

TER 2019 - Spécifications

Maxime Gonthier - Benjamin Guillot - Laureline Martin

9 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

2 Contraintes

- 2.1 Contraintes dures
- 2.2 Contraintes faibles
- 2.3 Sélections par heuristiques

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. C'est elle que l'on va déplacer lors de l'affectation. Elle contient :

1. Durée : un entier
2. Une liste des étudiants assisants à ce cours :

un tableau? un char?*

3. Le type de salle utilisé : un entier, 0 pour une salle de TP et 1 pour un autre salle

4.

Un indice de flexibilité?

5. Le nombre d'étudiant : un entier
6. Un numéro de salle : un entier
7. Le numéro du professeur donnant ce cours : un entier
8. L'horaire de début : un float
9. L'horaire de fin : un float

1.2 Module salle de classe

DESCRIPTIONSALLEDECLASSE

1. Numéro de salle : un entier
2. Une localisation :

*SOIS UN ENTIER SOIS UN CHAR * JSP*

3. Un type de salle, c'est à dire une salle de TP, ou une salle classique : un entier, 0 pour une salle de TP et 1 pour un autre salle
4. Une capacité maximale : un entier

1.3 Module etudiant

DESCRIPTION ETUDIANT

1. Temps de trajet : un entier
2. 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
- 3.

Flexibilit?

1.4 Module professeur

DESCRIPTION PROFESSEUR

1. Un numéro de professeur : un entier
2. Une plage de disponibilité :

JSP

Concluresurlesliensentrechaqueclasses

2 Contraintes

Pour optimiser, nous faisons face à plusieurs contraintes, toutes ne sont pas de même "importance". 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. 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. Avoir un travail, cela impose la même chose que la contraintes précédentes
3. Salle utilisée par deux cours différents pour des horaires qui se chevauchent
4. Un élève qui suit deux cours dont les horaires se chevauchent

2.2 Contraintes faibles

1.

Jspj'aitoutmisencontraintesdurs

Concluresurlescontraintesetleursfutureutilisation

2.3 Sélections par heuristiques

Les différentes heuristiques que nous allons utiliser pour jouer sur les contraintes.

surementtabou

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.

4 Evaluation de chaque affectation

blabla

5 Début de reflexion sur l'implémentation des transports

blabla