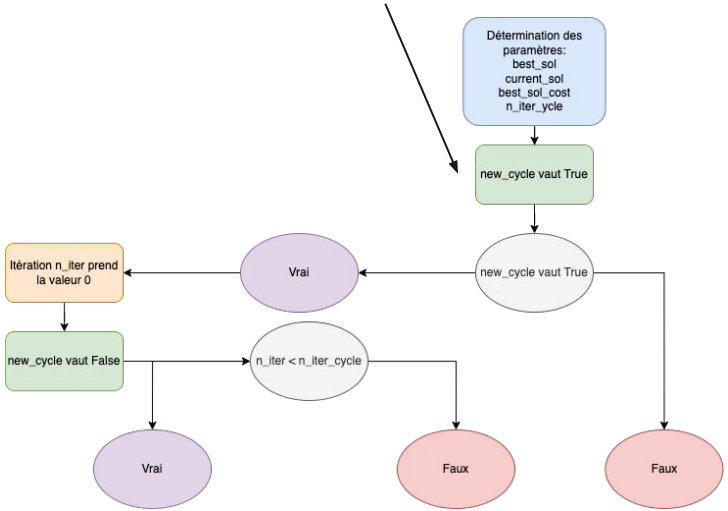
Capacité maximale des camion : 100,

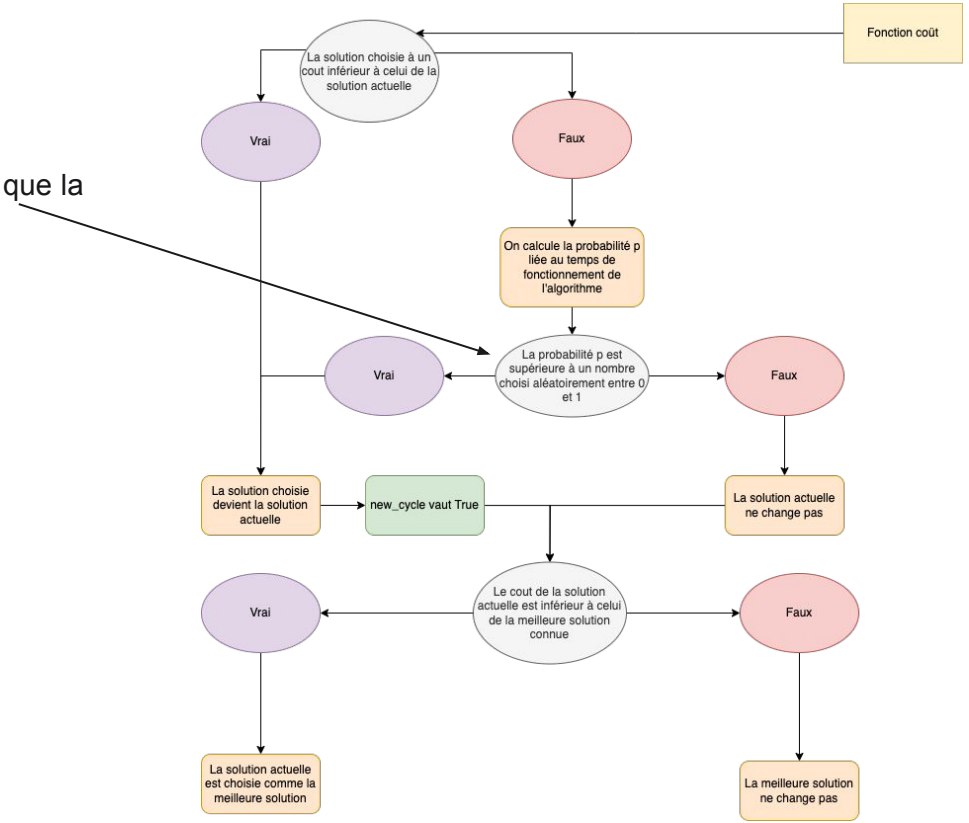
Taille de demande client : 1,

Coût obtenu : 370

Méthode Recuit simulé

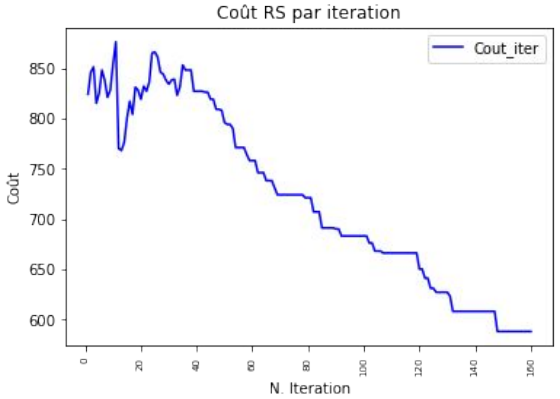
- Méthode similaire à celle de Tabou;
- Accepte de garder une solution qui n'est pas meilleure que la précédente : à maximum locale;
- Probabilité liée au temps de recherche de la solution.

