

Métaheuristiques

Master 1 Informatique

Projet - Problème du voyageur de commerce

Indications générales

Date de rendu : 15 mars 2020 avant 8 h 00

Travail en groupes de deux

Les fichiers seront déposés sur ARCHE dans une archive `.zip`

Problème du voyageur de commerce

Le problème du voyageur de commerce est un problème d'optimisation classique. Un voyageur doit passer par diverses villes, exactement une fois chacune et retourner à son point de départ, en choisissant le parcours le plus court possible. Le problème est NP-Complet et a été étudié depuis de longues années.

Le projet s'appliquera sur quatre instances du problème. Trois d'entre elles (`bier127`, `fn14461` et `gr666`), répertoriées par l'université de Heidelberg (TSPLIB), sont mises à disposition pour le développement du projet. La dernière (`surprise1350`) ne sera utilisée que lors de l'évaluation.

Le projet

On cherche à développer un algorithme métaheuristique permettant de trouver une solution satisfaisante au problème du voyageur le plus efficacement possible. Chaque projet disposera de 60 secondes de calcul pour trouver une solution. Les méthodes étant stochastiques, une moyenne sur 10 exécutions sera utilisée pour évaluer les solutions obtenues.

Le code existant

Le projet devra être développé en langage Java en se basant sur les classes fournies. Tout le code développé se trouvera dans un **unique package** nommé `tsp.projects.nomduprojet` qui devra être créé. Ce *package* devra contenir **une seule classe étendant `Project.java`**. L'**unique** constructeur de la classe devra nécessairement comporter **un seul** paramètre de classe `Evaluation` et exécuter obligatoirement et uniquement les méthodes `addAuthor` et `setMethodName` afin d'identifier le groupe d'étudiants pour l'évaluation du projet.

La classe `Project.java` contient deux méthodes abstraites qu'il faut définir. La méthode `initialization` est exécutée une fois au début de l'optimisation. L'initialisation doit se faire dans cette méthode et pas dans le constructeur. La méthode `loop` est la boucle principale de l'algorithme. Cette boucle est répétée pendant 60 secondes. La classe `Project.java` contient également deux attributs : `Problem problem` (liste des coordonnées des villes à parcourir) et `Evaluation evaluation` (fonction d'évaluation).

Une solution potentielle prendra la forme d'une instance de la classe `tsp.evaluation.Path` et peut être évaluée par la méthode `tsp.evaluation.Evaluation.evaluate (Path path)`. Cette méthode met automatiquement à jour la meilleure solution découverte. Une fois le temps écoulé, la solution n'est plus mise à jour.

Il **n'est pas** autorisé d'utiliser des bibliothèques Java existantes sur les algorithmes métaheuristiques. D'autres classes peuvent bien entendu être créées dans le *package*.

Les étudiants pourront s'inspirer du package `tsp.projects.exemple` et de la classe `RandomSearch`.

Le travail demandé

Il est demandé de réaliser, en langage de programmation Java, un unique algorithme d'optimisation méta-heuristique qui donne une solution satisfaisante au problème du voyageur de commerce. Il n'est pas autorisé de faire du calcul distribué, que ce soit en local (multi-threads) ou à distance (serveurs de calculs), de charger des informations autres que les données fournies, d'instancier les algorithmes adverses.

Il conviendra d'utiliser l'encodage UTF-8 (si les chaînes de caractères comportent des caractères spéciaux, par exemple). **Seul** le dossier contenant le *package* `tsp.projects.nomduprojet` devra être envoyé. Le code devra être rendu dans une archive `.zip`. **Aucune** modification du code existant n'est tolérée.

Un rapport de dix pages maximum est également demandé. Ce rapport devra être convenablement structuré (page de garde, introduction, conclusion, etc.). Il développera la recherche bibliographique réalisée et décrira le principe de la méthode d'apprentissage mise en œuvre.

La notation

La notation sera décomposée en quatre parties :

- l'ingéniosité et la pertinence la solution mise en œuvre (9 points) ;
- la qualité de la programmation en langage Java (3 points) ;
- la qualité de la rédaction du rapport (3 points) ;
- l'efficacité face aux solutions proposées par les autres groupes d'étudiants sur les quatre instances du problème (5 points).

Les méthodes des différents groupes d'étudiants seront comparées entre elles. Le classement obtenu déterminera la note entre 1 (dernier du classement) et 5 (premier du classement). Une note de 0 sera attribuée aux groupes disqualifiés (méthodes non conformes au cahier des charges).

Remarques

Les questions concernant le projet sont à poser par e-mail à l'adresse suivante : `alexandre.blanche@univ-lorraine.fr`.