

Outil d'appariement pour l'attribution des projets Androide

Perotti-Valle Rayan Savarit Felix Thomasson Malo

February 28, 2024

Contents

1	Introduction	3
2	Objectifs	3
2.1	Objectif 1	3
2.2	Objectif 2	4
3	Description des besoins	4
4	Contraintes	4
5	Délais	5

1 Introduction

Ce projet a pour but de récupérer les préférences des étudiants pour leur choix des projets Androïde dans le cadre de leur UE de projet. Dans un premier temps nous recueillirons tous les choix des étudiants et professeurs à l'aide d'une interface web liée à une base de données. Cela nous donnera une liste. Dans un second temps nous utiliserons un programme linéaire sur les listes de préférences des étudiants et professeurs pour maximiser le nombre d'étudiants affectés à un projet tout en minimisant la somme des **rangs** des étudiants. On appelle le **rang** d'un étudiant la position de son projet dans sa liste de préférences.

2 Objectifs

Pour ce projet nous avons plusieurs objectifs. Dans un premier temps, la mise en place d'une interface web liée à une base de données servant à recueillir tous les choix des étudiants et professeurs. Cela nous donnera un ensemble d'informations. Dans un second temps, l'élaboration d'un programme linéaire sur les listes de préférences des étudiants et professeurs pour minimiser la somme des rangs des étudiants afin de maximiser le nombre d'étudiants affectés à un projet. On appelle le rang d'un étudiant la position de son projet dans sa liste de préférences. Pour cela, nous allons suivre le programme suivant:

2.1 Objectif 1

Objectif :

Variables :

$$x_{ij}$$

$$x_{Aj}$$

- j projet
- A groupe
- i étudiant
- p nombre d'étudiants

Minimiser la somme de tous les rangs :

$$\text{Minimiser } \sum_A \sum_j x_{Aj} \cdot \text{rang}(A, j)$$

Contraintes :

1. Chaque groupe d'étudiants est affecté à au plus un projet :

$$\forall i \sum_j x_{ij} \leq 1,$$

$$\forall j \sum_i x_{ij} \leq 1,$$

2. La capacité de chaque projet ne peut être dépassée :

$$\sum_{i=1}^n x_{ij} \leq \text{capacité}(j), \quad \forall j$$

3. Chaque étudiant doit être affecté à un projet :

$$\sum_i \sum_j x_{ij} \geq p$$

4. un étudiant ne doit pas avoir plusieurs projet :

$$\sum_A \sum_j x_{Aj} = x_{ij}$$

5. Les variables de décision sont binaires :

$$x_{ij} \in \{0, 1\}$$

$$x_{Aj} \in \{0, 1\}$$

2.2 Objectif 2

3 Description des besoins

Nous devons donc approfondir nos connaissances en interface dans les langages web afin de créer une interface simple et efficace à utiliser, nous devons aussi en connaître plus sur les problèmes d'affectations comme Hôpital-Résident afin de faire le programme linéaire le plus optimal. Il nous faut donc de bonne connaissances de plusieurs langages de programmation, en optimisation d'interface et l'algorithme d'affectation couplage stable et aussi des bonne connaissance des outils de programme linéaire.

4 Contraintes

Nous devons faire face à un problème complexe avec plusieurs acteurs dans ce projet. Il est donc nécessaire de respecter certaines contraintes :

- Affecter le bon nombre étudiants à un projet pour ainsi respecter sa capacité
- Un étudiant peut postuler à différent projets avec différents groupe
- Chaque étudiant doit postuler à un projet avec seul groupe

- Un étudiant sera assigné à un seul et unique groupe pour un seul et unique projet
- Enfin il faudra dans un second temps gérer le mixage de plusieurs groupe

5 Délais

Semaine du 20 Mai