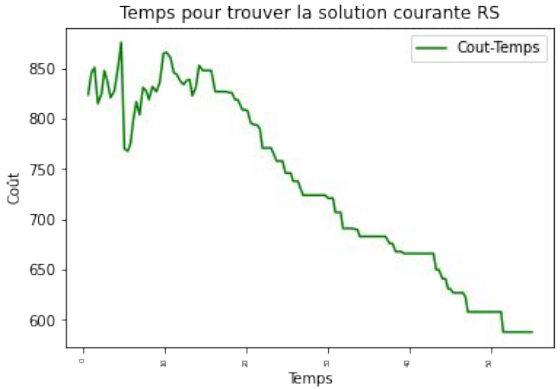


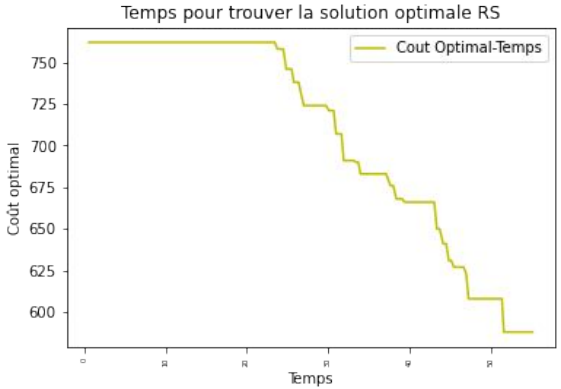


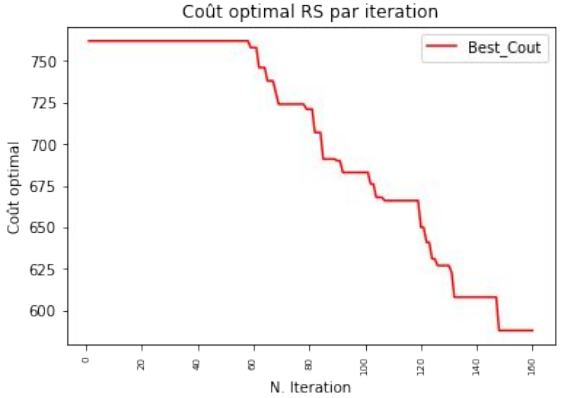
Courbes et Tableaux avec capacité

Exemple (même que précédemment)







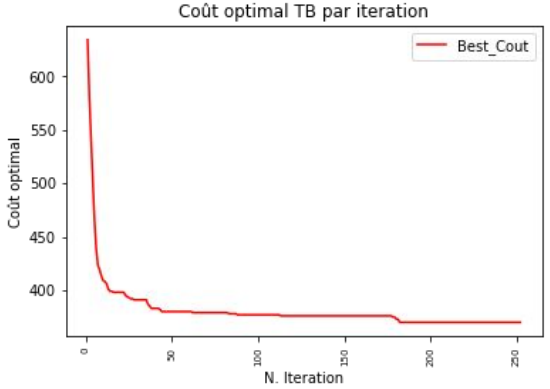


Nombre de clients : 51

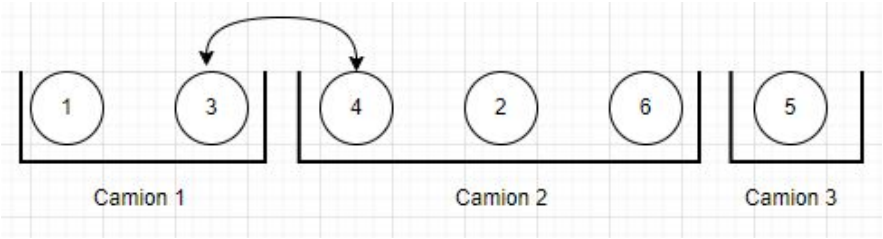
Poids : ω=5 pour le calcul du coût

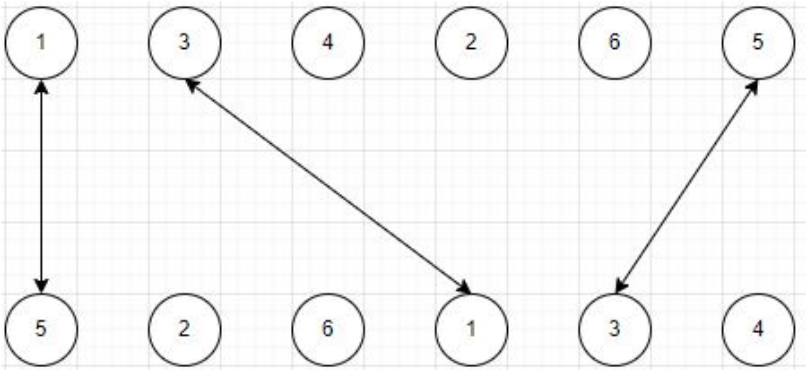
Nombre d'itération maximum : 100

Facteur d'aspiration : 3



Méthode Algorithme génétique





Etape 0 : Définir un codage du problème

Étape 1 : t:=0, créer une population initiale de N individus P(0) = x1,x2,...,xN

Etape 2 : Evaluation

- Calculer la force F(xi) de chaque individu xi, i=1...N

Etape 3 : Sélection

- Sélectionner N individus de P(t) et les ranger dans un ensemble S(t). Un même individu de P(t) peut apparaître plusieurs fois dans S(t)

Etape 4 : Recombinaison Grouper les individus de S(t) par paire, puis, pour chaque paire d'individus :

