

## 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 devrait pouvoir s'exécuter sous Windows (je recommande le langage Java).
- Pour évaluer vos algorithmes, vous disposez de 5 instances comprenant entre 50 et 199 clients, en plus du dépôt (disponibles dans StudiUM).
- La valeur d'une solution correspond à la distance totale parcourue par les véhicules et celle-ci doit être arrondie à la deuxième décimale à la fin. 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 objectif, 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) Toute information technique digne d'intérêt: structures de données, astuces d'implantation, ...
- (c) Résultats expérimentaux
  - Indiquer le type d'ordinateur et les valeurs des paramètres utilisées. Il est permis de faire appel à une fonction qui détermine les valeurs des paramètres en fonction des caractéristiques d'une instance, mais vous ne pouvez pas ajuster les valeurs des paramètres « à la main » pour chaque instance.
  - Dans le cas d'un algorithme déterministe, décrire la solution obtenue pour chaque instance (séquences de clients dans les tournées) et indiquer la valeur de la solution ainsi que le temps de calcul.
  - Dans le cas d'un algorithme contenant 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, ainsi que le temps de calcul moyen. Indiquer aussi le «germe» utilisé pour générer la séquence de nombres aléatoires dans chacun des cas. Décrire la meilleure solution obtenue pour chaque instance (séquences de clients dans les tournées).
  - 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, le jeudi 12 mars (30% de la note finale).

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