

Projet Avancé 2025 – Problème du Voyageur de Commerce (TSP)

Présentation

Ce projet a pour objectif d'implémenter différentes méthodes de résolution du **Problème du Voyageur de Commerce (TSP)** en langage **C**, en respectant les spécifications données (lecture de fichiers TSPLIB, comparaison d'algorithmes, normalisation des résultats).

Équipe 12

- **Fourat Ammari** : Lecture TSPLIB, structures de données, matrice des distances
- **Oleksandra Filiushkina** : Brute Force, Nearest Neighbor
- **Mohamad Rammal** : Random Walk, 2-opt
- **Mohamed Yassine Bouzaïene** : Algorithme génétique, paramètres et tests

Objectifs

- Développer et comparer plusieurs algorithmes (Brute Force, Nearest Neighbor, Random Walk, 2-opt, Algorithme génétique).
- Respecter les contraintes de lecture de données (TSPLIB) et de normalisation des résultats.
- Produire un code documenté et testé, accompagné d'un rapport final.

Compilation et exécution

Un Makefile sera fourni pour compiler facilement le projet.

Exemple :

```
make
./tsp -f instances/att10.tsp -m bf    # exécution brute force
./tsp -f instances/att10.tsp -m nn    # exécution nearest neighbor
```

Organisation Git & Workflow Collaboratif

1. Cloner le dépôt

Chaque membre commence par cloner le dépôt :

```
git clone https://github.com/tonpseudo/projet-tsp.git
cd projet-tsp
```

2. Créer une branche de travail

Toujours travailler sur une branche personnelle/fonctionnelle :

```
git checkout main
git pull origin main
git checkout -b mohamad-rw  # exemple pour Random Walk
```

3. Committer son travail

```
git add fichier.c
git commit -m "Implémentation initiale de Random Walk"
```

4. Pousser la branche sur GitHub

```
git push origin mohamad-rw
```

5. Créer une Pull Request (PR)

- Aller sur GitHub
- Cliquer sur **Compare & pull request**
- Vérifier que la PR va de **ta branche** → **main**
- Ajouter un commentaire descriptif
- Soumettre la PR pour validation

6. Mettre à jour son dépôt

Avant de commencer une nouvelle tâche :

```
git checkout main
git pull origin main
git checkout -b nouvelle-branche
```

7. Supprimer une branche (optionnel après merge)

```
git branch -d mohamad-rw
git push origin --delete mohamad-rw
```

☒ Résumé des commandes utiles

- **Mettre à jour main :**

```
git checkout main
git pull origin main
```
- **Créer une branche :**

```
git checkout -b nom-de-branche
```
- **Commit & Push :**

```
git add .  
git commit -m "Message"  
git push origin nom-de-branche
```