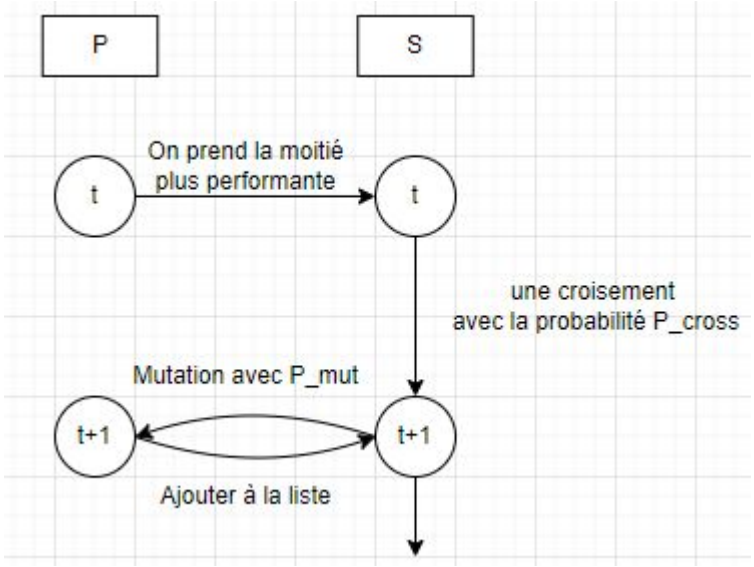
- avec la probabilité Pcross, appliquer le croisement à la paire et recopier la progéniture dans S(t+1) (la paire d'individus est éliminée, elle est remplacée par sa progéniture),
- avec la probabilité 1-Pcross, recopier la paire d'individus dans S(t+1)

Pour chaque individu de S(t+1) :

- avec la probabilité Pmut, appliquer la mutation à l'individu le recopier dans P(t+1)
- avec la probabilité 1-Pmut, recopier l'individu dans P(t+1)
- Etape

5 : Incrémenter t et reprendre à l'étape 2 jusqu'à un critère d'arrêt

Méthode Algorithme génétique



Etape 0 : Définir un codage du problème

Étape 1 : t:=0, créer une population initiale de N individus P(0) = x1,x2,...,xN

Etape 2 : Evaluation

- Calculer la force F(xi) de chaque individu xi, i=1...N

Etape 3 : Sélection

- Sélectionner N individus de P(t) et les ranger dans un ensemble S(t). Un même individu de P(t) peut apparaître plusieurs fois dans S(t)

Etape 4 : Recombinaison Grouper les individus de S(t) par paire, puis, pour chaque paire d'individus :

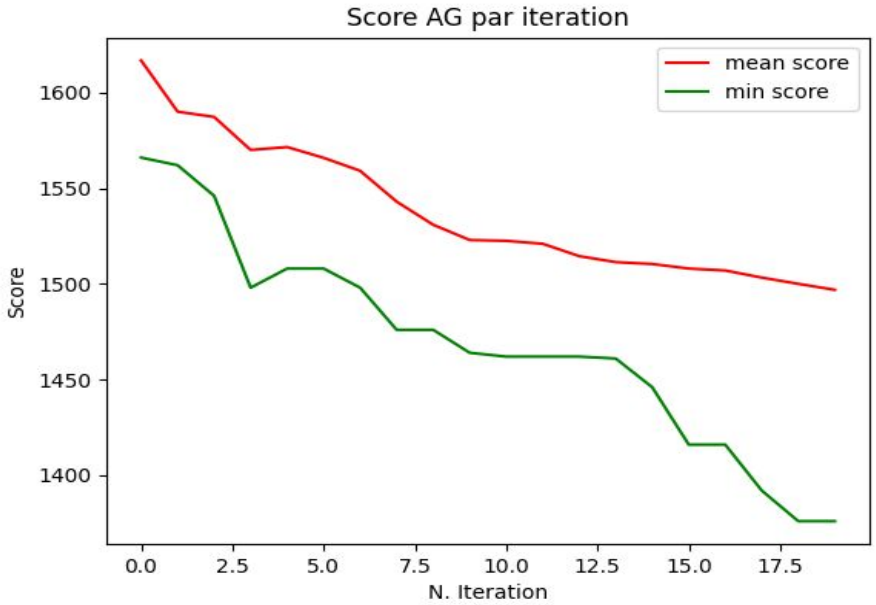
- avec la probabilité Pcross, appliquer le croisement à la paire et recopier la progéniture dans S(t+1) (la paire d'individus est éliminée, elle est remplacée par sa progéniture),
- avec la probabilité 1-Pcross, recopier la paire d'individus dans S(t+1)

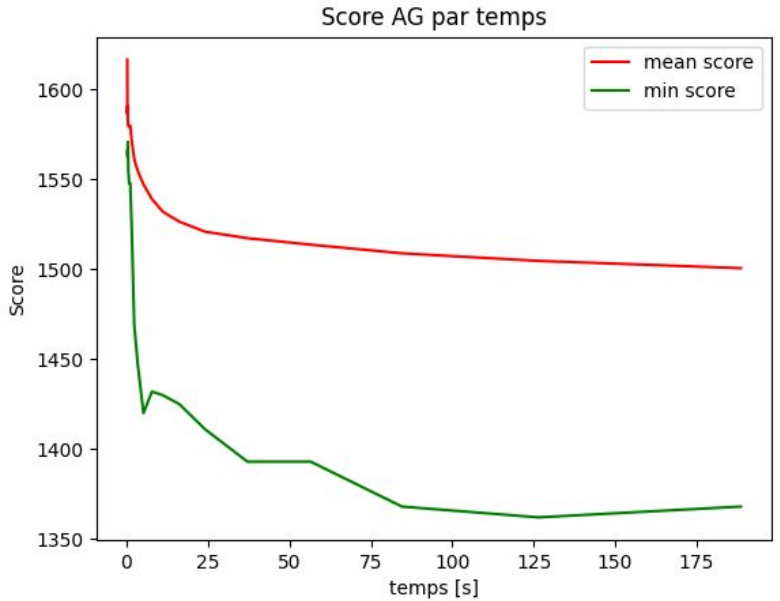
Pour chaque individu de S(t+1) :

