

Devoir 2

Dans ce devoir, vous devez concevoir et réaliser une implantation d'une recherche tabou et d'un algorithme génétique pour le problème de tournées de véhicules avec contraintes de capacité. Voici quelques remarques importantes.

- L'implantation doit pouvoir s'exécuter sous Windows (je recommande le langage Java).
- Pour évaluer vos algorithmes, vous disposez de 5 problèmes Euclidiens comprenant entre 50 et 199 clients (disponibles sur la page web du cours).
- Pour le calcul des distances Euclidiennes entre les sommets, vous devez arrondir ces distances à la quatrième décimale. La valeur finale d'une solution correspond à la distance totale parcourue par les véhicules et celle-ci doit être arrondie à la deuxième décimale. Évidemment, l'objectif consiste à minimiser la distance totale.

Dans votre rapport, vous devez inclure les informations suivantes:

- (a) Description des deux algorithmes, comprenant entre autres les éléments suivants :
 - recherche tabou* : espace des solutions admissibles, fonction objective, structure de voisinage, liste tabou, critères d'aspiration, intensification, diversification, critère d'arrêt, ...
 - algorithme génétique* : encodage d'une solution, sélection, croisement, mutation, remplacement de la population, critère d'arrêt, ...
- (b) Informations techniques dignes d'intérêt: structures de données, astuces d'implantation, ...
- (c) Résultats expérimentaux
 - Indiquer le type d'ordinateur et les valeurs des paramètres utilisées pour obtenir les solutions rapportées. Pour chaque problème, indiquer la solution obtenue (séquence de clients dans chacune des tournées et distance totale) ainsi que le temps de calcul. Indiquer aussi le «germe» utilisé pour générer la séquence de nombres aléatoires si votre implantation comprend des éléments probabilistes.
 - Dans le cas d'un algorithme avec des éléments probabilistes, exécuter l'algorithme cinq fois sur chaque problème et indiquer la valeur de la meilleure solution, de la pire solution et la valeur moyenne des solutions obtenues (ainsi que le temps moyen de calcul).
 - Présenter tout autre résultat digne d'intérêt (e.g., étude de sensibilité des paramètres).
 - Établir une comparaison entre les deux algorithmes : qualité des solutions obtenues, temps d'exécution, facilité d'implantation, robustesse, ...
- (d) Inclure un guide de l'utilisateur (en particulier, spécifier les entrées et les sorties de vos programmes).

Critères de correction (pour chacun des deux algorithmes)

Présentation	10%
Algorithme	60%
Résultats expérimentaux	30%

A remettre, mardi le 16 mars (40% de la note finale).

Le code source et le code compilé devront être envoyés par courrier électronique à:
potvin@iro.umontreal.ca.