HAI902I - Aide à la décision

Conception et implantation d'un système d'aide à la décision

Réalisé par :
Wided MEFLAH

Encadré par:
Dr.Souhila KACI

26 novembre 2023

Table des matières

1 Introduction :

L'algorithme du mariage stable a émergé comme une solution remarquablement efficace pour résoudre des problèmes complexes d'appariement dans divers domaines. L'idée principale derrière l'algorithme est de permettre à chaque personne de classer les membres de l'autre groupe par ordre de préférence. Ensuite, en utilisant ces listes de préférences, l'algorithme va tenter de former des couples où aucune paire d'individus de groupe opposé ne préférerait être avec une autre personne.

Dans ce projet, nous implémentons l'algorithme du mariage stable pour le problème d'assignation des étudiants aux établissements. Les objectifs de ce projet peuvent être résumés comme suit :

- Implémenter l'algorithme du mariage stable avec des préférences générées aléatoirement.
- Proposer une méthode pour mesurer la satisfaction des étudiants ainsi que celle des établissements.
- Tester le système avec plusieurs jeux de données.
- Proposer des extensions au système pour intégrer les représentations compactes des préférences.

2 Mode d'emploi du projet :

Le projet est contenu dans une archive nommée **"ProjetMeflahWided.zip"**. Pour le faire fonctionner, commencez par décompresser l'archive. Avant de poursuivre, assurez-vous d'avoir Python 3 installé, idéalement dans une version égale ou supérieure à la version 3.10. Ensuite, installez le package "eel" en utilisant la commande **pip install eel**. Une fois l'archive décompressée, lancez le fichier aide.py, puis ouvrez votre navigateur web local et accédez à l'adresse **localhost :5000**.

3 Conception :

3.1 Algorithme du mariage stable :

Le système prend comme entrée des listes de préférences générées aléatoirement pour chaque étudiant concernant les établissements, ainsi que des listes de préférences générées aléatoirement pour chaque établissement concernant les étudiants.

Le système prend en compte deux versions : l'une qui priorise les étudiants, dans ce cas on commence par affecter des établissements aux étudiants ; l'autre version priorise les établissements en faisant l'inverse.

Ci-dessous, nous présentons la version de l'algorithme qui donne la priorité aux étudiants.

Algorithm 1 Fonction MariageStable_v1

```
1: function MariageStable_v1(liste_etudiants, liste_etablissements)
2:     N ← nombre d'étudiants ou d'établissements dans les listes
3:     etudiant_propositions ← tableau de N booléens initialisés à faux
4:     affectation_etablissement ← tableau de N éléments initialisés à NULL
5:     etudiants_non_affectes ← liste_etudiants
6:     while il reste des étudiants non affectés do
7:         for chaque étudiant dans etudiants_non_affectes do
8:             if l'étudiant n'a pas fait de proposition then
9:                 etablissement ← obtenir l'établissement le plus haut dans la liste de
   préférences de l'étudiant auquel il n'a pas encore postulé
10:                etudiant_propositions[tudiant] ← vrai
11:                if l'établissement a encore de la place then
12:                    affectation_etablissement[tudiant] ← etablissement
13:                    retirer étudiant de etudiants_non_affectes
14:                else if l'établissement préfère l'étudiant à l'un de ses affectés actuels
   then
15:                    ancien_affecte ← affectation_etablissement[tudiant]
16:                    affectation_etablissement[tudiant] ← etablissement
17:                    etudiant_propositions[ancien_affecte] ← faux
18:                    retirer étudiant de etudiants_non_affectes
19:                    ajouter ancien_affecte à etudiants_non_affectes
20:                end if
21:            end if
22:        end for
23:    end while
24:    Retourner affectation_etablissement
25: end function
```

3.2 Mesure de la satisfaction :

Après l'assignation des étudiants aux établissements et vice versa par le processus de mariage stable, il est nécessaire de mesurer la satisfaction de chaque individu concernant

son affectation. Ensuite, il convient de mesurer la satisfaction moyenne pour l'ensemble des étudiants et pour l'ensemble des établissements, pour chacune des deux versions de l'algorithme.