- avec la probabilité Pmut, appliquer la mutation à l'individu le recopier dans P(t+1)
- avec la probabilité 1-Pmut, recopier l'individu dans P(t+1)
- Etape

5 : Incrémenter t et reprendre à l'étape 2 jusqu'à un critère d'arrêt

Méthode Algorithme génétique

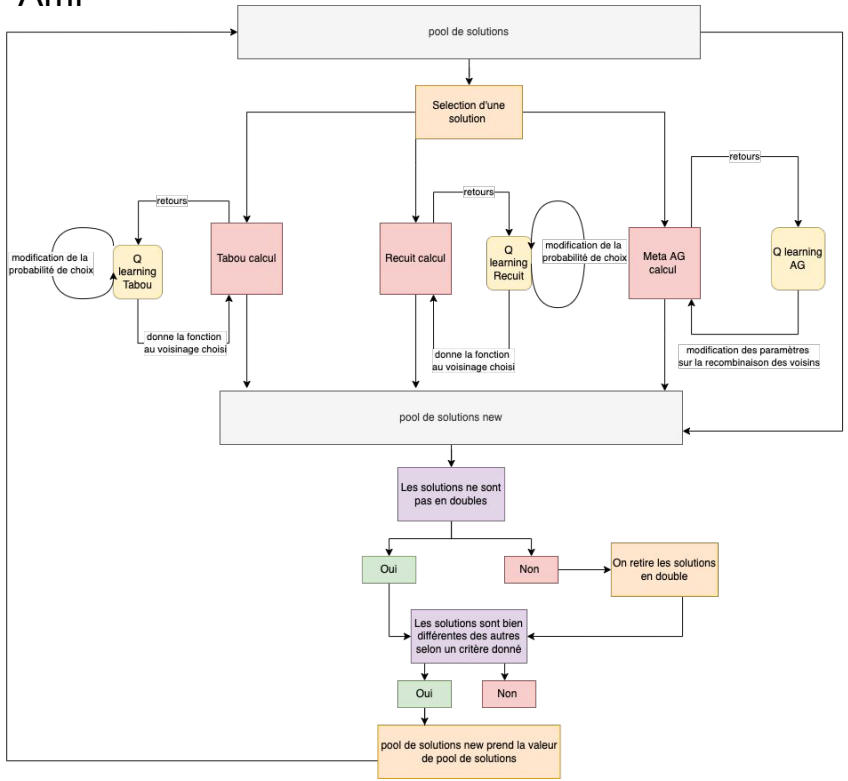




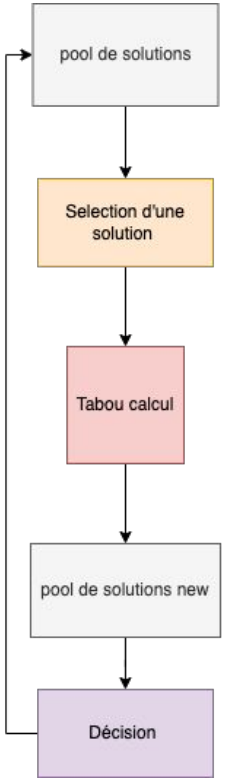
Système Multi-Agent

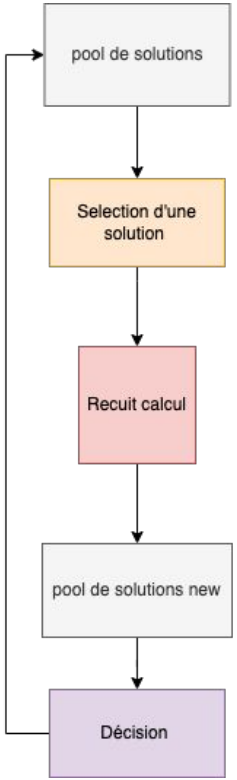
Interaction Ennemi - Ami

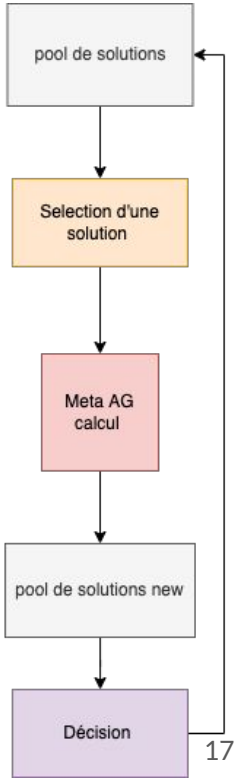
Ami



Ennemi

