

---

# Perfectionnement à la programmation en C :

## Fiche de TP 6

Licence 2<sup>e</sup> année - 2017/2018

MALEK Akram  
LIEGEY Armand

---

### I.Introduction

Le problème du voyageur de commerce, étudié depuis le 19<sup>e</sup> siècle, est l'un des plus connus dans le domaine de la recherche opérationnelle. En effet il décrit une situation qui prend en entrée un ensemble de villes avec leur position géographique, et qui consiste à fournir en réponse un ordre de visite de ces villes qui soit le plus court possible.

### II.Corps du rapport

Pour commencer on a fixé par des valeurs globales un nombre maximum de villes puis l'utilisateur choisi en fonction de celui-ci combien de villes il souhaite implémenter. Après cela on tire au hasard les coordonnées des villes choisies par l'utilisateur. A la suite on génère un certain nombre  $k$  de visites de façon aléatoire puis sur ces visites on va calculer la distance totale au carré pour un souci d'efficacité. Une fois que cela est fait on va créer un certain nombre de nouvelles visites avec des mutations de la première et on va ensuite trier toutes ces visites et seulement retenir la meilleure d'entre elles pour continuer l'algorithme. Après ça les visites sont affichées dans une fenêtre graphique grâce à la bibliothèque MLV ainsi que la distance totale est aussi affichée dans le terminal.

Concernant les mutations nous avons décidé de garder un alpha bêta gamma de respectivement 50 % 25 % et 25 %. Ces trois valeurs représentent les modifications que la mutation va apporter.

Pour un grand nombre de villes nous pouvons faire l'affichage par exemple de une fois toute la centaine d'améliorations pour réduire le temps d'exécution du programme car l'affichage utilise des fonctions relativement longues par rapport au temps d'exécution des calculs. Nous avons aussi enlevé la racine carrée pour chaque calcul de distance car cette fonction est aussi longue à exécuter nous l'avons donc seulement effectuée au moment de l'affichage dans le terminal de la valeur.

---

## III. Conclusion

Ce TP permet de mieux comprendre un des grands problème qu'on peut rencontrer en informatique c'est à dire les complexification exponentiel et à nous montrer que nous pouvons le contourner par une gestions pseudo aléatoire et en cherchant un bon chemin et non forcément le meilleur.

D'autre algorithme sont efficace afin de résoudre ce genre de problème tels que l'algorithme glouton ou on maximise et minimise la solution sur un sous-problème et on agrandit les données d'entrée petit à petit ou bien l'algorithme génétique.

## IV. Annexe 1

Remarque : si on change les valeurs globales il ne faut pas oublier de faire « make mrproper » avant de l'exécuter (elles se trouvent dans “main.c” et “algo\_jeu.h”).

Le Makefile est joint et crée l'exécutable Projet

Afin de le lancer il faut faire ./Projet.