

---

**Travail pratique**

**Affectation de travaux de bachelor**

---

**Objectifs**

L'objectif de ce travail pratique est de modéliser un problème d'affectation de travaux de bachelor et de le résoudre à l'aide de la suite logicielle OR-Tools. Une phase de postoptimisation sera également effectuée afin d'adapter au mieux la solution initialement trouvée suite à des modifications de données.

**Données et paramètres**

Les données du problème sont stockées dans deux fichiers textes :

- ▷ Fichier `projetcst.txt` : ce fichier contient la liste des travaux de bachelor disponibles avec, pour chacun, le professeur encadrant.

La première ligne du fichier contient le nombre de projets suivi du nombre de professeurs encadrants. Les projets et les professeurs sont identifiés par des entiers consécutifs en partant de zéro.

Les lignes suivantes correspondent chacune à un travail de bachelor et contiennent deux entiers, le premier correspondant à l'identifiant du projet et le second à l'identifiant du professeur encadrant.

- ▷ Fichier `preferences.txt` : ce fichier contient les préférences émises par les étudiants et les étudiantes pour les travaux proposés.

La première ligne du fichier contient le nombre d'étudiants et d'étudiantes, identifiés par des entiers consécutifs en partant de zéro.

Les lignes suivantes contiennent trois entiers, le premier est l'identifiant d'un étudiant, le deuxième celui d'un projet et le troisième est un entier compris entre 1 et 5 et correspondant à la préférence de l'étudiant pour le travail de bachelor (1 étant la préférence la plus faible et 5 la plus grande).

En plus des données précédentes un paramètre, unique et valable pour tous les professeurs, définit le nombre maximal de travaux de bachelor que peut encadrer un professeur.

**Partie 1**

La première partie du travail consiste à modéliser le problème de la recherche d'une affectation d'un travail de bachelor à chaque étudiant. Chaque projet ne peut être donné qu'à un seul étudiant et on ne peut attribuer à un étudiant qu'un projet pour lequel il a émis une préférence positive (et qui est donc présent dans le fichier des préférences). L'affectation devra maximiser la somme des préférences des travaux affectés tout en respectant le nombre maximal de projets encadrés par un même professeur.

Vous compléterez le fichier source fourni afin de résoudre votre modèle avec le solveur **SCIP**. Vous considérerez successivement un encadrement maximal de 4, puis 3, puis 2 projets par professeur.

Il est suffisant de modifier le paramètre d'encadrement dans les sources et de relancer la résolution pour chaque valeur. Vous afficherez dans chaque cas la valeur maximale pour la somme des préférences des travaux sélectionnés. Vous créerez aussi un fichier texte contenant la solution optimale (si elle existe). Le format de chaque ligne de ce fichier sera simplement l'identifiant de l'étudiant suivi de celui du projet affecté.

## **Partie 2**

**Pour cette partie et la suivante, l'encadrement maximal est fixé à 3 projets par professeur.**

À peine l'affectation publiée que l'on apprend que les projets 68, 123 et 151 ne seront finalement pas disponibles.

Vous devez modifier votre modèle afin de trouver une nouvelle affectation qui prenne en compte les informations précédentes tout en conservant un nombre maximal d'affectations présentes dans la solution initiale.

Peu importe les préférences émises pour les travaux retenus, il s'agit de retrouver une affectation admissible qui n'utilise pas les projets 68, 123, et 151, et qui maximise le nombre d'affectations déjà présentes dans la solution initiale.

Vous afficherez le nombre d'étudiants qui conservent leur affectation d'origine et créerez un fichier texte avec la solution correspondante.

## **Partie 3**

Maintenant que le nombre maximal d'affectations qui peuvent être conservées est connu, il s'agit de déterminer une affectation admissible qui

- 1) n'utilise pas les projets 68, 123 et 151,
- 2) maintient le nombre maximal d'affectations (trouvé en partie 2) de la solution initiale (trouvée en partie 1),
- 3) maximise la somme des préférences des affectations.

Vous afficherez la valeur maximale pour la somme des préférences et créerez, ici aussi, un fichier texte avec la solution optimale correspondante.

## **Remarques et indications**

Pour la partie 2 :

- ▷ Il ne faut pas modifier les données mais ajouter des contraintes interdisant la sélection des projets indisponibles.
- ▷ Il est possible de changer la fonction objectif tout en conservant les contraintes en appelant la méthode `clear()` de l'objet `MPObjective`.

Pour la partie 3 :

- ▷ Il faut rétablir la fonction objectif de la partie 1 et ajouter une contrainte pour maintenir le nombre maximal d'affectations d'origine.

#### **Modalités et délais**

- ▷ Le travail de programmation est à effectuer par groupe de deux, en Java, version 21 ou 23.
- ▷ L'archive contenant les sources du projet et les jeux de données à étudier, est disponible sur le site Cyberlearn du cours.
- ▷ Vous devez rendre une archive (au format **zip**) contenant le répertoire correspondant à votre groupe qui contiendra votre fichier source complété et commenté ainsi que les fichiers textes contenant les solutions optimales pour les différentes parties. ...
- ▷ Vous devez rendre votre travail sur Cyberlearn au plus tard le **dimanche 25 janvier 2026** (avant minuit).