Pénélope Delabrière
M1 SDL – TPA42
n° 21902367

UNIVERSITÉ TOULOUSE III PAUL SABATIER Université de Toulouse

Compte rendu tp3
– voyageur de commerce –

2022 – 2023

Algorithmes implémentés

– **read(file_m) :** lit un ficher donné en entrée et retourne une liste de couples. Chaque couple (numéro_ville, coord_x, coord_y) représente une ville.

– **random_solution(file_m) :** Retourne une solution initiale à partir du fichier donné en entrée, grâce à la méthode constructive décrite dans l'énoncé.

– **liste_villes(solution) :** Retourne un liste d'entier, où chaque entier est le numéro d'une ville tirée à partir d'une solution de liste de couples donnée en entrée. (Permet de retourner des solutions plus lisibles sans les coordonnées des villes).

– **liste_villes_tabou(tabou) :** Effectue le même traitement que liste_ville, pour une liste de solutions donnée en entrée.

– **euclidian_distance(v, w) :** Retourne la distance euclidienne entre 2 vecteur à 2 coordonnées.

– **distance(solution) :** Calcule la valeur d'une solution, càd la distance que le voyageur de commerce doit parcourir en partant de (0,0) puis en passant par toutes les villes, puis en revenant en (0,0).

– **voisins_non_tabou(solution, tabou) :** Calcule tous les voisins d'une solution qui ne se trouvent pas dans la liste tabou

– **voisins(solution) :** Calcule tous les voisins d'une solution donnée en entrée.

– **meilleur_voisin(voisins) :** Retourne le voisin dont la valeur de la distance est minimale.

– **steepest_Hill_Climbing(file_m, max_depl)**

– **steepest_Hill_Climbing_redemarrage(file_m, max_depl, max_essais)**

– **tabou(file_m, max_depl, taille_tabou)**

Voisinage implémenté

Un voisin X' d'une solution X est une solution telle que 2 villes de X ont été permutées.

Instances testées

- Pour le fichier tsp5.txt :

Steepest Hill Climbing

On effectue au plus 4 déplacements pour arriver à un optimum local. Le nombre de déplacements moyen est de 2 à 3.

Les optimums locaux trouvés avec max_depl ≥ 4 sont
– [1, 2, 4, 5, 3] et d = 194,
– [3, 5, 4, 2, 1] et d = 194,
– [5, 4, 2, 3, 1] et d = 196,
– [1, 3, 2, 4, 5] et d = 196

Temps effectué moyen en < 0,001 sec

Steepest Hill Climbing avec redémarrage

Pour max_essais suffisamment grand et max_depl ≥ 4, on obtient les optimums globaux suivants :
– [1, 2, 4, 5, 3] et d = 194,
– [3, 5, 4, 2, 1] et d = 194 Temps effectué moyen en < 0,01 sec avec max_essais ≤ 99

Tabou

Si la taille de la liste tabou >=3 avec max_depl >=7, on est sûr d'obtenir les optimums globaux. On ne pourra jamais garantir l'optimalité sinon. Les optimums globaux trouvés apparaissent en rouge.

– Pour max_depl = 10 , taille_tabou = 3:

solution initiale : [1, 5, 4, 3, 2] solution atteinte : [3, 5, 4, 2, 1] meilleure solution rencontrée : [3, 5, 4, 2, 1] distance de la solution : 194.04 nombre de déplacement parcouru : 10 tabou = [[1, 3, 5, 2, 4], [1, 3, 4, 2, 5], [1, 5, 4, 2, 3]] temps effectué : 0.00090 sec	solution initiale : [2, 3, 5, 1, 4] solution atteinte : [2, 4, 5, 3, 1] meilleure solution rencontrée : [1, 2, 4, 5, 3] distance de la solution : 194.04 nombre de déplacement parcouru : 10 tabou = [[1, 4, 2, 5, 3], [3, 4, 2, 5, 1], [5, 4, 2, 3, 1]] temps effectué : 0.00090 sec

