Voyageur de commerce – Méthode de descente et Recuit Simulé

Implantation d'un algorithme de recuit simulé pour le problème du voyageur de commerce.

Le choix du langage de programmation est laissé libre.

Données

n le nombre de villes. Les villes sont numérotées de 1 à n ou de 0 à n-1 selon le langage utilisé.

C matrice $n \times n$ symétrique telle que C[i,j] = C[j,i] = le coût pour aller de i à j = le coût pour aller de j à i.

Structure de données

T un tableau de n éléments telle que T[i] = la $i^{ème}$ ville parcourue dans le tour i=1 à n (ou i=0 à n-1)

Par exemple T=[4 5 1 2 3 6 7 8 9] ; en position 1 est la ville 4, en position 2 la ville 5, en position 3 la ville 1, etc…

Procédures à écrire

Ecrire une procédure qui fait la transformation 2-opt (voir le cours)

- En entrée : la position de 2 villes i et j avec i<j, T
- En sortie : T modifié

Exemple : T=[1 2 3 4 5 6 7 8 9], i=2, j=7

On obtient en sortie T=[1 2 7 6 5 4 3 8 9]

Ecrire une procédure qui calcule le coût d'un tour T

- En entrée T
- En sortie : coût de T

Ecrire une procédure qui choisit de façon aléatoire 2 positions, i et j , dans T avec i<j.

- En sortie : i et j

Ecrire la fonction de Metropolis

- En entrée : δ (la différence de coût entre 2 solutions successives), Température
- En sortie : vrai ou faux (vrai=mouvement accepté)

Programme principal

- Dans un premier temps, faire une méthode de descente
- Dans un deuxième temps, faire le recuit simulé. Quand le mouvement est refusé, il faut restaurer le solution (celle avant transformation). Pour la transformation 2-opt, cela peut se faire très simplement.