



UNIVERSITÉ
DE MONTPELLIER

Rapport du projet Conception et implantation d'un système d'aide à la décision

El Houiti Chakib
Kezzoul Massili

30 novembre 2021

Introduction

Objectif du projet

L'objectif principal du projet est de réaliser une analyse critique de l'algorithme du mariage stable. Dans un premier temps l'objectif est d'implémenter l'algorithme de Gale et Shapley (mariage stable), pour l'affectation des étudiants aux instituts, ensuite, de proposer une méthode de satisfaction pour les deux côtés et de tester cet algorithme sur plusieurs jeux de données. Finalement, c'est de proposer une représentation compacte des préférences.

Environnement de développement

Le projet a été développé sur notre propre environnement de travail. On a utilisé le langage Python, pour l'implémentation des différentes fonctionnalités, en utilisant plusieurs bibliothèques propres à Python.

Structure du projet

Pour une meilleure compréhension de l'environnement du projet, voici ci-dessous différentes informations sur les différents fichiers et répertoires du projet :

src/ Répertoire contenant les fichiers sources du projet.

generate.py Fichier pour la génération automatique de jeux de données.

algorithmes.py Fichier implémentant les différents algorithmes, notamment celui de Gale et Shapley.

graphviz.py Fichier de visualisation.

main.py Fichier contenant le programme principale du projet.

data/ Répertoire contenant les différents jeux de données de différentes tailles.

output/ Répertoire contenant toutes les sorties du programme.

README.md Fichier expliquant la manière d'utiliser le programme (initialisation, compilation et exécution). Reférez-vous à la section *Utilisation* de ce dernier pour plus d'informations.

Makefile Fichier qui spécifient les commandes de compilation, initialisation et autres.

1 Modélisation et implémentation

1.1 Programme de génération de préférences aléatoires

La première partie du travail est la génération de préférences aléatoires pour les étudiants ainsi que les institutions. L'objectif ici est de générer un fichier contenant ces préférences qui pourra ensuite être interpréter par un programme. Nous avons choisis de représenter les préférences par un fichier *JSON*¹. En effet, ce format est très expressif et facile à manipuler.

1. JavaScript Object Notation est un format de données textuelles dérivé de la notation des objets du langage JavaScript. il permet de représenter de l'information structurée comme le permet XML par exemple.

```

0  {
1    "students": {
2      "E1": ["I3", "I1", "I2"],
3      "E2": ["I1", "I2", "I3"],
4      "E3": ["I3", "I2", "I1"]
5    },
6    "institutions": {
7      "I1": {
8        "capacities": 1,
9        "preferences": ["E1", "E2", "E3"]
10     },
11     "I2": {
12       "capacities": 1,
13       "preferences": ["E1", "E3", "E2"]
14     },
15     "I3": {
16       "capacities": 1,
17       "preferences": ["E3", "E2", "E1"]
18     }
19   }
20 }

```

Listing 1 – "Exemple d'un fichier de préférences"

Cette Partie est faite pour la génération aléatoires des préférences des étudiants et des instituts. Pour cela, on a réaliser une méthode prenant n étudiants et k instituts, qui permet de générer n étudiant avec k préférences d'instituts pour chaque étudiant, qui seront stockés dans un fichier *json*, pour avoir la possibilité de générer plusieurs jeux de données. Pour les institus, on génère k instituts avec n préférences pour chaque institut et une capacité q . la capacité de chaque instituts est aussi générer aléatoirement, en respectant, la propriété que la somme des capacités de tout les instituts soit égale au nombre n d'étudiants.

1.2 Implémentation de l'algorithme du mariage stable

1.3 Interface de visualisation

Pour pouvoir visualiser, les affectations des étudiants aux instituts, on a pensé à une structure de graphes, avec des noeuds et des arêtes, pour mieux visualiser tout cela. Chaque institut et chaque étudiants sont représenté par un noeud, étiqueté par le numéro de ces derniers (ex : E1 ou I1). Chaque affectation est représenté par une arête, donc une arête entre un étudiant et son affectation d'instituts.

Cette visualisation est une petite application web, faite avec la bibliothèque *Dash* disponible sous python, qui permet de projeter des graphes avec un serveur web.

Sachant que nos résultats des affectations sont exportés dans des fichiers sous fromat *JSON* ou *CSV*, donc les résultats restent toujours, consultables via fichiers.

1.4 Méthodes de satisfaction

La satisfaction de chaque côté est nécessaire, pour évaluer nos algorithmes et les classés. Les problèmes de satisfaction apparaissent toujours dans les systèmes d'aid à la décison ou plus précisément dans les

FIGURE 1 – Visualisation via app web

Dash Cytoscape:

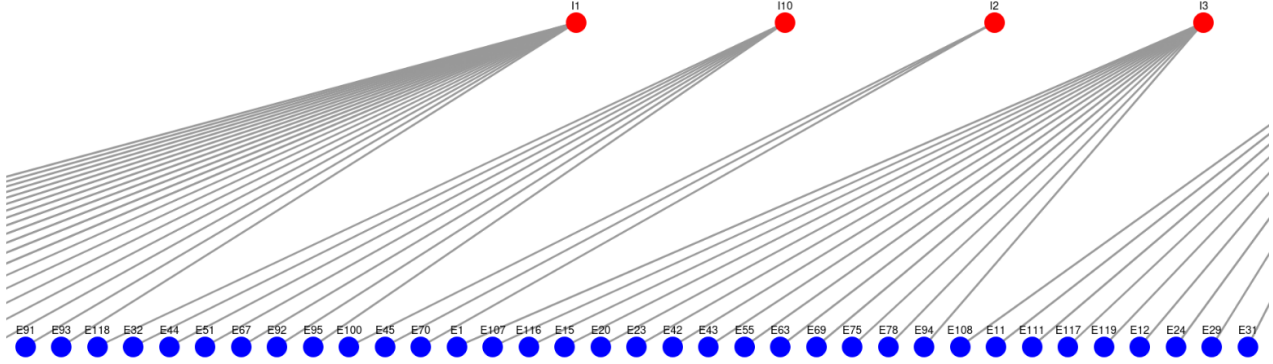


FIGURE 2 – Visualisation via fichier JSON

```
{
  "I1": [
    "E1"
  ],
  "I2": [
    "E2"
  ],
  "I3": [
    "E3"
  ]
}
```

