

1 Introduction

L'objectif de ce travail est de compléter l'initiation aux notions abordées pendant les séances de cours du module autour de l'intérêt d'utiliser des heuristiques et/ou des métaheuristiques lorsque le problème à résoudre est "*difficile*".

Pour cela vous allez travailler en équipe.

Une équipe est, par définition, composée de plusieurs individus. Il ne s'agit pas d'un travail individuel mais bien d'une collaboration qui doit notamment contribuer à renforcer vos compétences organisationnelles au sein d'un groupe.

L'organisation au sein de l'équipe n'est pas du ressort de l'équipe enseignante qui peut néanmoins intervenir en cas de difficultés ou problèmes. Il est **très important** que chaque membre de l'équipe puisse mettre en avant son rôle et ses contributions dans la réalisation du projet et son aboutissement. Ces éléments seront pris en compte dans la notation et pourront déboucher sur une note différente pour chacun.

D'un point de vue pratique, toutes les séances restantes du module seront consacrées à ce travail d'équipe.

Toutes les informations et échanges de documents se feront via Moodle sur la page du cours : <https://moodle.uphf.fr/course/view.php?id=5628>

2 Description du problème

En informatique théorique, le problème de couverture par ensembles (Set Cover problem en anglais) est un problème d'algorithmique particulièrement important car c'est l'un des 21 problèmes NP-complets de Karp.

Étant donné un univers U constitué des éléments $\{1, 2, \dots, m\}$ et une collection S de n sous-ensembles de U , où chaque sous-ensemble j de S est associé à un poids positif c_j , le problème de la couverture d'ensemble consiste à trouver une sous-collection de S dont l'union est égale à U et dont la somme des poids est minimale.

Le problème peut être modélisé de la manière suivante. Considérons une matrice binaire $A = (a_{ij})$ de taille $m \times n$ où $a_{ij} = 1$ si et seulement si l'élément i appartient au sous-ensemble j . En utilisant des variables binaires x_j , telles que $x_j = 1$ si et seulement si le sous-ensemble j est sélectionné, le problème de la couverture d'ensembles peut être modélisé de la manière suivante :

$$\min \sum_{j=1}^n c_j x_j \tag{1}$$

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i \in \{1, 2, \dots, m\}; \tag{2}$$

$$x_j \in \{0, 1\}, j \in \{1, 2, \dots, n\}. \quad (3)$$

Dans ce modèle, la fonction objective vise à minimiser le poids total de tous les sous-ensembles sélectionnés. La contrainte (2) garantit que chaque élément i appartient à au moins un sous-ensemble sélectionné.

3 Travail attendu - tâches à réaliser

Le but de ce projet est de développer une approche basée sur les principes des métaheuristiques pour résoudre ce problème. Afin d'atteindre cet objectif, il faut développer les algorithmes suivants :

- Une procédure de lecture des données ; Les données sont accessibles via le lien suivant <https://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html>
- Une procédure de construction d'une solution réalisable ;
- Une procédure de vérification de la faisabilité d'une solution ;
- Différentes recherches locales :Ajouter un sous-ensemble ; Retirer un sous-ensemble ; Échanger deux sous-ensembles ;
- Développer une approche basée sur les procédures précédentes (VNS ou Tabou Search).

La complexité des structures de données utilisées et des différents sous-programmes de l'algorithme doit être estimée et expliquée. De plus, il est nécessaire de fournir des tableaux de résultats. Ces tableaux doivent permettre de comparer la valeur de la solution initiale, les valeurs des solutions obtenues après les procédures de recherche locale (LS) et la valeur de la solution obtenue par l'approche métaheuristique. Enfin, les tableaux doivent contenir les temps CPU en secondes nécessaires pour atteindre chaque valeur de solution.

4 Dépôt et modalités

Le langage de programmation est le langage C ou le langage C++ qui sont conseillés pour des questions de performance.

Un rapport sera à fournir par équipe. Ce rapport doit, à minima, présenter le problème, présenter et expliquer les algorithmes développés (en donnant le pseudo-code des procédures), présenter et commenter les expériences numériques réalisées et les résultats obtenus, donner une conclusion et des pistes d'amélioration. Il doit aussi clairement indiquer les contributions de chacun.

Le rapport et le code source devront être déposés sur Moodle avant la date limite.

Il est fortement conseillé de ne pas attendre la dernière minute pour déposer votre travail, afin de limiter les risques liés à un problème de réseau par exemple.

Le dépôt se fera sous la forme d'une archive au format .zip, .rar ou .7z.