– Pour max_depl = 6 , taille_tabou = 4:

solution initiale : [1, 4, 3, 2, 5] solution atteinte : [1, 4, 2, 5, 3] meilleure solution rencontrée : [1, 3, 2, 4, 5] distance de la solution : 196.12 nombre de déplacement parcouru : 6 tabou = [[1, 3, 2, 4, 5], [1, 3, 5, 4, 2], [1, 3, 5, 2, 4], [1, 4, 5, 2, 3]] temps effectué : 0.00057 sec	solution initiale : [2, 4, 5, 3, 1] solution atteinte : [3, 4, 2, 5, 1] meilleure solution rencontrée : [1, 2, 4, 5, 3] distance de la solution : 194.04 nombre de déplacement parcouru : 6 tabou = [[5, 2, 4, 3, 1], [3, 2, 4, 5, 1], [1, 2, 4, 5, 3], [1, 4, 2, 5, 3]] temps effectué : 0.00061 sec

- Pour le fichier tsp101.txt :

Steepest Hill Climbing

Temps de calcul moyen de 205 secondes pour effectuer l'algorithme. Le nombre de déplacements varie entre 90 et 150 par essai. Optimum global non trouvé. Optimums locaux nombreux.

Voici les données du meilleur optimum local trouvé ayant 200 pour nombre de déplacements maximal avec l'algorithme ;

solution initiale : [44, 70, 18, 37, 64, 5, 87, 22, 28, 88, 96, 7, 10, 67, 6, 93, 32, 46, 26, 9, 43, 27, 13, 53, 47, 60, 4, 77, 30, 63, 33, 65, 39, 40, 99, 34, 52, 42, 73, 51, 80, 86, 49, 19, 14, 61, 23, 24, 21, 57, 8, 59, 68, 35, 11, 91, 2, 101, 97, 94, 16, 45, 20, 58, 72, 66, 25, 83, 50, 78, 54, 31, 76, 82, 48, 1, 62, 17, 85, 74, 29, 56, 75, 89, 3, 15, 41, 36, 55, 92, 98, 69, 38, 100, 71, 95, 84, 90, 79, 81, 12]
solution atteinte : [76, 98, 10, 7, 6, 4, 2, 62, 55, 72, 94, 83, 11, 12, 13, 8, 9, 47, 46, 5, 3, 54, 100, 53, 87, 75, 58, 85, 35, 29, 27, 28, 30, 32, 31, 33, 34, 64, 86, 52, 77, 90, 19, 49, 22, 20, 23, 25, 84, 66, 67, 81, 1, 91, 70, 99, 56, 101, 69, 82, 97, 95, 93, 92, 65, 21, 50, 24, 26, 78, 60, 88, 14, 16, 17, 18, 48, 15, 79, 74, 80, 61, 89, 71, 40, 39, 38, 36, 37, 41, 44, 45, 43, 42, 73, 68, 51, 63, 96, 57, 59]
nombre de déplacement parcouru : 126
distance de la meilleure solution : 1001.91 sec
temps effectué : 214.07 sec

Les distances autres optimums locaux trouvés varient en 1001 et 1400 km.

Steepest Hill Climbing avec redémarrage

Ci-dessous les résultats des 3 instances testées sur l'algorithme avec max_depl = 200 sur chaque instance :

max_essais = 10	max_essais = 30	max_essais = 50
meilleure solution rencontrée : [76, 59, 98, 60, 88, 53, 84, 67, 66, 89, 7, 6, 4, 2, 82, 72, 35, 33, 34, 64, 86, 85, 57, 65, 23, 19, 49, 22, 24, 87, 10, 14, 11, 100, 58, 25, 26, 78, 75, 54, 99, 70, 91, 1, 81, 92, 93, 95, 68, 96, 52, 77, 90, 20, 50, 21, 63, 51, 32, 30, 28, 27, 29, 31, 94, 97, 55, 73, 40, 41, 44, 45, 43, 62, 56, 61, 17, 16, 12, 13, 15, 48, 18, 79, 74, 80, 8, 9, 47, 46, 5, 3, 101, 71, 42, 39, 38, 36, 37, 69, 83] distance de la solution : 1083.03 temps effectué : 6133 sec	meilleure solution rencontrée : [76, 98, 60, 75, 57, 96, 63, 51, 68, 72, 73, 40, 42, 55, 97, 95, 93, 85, 86, 64, 34, 33, 31, 29, 27, 28, 30, 32, 35, 94, 82, 62, 43, 45, 39, 38, 36, 37, 41, 44, 2, 4, 6, 46, 5, 3, 101, 56, 99, 89, 61, 74, 80, 8, 7, 1, 92, 81, 69, 71, 47, 9, 70, 91, 66, 67, 84, 58, 25, 23, 21, 50, 20, 90, 77, 52, 65, 11, 13, 16, 17, 18, 48, 15, 79, 54, 83, 100, 53, 87, 88, 10, 12, 14, 59, 26, 24, 22, 49, 19, 78] distance de la solution : 978.69 temps effectué : 6305 sec	meilleure solution rencontrée : [76, 98, 17, 18, 48, 16, 60, 59, 78, 26, 33, 31, 32, 35, 51, 86, 64, 77, 90, 34, 29, 27, 28, 30, 52, 21, 50, 20, 24, 22, 49, 19, 23, 25, 69, 62, 55, 82, 1, 91, 66, 58, 84, 67, 81, 92, 96, 85, 65, 57, 93, 95, 94, 72, 97, 56, 70, 99, 89, 54, 11, 10, 14, 12, 13, 79, 61, 7, 3, 101, 71, 45, 44, 41, 43, 42, 40, 39, 37, 36, 38, 73, 68, 63, 87, 75, 88, 53, 100, 83, 2, 4, 6, 46, 5, 47, 9, 8, 80, 74, 15] distance de la solution : 1055.76 temps effectué : 10440 sec

