

---

Travail pratique                      Heuristiques d'insertion pour le voyageur de commerce

---

## Objectifs

L'objectif de ce travail pratique est de programmer trois heuristiques d'insertion pour le problème du voyageur de commerce euclidien et comparer leurs performances sur plusieurs jeux de données fournis à l'aide de statistiques sur les longueurs des tournées obtenues en faisant varier la ville de départ.

## Travail de programmation à effectuer

Vous devez compléter les sources fournies afin de mettre en œuvre trois méthodes d'insertion différentes : « Insertion aléatoire », « Insertion la plus proche » et « Insertion la plus éloignée ». Les trois variantes procèdent globalement de la même manière. En partant d'une sous-tournée réduite à une ville de départ fournie et tant que toutes les villes n'ont pas été insérées dans la tournée on choisit une ville hors tournée et on l'insère entre deux villes de la tournée actuelle, à l'endroit qui provoque la plus petite augmentation de longueur de la tournée.

- a) Pour l'heuristique « Insertion aléatoire », la prochaine ville ajoutée à la tournée est choisie au hasard parmi toutes les villes hors tournée.
- b) Dans les deux autres méthodes on commence par associer à chaque ville hors tournée la ville de la tournée qui lui est la plus proche. Ensuite
  - ▷ dans la variante « Insertion la plus proche » on sélectionne la ville hors tournée dont la distance à la ville la plus proche de la tournée est minimale,
  - ▷ alors que dans la variante « Insertion la plus éloignée » on sélectionne la ville hors tournée dont la distance à la ville la plus proche de la tournée est maximale.

Vous veillerez à la structure de vos mises en œuvre qui devront implémenter l'interface `ObservableTspConstructiveHeuristic` mais, également à leur efficacité et, en particulier, à leur complexité dans le pire des cas qui devra être en  $O(n^2)$  pour un problème comptant  $n$  villes.

## Format des données

Les jeux de données sont stockés dans des fichiers texte d'extension `.dat`. La première ligne de chaque fichier contient un seul entier égal au nombre  $n$  de villes de l'instance qui doit être

au moins égal à 3. Les  $n$  lignes suivantes contiennent chacune l'identifiant d'une ville suivi de ses deux coordonnées cartésiennes. Les identifiants sont des entiers allant de 0 à  $n - 1$  et ils apparaissent dans l'ordre croissant dans chaque fichier. Les coordonnées cartésiennes de chaque ville sont des entiers (normalement positifs ou nuls).

La distance entre deux villes est égale à la distance euclidienne entre les points du plan correspondant aux deux villes mais arrondie, par excès, à l'entier le plus proche.

La classe `TspData` fournie permet la lecture, le stockage et l'exploitation d'un jeu de données à partir d'un fichier respectant le format précédent.

### Travail d'analyse à effectuer

Vous évaluerez les trois heuristiques sur les six jeux de données fournis avec l'archive des sources à compléter. Pour chacun d'eux, la longueur de la tournée optimale est donnée dans la table qui suit.

Nom de l'instance	pcb442	att532	u574	pcb1173	nrv1379	u1817
Longueur optimale	50 778	86 729	36 905	56 892	56 638	57 201

Pour chaque instance et chaque heuristique vous calculerez les longueurs des tournées obtenues en utilisant successivement chaque ville du problème comme ville de départ. À partir de ces résultats vous calculerez quelques mesures de performances, absolues et relatives, permettant de comparer les trois heuristiques, statistiques que vous afficherez dans la console..

Pour l'heuristique d'insertion aléatoire, vous initialiserez, au début du traitement de chaque jeu de données, votre générateur de nombres pseudo-aléatoires avec la graine `0x134DA73`.

### Modalités et délais

- ▷ Le travail de programmation est à effectuer par groupe de deux, en Java, version 21 (ou 23).
- ▷ L'archive contenant les sources du projet et les jeux de données à étudier, est disponible sur le site Cyberlearn du cours.
- ▷ Vous devez rendre une archive (au format `zip`) contenant toutes les sources de votre projet complété. Vous prêterez une attention toute particulière aux commentaires de vos implémentations des trois heuristiques. Votre archive contiendra également un fichier texte (ou markdown) contenant les statistiques sur les performances des trois heuristiques pour chacun des jeux de données fournis.
- ▷ Vous devez rendre votre travail sur Cyberlearn au plus tard le **dimanche 3 novembre 2024** (avant minuit).