Travail pratique **Heuristique 2-opt pour le TSP**

Objectifs

L'objectif de ce travail pratique est de programmer la variante « meilleure amélioration » de l'heuristique 2-opt pour le problème du voyageur de commerce symétrique et d'observer et évaluer ses performances pour les jeux de données fournis et pour plusieurs méthodes de construction d'une tournée initiale.

Heuristique 2-opt

Rappelons que l'heuristique 2-opt est un algorithme d'amélioration basé sur des modifications locales d'un cycle hamiltonien représentant une tournée dans un problème de voyageur de commerce. Une modification locale, appelée « 2-échange », consiste à retirer deux arêtes du cycle actuel et à les remplacer par deux nouvelles arêtes de manière à conserver un cycle hamiltonien.

De manière plus pragmatique, votre tournée sera vraisemblablement représentée par une permutation circulaire imposant de fait un sens de parcours (arbitraire). Cette permutation pourra être stockée dans un simple tableau (comme dans `TspTour`) dont la valeur à l'indice i est égale au numéro de la ville en position i dans la tournée. Dans un tel cadre, un 2-échange (i, j) consiste à retirer le trajet reliant les villes aux positions i et $i+1$ dans la tournée actuelle ainsi que celui reliant les villes aux positions j et $j+1$ et à les remplacer par les trajets entre les villes aux positions i et j d'une part et $i+1$ et $j+1$ de l'autre. Un 2-échange est améliorant si la longueur de la nouvelle tournée est plus petite que celle de la tournée avant échange. Le procédé est illustré en figure 1.

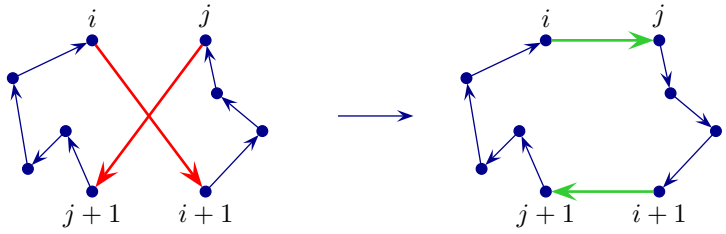

FIGURE 1 – Illustration graphique d'un 2-échange. Les indices en regard des sommets dénotent leur position dans la tournée de départ (celle de gauche). L'échange est améliorant si la somme des longueurs des deux trajets retirés (en rouge) est plus grande que celle des deux trajets ajoutés (en vert).

La variante « meilleure amélioration » de l'heuristique 2-opt consiste à rechercher et appliquer à chaque itération le 2-échange entraînant la plus grande diminution de la longueur de la tournée actuelle. L'algorithme s'arrête lorsqu'il n'est plus possible de diminuer la longueur de la tournée, c.-à-d. lorsque l'on ne trouve plus de 2-échanges améliorants pour la solution courante. Un pseudocode de l'algorithme est présenté en table 1.

Algorithme 1 Heuristique TwoOptBestImprovement

Données : Un cycle hamiltonien représentant une tournée dans un problème de voyageur de commerce symétrique.

Résultat : Un nouveau cycle hamiltonien, correspondant à un minimum local de la structure de voisinage définie par les 2−échanges.

1: **procédure** TwoOptBestImprovement
2: **répéter**
3: Rechercher le meilleur 2-échange pour la tournée actuelle
4: Effectuer le 2-échange s'il est améliorant
5: **jusqu'à ce qu'**il n'existe plus de 2-échanges améliorants
6: Retourner la nouvelle tournée
7: **fin procédure**

Quelques considérations pratiques

- ▷ Les problèmes de voyageur de commerce considérés étant symétriques, les 2-échanges (i, j) et (j, i) correspondent à la même modification, la seule différence étant le sens de parcours de la nouvelle tournée. Il est donc suffisant de ne considérer que des 2-échanges (i, j) avec $i < j$.
- ▷ Les 2-échanges (i, j) avec $j = i + 1$ ne modifient pas la tournée, il est donc inutile de les considérer.
- ▷ Faites attention à tester tous les 2-échanges possibles sans oublier ceux impliquant les villes aux dernières positions dans votre tournée.
- ▷ Évaluer si un 2-échange est améliorant ou non se fait en temps constant, mais il s'agit de l'opération la plus fréquente dans l'algorithme. Essayez de trouver une approche aussi rapide que possible.
- ▷ La mise à jour de la permutation représentant une tournée nécessite de renverser le sens de parcours d'une des deux portions de tournée (en bleu dans la figure 1).
- ▷ Aucun départage particulier des égalités n'est demandé.

Travail de programmation et d'analyse à effectuer

Premièrement vous devez compléter les sources fournies afin de mettre en œuvre la variante « meilleure amélioration » de l'heuristique 2-opt présentée en table 1.

Ceci fait vous mettrez en place un programme principal permettant d'évaluer la qualité et les performances de l'heuristique en fonction de la tournée de départ fournie. Plus précisément, **pour chacun des six jeux de données à disposition** (les mêmes que ceux du travail pratique précédent),

- ▷ vous évaluerez votre algorithme en partant d'une tournée initiale aléatoire (fournie par la classe `RandomTour`) ou d'une tournée obtenue à l'aide des heuristiques « Insertion la plus proche » et « Insertion la plus éloignée » développées lors du laboratoire précédent ;
- ▷ **pour chacune des trois possibilités**, vous mesurerez et calculerez les performances (longueurs minimale, moyenne et maximale, temps moyen) obtenues en effectuant 50 optimisations ;
- ▷ vous établirez les statistiques sur les longueurs des tournées obtenues (minimum, moyenne, maximum) à la fin de la construction de la tournée de départ et à la fin de la phase d'amélioration, de plus vous afficherez uniquement des performances relatives ;

▷ pour le temps moyen d'exécution, vous considérerez uniquement le temps total d'optimisation (initialisation et amélioration) ;

Afin d'obtenir vos mesures de performance vous créerez, pour chaque jeu de données, deux objets `RandomTour`, tous deux initialisés avec la graine `0x134DAE9`.

▷ Le premier vous permettra de créer, successivement, 50 tournées initiales aléatoires pour votre algorithme 2-opt.

▷ Vous utiliserez le second pour créer une permutation aléatoire des villes et vous utiliserez les 50 premières valeurs de cette permutation comme villes de départ pour vos heuristiques d'insertion.

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 votre implémentation de l'heuristique 2-opt. Votre archive contiendra également un fichier texte (ou markdown) contenant les statistiques sur les performances de l'heuristique pour chacun des jeux de données fournis et chacune des trois méthodes constructives d'initialisation retenues.

▷ Vous devez rendre votre travail sur Cyberlearn au plus tard le **dimanche 15 décembre 2024** (avant minuit).