Tabou

Algorithme testé sur 3 instances décrits ci-dessous :

Avec max_depl = 500, liste_tabou = 10, on a :

- meilleure solution rencontrée : [75, 53, 81, 97, 82, 69, 101, 71, 43, 40, 37, 36, 38, 55, 1, 91, 66, 67, 92, 95, 68, 63, 51, 94, 72, 73, 39, 42, 62, 70, 99, 83, 84, 52, 64, 86, 85, 96, 93, 57, 65, 21, 23, 25, 58, 54, 89, 3, 7, 80, 74, 61, 56, 45, 41, 44, 2, 4, 6, 46, 5, 47, 9, 8, 79, 13, 12, 10, 88, 87, 35, 32, 30, 28, 27, 29, 77, 50, 20, 49, 19, 90, 34, 31, 33, 100, 11, 15, 48, 18, 17, 22, 24, 26, 78, 59, 60, 14, 16, 98, 76]
- distance de la solution : 1199.76
- nombre de déplacement parcouru : 500
- temps effectué : 866 sec

Avec max_depl = 500, taille_tabou = 15, on a :

- meilleure solution rencontrée : [76, 60, 54, 99, 69, 82, 62, 41, 37, 36, 55, 97, 93, 95, 94, 73, 40, 39, 38, 32, 30, 28, 27, 29, 31, 33, 34, 96, 57, 92, 81, 71, 101, 7, 47, 9, 8, 80, 74, 88, 59, 78, 26, 24, 21, 65, 67, 66, 91, 70, 83, 100, 84, 52, 77, 90, 64, 86, 85, 58, 53, 11, 12, 13, 15, 79, 61, 89, 1, 68, 63, 51, 35, 72, 42, 43, 45, 44, 2, 4, 6, 46, 5, 3, 56, 25, 23, 50, 20, 19, 49, 22, 75, 87, 10, 48, 18, 17, 16, 14, 98]
- distance de la solution : 1108.46
- nombre de déplacement parcouru : 500
- temps effectué : 880 sec

Avec max_depl = 500, taille_tabou = 30, on a :

- meilleure solution rencontrée : [78, 26, 24, 22, 49, 19, 20, 50, 21, 66, 83, 54, 47, 9, 8, 61, 89, 99, 70, 91, 1, 82, 62, 71, 3, 7, 101, 69, 81, 92, 95, 97, 55, 43, 45, 44, 73, 72, 94, 86, 64, 90, 77, 52, 84, 53, 10, 14, 16, 17, 18, 74, 80, 46, 6, 4, 2, 5, 56, 67, 65, 57, 85, 96, 93, 68, 63, 51, 35, 32, 30, 28, 27, 29, 31, 33, 34, 87, 60, 98, 88, 100, 42, 40, 39, 38, 36, 37, 41, 79, 48, 15, 13, 12, 11, 58, 23, 25, 75, 59, 76]
- distance de la solution : 1068.22
- nombre de déplacement parcouru : 500
- temps effectué : 930 sec