

README.md : Solveur Modulaire du Problème du Voyageur de Commerce (TSP)

Vue d'Ensemble du Projet

Ce projet implémente un solveur polyvalent pour le **Problème du Voyageur de Commerce (TSP)** en langage C.

Le cœur du programme utilise un design modulaire basé sur des **pointeurs de fonctions génériques** (`types.h`) pour permettre l'échange dynamique des algorithmes de résolution et des fonctions de calcul de distance.

Stratégies de Résolution

Le solveur offre quatre stratégies de résolution distinctes :

1. **Force Brute (FB)** : Solution **optimale** pour $N \leq 12$.
 2. **Nearest Neighbor (NN)** : Heuristique constructive **rapide** ($O(N^2)$).
 3. **Random Walk (RW)** : Heuristique constructive **aléatoire** ($O(N)$).
 4. **Algorithme Génétique (GA)** : Méta-heuristique **performante** pour $N > 50$.
-

Équipe et Rôles

Rôle	Membre(s)
Responsable Projet (RP)	Diogo CRUZ
Responsable Test (RT)	Diogo CRUZ
Responsable Développeur 3 (RD3)	Diogo CRUZ
Responsable Git (RG)	Camille MAGNE
Responsable Développeur 1 (RD1)	Camille MAGNE
Responsable Canevas (RCA)	Esso-Ylow-Noyou Trésor PANA
Responsable Développeur 2 (RD2)	Esso-Ylow-Noyou Trésor PANA

🔧 □ Instructions de Compilation

Prérequis

- Compilateur C (GCC ou Clang) compatible avec C11.
- Librairie mathématique standard (-lm).

Commandes `make`

Pour compiler l'exécutable `./main` :

```
make
```

Pour supprimer l'exécutable et les fichiers objets :

```
make clean
```

🔧 Scripts et Exécution

📄 Format du Fichier d'Entrée

Les fichiers doivent être au format **TSPLIB** (e.g., NAME, DIMENSION, NODE_COORD_SECTION).

Les fichiers d'instance doivent être placés dans le sous-répertoire `tests/`.

Syntaxe de la Ligne de Commande

L'exécution est pilotée par les options suivantes :

Flags	Description	Valeurs
<code>-f</code>	Fichier d'entrée TSPLIB (Obligatoire).	<code><file.tsp></code> (dans <code>tests/</code>)
<code>-m</code>	Méthode de résolution .	<code>bf / bfm / nn / rw / 2optnn / 2optnw / ga / gadpx</code>
<code>-d</code>	Type de Distance (surcharge l'auto-détection).	<code>{eucl2d att geo}</code>
<code>-o</code>	Fichier de sortie pour les résultats.	<code><output.txt></code>

Flags	Description	Valeurs
-c	Afficher la longueur de la tournée canonique .	(Aucune valeur)
-h	Affiche l'aide et quitte le programme.	(Aucune valeur)

Paramètres GA (-m ga ou -m gadpx)

Si la méthode GA est choisie, les paramètres suivants doivent suivre l'option -m :

```
<nombre_individus> <nombre_generations> <taux_mutation> <taille_tournoi>
[Crossover: D ou O]
```

Interruption Contrôlée (Ctrl+C)

Les méthodes **bf** et **bfm** supportent le gestionnaire de signal. En cas de **Ctrl+C**, le programme affiche la **meilleure tournée trouvée jusqu'à présent** et demande si l'utilisateur souhaite continuer ou s'arrêter.

Algorithmes, Performances et Limitations

Mesure des Performances

Le temps d'exécution réel (en secondes) est mesuré pour chaque méthode de résolution (`run()` dans `exec.c`) et est inclus dans la sortie formatée.

1. Méthodes de Résolution

Algorithme	Option(s)	Type / Complexité	Limite typique (N)	Notes
Force Brute	bf, bfm	Optimal / $O(N!)$	≤ 13	bfm utilise une demi-matrice pour l'optimisation.
Heuristiques	2optnn, 2optrw	Amélioration locale / $O(N^3)$	Grande	Le 2-opt utilise <code>reverse_segment</code> pour minimiser les croisements.

Algorithme	Option(s)	Type / Complexité	Limite typique (N)	Notes
Algorithme Génétique	ga, gadpx	Méta-heuristique	>=50	gadpx intègre le 2-opt pour affiner les descendants.

2. Types de Distance Supportés

Le solveur gère les normes de distance :

- **Euclidienne 2D** (`euc12d`)
- **Pseudo-Euclidienne ATT** (`att`)
- **Géographique GEO** (`geo`)

La distance est **détectée automatiquement** à partir du fichier TSPLIB, mais peut être forcée par l'option `-d`.

Instances de Test

Instance	N	Type de Distance	Objectif de Test	Méthode Recommandée
<code>att10</code>	10	ATT	Tester l' Optimum et la fonction <code>dist_att</code> .	<code>bfm -d att</code>
<code>burma14</code>	14	GEO	Tester la limite de <code>bfm</code> et la fonction <code>dist_geo</code> .	<code>bfm -d geo (long)</code> ou <code>2optnn -d geo</code>
<code>ali535</code>	535	GEO	Tester les performances de <code>gadpx</code> sur une très grande instance GEO .	<code>gadpx -d geo</code>

Documentation API (Doxygen)

La documentation complète de l'API (fonctions, structures, graphes d'appel/inclusion) peut être générée en HTML :

- **Commande** : `doxygen Doxyfile, make docs, ou juste make.`
- **Accès** : Ouvrez `docs/html/index.html` après génération.
 - **Astuce** : Une fois dans le dossier html, utilisez `Crtl f` pour trouver `index.html`.