

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

In this project we consider the knapsack problem with setups which is defined as follows. Let us consider  $m$  families of items. Each family  $j \in \{1, \dots, m\}$  has a setup cost  $f_j$  and a capacity consumption  $b_j$ . In addition, each family  $j$  contains  $n_j$  items. Each item  $i$  of family  $j$  is characterized by a profit  $p_{ij}$  and capacity consumption  $a_{ij}$ . The knapsack has a capacity  $Q$ . The goal is to maximize the net profit (total profit of selected elements minus total set-up cost) of the families and their items selected to be included in the knapsack. More formally, let us define the following binary variables :

- $x_{ij} = 1$  if and only if element  $i$  from family  $j$  is included in the knapsack ;
- $y_j = 1$  if and only if family  $j$  is placed in the knapsack, that is at least one element from family  $j$  is included in knapsack ;

Given these definitions, the knapsack problem with setups (KPS) can be formulated as follows

$$\max \sum_{j=1}^m \sum_{i=1}^{n_j} p_{ij} x_{ij} - \sum_{j=1}^m f_j y_j \quad (1)$$

subject to

$$x_{ij} \leq y_j, \quad j \in \{1, 2, \dots, m\}, i \in \{1, 2, \dots, n_j\} \quad (2)$$

$$\sum_{j=1}^m \sum_{i=1}^{n_j} a_{ij} x_{ij} + \sum_{j=1}^m b_j y_j \leq Q, \quad (3)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad j \in \{1, 2, \dots, m\}, i \in \{1, 2, \dots, n_j\}. \quad (4)$$

The objective function maximizes the total profit that includes the sum of profits of all elements selected minus the sum of setup costs. The set up cost of a family does not depend on the number of elements selected from the family and therefore it is paid only once. Constraints (2) imply that if we select at least one element, the corresponding family is setup. Constraint (3) ensures that the knapsack capacity is not exceeded. The knapsack capacity is consumed by both set-up families and selected items. The last set of constraints describe the decision variables.

### 3 Travail attendu - tâches à réaliser

Design an algorithm which determines (near-)optimal solution of the knapsack problem with setups. To achieve this objective you will develop algorithms in order to follow the steps below :

- Reading the data and representing a solution ;
- Procedure for constructing an initial feasible solution (Hint : greedy initial solution)
- Procedure to verify if a given solution is feasible or not ;
- Local search (LS) procedures (Hint : 1-flip LS and LS swap) ;
- Developing a genetic algorithm that uses above procedures.

The complexity of used data structures and different sub-routines of algorithm need to be estimated and explained. In addition, the table(s) with computational results needs to be provided. The table needs to provide comparison among initial solution value ; solution values after LS procedures ; and solution value obtained by genetic algorithm. Also, the table needs to contain CPU times in seconds needed to reach each solution value.

### 4 Dépôt et modalités

Le code doit être écrit en langage C.

Un rapport (compte-rendu étendu de TP) 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**.