جامعة هواري بومدين للعلوم والتكنولوجيا

Université des Sciences et de la Technologie Houari-Boumediene

Faculté d'Informatique

Rapport de Projet

Thème

Δ-Problème du voyageur de commerce

Présenté par:

- Kerdjidj Mohamed Seddik
- Zeghouf Samir

Table des matières

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 (‘**kerdjidj_mohamedSeddik_zeghouf_samir.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 **('kerdjidj_mohamedSeddik_zeghouf_samir.csv').**

Analyse Théorique :

1. 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 :$O(V^2 \log V)$.

2. 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 :

kerdjidj_mohamedSeddik_zeghouf_samir.csv > data

```
=== Graph with 10 nodes ===
Distance Matrix:
,1,2,3,4,5,6,7,8,9,10
1,0.00000,5.51056,0.56148,6.21440,5.02626,6.30661,6.86696,1.89948,3.92283,3.10547
2,5.51056,0.00000,5.71289,0.75869,2.90903,2.43458,4.02998,4.54301,6.96255,4.07517
3,0.56148,5.71289,0.00000,6.43872,4.92840,6.68791,7.33054,2.43160,4.44829,3.62311
4,6.21440,0.75869,6.43872,0.00000,3.51253,2.14737,3.78466,5.13081,7.39724,4.53195
5,5.02626,2.90903,4.92840,3.51253,0.00000,5.28135,6.78737,5.12186,7.96270,5.35508
6,6.30661,2.43458,6.68791,2.14737,5.28135,0.00000,1.63740,4.71302,6.25605,3.70579
7,6.86696,4.02998,7.33054,3.78466,6.78737,1.63740,0.00000,5.05042,5.80843,3.86293
8,1.89948,4.54301,2.43160,5.13081,5.12186,4.71302,5.05042,0.00000,2.85118,1.21544
9,3.92283,6.96255,4.44829,7.39724,7.96270,6.25605,5.80843,2.85118,0.00000,2.88765
10,3.10547,4.07517,3.62311,4.53195,5.35508,3.70579,3.86293,1.21544,2.88765,0.00000
```

```
15  l'arbre couvrant minimal:
16  from,to,weight
17  1,3,0.56148
18  2,4,0.75869
19  8,10,1.21544
20  6,7,1.63740
21  1,8,1.89948
22  4,6,2.14737
23  8,9,2.85118
24  2,5,2.90903
25  6,10,3.70579
26  total,,17.68586
```

kerdjidj_mohamedSeddik_zeghouf_samir.csv > data

```
1   === Graph with 10 nodes ===
28  parcours eulerien:
29  from,to,distance
30  1,3,0.56148
31  3,1,0.56148
32  1,8,1.89948
33  8,10,1.21544
34  10,6,3.70579
35  6,7,1.63740
36  7,6,1.63740
37  6,4,2.14737
38  4,2,0.75869
39  2,5,2.90903
40  5,2,2.90903
41  2,4,0.75869
42  4,6,2.14737
43  6,10,3.70579
44  10,8,1.21544
45  8,9,2.85118
46  9,8,2.85118
47  8,1,1.89948
48  total,,35.37172
```

kerdjidj_mohamedSeddik_zeghouf_samir.csv > data

```
1   === Graph with 10 nodes ===
50  cycle hamiltonien:
51  from,to,distance
52  1,3,0.56148
53  3,8,2.43160
54  8,10,1.21544
55  10,6,3.70579
56  6,7,1.63740
57  7,4,3.78466
58  4,2,0.75869
59  2,5,2.90903
60  5,9,7.96270
61  9,1,3.92283
62  total,,28.88962
```

Et y en a aussi pour un ensemble de 20 et 50 points générés aléatoirement, les résultats obtenus sont dans le fichier Excel **(‘kerdjidj_mohamedSeddik_zeghouf_samir.csv’).**

Conclusion :

Cet algorithme offre une solution approchée efficace pour le TSP, avec une complexité de **$O(V^2 \log V)$** 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.