

TER 2019 - Spécifications

Maxime Gonthier - Benjamin Guillot - Laureline Martin

13 mars 2019

Table des matières

1 Les données

- 1.1 Module Cours
- 1.2 Module Salle de classe
- 1.3 Module Etudiant
- 1.4 Module Professeur
- 1.5 Conclusion des liens entre les modules

2 Contraintes

- 2.1 Contraintes dures
- 2.2 Contraintes faibles
- 2.3 Module Contraintes

3 Affectation

4 Evaluation de chaque affectation

5 Début de reflexion sur l'implémentation des transports

1 Les données

On va répartir les données concernant la fac en 4 modules.

- Module cours
- Module salle de classe
- Module étudiant
- Module professeur

1.1 Module Cours

Le module cours est une classe représentant un cours sur un temps donné. Lors des affectations, on modifiera pour chaque instance a deplacer l'horaire de début.

Elle contient :

1. Durée : un entier -représente la durée du cours en minutes-
2. Une liste des étudiants : un tableau d'entier qui contient les numero des étudiants participant à ce cours.
3. Le type de salle utilisé : un entier, 0 pour une salle de TP et 1 pour un autre salle (on pourra ajouter d'autres type de salle si necessaire).
4. Un indice de flexibilité calculé en fonction de celle des etudiants : somme des flexibilité de chaque étudiants (cette modelisation a pour but de favoriser les cours ayant le plus d'etudiants).
5. Le nombres d'étudiant : un entier
6. Un numéro de salle : un entier (identifiant unique représentant la salle)
7. Le numéro du professeur donnant ce cours : un entier (identifiant unique représentant le professeur)
8. L'horaire de début : un entier pour l'heure, un entier pour la minute (structure implémentée specialement pour le projet)
9. L'horaire de fin : un entier pour l'heure, un entier pour la minute.

1.2 Module Salle de classe

Le module salle représente la localisation du cours sur un temps donné.

1. Numéro de salle : un entier (identifiant unique)
2. Une localisation : un entier -0 pour proche de l'arrêt, 1 pour modérément éloigné, 2 pour éloigné-.
3. Un type de salle : un entier, 0 pour une salle de TP et 1 pour une autre salle (on pourra ajouter d'autres type de salle si necessaire).
4. Une capacité maximale : un entier

1.3 Module Etudiant

Le module étudiant représente un élève sur le temps d'exécution du programme.

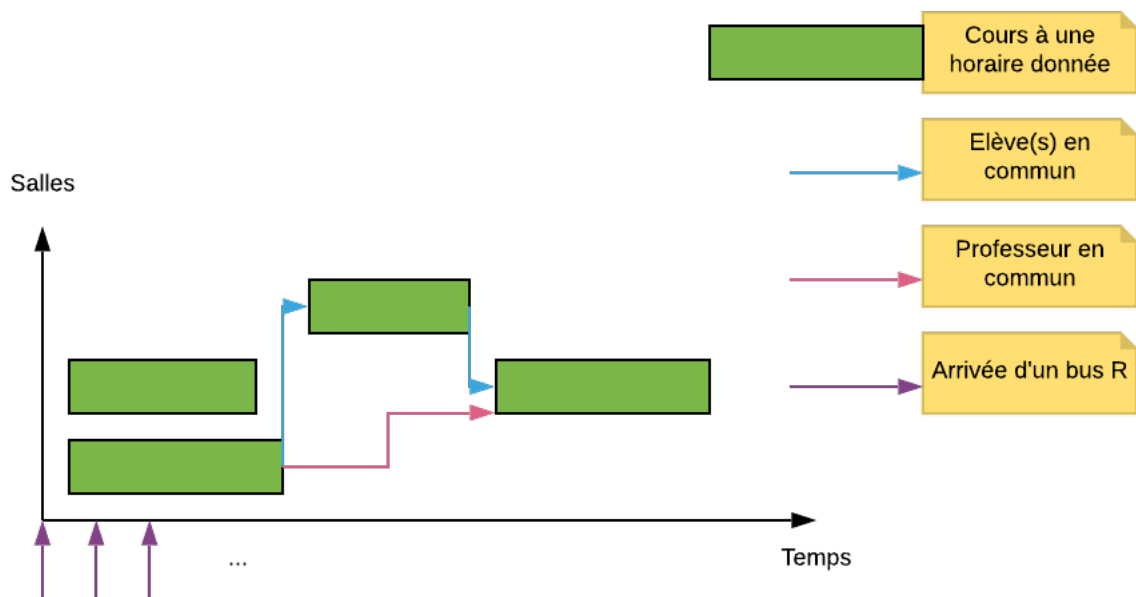
1. Distance entre son domicile et l'université : un entier, 0 si l'étudiant habite a moins de 15 min de la fac, 1 si il habite entre 15 et 45 min de la fac, 2 sinon.
2. Flexibilité : un entier calculé en fonction de la distance de trajet et des contraintes forte et faible.

