



Université Toulouse 3 Paul Sabatier - Master Informatique

## **Rapport TP Métaheuristique**

**Présenté par :**

**DAOULATLI Nour**

# Sommaire

## 1- Les algorithmes implémentés et les voisinages implémentés

Le fichier voyageur.py contient plusieurs fonctions qui répondent aux questions de l'exercice deux du tp. Pour commencer, j'ai récupéré les données du fichier ensuite je code la fonction `solution_initiale` qui prend en paramètre la taille `n` qui renvoie un vecteur de `n` villes et la fonction `calcul_val` qui calcule la distance parcourue pour une certaine solution. A chaque itération on calcule la distance entre la ville `i` et la ville précédente. A la fin, il faut faire attention à rajouter le chemin entre la dernière ville et le point de départ. Ensuite, la fonction du meilleur\_voisin qui renvoie la meilleure solution voisine. Concernant la méthode `steepestHillClimbing`, j'ai codé deux versions : la première il s'agit de la méthode classique. On part d'une solution initiale et on cherche son meilleur voisin. Ensuite on itère pour chaque voisin retrouvée jusqu'à avoir une solution qui n'a pas de meilleur voisin qu'elle-même. Enfin l'algorithme du `SteepestHillClimbing` avec répétition consiste simplement à mettre l'algorithme précédent dans une boucle pour refaire l'algorithme `Max_essais` fois. Pour finir, la dernière fonction est `liste_tabou` qui implémente la méthode Tabou. Afin de coder cette dernière, j'ai dû utiliser deux fonctions: `voisins_non_tabou` qui renvoie une liste contenant les voisins d'une solution qui ne sont pas dans la liste tabou et `meilleur_voisin_non_tabou` renvoie la meilleure solution de la liste des voisins.

## 2- Tests et résultats

Voici un exemple de résultat obtenue avec le fichier tsp5.txt :

```
● nour@nour-VirtualBox:~/Bureau/AA/tpMeta$ python3 voyageur.py tsp5.txt 3 2 5
solution initiale = [4, 5, 1, 2, 3]
valeur de la solution = 255.92085287568437
le meilleur voisin [1, 5, 4, 2, 3]
valeur du meilleur voisin 197.07231856368998
Recherche avec l'algorithme steepestHillClimbing classique:
Solution initiale: [3, 4, 5, 1, 2]
Solution atteinte: [3, 5, 4, 2, 1] en 2 déplacements
Recherche avec l'algorithme steepestHillClimbing avec redémarrages:
  Essaie n° 0
    Solution initiale: [5, 4, 3, 1, 2]
    Solution atteinte: [5, 4, 2, 3, 1] en 2 déplacements
  Essaie n° 1
    Solution initiale: [1, 2, 3, 5, 4]
    Solution atteinte: [1, 2, 4, 5, 3] en 1 déplacements
La methode de tabou
  Solution initiale: [5, 2, 1, 3, 4]
  Solution atteinte: [5, 4, 2, 3, 1] en 3 déplacements
○ nour@nour-VirtualBox:~/Bureau/AA/tpMeta$ ^C
○ nour@nour-VirtualBox:~/Bureau/AA/tpMeta$ █
```