جامعة هواري بومدين للعلوم والتكنولوجيا

***Université des Sciences et de la Technologie Houari-Boumediene***

***Faculté d’Informatique***

#### Rapport de Projet

**Thème**

∆-Problème du voyageur de commerce

**Présenté par:**

* Kerdjidj Mohamed Seddik
* Zeghouf Samir

**Table des matières**

[**Introduction :** 3](#_Toc197291013)

[**Méthodologie :** 3](#_Toc197291014)

[**Étapes de l'algorithme :** 3](#_Toc197291015)

[**1. Génération de la matrice des distances :** 3](#_Toc197291016)

[**2. Calcul de l'arbre couvrant minimal (MST) :** 3](#_Toc197291017)

[**3. Duplication des arêtes :** 4](#_Toc197291018)

[**4. Parcours eulérien :** 4](#_Toc197291019)

[**5. Transformation en cycle hamiltonien :** 4](#_Toc197291020)

[**6. Calcul des longueurs :** 4](#_Toc197291021)

[**Analyse Théorique :** 4](#_Toc197291022)

[**1.** **Complexité de l'algorithme :** 4](#_Toc197291023)

[**2.** **Preuve de l’approximation à un facteur 2 :** 5](#_Toc197291024)

[**Résultats :** 5](#_Toc197291025)

[**Exemple de sortie pour 10 points :** 5](#_Toc197291026)

[**Exemple de sortie pour 20 points :** 6](#_Toc197291027)

[**Exemple de sortie pour 50 points :** 7](#_Toc197291028)

[**Conclusion :** 8](#_Toc197291029)

# **Introduction :**

Le problème du voyageur de commerce (TSP) est un problème classique d'optimisation combinatoire. Il consiste à trouver le plus court chemin passant par un ensemble de points (ou villes) exactement une fois, avant de revenir au point de départ. Ce problème est NP-difficile, ce qui signifie qu'il est difficile de trouver une solution optimale en un temps raisonnable pour un grand nombre de points.

Dans ce projet, nous avons implémenté un algorithme d'approximation basé sur l'arbre couvrant minimal (MST) pour résoudre le TSP. Cet algorithme garantit une solution approchée avec un facteur d'approximation de 2.

# **Méthodologie :**

## **Étapes de l'algorithme :**

### **1. Génération de la matrice des distances :**

- Une matrice des distances est générée à partir des coordonnées des points. Chaque élément de la matrice représente la distance euclidienne entre deux points.

- Cette matrice est exportée dans un fichier Excel (‘distance\_matrix.csv’) pour une meilleure visualisation.

### **2. Calcul de l'arbre couvrant minimal (MST) :**

- L'algorithme de Kruskal est utilisé pour construire un arbre couvrant minimal à partir de la matrice des distances.

- Complexité : O(E log V), où E est le nombre d'arêtes et V le nombre de sommets.

### **3. Duplication des arêtes :**

- Les arêtes de l'arbre couvrant minimal sont dupliquées pour former un graphe eulérien.

### **4. Parcours eulérien :**

- Un parcours eulérien est effectué sur le graphe eulérien pour obtenir une séquence de sommets visités.

- Complexité : O(E).

### **5. Transformation en cycle hamiltonien :**

- Les sommets déjà visités sont ignorés pour transformer le parcours eulérien en un cycle hamiltonien.

- Complexité : O(V).

### **6. Calcul des longueurs :**

- Les longueurs de l'arbre couvrant minimal, du parcours eulérien et du cycle hamiltonien sont calculées.

- Ces résultats sont exportés dans un fichier Excel (‘results.csv’).

# **Analyse Théorique :**

## **Complexité de l'algorithme :**

L'algorithme suit les étapes suivantes :

- Calcul de l'arbre couvrant minimal (Kruskal) :

- Complexité ≈ O(E log V) avec Union-Find optimisé.

- Duplication des arêtes :

- Complexité linéaire O(E).

- Parcours eulérien :

- Pour un graphe connexe avec des degrés pairs, complexité O(E) (via l'algorithme de Hierholzer).

