

## Sujet de Projet

---

# 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.

# 2 Description du problème

Le problème traité est le problème du sac-à-dos **multidimensionnel**, qui se base sur le problème du sac-à-dos. A partir d'un ensemble  $N = \{1, 2, \dots, n\}$  de  $n$  objets, caractérisés par un poids ( $a_j$ ) et un profit ( $c_j$ ), on cherche le sous-ensemble d'objets pouvant être sélectionné pour être mis dans un sac d'une certaine capacité ( $b$ ) en maximisant la somme des profits. Rappelons la formulation mathématiques du problème du sac-à-dos (SAD) en variables 0-1 :

$$(\text{SAD}) \quad \begin{cases} \max & \sum_{j=1}^n c_j x_j \\ \text{s.c. :} & \sum_{j=1}^n a_j x_j \leq b \\ & x_j \in \{0, 1\} \quad j \in N = \{1, \dots, n\} \end{cases}$$

Dans la variante multidimensionnelle (SADM) on considère un ensemble  $M = \{1, 2, \dots, m\}$  de contraintes de ressources (avec  $m > 1$ ). Ainsi chaque objet est caractérisé par  $m$  poids ( $a_{ij}$ ), un pour chaque contrainte. On obtient la formulation suivante :

$$(\text{SADM}) \quad \begin{cases} \max & \sum_{j=1}^n c_j x_j \\ \text{s.c. :} & \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i \in M = \{1, \dots, m\} \\ & x_j \in \{0, 1\} \quad j \in N = \{1, \dots, n\} \end{cases}$$

Tous les coefficients ( $a_{ij}$ ,  $c_j$ ,  $b_i$ ) sont supposés non négatifs.

Le SADM est un problème NP-Difficile. Votre travail consiste à implémenter une ou plusieurs méthodes approchées (heuristiques, métaheuristiques) de manière à obtenir les meilleures solutions **faisables** possibles, dans un temps imparti. Votre programme **devra donc s'arrêter à la fin de ce temps** (voir section Modalités).

Vous trouverez sur Moodle quelques articles scientifiques dont vous avez le droit de vous inspirer, ainsi qu'un ensemble de jeux de données que vous **devez** utiliser pour tester vos approches.

## Format des données (en entrée et en sortie) :

Un fichier texte est associé à chaque instance à traiter. Tous les fichiers respectent le même format, excepté pour les espaces ou lignes blanches qui peuvent différer d'un fichier à l'autre. Le format est le suivant :

- 1<sup>ère</sup> ligne : le nombre d'objets ( $n$ ) et le nombre de contraintes ( $m$ ) (séparés par au moins un espace)
- $n/10$  lignes suivantes : les coefficients des objets dans la fonction objective ( $c_j$ ), regroupés par 10 sur une ligne et dans l'ordre des objets (de 1 à  $n$ )
- $\lceil m/10 \rceil$  lignes suivantes : les capacités des  $m$  contraintes, regroupées par 10 sur une ligne et dans l'ordre (de 1 à  $m$ )
- $m \times n/10$  lignes suivantes : les poids des objets dans les contraintes, contrainte par contrainte (1<sup>er</sup> paquet de  $n/10$  lignes pour les poids des objets de 1 à  $n$  dans la première contrainte, puis idem pour la seconde contrainte, ..., jusque la  $m^{\text{ème}}$  contrainte).

La meilleure solution obtenue par votre algorithme doit être **sauvegardée dans un fichiers texte** dont le format est **scrupuleusement** le suivant :

- 1<sup>ère</sup> ligne : valeur de la solution (fonction objective) et nombre d'objets sélectionnés (séparés par un espace)
- 2<sup>ème</sup> ligne : liste des indices des objets sélectionnés (séparés par un espace).

## 3 Travail attendu - tâches à réaliser

Concevez un algorithme permettant de déterminer une solution (quasi-)optimale du problème du sac à dos multidimensionnel 0-1. Pour atteindre cet objectif, vous développerez des algorithmes pour répondre au points suivants :

- Lecture des données, représentation du problème et d'une solution ;
- Fonction de construction d'une solution initiale faisable (solution initiale gloutonne ou solution initiale aléatoire) ;
- Fonction de vérification de la faisabilité d'une solution ;
- Fonction de recherche locale (LS) (Indice : LS 1-flip et LS échange) ;
- Développement d'une fonction de descente de voisinage variable (VND) utilisant deux procédures de recherche locale.
- Développement d'un algorithme de recherche à voisinage variable (VNS) utilisant la VND précédente et une procédure de perturbation (au choix).

La complexité spatiale des structures de données utilisées et la complexité temporelle des différentes sous-routines de l'algorithme doit être estimée et expliquée. De plus, un ou plusieurs

tableaux de résultats expérimentaux doivent être fournis. Ces tableaux doivent comparer la valeur de la solution initiale, les valeurs des solutions obtenues après les procédures de recherche locale, et les valeurs des solutions obtenues par VND et VNS<sup>1</sup>. Le tableau doit également indiquer le temps CPU nécessaire pour atteindre chaque solution, en fonction des caractéristiques de la machine que vous aurez utilisée pour les tests.

Bonus : vous implémenter un algorithme génétique (GA) et/ou un algorithme hybride GA-VNS.

## 4 Dépôt et modalités

Le code doit être écrit en langage C.

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 de façon rigoureuse 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, fixée pendant le déroulement du module. Vous devrez fournir, par l'intermédiaire de Moodle, l'ensemble des fichiers **sources** permettant de générer l'exécutable correspondant à votre programme, avec à minima un **ReadMe** expliquant comment générer cet exécutable (en précisant notamment les options de compilation à utiliser).

Votre exécutable doit **obligatoirement** pouvoir être utilisé en ligne de commande avec comme paramètres : le nom de l'instance (i.e., le nom du fichier texte contenant les données du problème) et le temps d'exécution autorisé (en secondes). Vous devez donc vous assurer que votre algorithme s'arrête une fois ce temps écoulé.

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**.

La note finale pourra être différente pour des étudiants d'un même groupe, selon le travail fourni par chacun.

Le non respect de tout ou partie des consignes sera pris en compte dans la notation.

---

1. Les meilleures valeurs connues des instances à traiter vous seront fournies sur Moodle pour comparaison.