3.2.1 Méthode proposée :

La méthode proposée pour mesurer la satisfaction de chaque individu i consiste à calculer le rang de son affectation dans sa liste de préférences. L'ordre décroissant des rangs représente le degré de satisfaction de l'individu i. En conséquence, la fonction calculant la satisfaction de l'individu i est de la forme suivante : Score = n - rang + 1, où n représente la taille de la liste de préférences (le nombre d'établissements ou le nombre d'étudiants) et rang représente le rang de l'affectation dans la liste de préférences. Il convient de noter que le rang varie de 1 à n. Le graphe ci-dessous présente le degré de satisfaction en fonction du rang d'affectation dans une liste de préférences de taille 10.

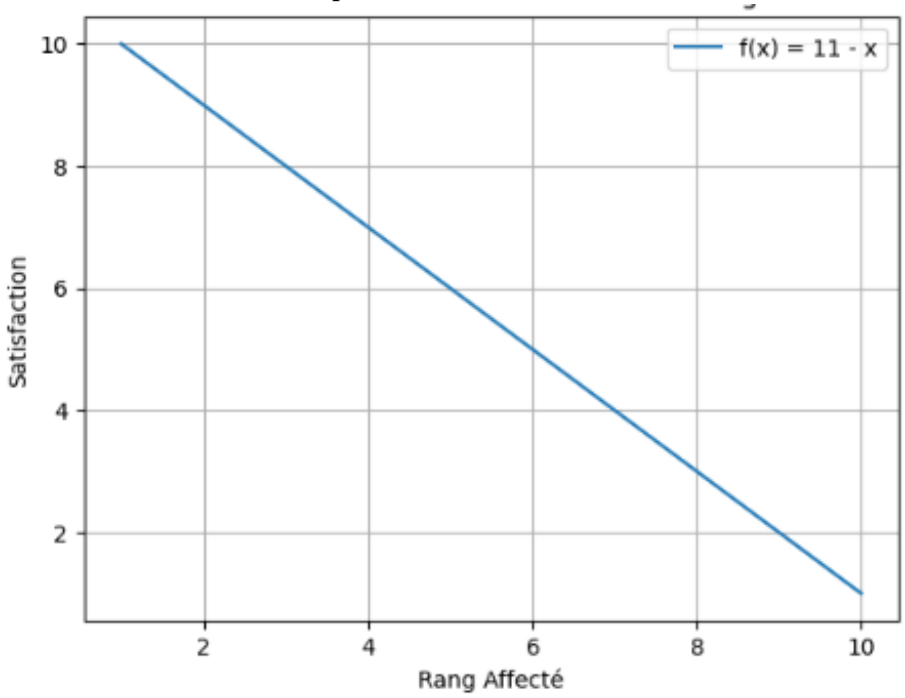

FIGURE 1 – Variation de la satisfaction en fonction du rang d'affectation

Pour le calcul de la satisfaction globale par catégorie (des étudiants ou des établissements), la méthode proposée consiste à sommer les degrés de satisfaction des individus de la catégorie et à diviser cette somme par le degré maximal de satisfaction globale. Le degré maximal de la satisfaction globale correspond au cas où chaque individu a eu son premier choix. Ainsi, chaque individu aurait un degré de satisfaction de n. En sommant les degrés maximaux de satisfaction des individus, on obtient : n^2.

Pour rendre la satisfaction plus compréhensible pour les utilisateurs de le système, elle sera ajustée en étant multipliée par 100, présentant ainsi la satisfaction sous forme de pourcentage.

