Modèle Mathématique pour la Génération Automatique d'Emplois du Temps Universitaires

Tchami Tamen Sorelle

Mai 2025

1 Introduction

Ce document présente le modèle mathématique développé pour générer automatiquement un emploi du temps pour le département d'informatique de l'Université de Yaoundé I. Le problème est formulé comme un problème d'optimisation sous contraintes, résolu à l'aide de la programmation par contraintes.

2 Formulation du Problème

2.1 Ensembles et Indices

- $L = \{1, 2, \ldots, |L|\}$: ensemble des classes (niveau-semestre)
- $C = \{1, 2, \ldots, |C|\}$: ensemble des cours
- $R = \{1, 2, \ldots, |R|\}$: ensemble des salles
- $T = \{1, 2, \ldots, |T|\}$: ensemble des enseignants
- $D = \{1, 2, \ldots, 6\}$: ensemble des jours
- $P = \{1, 2, \ldots, 5\}$: ensemble des périodes

2.2 Paramètres

- w_p : poids associé à la période $p \in P$, où $w_1 > w_2 > \ldots > w_5 > 0$ pour favoriser les périodes du matin
- $\text{programme}_{l,c}$: 1 si le cours c fait partie du programme de la classe l, 0 sinon
- $\text{courseTeacher}_{c,t}$: 1 si l'enseignant t peut donner le cours c, 0 sinon
- credit_c : nombre de crédits associés au cours c

2.3 Variables de Décision

$$x_{l,c,r,d,p,t} = \begin{cases} 1, & \text{si la classe } l \text{ suit le cours } c \text{ dans la salle } r \text{ pendant} \\ & \text{la période } p \text{ du jour } d \text{ avec l'enseignant } t \\ 0, & \text{sinon} \end{cases} \tag{1}$$

Pour le modèle relâché, nous introduisons une variable supplémentaire :

$$y_{l,c} = \begin{cases} 1, & \text{si le cours } c \text{ de la classe } l \text{ est programmé quelque part} \\ 0, & \text{sinon} \end{cases} \tag{2}$$

3 Modèle d'Optimisation

3.1 Contraintes

3.1.1 Contrainte 1: Pas de Conflits d'Horaire pour une Classe

Une classe ne peut pas suivre deux cours différents ou être à deux endroits différents au même moment:

$$\sum_{c\in C}\sum_{r\in R}\sum_{t\in T} x_{l,c,r,d,p,t} \leq 1, \quad \forall l \in L, d \in D, p \in P \tag{3}$$

3.1.2 Contrainte 2: Programmation des Cours

Chaque cours du programme doit être programmé exactement une fois:

$$\sum_{r\in R}\sum_{d\in D}\sum_{p\in P}\sum_{t\in T} x_{l,c,r,d,p,t} = 1, \quad \forall l \in L, c \in C : \text{programme}_{l,c} = 1 \tag{4}$$

Dans le modèle relâché, cette contrainte devient:

$$\sum_{r\in R}\sum_{d\in D}\sum_{p\in P}\sum_{t\in T} x_{l,c,r,d,p,t} \leq 1, \quad \forall l \in L, c \in C : \text{programme}_{l,c} = 1 \tag{5}$$

Avec la relation:

$$y_{l,c} = \sum_{r\in R}\sum_{d\in D}\sum_{p\in P}\sum_{t\in T} x_{l,c,r,d,p,t}, \quad \forall l \in L, c \in C : \text{programme}_{l,c} = 1 \tag{6}$$

3.1.3 Contrainte 3: Respect du programme

Cette contrainte est implicitement respectée en ne créant des variables que pour les cours qui font partie du programme de chaque classe.

3.1.4 Contrainte 4: Pas de Conflit d'Enseignant

Un enseignant ne peut pas donner deux cours simultanément:

$$\sum_{l\in L}\sum_{c\in C:\text{courseTeacher}_{c,t}=1}\sum_{r\in R} x_{l,c,r,d,p,t} \leq 1, \quad \forall t \in T, d \in D, p \in P \tag{7}$$

3.1.5 Contrainte 5: Pas de Conflit de Salle

Une salle ne peut pas accueillir deux cours au même moment:

$$\sum_{l \in L} \sum_{c \in C} \sum_{t \in T} x_{l,c,r,d,p,t} \leq 1, \quad \forall r \in R, d \in D, p \in P \tag{8}$$

3.2 Fonction Objectif

3.2.1 Modèle Standard

Maximiser le nombre de cours programmés le matin:

$$\text{Maximiser} \sum_{l \in L} \sum_{c \in C} \sum_{r \in R} \sum_{d \in D} \sum_{p \in P} \sum_{t \in T} w_p \cdot x_{l,c,r,d,p,t} \tag{9}$$

3.2.2 Modèle Relâché

Maximiser le nombre de cours programmés, en priorité ceux avec plus de crédits, et préférer les créneaux du matin:

$$\text{Maximiser} \sum_{l \in L} \sum_{c \in C} (1000 \cdot y_{l,c} + \text{credit}_c \cdot y_{l,c}) + \sum_{l \in L} \sum_{c \in C} \sum_{r \in R} \sum_{d \in D} \sum_{p \in P} \sum_{t \in T} w_p \cdot x_{l,c,r,d,p,t} \tag{10}$$

4 Traitement des Cas Insolubles

Lorsque le nombre de cours à programmer est supérieur au nombre de créneaux disponibles, le problème devient mathématiquement insoluble. Dans ce cas, nous adoptons une approche relâchée qui :

1. Permet de ne pas programmer tous les cours (contrainte ≤ 1 au lieu de $= 1$)
2. Maximise le nombre de cours programmés
3. Priorise les cours avec plus de crédits
4. Garde la préférence pour les créneaux du matin

5 Implémentation

Le modèle mathématique est implémenté à l'aide de la bibliothèque Google OR-Tools, qui utilise un solveur CP-SAT (Constraint Programming - SATisfiability) pour résoudre efficacement des problèmes d'optimisation combinatoire. L'implémentation comprend:

- lien de mon github

1. Extraction des données à partir de fichiers JSON
2. Construction du modèle de programmation par contraintes
3. Résolution du modèle avec une limite de temps
4. Traitement et affichage des résultats
5. Analyse du problème en cas d'insolvabilité

6 Conclusion

Le modèle mathématique présenté permet de générer un emploi du temps optimisé qui respecte toutes les contraintes imposées. En cas d'impossibilité mathématique, une approche relâchée fournit une solution partielle qui maximise le nombre de cours programmés en respectant les priorités établies.