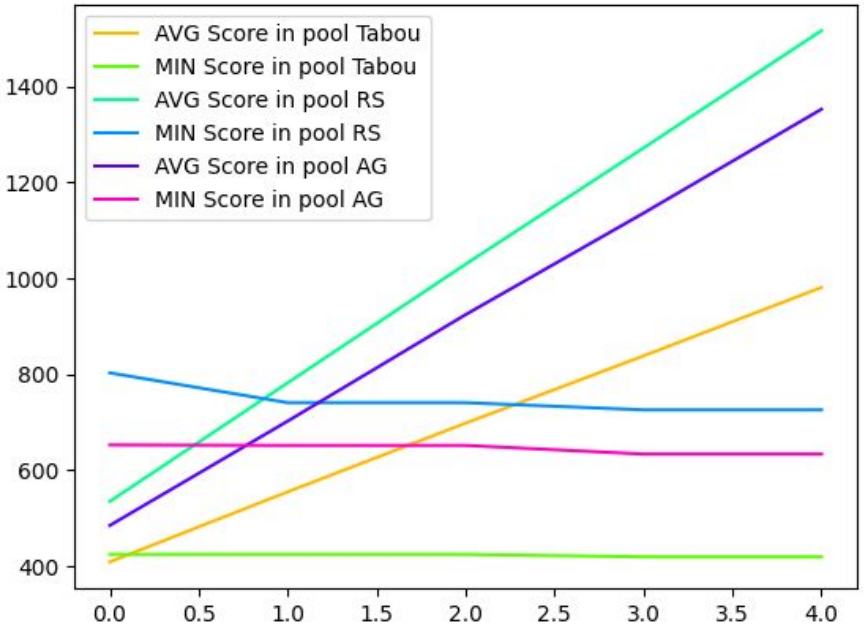




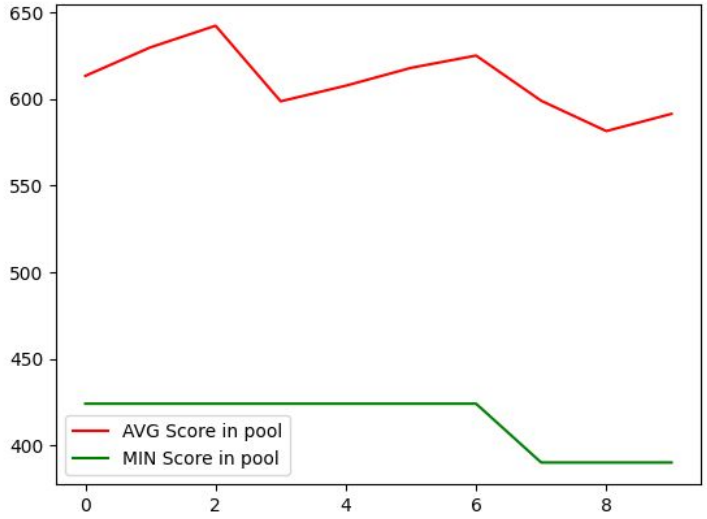


Résultat

Volume pour chaque client: 3
Nombre de clients: 50
Coût associé au nombre de camion: 5



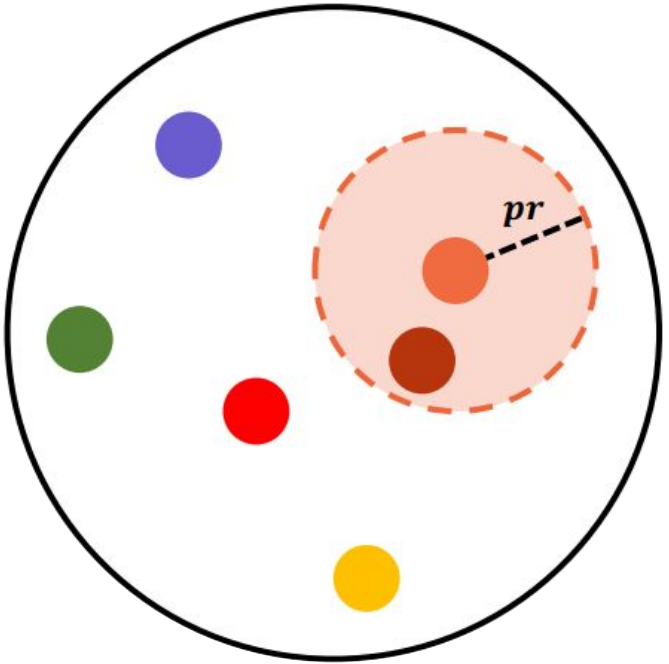
Approche Ennemie



Approche Collaborative

Distance de solution pool

$$g(\phi_i) = \sum_{j=1}^{P} \phi(\lambda_{ij}) \qquad \text{où} \qquad \phi(\lambda_{ij}) = \begin{cases} 1 - \dfrac{\lambda_{ij}}{pr} & \text{si } \lambda_{ij} \leq pr \\ 0 & \text{si } \lambda_{ij} > pr \end{cases}$$

