

# Tabou

---

## Choix des opérateurs

Il ne faut pas que les opérateurs se superposent. Besoin d'un opérateur qui permet de supprimer un camion : le relocate

## Détection de schéma

**Mettre capture d'écran d'un schéma qui se répète** Une fois le développement de la liste tabou achevé, nous avons remarqué que dans la quasi totalité de nos premiers tests, nous pouvions visuellement voir une répétition de fitness. Cette répétition arrivait assez rapidement et nous avons donc penser qu'il serait utile de réaliser une détection de schéma afin d'éviter un grand nombre d'itération inutile

## Développement

Pour réaliser cette détection, nous avons d'abord décider du système suivant : une liste contient des dictionnaire, qui eux même stockent en clé la fitness et en valeur la liste tabou converti en chaîne de caractère. L'idée derrière ce système était que si nous revenions à la même fitness avec la même liste tabou, alors nous sommes dans le même état que le début du cycle, et donc que nous allons tourner en boucle. Toute cette idée repose sur le fait que la fitness est unique pour chaque solution. Or, nous nous sommes rendu compte que ce n'était pas le cas et que deux solutions pouvaient avoir la même fitness. Ainsi, nous avons donc développer une fonction de hashage de la solution qui nous permet d'avoir une clé réellement unique et de rendre cette détection de schéma fonctionnel

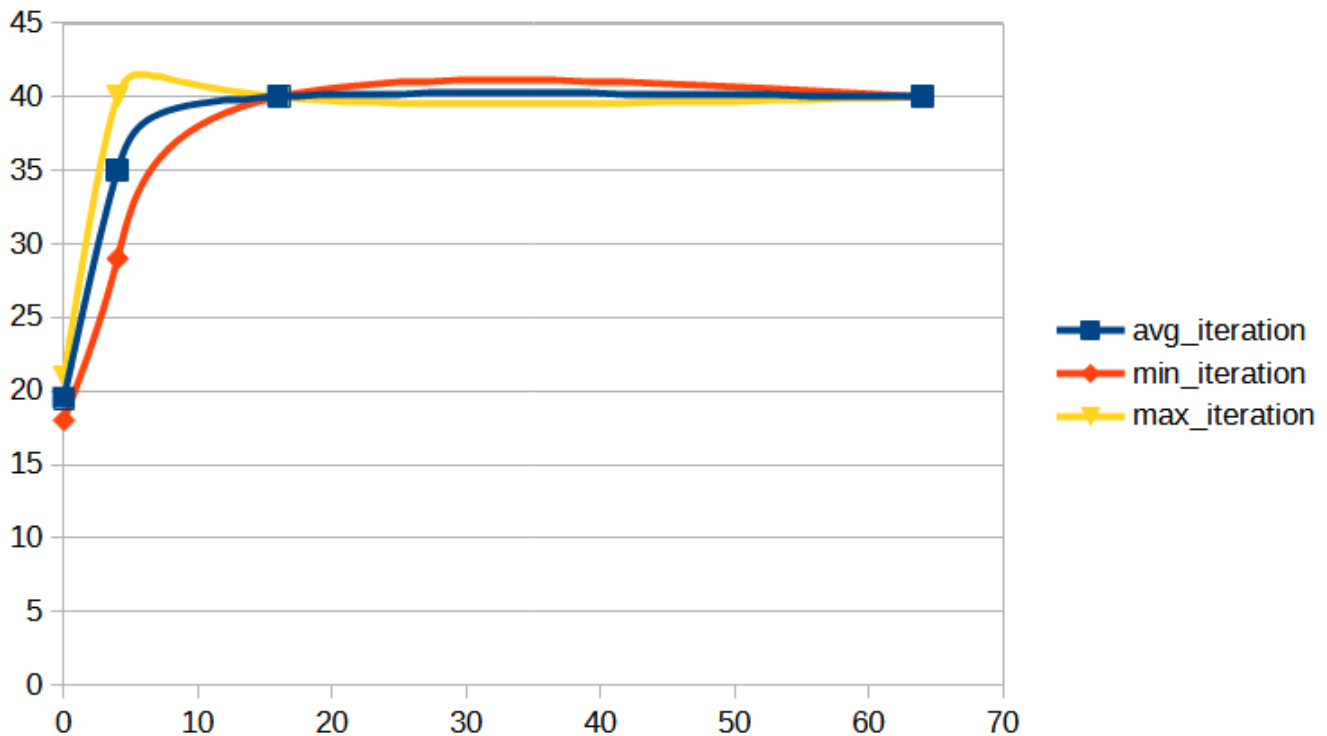
## Analyse des résultats liés

Cette fonctionnalité nous a amené à la métrique suivante : le nombre d'itération réel (en opposition avec le nombre d'itération prévu au départ). Dans la suite du rapport, nous noterons les itération réelle avec la notation IR et le nombre d'itération prévu avec la notation IP

