

# Résolution pratique du TSP

## I. Heuristiques globales

### Question 1 :

#### Implémentation de la construction itérative par ajout du plus proche voisin

J'ai choisi d'implémenter la version décrite dans le TP ainsi qu'une variante.

*Celle présente dans le TP (heuristique\_iter\_alea):*

- Elle choisit un premier sommet aléatoirement.
- Puis fait appel à la fonction `heuristique_iter_premier_sommet` qui construit une tournée en ajoutant à chaque itération le sommet le plus proche suivant le premier sommet qu'on lui a donné tant qu'il y a encore des villes non visitées.

*L'alternative (heuristique\_iter\_chaque\_sommet\_debut):*

- Pour chaque sommet possible, on appelle la fonction `heuristique_iter_premier_sommet` et on prend la meilleure tournée parmi celle trouvées à chaque appel de cette fonction.

### Question 2 :

La première version ne retourne pas toujours la même valeur car cela dépend du choix du premier sommet.

Résultats obtenus avec la solution alternative sur les différentes données :

Fichiers de données	Tournée	Valeur de la tournée
exe5.atsp	[1, 3, 2, 0, 4]	22
exe7.atsp	[3, 4, 5, 6, 0, 1, 2]	92
exe7b.atsp	[0, 2, 1, 5, 6, 3, 4]	70
br17.atsp	[3, 4, 5, 6, 14, 15, 0, 11, 1, 9, 10, 12, 2, 13, 7, 8, 16]	56
bays29.atsp	[22, 26, 23, 7, 0, 27, 5, 11, 8, 4, 25, 28, 2, 1, 20, 19, 9, 3, 14, 17, 13, 21, 16, 10, 18, 15, 12, 24, 6]	2134
bayg29.atsp	[20, 1, 19, 9, 3, 14, 18, 24, 6, 22, 26, 23, 7, 27, 0, 5, 11, 8, 4, 25, 28, 2, 12, 15, 10, 21, 13, 17, 16]	1935

## II. Méthodes locales

### Question 1 :

#### Implémentation de HillClimbing

J'ai choisi d'implémenter une version pour les problèmes ayant une matrice de distance symétrique et une autre pour les autres cas.

*Versión avec matrice symétrique (optium\_local\_tsp):*

- On trouve une tournée grâce à l'algorithme précédent
- On essaye de faire différentes permutations afin de voir si on en trouve une meilleure
- Si c'est le cas, on recommence les permutations avec la nouvelle tournée
- Sinon, on s'arrête là

*Version avec matrice non symétrique (optium\_local\_atsp):*

Algorithme semblable au précédent mais avec la permutation 'inverse' (ex : ABC => CBA)

### Question 2 :

Résultats obtenus avec utilisation de l'algorithme précédent (heuristique globale) suivant chaque sommet :

Fichiers de données	Tournée	Valeur de la tournée
exe5.atsp	[1, 3, 2, 0, 4]	22
exe7.atsp	[3, 4, 5, 6, 0, 1, 2]	92
exe7b.atsp	[0, 2, 1, 5, 6, 3, 4]	70
br17.atsp	[3, 4, 5, 6, 14, 15, 0, 11, 1, 9, 10, 12, 2, 13, 7, 8, 16], 56)	56
bays29.atsp	[22, 26, 23, 7, 0, 27, 5, 11, 8, 4, 25, 28, 2, 1, 20, 19, 9, 3, 14, 17, 13, 21, 16, 10, 18, 15, 12, 24, 6]	2134
bayg29.atsp	[20, 1, 19, 9, 3, 14, 18, 24, 6, 22, 26, 23, 7, 27, 0, 5, 11, 8, 4, 25, 28, 2, 12, 15, 10, 21, 13, 17, 16]	1935

## III. Méthodes exactes

### Question 1 : (*fichier methode\_exacte*)

Cette méthode permet de n'explorer un noeud que si sa borne inférieure est inférieure à la borne sup (borne trouvée grâce à l'algorithme de l'heuristique globale). Cela permet de réduire le nombre d'explorations. Cependant, cet algorithme n'est pas praticable sur de grandes données. Pour cela, il faudrait modifier l'algorithme afin de descendre un noeud et éventuellement mettre à jour la borne sup, afin de pouvoir après explorer et descendre dans les autres noeuds.

### Question 3 :

$\text{solution\_initiale} = \text{solution\_reduit} + \text{borne\_inf}$

### Question 4 :

Fonction *borne\_inf\_simplifie\_donnee* dans le *fichier methode\_exacte\_deuxieme* :

- Parcourt chaque ligne, et soustrait la valeur la plus petite trouvée pour chaque ligne
- Parcourt toute la matrice, en retenant la valeur la plus petite de chaque colonne, pour chaque colonne, soustrait la valeur trouvée précédemment

### Question 5 :

Méthode en  $O(k^2)$  avec  $k$  le nombre de colonnes (= lignes) dans la matrice à étudier.

### Question 6 :

Si on sélectionne l'arc  $(i,j)$ , on interdit alors les arcs:

- $(i, k)$  avec  $k \neq j$  (ie tous les autres arcs ayant comme départ  $i$ )
- $(k, j)$  avec  $k \neq i$  (ie tous les autres arcs ayant comme arrivée  $j$ )
- $(j, deb)$  avec  $deb$  la ville de départ de la tournée

Même si ce n'est pas explicite, l'arc  $(j,i)$  est interdit. Dans le premier cas, cela est fait grâce à l'interdiction de l'arc  $(j, deb)$ . Dans les cas suivants, il est fait lors de l'étape précédente (i.e. lors de l'interdiction des arcs précédents)