3.2.2 Justification des choix :

- Pour la mesure de satisfaction proposée le rang commence par 1, donc si un individu est affecté à son dernier choix, il aura une satisfaction de 1 et non de 0. Cela est

justifié par le fait que même si l'individu obtient son choix le moins préféré, il a quand même été affecté à un établissement qu'il a exprimé dans sa liste de préférences.

- Bien que cette mesure semble simple pour un scénario assez complexe, son utilisation est justifiée car un système linéaire nous semblait être le meilleur modèle pour représenter au mieux le degré de satisfaction car nous voulions que l'écart de satisfaction entre le choix numéro 1 et le choix numéro 2 soit le même que celui entre le choix numéro 2 et le choix numéro 3.
- Il était envisageable d'intégrer une indication du degré de satisfaction des étudiants. Cependant, cela pourrait compromettre la manipulation des choix. Un individu pourrait attribuer un score élevé à son premier choix et des scores bas pour les autres choix afin de maximiser ses chances d'obtenir son choix préféré. Cela pourrait potentiellement fausser la représentation réelle des préférences de l'individu et compromettre l'équité du système de l'affectation.

3.3 Extensions du système :

3.3.1 Extension 1 :l'utilisation de la logique des poids

Chaque élève et chaque établissement exprime ses préférences sous forme de couples où la relation d'ordre est accompagnée d'une valeur représentant le degré de la préférence ce qui correspond à la logique possibiliste. on applique pour chaque étudiant et chaque établissement la règle de cramer simpson pour effectuer un ordre total qui permet par la suite d'appliquer l'algorithme du mariage stable.

Exemple :

Préférences d'un étudiant :

$$e1 : \langle (u1 > u2), 0.6\rangle; \langle (u1 > u3), 0.5\rangle; \langle (u1 > u4), 0.8\rangle; \langle (u1 > u5), 0.3\rangle;$$

$\langle (u2 > u5), 0.5\rangle; \langle (u3 > u4), 0.7\rangle; \langle (u4 > u2), 0.8\rangle$

Application de la règle de Cramer-Simpson

	$u1$	$u2$	$u3$	$u4$	$u5$	score
$u1$	$-$	0.6	0.5	0.8	0.3	2.2
$u2$	0.4	$-$	0	0.2	0.5	1.1
$u3$	0.5	0	$-$	0.7	0	1.2
$u4$	0.2	0.8	0.3	$-$	0	1.3
$u5$	0.7	0.5	0	0	$-$	1.2

Selon l'ordre décroissant des scores calculés, on peut déduire que l'ordre total des préférences de l'étudiant 1 est : $u1 > u4 > u3 = u5 > u2$.

Une deuxième version de cette extension pourrait être représentée en utilisant la logique de pénalité. Cette fois-ci, on utilise la règle de Cramer pour ordonner les préférences selon l'ordre croissant des scores calculés.

3.3.2 Extension 2 :l'utilisation de la logique conditionnelle

En réalité, les étudiants sélectionnent leurs établissements en fonction de certains critères, tout comme les établissements sélectionnent leurs étudiants selon des critères spécifiques.

Dans cette extension, les étudiants expriment leurs préférences conditionnelles en fonction de différentes combinaisons de critères de sélection des établissements. Ensuite, le graphe CPNet est utilisé pour établir une relation de préférence associée à ces préférences conditionnelles. Après cela, nous associerons chaque établissement aux critères qu'il satisfait, permettant ainsi d'établir un ordre total des établissements pour chaque étudiant. Cette méthode ouvre la voie à l'application de l'algorithme du mariage stable. Ce même processus est effectué pour les établissements pour effectuer un ordre total sur les étudiants , cette fois en considérant leurs propres critères de sélection.