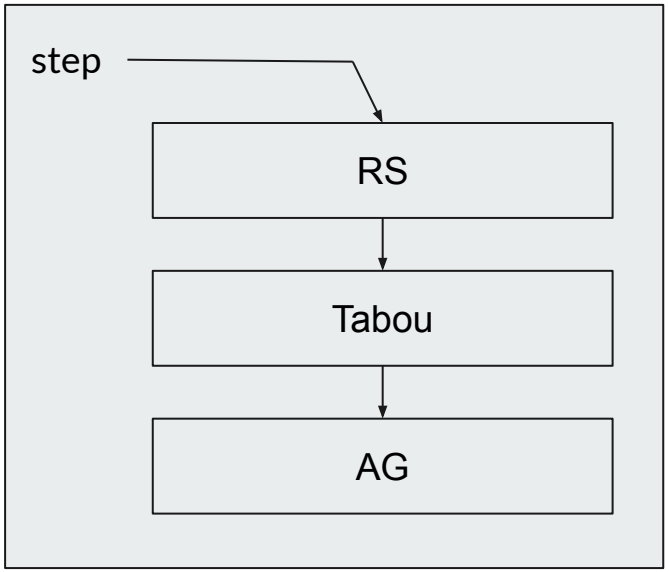


Comparaison entre deux solutions: Nombre d'arrêts en commun.

Modification du scheduler de MESA

Avant

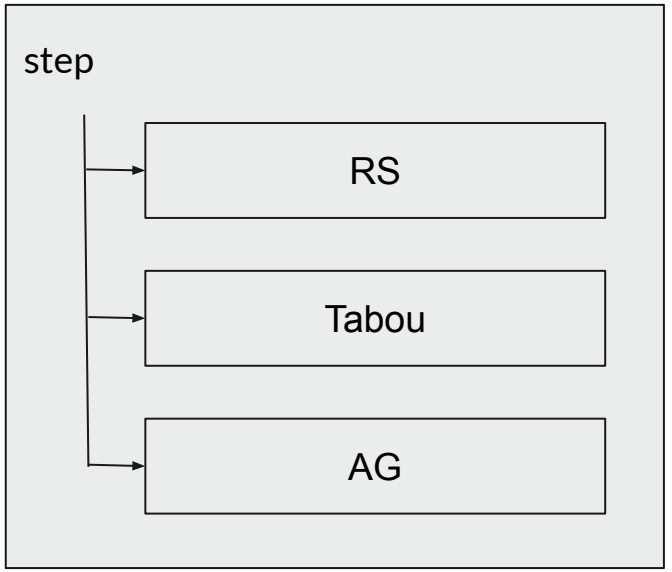
RandomActivation



Utilisation d'un seul coeur du processeur

Après

MultiProcessActivation



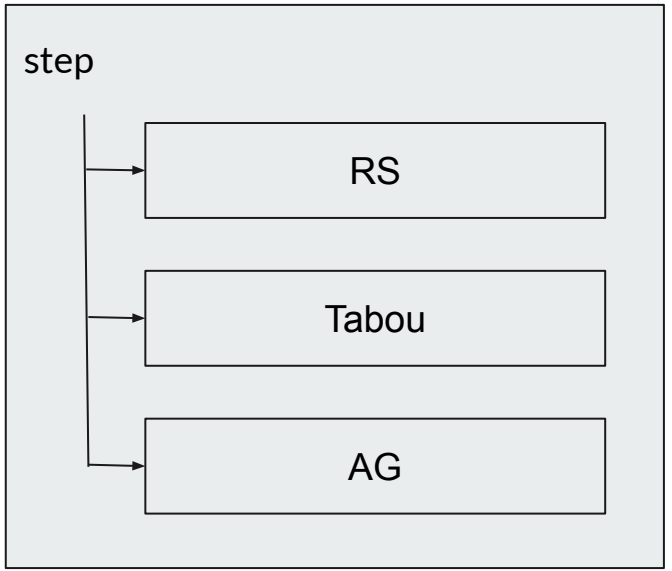
Utilisation de 3 coeurs du processeur en simultané

Modification du scheduler de MESA

```
S  0.0  1.1  0:01.53 └─ python3 big_example.py
S  0.0  0.7  0:00.00    ├─ python3 big_example.py
S  0.0  0.7  0:00.00    │  └─ python3 big_example.py
R 97.8  0.8  0:02.43    ├─ python3 big_example.py
R 97.8  0.7  0:02.43    ├─ python3 big_example.py
R 100.  0.9  0:02.49    └─ python3 big_example.py
S  1.9  0.9  0:00.05       └─ python3 big_example.py
```

Utilisation de 3 coeurs du processeur en simultané

MultiProcessActivation



QLearning

QLearning Tabou - Recuit simulé

8 façons de modifier les voisins:

-intra_route_swap,

-intra_route_shift,

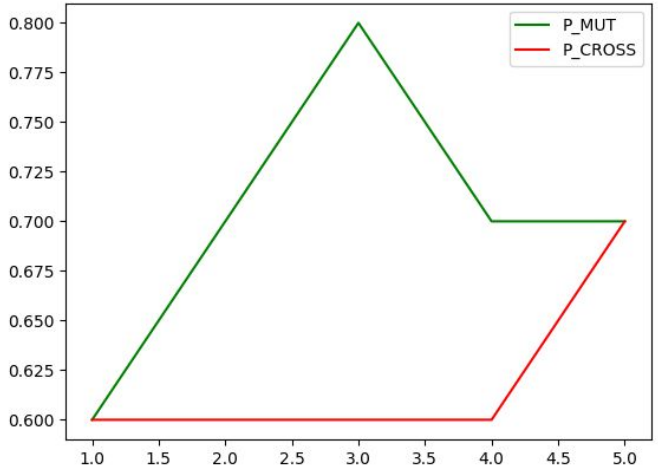
-two_intra_route_swap,

-two_intra_route_shift,

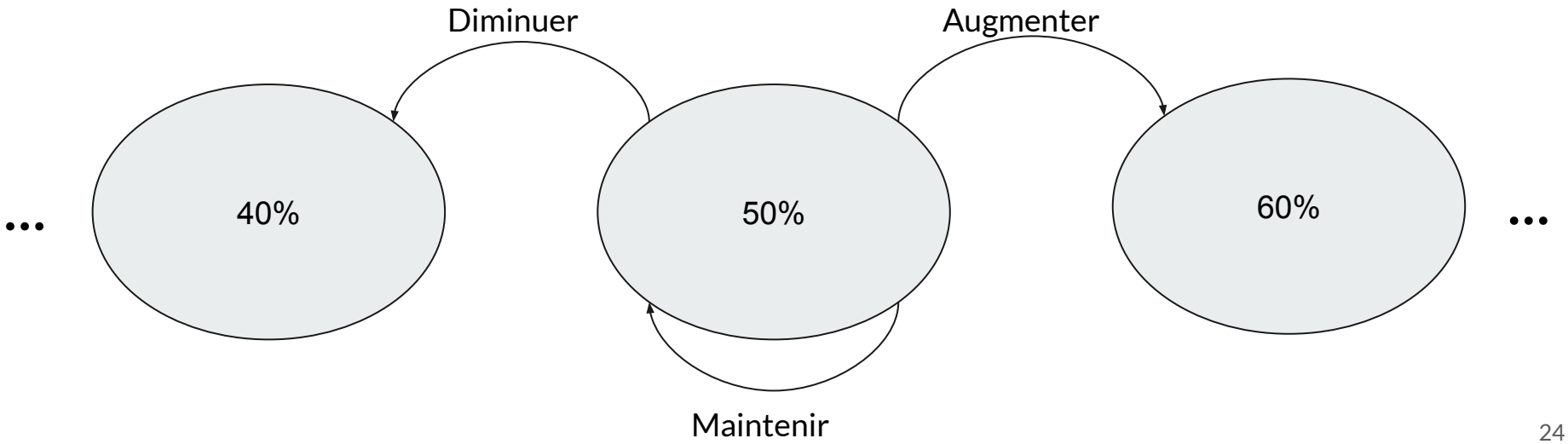
- del_small_route_w_capacity,
- del_random_route_w_capacity
- inter_route_swap_w_capacity
- inter_route_shift_w_capacity