### Analyse sur les version short (30 premiers clients)

Nous pouvons remarqué un fait intéressant sur cette donnée. Prenons par exemple le fichier output\_data101\_short.csv. Dans ce fichier, nous remarquons d'abord que pour le hill climbing (càd taille tabu = 0), le nombre de ligne effective est globalement toujours le même, et toujours inférieur au nombre d'itération prévu (environ 18 itérations réels). Cela est assez simplement interpretable : le hill climbing converge toujours en un nombre d'itération similaire sur ce jeu de données. Ainsi, nous pouvons déduire que le bon hyperparamètre pour le hill climbing est de 18 itérations. TODO : il faudrait plutot analyser le fichier de moyenne pour cela

Pour revenir au tabou,



## Choix des hyper-parametres

Quels hyper paramètres tester ?

on fait 0 itération pour voir la fitness moyenne on faire 0 tabu pour avoir les données du hill climbing

### Choix de la métrique

Pour le tabou, nous avons deux hyper-paramètres à sélectionner : la taille de la liste et le nombre d'itération visé. Pour trouver les hyper-paramètres les plus adaptés, il faut ainsi voir le couple qui donne le meilleur résultat en terme de fitness. Cependant, nous pouvons voir dans l'annexe 1 (**TODO : créer annexe 1, càd le fichier moyenne pour les tilles 30**) que, pour 30 éléments, la fitness avec les paramètres (16, 40) et (16, 160) donne des fitness moyenne très proche, mais que le deuxième couple prend beaucoup plus de temps à calculer. Ainsi, pour évaluer la qualité d'une solution, nous allons également mettre en jeu cette metrique de temps en faisant un ratio :

$q = \frac{1}{f \times 2 + d} \times 1000000$  avec :

- $q$  la qualité de la solution
- $f$  la fitness moyenne (à laquelle nous donnons deux fois plus de poids)
- $d$  la durée moyenne
- la multiplication par 100000 nous permettant de revenir dans des unité plus simple à analyser

### Analyse des résultats du ratio

Ainsi, avec ce calcul, nous obtenons ces résultats :

- (16, 40) :

- (16, 160) : Avec ce calcul, nous pouvons voir que la qualité des hyperparamètres (16, 40) est supérieur. Pour que ce ration s'inverse, il faut donner à la fitness un poid de 8. Nous pouvons donc interpréter les résultat comme cela : si la fitness est 8 fois plus important que le temps d'exécution, alors le couple (16, 160) est plus intéressant. Dans le cas contraire, nous choisirons plutot le couple (16, 40) **faire aussi pour le couple 64,640, et même peut être automatiser**

## Analyse des choix d'opérateurs

Comme dit précédement, nous avons décider d'utiliser les opérateur relocate et 2-opt. Ainsi, une métrique intéressante est la proportion de choix de chacun de ces opérateurs. Nous pouvons ainsi observer une évolution de la proportion de 2opt en fonction du nombre d'IP et de la taille de la liste **graphique evolution en fonction de la taille graphique evolution en fonction de l'IP potentiellement un seul graphe avec 4 courbes (3 courbes en fait, car on ne prend pas le hill climbing)** pourquoi quand on augmente la taille de la liste on a plus de pourcentage de 2opt : théorie, parce que les bonnes solutions du relocate sont dans la liste, donc impossible de les sélectionner. Cela se confirme si nous regardons le graphique de l'évolution en fonction de la taille. En effet, nous voyons que pour un même nombre d'itération, la proportion de 2-opt est plus grande. Ainsi, nous pouvons penser que toutes les bonnes solutions de relocate ont été sélectionnées et mise dans la liste, laissant plus de place au deuxième opérateur

## Comparaison tabou / hill climbing

Comme dit précédement, le choix des hyper-paramètres (0, x) nous permettent d'avoir les résultat du hill climbing. Ainsi, nous allons comparer ces résultats avec ceux du tabou afin de voir si la mise en place d'une liste tabou à un impact notable.

IP/IR

**TODO**

Qualité

Nous allons réutiliser notre ratio précédent afin de comparer les qualités pour le même nombre d'IP

**TODO**

## Limites du tabou

Croissance exponentielle (mettre calcul et graph)

difficile d'enlever complètement les dernières routes, car il faut voir long terme plutot que de prendre des décisions sur une seule itération (bloquage à 4 routes pour le premier fichier en taille 30)

## Notes à développer

pourquoi relocate et 2 opt :

- relocate car il faut pouvoir enlever des routes et que c'est celui qui s'est montré le plus efficace dans le recuit
- 2-opt car il est cohérent avec les time windows (pas beaucoup de possibilité et des possibilités qui renvoient souvent true, donc ça évite pleins de calculs pour rien)

faire une partie sur les échecs

Mettre les graphes d'évolution de la fitness pour montrer que le tabu marche bien (et monter ce que ça donnait quand ça marchait pas)

regarder les anciens rapports d'autres élèves

si la taille de la tabu est trop grande, on ne trouve plus d'action possible