FIGURE 3 – Visualisation via fichier CSV

```
E1;I1
E2;I2
E3;I3
```

systèmes d'affectations. Dans notre projet, on s'est concentré sur la satisfaction des étudiants, qui est un problème très fréquent dans la vie réelle.

Satisfaction des étudiants

Il existe plusieurs manières pour calculer la satisfaction des étudiants, on a choisi des méthodes qui sont significatives. Ces méthodes ont de même des points forts et des points faibles. Toutes les méthodes sont faites, d'une façon à donner une note à chaque étudiant, selon son affectation par rapport à sa liste de préférences. Une moyenne de ces notes permet de mesurer la satisfaction globale de tout les étudiants. La note de chaque étudiant est comprise entre 0 et 1, c.à.d si un étudiant a eu son premier choix, il aura une note de 1 et s'il a eu son dernier choix, il aura une satisfaction égale à 0. Les notes des autres choix sont calculés avec des fonction mathématiques du genre $y = f(i)$, tel que, y est la note de satisfaction et i est la position de l'affectation de l'étudiant dans sa liste de préférences.

Linéaire En premier, une fonction linéaire, une façon très simple de calculer les satisfactions. Son principe est de donner une note à un choix i d'un étudiant qui est inférieure à la note du choix $i-1$. Cette note diminue de 1 vers 0 d'une façon constante (comme le montre le graph ci-dessous) et cela selon le nombre de choix k des étudiants. Donc, par exemple si $k = 10$, on aura une liste de valeurs = (1, 0.9,

0.8, 0.7, ..., 0). Cette méthode de calculer est efficace pour un petit jeu de données, mais si on a un grand jeu de données, par exemple 100000 étudiant et 100 instituts, on a constaté que la satisfaction globale approche de 1, car les cinq premiers choix auront des notes élevées égales à (1, 0.99, 0.98, 0.97, 0.96), alors que dans un jeu de données pareil, 90% des étudiants ont eu un de leurs trois premiers choix et que le pire étudiant a eu son 11ème choix avec une note de 0.9, tout cela affecte la moyenne globale des satisfactions et du fait quelle soit proche de 1. Aussi, vu que notre fonction diminue d'une façon linéaire, la différence entre la note du 1^{er} et du 2^{ème} choix est égale à la différence entre la note du 66^{ème} et du 67^{ème} choix. Ce qui nous a mener à penser à d'autres fonctions.

Polynomiale Une fonction qui diminue d'une façon Polynomiale, c.à.d, la différence entre chaque deux choix décroît en allant du premier au dernier choix. ce qui limite le problème de la fonction linéaire. Par exemple, la différence entre la note du 1^{er} et du 2^{ème} choix est supérieure à la différence entre la note du 66^{ème} et du 67^{ème} choix. Ce qui donne une forme d'importance à la différence entre les premiers choix, contrairement aux derniers choix, Car dans un cas pratique, un étudiant qui aura son 50^{ème} choix ou son 60^{ème} choix, ça reste presque la même chose, du côté de ça satisfaction, par contre, un étudiant, qui aura son 1^{er} choix ou son 5^{ème} choix, affecte plus ça satisfaction. Le graph ci-dessous montre la décroissance non linéaire entre chaque deux choix.

Inverse Une fonction Inverse du genre $f(i) = 1/i$, Cette fonction permet de donner, une plus grande importance entre les premiers choix, dans ce cas là le 2^{ème} choix aura une note de 0.5, qui est une différence énorme entre le 1^{er} et le 2^{ème} choix. on a pensé a cette fonction pour donner une satisfaction globale qui accorde plus d'importance au 1^{er} choix, Donc plus la moyenne est élevée, plus on saura qu'un grand nombre d'étudiants ont eu leurs 1^{er} choix. Le graph ci-dessous montre la décroissance de la fonction inverse.

Satisfaction des instituts

Contrairement à la satisfaction des étudiants, celle des instituts doit respecter une contrainte supplémentaire, qui est du au faite qu'on affecte à chaque institut plusieurs étudiants selon sa capacité Q . Pour palier à cette contrainte et avoir une valuer de satisfaction significative, Nous attribuons une note maximale de 1 pour un institut donné, si et seulement si les étudiants affectés à cet institut sont les Q **premiers** choix de ce dernier. inversement, une note de 0 est attribué si et seulement si les étudiants affectés à cet institut sont les Q **derniers** choix de ses préférences. Toutes les affectations qui sont au milieu suivent une fonction qui diminue linéairement. Une moyenne est calculé pour avoir une satisfaction globale de tout les instituts.

2 Extension du système

2.1 Random assignement

2.2 Les sans instituts

3 Conclusion

3.1 Utilisation du programme

3.2 Perspectives

3.2.1 Pré-matching-Problem and Theroem de HALL