Prise en compte de la capacité

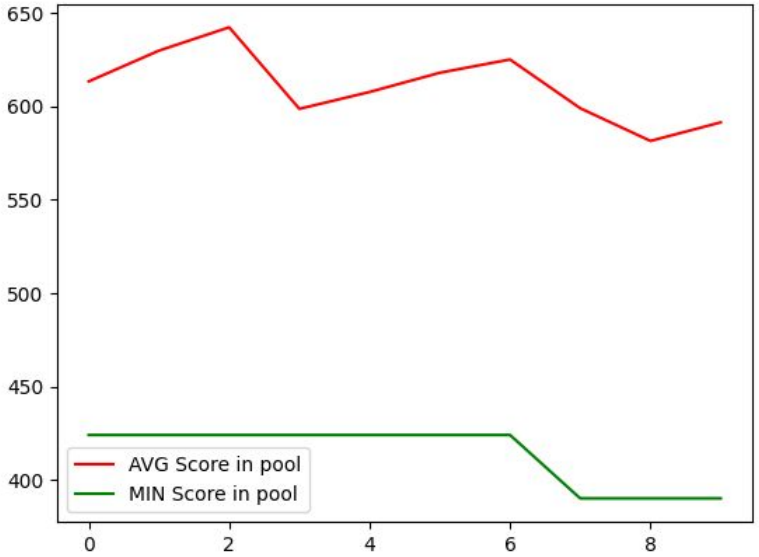
QLearning Algorithme génétique



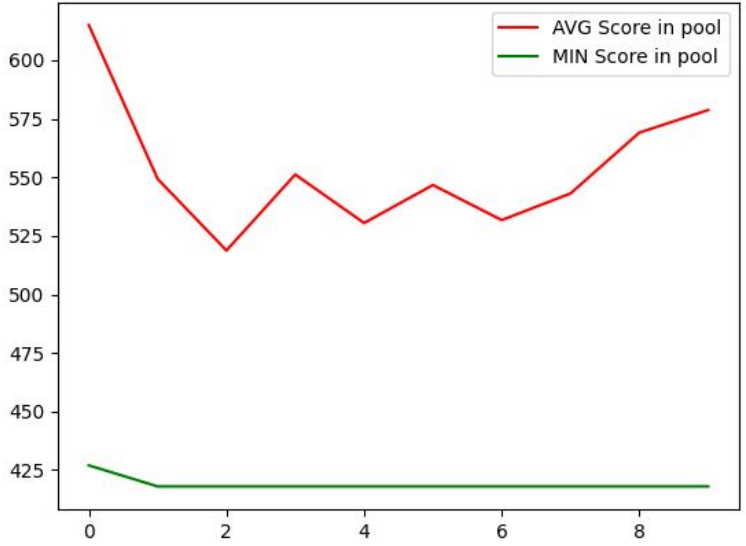
Probabilité mutation et cross



Résultats



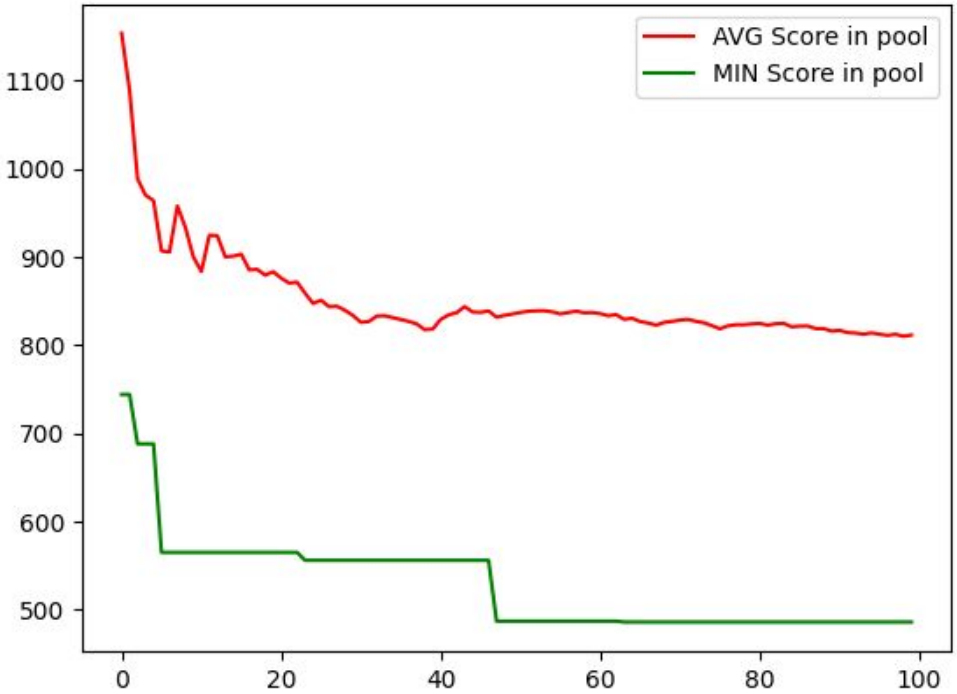
Sans QLearning



Avec QLearning

Volume pour chaque client: 3
Nombre de clients: 50
Coût associé au nombre de camion: 5

Résultats finaux



Volume pour chaque client: de 1 à 3 (aléatoirement)
Nombre de clients: 50
Coût associé au nombre de camion: 100
Avec Q-Learning

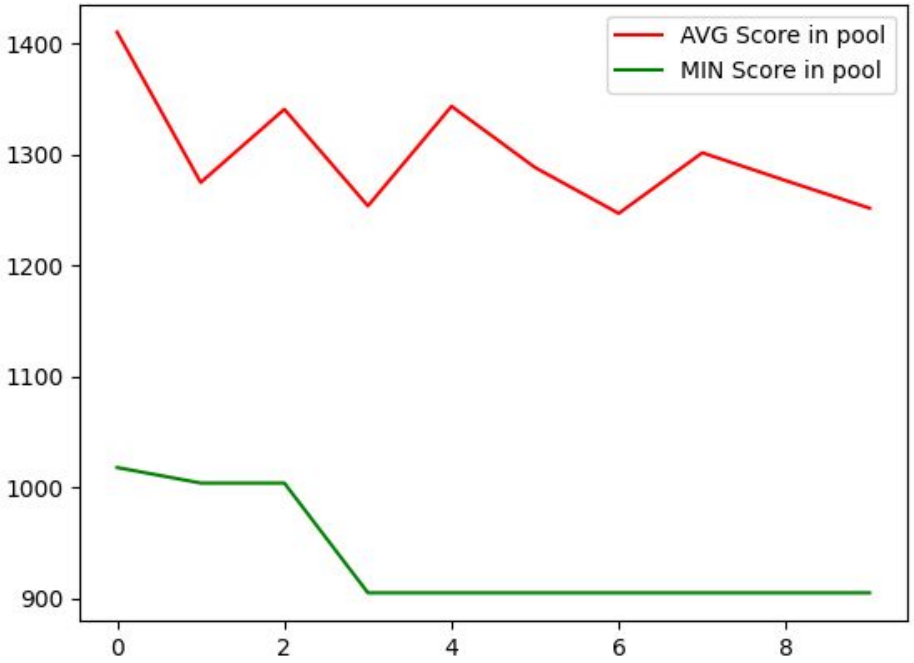
Limites du modèle

- Dans notre système nous avons des camions qui avaient une capacité constante: **prendre en compte des camions de capacité différente**

- **Le temps d'exécutions des algorithmes** aurait pu être amélioré même si on a amélioré le scheduler pour faire du travail multicore

Conclusion

Résultats finaux avec Q-learning



Volume pour chaque client: de 1 à 10 (aléatoirement)
Nombre de clients: 50
Coût associé au nombre de camion: 100
Avec Q-Learning