- Transformation en cycle hamiltonien :

- Complexité O(V).

Complexité totale : O(E log V)

Pour un graphe complet, où E = V(V-1)/2, la complexité finale est :O(V² log V).

## **Preuve de l’approximation à un facteur 2 :**

L'algorithme est 2-approximatif pour le TSP respectant l'inégalité triangulaire. Voici la preuve :

- Soit **OPT** la longueur de la tournée optimale.

- L'arbre couvrant minimal (MST) a une longueur ≤ OPT (car supprimer une arête de la tournée optimale donne un arbre).

- Le doublement des arêtes donne une longueur ≤ 2 × OPT.

- La transformation en cycle hamiltonien (par "shortcutting") ne rallonge pas le parcours grâce à l'inégalité triangulaire.

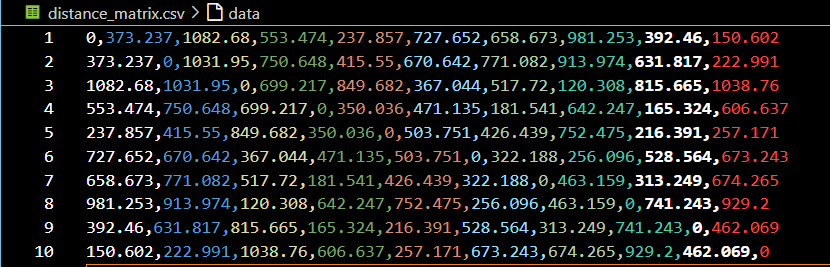
Ainsi, la longueur finale est ≤ 2 × OPT, ce qui prouve que l'algorithme est 2-approximatif.

# **Résultats :**

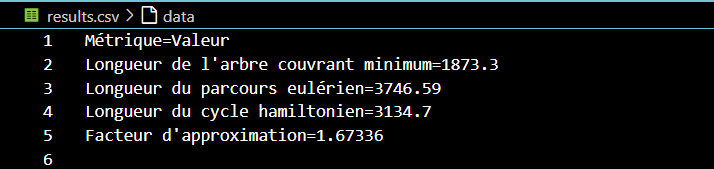
## **Exemple de sortie pour 10 points :**

Pour un ensemble de 10 points générés aléatoirement, les résultats obtenus sont les suivants :

