

## RECUIT SIMULE APPLIQUE AU PROBLEME DU VOYAGEUR DE COMMERCE

### 1. DESCRIPTION DU PROGRAMME FOURNI

**Un fichier script Scilab fourni** : Recuit simulé 2025.sce

Le script est largement commenté, on trouve successivement :

- 1) les paramètres de réglage de l'algorithme.
- 2) le nombre, la définition des villes, soit réparties aléatoirement sur un carré de côté 2, soit sur un cercle (variable booléenne « Cercle ») puis le calcul des distances (on rappelle euclidiennes). On utilise une graine afin de préserver les positions. Pour d'autres positions de villes changer la valeur ligne 16 dans `rand("seed",11)`;
- 3) pour le démarrage de l'algorithme, on impose une graine identifiée dans la variable « cas ». Changer la valeur permet de changer la séquence du tirage aléatoire du recuit simulé.
- 4) les fonctions à savoir :
  - la fonction PERMUTATION de type 2-opt
  - la fonction d'ACCEPTATION selon Métropolis
  - la fonction DECROISSANCE DE LA TEMPERATURE ici une loi géométrique simple
  - une fonction GRAPHE pour afficher un cycle (position dans la figure n°cas)
- 5) Le premier cycle pris dans l'ordre d'apparition des villes lors du tirage
- 6) l'algorithme du recuit simulé proprement dit
- 7) l'affichage des évolutions du coût (figure n°1), de la température (figure n°2) et du taux d'acceptation (figure n°3).

Au cours de l'exécution, on peut suivre l'évolution du cycle sur la figure n°cas. Au terme de l'exécution, on obtient les 3 graphes ci-dessus, le graphe du meilleur circuit obtenu, le temps de calcul et la meilleure distance obtenue.

---

Le cas d'études envoyé correspond à une température initiale de 25, de gel de 1.00, un taux de décroissance de 0.95, d'un nombre de 10 itérations sur chaque palier, d'un nombre de villes de 50 sur un carré, d'une graine « ville » égale à 11, et d'une série aléatoire avec une graine de 4. **Ce jeu de paramètres n'est pas optimum !**

## 2. TRAVAIL PROPOSE : SUGGESTIONS

**Objectif** : trouver le meilleur choix de paramètres pour le recuit simulé codé selon le programme livré qui permettent en un moindre temps de calcul d'obtenir une solution acceptable.

### Recherche exhaustive

Le problème du voyageur de commerce est un problème NP-complet et difficile. Le seul algorithme permettant de déterminer à coup sûr le chemin optimal reste une recherche exhaustive où l'ensemble des trajets sont évalués (il existe des techniques pour restreindre l'espace de recherche, mais fondamentalement, ça ne change pas la problématique). Si  $N$  est le nombre de villes, combien de cycle hamiltonien peut-on dénombrer ? Que coûte l'évaluation d'un cycle ? comment évaluer sur votre ordinateur le coût temporel d'un cycle ? Connaissant cette donnée, évaluer le coût pour 5, 10, 20 puis 50 villes d'un algorithme exhaustif. Pour l'âge de l'univers, combien de villes vous pourriez traiter avec votre ordinateur ?

### Analyse du programme fourni

Description en amont. Vous pouvez identifier entre autres la partie même du recuit simulé, le critère d'acceptation codé dans le programme, analyser le voisinage proposé, etc.

Vous avez également toute liberté pour recoder l'algorithme dans un autre langage (Python ?).

### Premiers tests

Essayer différents jeux de paramètres, comprendre les résultats, en tirer des conclusions.

On conseille d'utiliser (pour les 50 villes) les 3 jeux de données suivantes afin de comprendre les problèmes liés à de mauvais choix de paramètres (on n'hésitera pas de faire des commentaires sur les résultats obtenus).

Temp	100	10	10
TempGel	10	0.01	0.01
alpha	0.99	0.75	0.99
kEquil	20	20	2

### Ajustements

Ajuster les paramètres en fonction des résultats précédents. Commenter. La solution est-elle optimale ? satisfaisante ?

### Etudes

Les paramètres bien réglés, la solution obtenue est-elle la meilleure ? Quand est-il si l'on change la séquence des tirages aléatoires ?

### Rapport partie 1

Ne pas hésiter d'illustrer vos choix, vos commentaires en illustrant des résultats obtenus.