1.4 Module Professeur

Le module professeur représente un professeur sur le temps d'exécution du programme.

1. Un numéro de professeur : un entier (identifiant unique)
2. plage de disponibilité début : un entier pour l'heure, un entier pour la minute.
3. plage de disponibilité fin : un entier pour l'heure, un entier pour la minute.

1.5 Conclusion des liens entre les modules



Modélisation.

2 Contraintes

Pour optimiser, nous faisons face à plusieurs contraintes, toutes ne sont pas de même "importances". Nous allons donc devoir définir un ordre de priorité sur les contraintes, ainsi lors de l'optimisation par notre algorithme, nous pourrons ajuster et obtenir de meilleurs résultats même si certaines contraintes "faibles" sont violées.

2.1 Contraintes dures

1. 0 : Salle utilisée par deux cours différents pour des horaires qui se chevauchent
2. 1 : Un élève qui suit deux cours dont les horaires se chevauchent
3. 2 : Avoir une personne à charge ce qui impose un horaire le matin et/ou le soir. Exemple : Sois X l'heure de début d'un cours, si un enfant doit être déposé à l'école à 9h on a : $X > 9 + (\text{indice de distance de cet étudiant}) * 30 \text{ min}$ Sois Y l'heure de fin d'un cours, si un enfant doit être récupéré à l'école à 17h on a : $Y < 17 - (\text{indice de distance de cet étudiant}) * 30 \text{ min}$

2.2 Contraintes faibles

1. 3 : Avoir un travail, cela impose la même chose que la contraintes précédentes

2.3 Module Contraintes

- identifiant : un entier non unique correspondant à une des contrainte précédement. L'identifiant représente une hierarchie dans les contraintes, de la plus forte a la plus faible
- un tableau d'entier contenant les identifiant des étudiants sujet à cette contrainte.

3 Affectation

On va affecter chaque cours à un horaires sans prendre en compte les transports. On ne va utiliser que les contraintes énoncés précédemment.

On va utiliser une heuristique tabou. On commence avec un emploi du temps par défaut. Le voisinage correspond à la modification d'un cours. A chaque itération on recalcule le nombre de contraintes violées. Un optimum local sera une solution viable, c'est à dire qui ne viole pas les deux contraintes les plus fortes.

4 Evaluation de chaque affectation

On évalue ensuite chaque optimum local trouvé afin de définir lequel viole le moins de contraintes. Elle correspond à la somme des contraintes violées. Les contraintes sont hiérarchisées comme vu précédemment, ainsi lors de la somme des contraintes on pondère en fonction de cette hiérarchisation.

5 Début de réflexion sur l'implémentation des transports

Pour les transports, on supposera que chaque élève arrivera par le bus précédant le début du premier cours de sa journée. De cette manière à chaque itération de l'affectation des cours, on pourra recalculer la congestion de chaque bus (qui sera utilisé comme fonction objectif du projet une fois les transports implémentés). La congestion peut être représentée par le pourcentage de remplissage du bus à chaque arrêt.