

**Etude comparative des heuristiques**

S2.02 - Exploration algorithmique d'un problème

Léo Bourdin / Antoine Lindimer / Romain Barabant

IUT DIJON AUXERRE  Département informatique

# Critères de comparaison

## Complexité algorithmique

La complexité algorithmique représente le nombre de calculs basiques que fera l’algorithme selon la taille du problème. Pour un même algorithme, la complexité peut parfois être légèrement différente selon la qualité du code.

**Pour ce critère** : le meilleur algorithme sera celui avec la complexité algorithmique la plus petite.

## Temps d’exécution

Le temps d’exécution représente la durée que prendra l’algorithme pour calculer l’ordre de la tournée finale. La Stopwatch mesurera et affichera cette durée en milliseconde pour chaque algorithme.

**Pour ce critère** : le meilleur algorithme sera celui avec le temps d’exécution moyen le plus petit.

*Remarque : pour que ce critère soit sensé, on exécutera chaque algorithme sur l’ensemble du jeu d’essai afin qu’il soit soumis à différentes situations.*

## Distance

La distance représente la distance totale de la tournée finale trouvée par l’algorithme.

**Pour ce critère** : le meilleur algorithme sera celui avec la distance moyenne le plus petit.

*Remarque : pour que ce critère soit sensé, on exécutera chaque algorithme sur l’ensemble du jeu d’essai afin qu’il soit soumis à différentes situations.*

# Jeu d’essai de graphes

Comme susdit, le jeu d’essai aura pour but de faire rencontrer aux algorithmes des situations diverses et variées afin de récolter des données plus précises, donc utiles pour la comparaison.

# Mesure qualitative des algorithmes

## Plus proche voisin

## Insertion proche

## Insertion loin

## Recherche locale

## Heuristiques personnelles

# Conclusion

# Correction (provisoire

inverser condition

stocker distanceTournee

renvoyer un couple de valeur (ne pas utiliser un attribut)

cas particulier pour jeu d'essais