**Matrice des distances (‘distance\_matrix.csv’)**



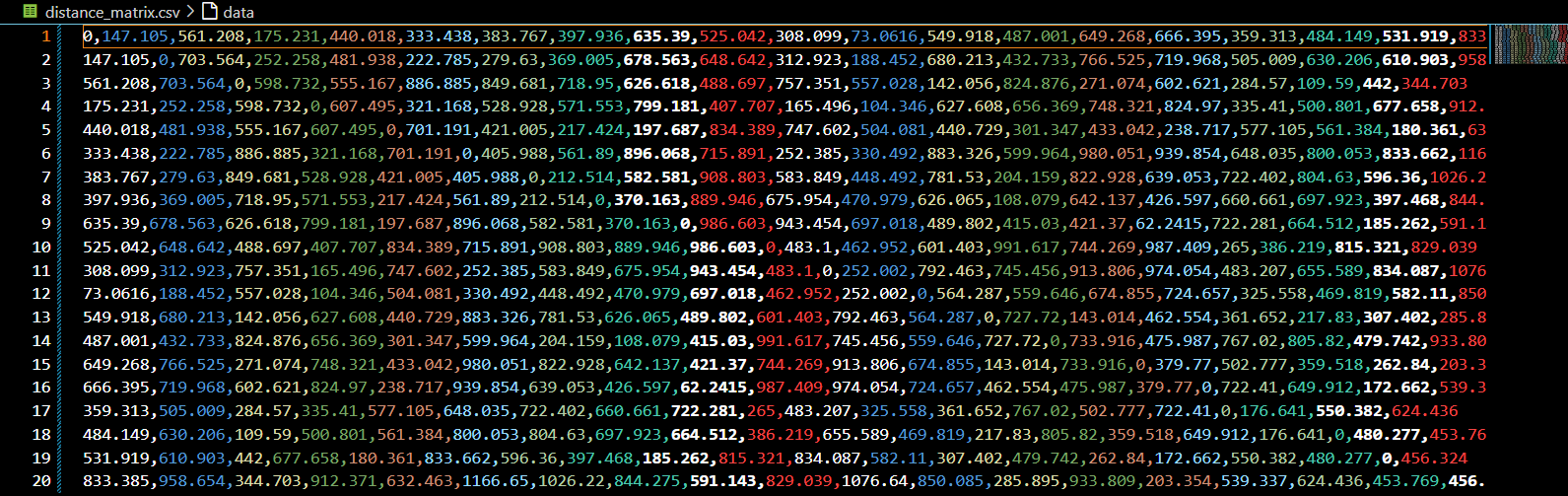
**Résultats (‘results.csv’)**



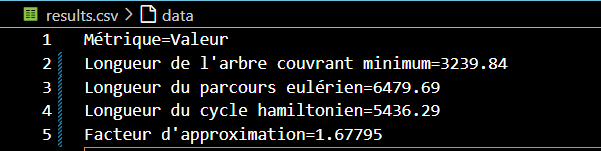
## **Exemple de sortie pour 20 points :**

Pour un ensemble de 20 points générés aléatoirement, les résultats obtenus sont les suivants :

