Rendu du TME 1-3
Intelligence Artificielle et Jeux
Gale-Shapley - LU3IN025

Nina Luc
21203220
Alejandra MORALES SAUCEDO
21214631
Enseignant: Nawal Benabbou

10 février 2025

1 Introduction

Ce projet porte sur l'affectation d'étudiants dans les différents parcours du master informatique de Sorbonne Université.
L'affectation dans les parcours dépend du dossier et voeux des étudiants.
Afin de trouver la solution optimale et stable pour effectuer l'affectation, nous avons utilisé l'algorithme de **Gale-Shapley** ainsi que les deux fichiers contenant le classement des parcours et des étudiants *PrefSpe.txt* et *PrefEtu.txt*

2 Description des structures et du code

2.1 Partie 1 : Affectation

La première partie du projet se concentre sur l'affectation des étudiants dans les différents parcours en prenant en compte les préférences de chaque.

On commence par faire la lecture des deux fichiers de données grâce à nos fonctions *lecture_preferences_etu* et *lecture_preferences_spe*, afin de les utiliser dans notre implémentation de l'algorithme de **Gale-Shapley**
Nous avons codé deux fonctions *GS_etudiants_nouv* et *GS_parcours_nouv* disponibles dans le fichier *fonctions.py* afin d'implémenter l'algorithme.

Afin d'optimiser notre code, nous avons décidé d'implémenter des dictionnaires pour donner la liste des marriages et d'utiliser des tas pour avancer dans l'affectation des étudiants grâce à la bibliothèque python *heapq*.

Après l'exécution du code, nous avons obtenu les résultats suivants :

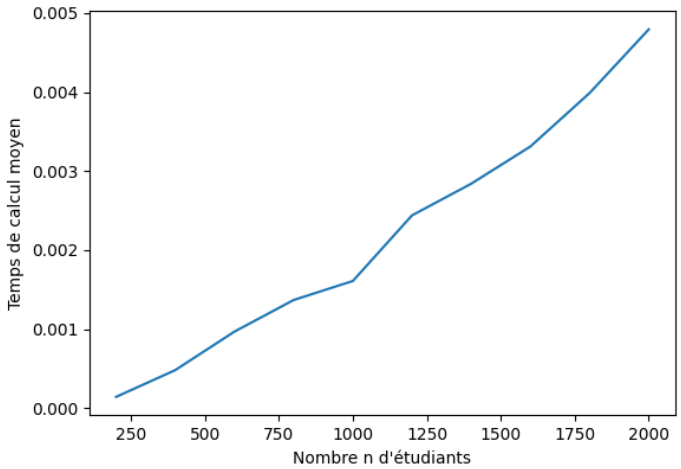
GS Etudiants : 6: [1], 2: [9], 5: [0], 7: [7], 4: [10], 0: [5, 3], 1: [4], 8: [6, 2], 3: [8]
GS Masters : 1: 5, 0: 6, 7: 7, 3: 0, 4: 1, 10: 4, 9: 2, 5: 0, 6: 8, 2: 8, 8: 3

Après avoir testé nos résultats pour voir si nous avions des paires instables nous avons obtenu des listes vides, donc nous avons pas de paires instables.
Cela nous permet de vérifier que notre algorithme marche correctement.

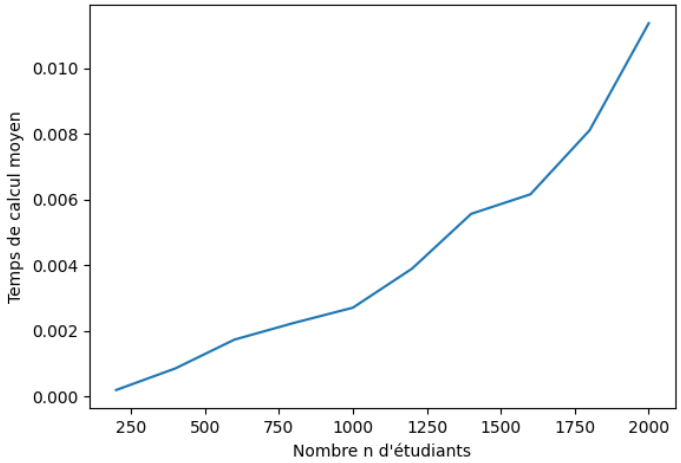
2.2 Partie 2 : Evolution du temps de calcul

Pour évaluer le temps de calcul de nos fonctions, nous avons d'abord généré des matrices de préférences des étudiants et des 9 parcours avec des préférences aléatoires. Les fonctions sont disponibles dans le fichier *matrices_aleatoires.py*.

Voici la courbe pour la fonction **GS_etudiants_nouv**:

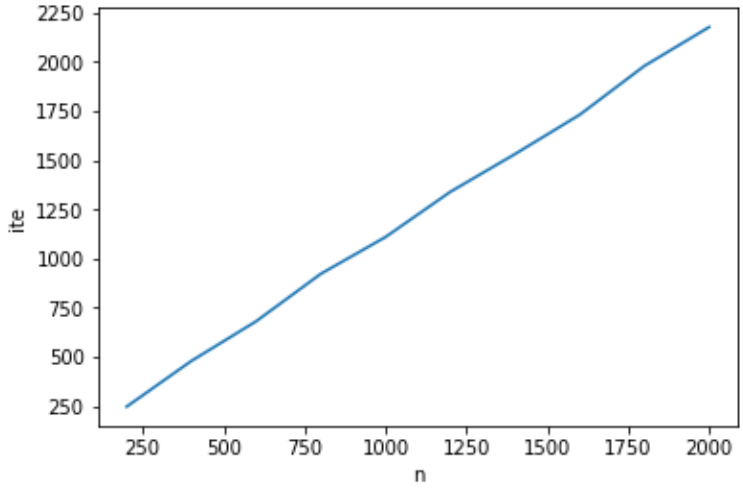


Et voici celle de la fonction **GS_parcours_nouv**:



On observe que le temps de calcul est **exponentiel** ce qui correspond bien à la complexité théorique démontrée en cours.

