

# Tabou

---

## Avant propos

Nombre d'itération : [0, ...] Taille de la liste : [ ... ] 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)

L'analyse se base principalement sur le fichier moyenne des 30 premiers Ce fichier Nous avons choisi d'analyser ce fichier plutôt que celui des 100 clients pour plusieurs raisons. D'abord, la quantité des données est bien supérieure. En effet, utiliser seulement les 30 premiers clients nous a permis de faire tourner 4 simulations pour chaque couple d'hyperparamètres. Ce point là est très important, car le résultat, particulièrement quand il y a peu d'itération ou bien beaucoup de camion, est très influencé par la solution aléatoire de départ. Ainsi, il est nécessaire de faire plusieurs simulations afin de lisser l'effet de l'aléatoire (bien que 4 soit assez peu, il est toujours préférable à l'unique solution trouvée pour les fichiers complets). Ensuite, les hyperparamètres choisis sont plus adaptés. En effet, les choix des hyperparamètres que nous allons utiliser ont été réalisés suite à une réflexion sur les premiers résultats. Or, ces premiers résultats ont été générés à partir des données des 30 premiers clients. Pour adapter les itérateurs

## 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 pensé 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 décidé du système suivant : une liste contient des dictionnaires, 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 développé une fonction de hachage de la solution qui nous permet d'avoir une clé réellement unique et de rendre cette détection de schéma fonctionnelle

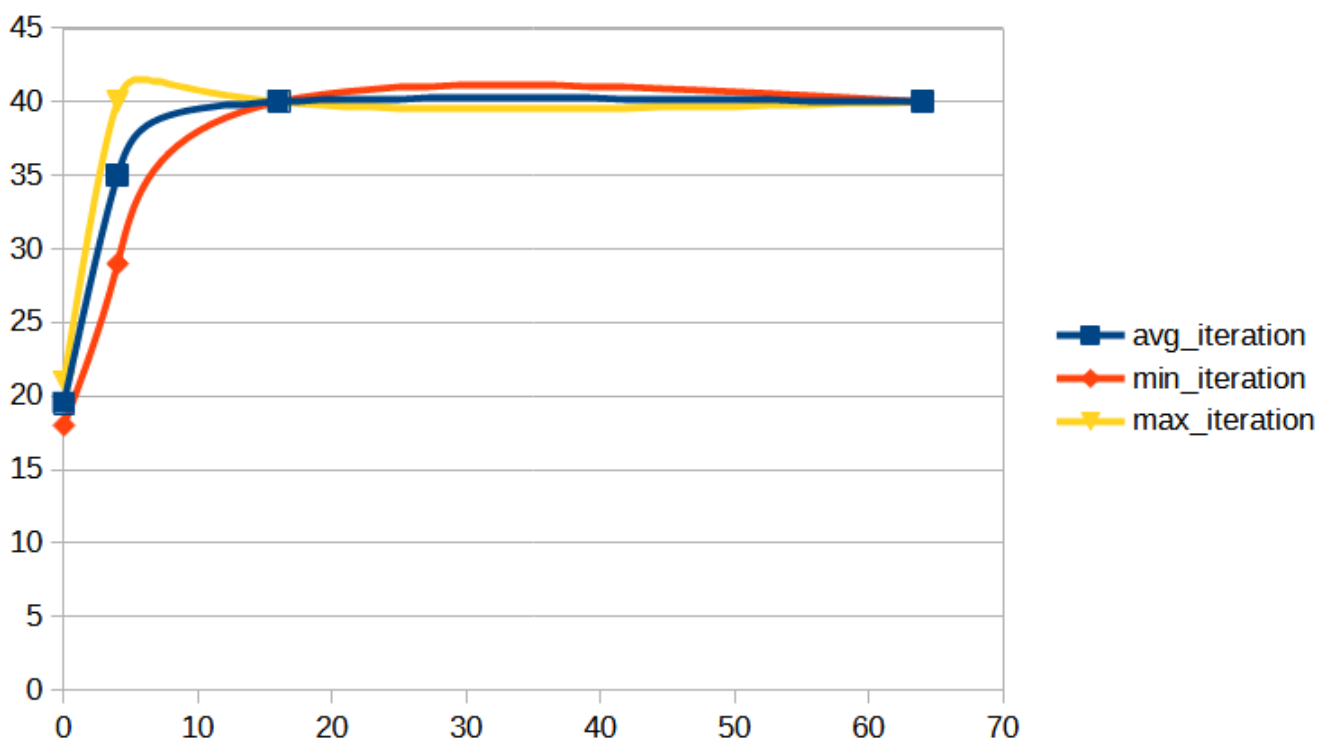
## Analyse des résultats liés

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 les 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 poids 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 utilisation des 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. En regardant la matrice de corrélation, nous pouvons observé un fort lien entre la proportion de 2 opt et le nombre d'IR et d'IP (0.48 et 0.86) 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