**Matrice des distances (‘distance\_matrix.csv’)**

****

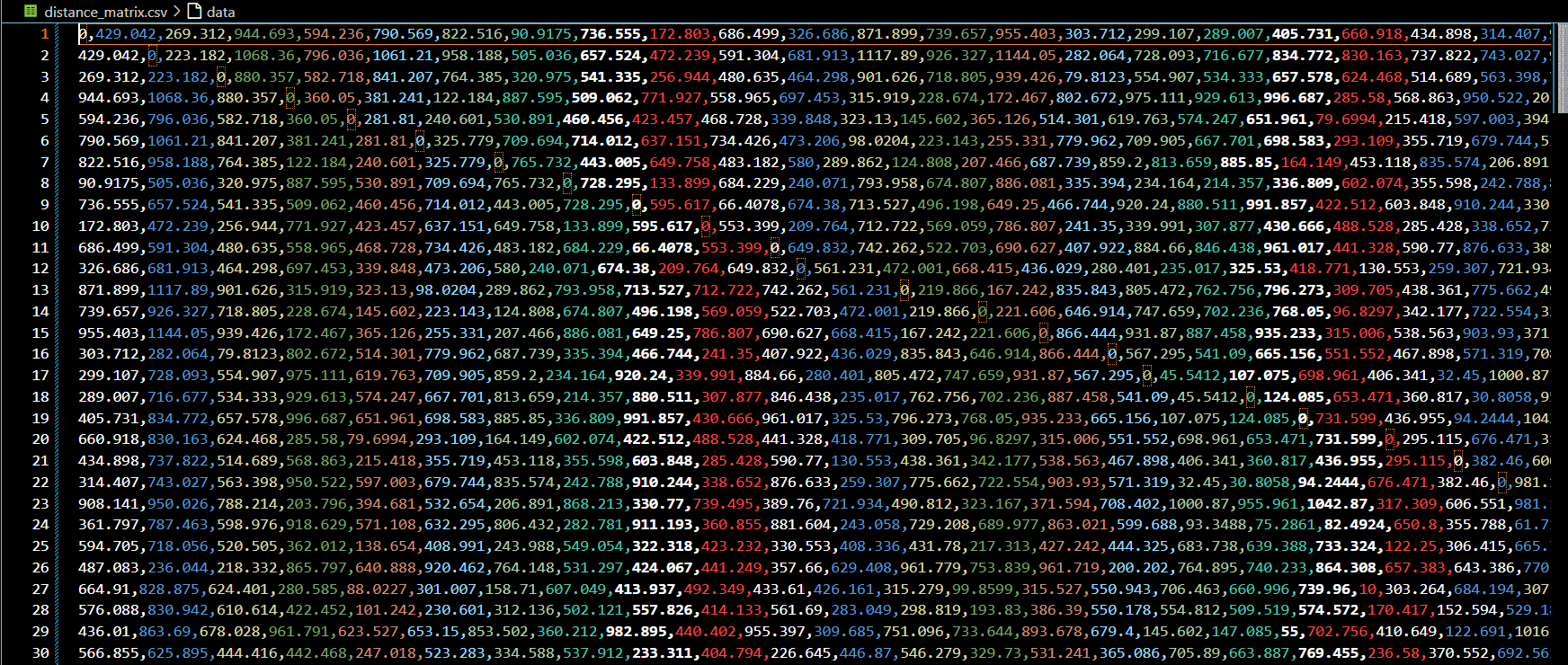
**Résultats (‘results.csv’)**

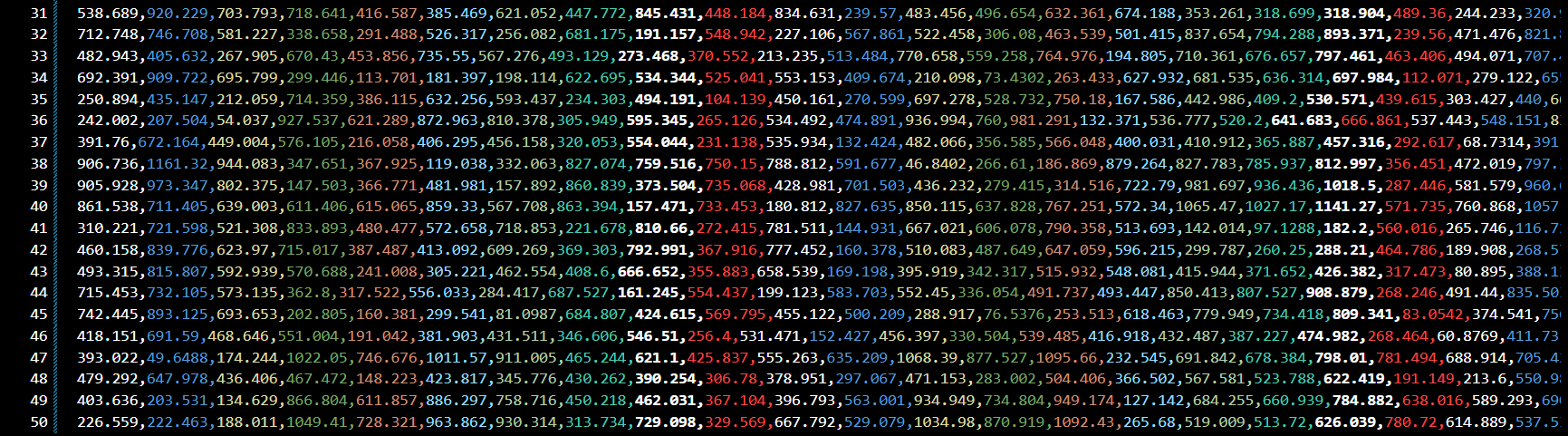


## **Exemple de sortie pour 50 points :**

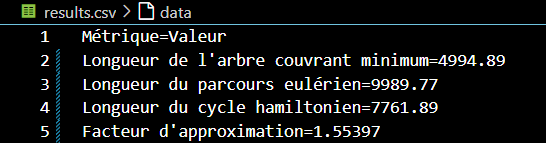
Pour un ensemble de 50 points générés aléatoirement, les résultats obtenus sont les suivants :

**Matrice des distances (‘distance\_matrix.csv’)**

****

****

**Résultats (‘results.csv’)**



# **Conclusion :**

Cet algorithme offre une solution approchée efficace pour le TSP, avec une complexité de **O(V² log V)** pour un graphe complet. Bien qu'il ne garantisse pas une solution optimale, il fournit une approximation avec un facteur de 2, ce qui est un compromis raisonnable entre qualité et performance.

Les résultats obtenus montrent que l'algorithme est capable de produire des solutions proches de l'optimum pour des ensembles de points générés aléatoirement. Les fichiers Excel générés permettent une analyse et une visualisation faciles des données.