

# Optimisation TD 4 - Partie 2

## Le voyageur de commerce

Carole Frindel

7 Novembre 2018

### 1 Le voyageur de commerce : cas de $\mathbb{R}^N$ dans $\mathbb{R}$

On se propose d'utiliser le recuit simulé pour résoudre le problème NP complet du voyageur de commerce introduit au début du cours. On se donne  $N$  villes dans un pays et on cherche comment parcourir toutes ces villes en minimisant la distance parcourue.

1. Faire un générateur aléatoire de villes (on peut écrire une fonction qui choisit aléatoirement des points dans un carré  $[0, L]^2$  et qui prend en paramètre le nombre de villes à créer). Indication : utiliser le générateur aléatoire `rand` du package `numpy.random`.
2. Faire une fonction qui calcule la distance euclidienne parcourue par le voyageur. La donnée d'entrée sera donc le trajet d'un voyageur sous la forme d'un vecteur de taille  $N$ . Celui-ci contiendra les entiers de 0 à  $N - 1$  qui seront les indices des villes parcourues.
3. Le "déplacement aléatoire de la solution courante" sera ici la permutation de deux villes. Indication : utiliser le générateur aléatoire `randint` du package `numpy.random`.
4. Implémenter l'algorithme.
5. Visualiser avec `matplotlib` l'évolution de la solution. Faire des captures d'écran à toutes les  $k$  itérations en donnant la distance parcourue. Commenter. Note : Commencer par des problèmes simples.
6. Tester différents schémas de décroissance de la température (décroissance en  $1/t$ ,  $1/t^3$  et  $1/\log(t)$ ). Le choix du schéma de décroissance est crucial dans cet algorithme. Comment l'observez-vous ? Indication : Pour vous en convaincre, saisir une petite carte d'une dizaine de villes, et observer le résultat à la fin de l'algorithme. Répéter l'opération plusieurs fois avec chaque schéma de décroissance. Qui donne les meilleurs résultats ? En combien de temps ?
7. Enfin, plus subtil, comment appliquer la méthode de descente de gradient à ce problème ?