## Lien entre les choix d'opérateurs et les autres variables

### Lien avec le nombre de camion

Dans la matrice de correlation, nous pouvons également voir que le choix de l'opérateur relocate est très fortement corrélé avec le nombre de camion enlevé (0.96). Cette correlation est logique, le relocate étant le seul opérateur pouvant enlevé un camion.

### Lien avec l'exploration

Une autre relation très marquée est celle entre le choix de l'opérateur relocate et la fitness. En effet, nous pouvons voir une relation corellé dans le négatif (-0.97). Ainsi, quand la proportion de relocate augmente, la fitness diminue. Cette corelation est liée à celle interprétée dans la partie précédente. En effet, la proportion de relocate diminue lorsque celle de 2-opt augmente. Or, la proportion de 2-opt augmente lorsque

l'algorithme à le temps et la place de mettre les très bons voisins dans la liste. Ainsi, la proportion de 2-opt reflète le niveau d'exploration qui a été réalisé. Logiquement, plus l'exploration augmente, plus il y a d'opportunité de trouver une solution possédant une bonne fitness. Cette hypothèse se confirme grâce à une autre corrélation : celle entre la durée moyenne et la proportion de 2-opt (0.86). En effet, une durée plus haute nous indique une exploration plus profonde.

## Analyse de la fitness

Nous allons maintenant voir quels variables sont en lien avec la fitness de la solution finale

### Lien avec le nombre de camion

Nous pouvons d'abord constaté un lien très fort entre le nombre de camion de la solution finale et sa fitness (-0.99). En parallèle, nous pouvons également voir la même relation entre le nombre de camion enlevées et la fitness (0.99). Cette relation s'explique intuitivement par le fait que les meilleurs solutions ont en générale moins de camion que les solutions aléatoirement générées.

### Lien avec le nombre d'IR

Nous pouvons ensuite voir un lien fort (mais toutefois moins que celui précédemment évoqué) entre la fitness et le nombre d'IR (-0.48). Ce lien nous montre que plus on réalise d'itération, moins la fitness sera haute (et donc meilleure sera la solution). Encore une fois, ce résultat est assez intuitif et ne nécessite pas une plus grande analyse.

## 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

La principale limite de l'algorithme tabou est la quantité de voisin à générer. En effet, à chaque itération, tous les voisins possibles avec les opérateurs choisi sont générés.

Croissance exponentielle (mettre calcul et graph)

Cette croissance liée au nombre peut nous pousser à limiter les opérateurs afin de limiter le nombre de voisin. De plus, une autre limite est le choix des opérateurs. En effet, comme vu précédemment, il n'est pas possible de choisir deux opérateurs qui ont une intersections non-nulles **voir si ça a été expliqué avant et, sinon, l'expliquer**

Une autre limite du tabou est la vision court terme. En effet, la décision de la solution choisi se base

exclusivement sur sa fitness. Or, il serait pertinent que le choix se base plutôt sur la potentielle fitness que cette solution apportera dans le futur. Ce manque de vision long terme se reflète à travers le nombre de camion. En effet, nous avons remarqué que l'algorithme trouvait parfois des solutions non optimales et si retrouvait bloqué. Dans les fichiers de 30 éléments, nous avons par exemple remarqué qu'il s'arrêtait parfois alors qu'il restait un camion en trop par rapport à la solution optimale. La solution avec le camion en trop possédait une très bonne fitness, mais il était parfois impossible de sortir de ce minimum local et d'explorer jusqu'à enlever la route en trop. Nous pourrions penser que ce problème est mitigé par la liste, qui permet une certaine exploration. Cela est en partie vrai, mais pas totalement. En effet, il est possible que l'algorithme suive une direction d'exploration complètement opposé à la solution optimale totale (bien qu'une liste plus grande permet une plus grande exploration).

## Solutions éventuelles

### Matrice

### Notes à développer

matrice de corrélation : voir ce qu'à fait greg et faire pareil

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

choix de mettre 0 itération car ça pose une base pour la matrice de corrélation