Finalement, voici la courbe du nombre d'itérations de nos algorithmes :



Le nombre d'itérations est en moyenne aux alentours de n.
Pourtant, la complexité théorique est de $O(n^2)$.

2.3 Partie 3 : Equité et PL(NE)

Dans cette partie, nous allons essayer de trouver la solution optimale en utilisant la programmation linéaire à la place de l'algorithme de Gale Shapley.
Nous utiliserons l'outil Gurobi pour nous aider à résoudre les programmes linéaires.

3 Réponses aux questions

Question 11
Pour garantir qu'un étudiant i a un de ses k premier choix, il faut qu'il soit associé à un master dont la position dans sa liste de préférence soir inférieure à m-k.
On a donc le PL suivant :

variables

$x_{ij} = 1$ "si l'étudiant i est affecté au master j"

$= 0$ sinon

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

c_j = capacité du master j

P_{ij} = position de j dans la liste de préférence de i.

K = nombre max de choix considéré

Contraintes

$\sum_{j=0}^{m} x_{ij} = 1 \quad \forall i \in \{0, \dots, n\}$ chaque etu à un master

$\sum_{i=0}^{n} x_{ij} \leq c_j \quad \forall j \in \{0, \dots, m\}$

si $P_{ij} > K_i \quad x_{ij} = 0$

Fonction objectif = $\sum_{i}^{max} \sum_{j} P_{ij} \times x_{ij}$ (meilleure affectation)

Question 12

Avec Gurobi, on obtient une somme maximisée égale à 8. Cela veut dire que uniquement 8 étudiants parmis les 11 ont un master, ce n'est donc pas une solution, et comme c'est la solution maximisée de notre programme linéaire, on peut en conclure qu'il n'existe pas de solution pour k = 3.

```
1   # Solution for model obj
2   # Objective value = 8
```

Question 13

Toujours à l'aide de Gurobi, nous avons trouvé que le plus petit k tel qu'il existe une solution est k = 5.

Voici les résultats :

Étudiant 0 affecté au parcours 7
Étudiant 1 affecté au parcours 6
Étudiant 2 affecté au parcours 4
Étudiant 3 affecté au parcours 8
Étudiant 4 affecté au parcours 1
Étudiant 5 affecté au parcours 2
Étudiant 6 affecté au parcours 5
Étudiant 7 affecté au parcours 0
Étudiant 8 affecté au parcours 8
Étudiant 9 affecté au parcours 3
Étudiant 10 affecté au parcours 0

Question 14

Nous avons créé les fichiers PNLE14.ld et PNLE15.ld pour tester nos nouveaux PL associés à ces différents problèmes. Au moment de devoir les tester, nous avons eu un problème de connexion avec la PPTI qui nous a empêché de tester ces PL à l'aide de Gurobi. Nous n'avons donc pas pu trouver les résultats avant la remise, mais comptons les calculer avant la soutenance. Nous nous excuser d'avance pour ce désagrément et espérons que vous pourrez comprendre.

variables

- $x_{ij} = \begin{cases} 1 \text{ si l'étudiant } e_i \text{ est avec le master } m_j \\ 0 \text{ sinon} \end{cases} \quad \forall i \in [0,n] \text{ et } j \in [0,m]$
- C_j = la capacité du master m_j
- P_{ij} = est la somme de la position de m_j dans la liste de préférences de e_i et de la position de e_i dans la liste de préférences de m_j

Fonction objectif :

$$\max \sum_{i=0}^{n} \sum_{j=m}^{m} x_{ij} P_{ij}$$

Contraintes :

- $\sum_{j=0}^{nm} x_{ij} = 1 \quad \forall i \in [0,\dots,n]$ chaque étudiant a un master
- $\sum_{i=0}^{m} x_{ij} \leq c_{ij} \quad \forall j \in [0,\dots,m]$

Question 15

Variables

- $OC_{ij} = \begin{cases} 1 \\ 0 \end{cases}$ si l'étudiant e_i est avec le master m_j $\quad \forall i \in [0, n]$ $\forall j \in [0, m]$
- C_j = la capacité du master m_j
- Pe_{ij} est la position de m_j dans la liste de préférences de e_i
- Pm_{ij} est la position de e_i dans la liste de préférences de m_j
- K^* est le nombre maximum de position dans les préférences de e considéré

Fonction objectif

$$\text{Max} \sum_{i}^{n} \sum_{j}^{m} x_{ij} \, (Pe_{ij} + Pm_{ij})$$

Contraintes

- $\sum_{j=0}^{m} x_{ij} = 1 \quad \forall i \in [0, \ldots, n]$ chaque étudiant a un master
- $\sum_{i=0}^{n} x_{ij} \leq e_{ij} \quad \forall j \in [0, \ldots, m]$
- Si $Pe_{ij} > K^*$, $x_{ij} = 0$

4 Description des jeux d'essais utilisés

Les jeux de tests sont en commentaire en bas de chaque fichier .py avec une description.