

Rapport de Projet de Graphe

Problème du voyageur de commerce (TSP)

4.a) Parcours optimale :

On énumère tous les parcours possibles qu'on sauvegarde dans un tableau.

On calcule tous la somme des poids pour chaque parcours qu'on sauvegarde dans un tableau.

On cherche le plus petit des parcours et on l'affiche (c'est le parcours optimale).

4.b) Plus proche voisin :

On part d'un sommet de départ et le relie à son voisin le plus proche par une arête. De ce nouveau sommet, nous recherchons son voisin le plus proche non visité et nous le relions par une nouvelle arête, et ainsi de suite, jusqu'à ce que tout les sommets ai été visité. Nous utilisons une file de priorité pour trouver le plus proche voisin d'un sommet.

PS : Bien que la file de priorité marche bien en générale, il y a de rare cas où elle ne prend pas le voisin le plus proche. Nous n'avons pas réussi à en trouver la cause.

4.c) Plus court détour :

Nous partons de trois sommets de départ que nous relions en triangle par trois arêtes. Nous cherchons ensuite parmi les sommets non visités, celui qui permettra d'avoir le plus petit détour entre les deux extrémités d'une des arêtes présentes, et ce pour toutes les arêtes présentes. Nous remplaçons ensuite l'une de ces arêtes par une deux nouvelles arêtes passant par le détour le plus minimal. Nous recommençons jusqu'à la reliure de tout les sommets par des arêtes.

4.d) ARPM – Prime + parcours en profondeur :

On choisit un sommet de départ. On cherche l'arrête de poids minimal qui permet de rajouter un nouveau sommet. Une fois qu'on a ajouté tous les sommets l'arbre de recouvrement est fait (l'ARPM). Nous effectuons ensuite un parcours en profondeur. Ce sera le parcours hamiltonien et nous affichons sa solution.

4.e) Christofides : non implémenté :

Exemple :

- Test 1 : « ./bin/exec 6 75 » : 6 sommet et graine : 75
- Test 2 : « ./bin/exec 7 94 » : 7 sommet et graine : 94
- Test 3 : « ./bin/exec 9 32 » : 9 sommet et graine : 32

Résultats :

4.a) Parcours optimale :

- Test 1 : (2 -3 -5 - 6 – 1 - 4) poids total : 124.00
- Test 2 : (1 – 6 – 2 – 3 - 7 – 4 - 5) poids total : 165.00
- Test 3 : Erreur ne marche pas pour $N > 7$, complexité trop élevée (A partir de 8 sommets la recherche met trop de temps et à partir de 9 sommets nous obtenons une erreur de segmentation).

4.b) Plus proche voisin :

Nous avons choisi en tant que sommet de départ le sommet 1 :

- Test 1 : (1 – 6 – 4 – 5 - 3 - 2)
- Test 2 : (1 – 6 – 2 – 4 - 5 – 7 - 3)
- Test 3 : (1 – 7 – 6 – 2 - 8 – 9 – 5 – 3 - 4)

4.c) Plus court detour :

Nous avons choisi en tant que trio de sommets de départs les sommets 1, 2 et 3 :

- Test 1 : (1 – 2 – 5 – 6 - 4 - 3)
- Test 2 : (1 – 5 – 4 – 2 - 7 – 6 - 3)
- Test 3 : (7 – 6 – 1 – 2 - 4 – 8 – 5 – 9 - 3)

4.d) ARPM – Prime :

Nous avons choisi en tant que sommet de départ le sommet 3 :

- Test 1 : (5 - 3 - 2 - 1 – 4 - 6)
- Test 2 : (5 - 4 - 2 - 1 - 3 - 6 - 7)
- Test 3 : (5 - 3 - 1 - 2 - 6 - 7 - 4 - 8 – 9)