

Génération d'emplois du temps avec Google OR-Tools

Université de Yaoundé I
Département d'informatique

INF4178

ASSONFACK YEMENE BERLAL 21T2501

Avril 2025

1. Énoncé du problème

L'objectif est de développer un système logiciel capable de générer automatiquement des emplois du temps hebdomadaires pour les classes du Département d'Informatique de l'Université de Yaoundé

I. Les emplois du temps doivent respecter les contraintes pédagogiques et logistiques suivantes :

- Une classe ne peut avoir qu'un seul cours à un instant donné.
- Un cours ne peut être dispensé qu'une seule fois dans la semaine.
- Aucun cours ne doit être affecté à une classe qui ne le suit pas.
- Il faut maximiser le nombre de cours planifiés avant midi.

2. Modèle mathématique

2.1 Ensembles

- C : ensemble des classes
- M : ensemble des matières
- T : ensemble des enseignants
- R : ensemble des salles
- D : ensemble des jours de la semaine ($|D| = 6$)
- P : ensemble des périodes dans une journée ($|P| = 5$)

2.2 Paramètres

- W_p : poids associé à la période $p \in P$, tel que $W_{P1} > W_{P2} > \dots > W_{P5}$
- $\text{mat}_c \subseteq M$: ensemble des matières suivies par la classe c
- ens_m : enseignant principal de la matière m

2.3 Variables de décision

$$x_{m,r,d,p}^c = \begin{cases} 1 & \text{si la classe } c \text{ suit le cours } m \text{ avec son enseignant dans la salle } r \text{ au jour } d \text{ et à la période } p \\ 0 & \text{sinon} \end{cases}$$

2.4 Contraintes

1. **Unicité du cours par semaine pour chaque matière et chaque classe :**

$$\sum_{r \in R} \sum_{d \in D} \sum_{p \in P} x_{m,r,d,p}^c = 1, \quad \forall c \in C, \forall m \in \text{mat}_c$$

2. **Pas de chevauchement de cours pour une classe :**

$$\sum_{m \in \text{mat}_c} \sum_{r \in R} x_{m,r,d,p}^c \leq 1, \quad \forall c \in C, \forall d \in D, \forall p \in P$$

3. **Pas de conflit de salle :**

$$\sum_{c \in C} \sum_{m \in \text{mat}_c} x_{m,r,d,p}^c \leq 1, \quad \forall r \in R, \forall d \in D, \forall p \in P$$

4. **Affectation uniquement si la matière est dans le programme de la classe :**

$$x_{m,r,d,p}^c = 0, \quad \text{si } m \notin \text{mat}_c$$

2.5 Fonction objectif

Maximiser la répartition des cours avant midi :

$$\max \sum_{c \in C} \sum_{m \in \text{mat}_c} \sum_{r \in R} \sum_{d \in D} \sum_{p \in P} W_p \cdot x_{m,r,d,p}^c$$

3. Implémentation avec OR-Tools

L'algorithme est implémenté en Python à l'aide du solveur **CP-SAT** de Google OR-Tools. Les données d'entrée (matières, enseignants, salles) sont chargées depuis les fichiers **subjects.json** et **rooms.json**.

Le fichier Python **timetables.py** :

- Crée les variables binaires pour chaque combinaison valide (cours, salle, jour, période).
- Ajoute les contraintes selon le modèle mathématique.
- Résout le modèle avec maximisation des cours le matin.
- Affiche les emplois du temps générés.

4. Résultat

Le solveur affiche une solution faisable où chaque cours est bien positionné sans conflit de salle ni de classe, en maximisant les cours programmés en matinée.