Exemple :

Critères de sélection des établissements :

- Formation : {Intelligence artificielle, Génie logiciel}
- Localisation : {Montpellier, Paris}
- Coût de la formation : {3770 €, 243 €}

Critères de sélection des étudiants :

- Formation effectuée : {même formation proposée, formation complémentaire, formation différente}
- Classement : {parmi les 10% premiers ; parmi les 25% premiers, parmi les 50% premiers}
- Age : {moins de 25 ans, plus de 26 ans}

L'étudiant peut exprimer ses préférences par rapport aux critères comme suit :

- Intelligence artificielle > Génie logiciel
- Montpellier et 243€ > Paris et 243€
- Montpellier et 3770€ > Paris et 3770€
- Montpellier et 243€ > 3770€ et Montpellier
- Paris et 243€ > Montpellier et 3770€

On construit à partir de ces préférences le graphe CPNet suivant :

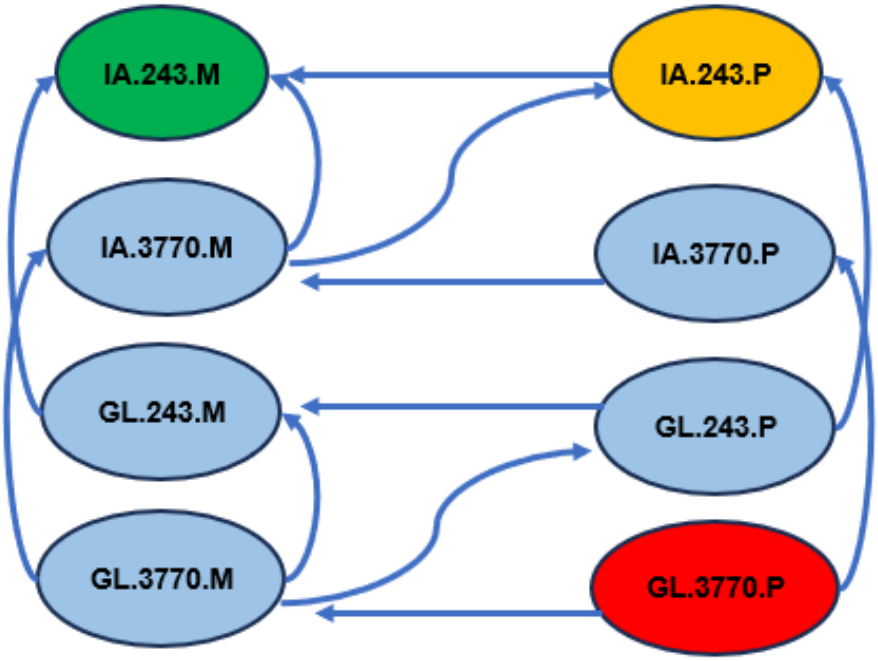

FIGURE 2 – Graphe CPNet

L'ordre total serait donc :

IA - Montpellier - 243€ > IA - Paris - 243€ > IA - Montpellier - 3770€ > IA - Paris - 3770€ >

GL - Montpellier - 243€ > GL - Paris - 243€ > GL - Montpellier - 3770€ > GL - Paris - 3770€.

On associe chaque combinaison des 8 à l'établissement associé, ce qui nous permettra d'obtenir un ordre total sur les établissements pour cet étudiant.

Une deuxième version de cette extension pourrait être représentée en utilisant la sémantique optimiste qui conduit au même ordre total.

4 Réalisation :

4.1 Langages de programmation :

Nous avons choisi d'implémenter l'algorithme en utilisant le langage Python, un langage que nous maîtrisons. Ce choix s'est avéré adéquat pour respecter la contrainte de temps et pour manipuler aisément les structures de données complexes. De plus, nous avons utilisé JavaScript, HTML et CSS pour créer l'interface graphique.

4.2 Méthodes :

Méthode	**Paramètres**	**Explication**
generate_names	n	À partir de la valeur n saisie par l'utilisateur, générer n noms d'étudiants et n noms d'établissements
generate_preferences	liste des étudiants, liste des établissements	Pour chaque individu d'une catégorie, générer des préférences aléatoires ordonnées de l'autre catégorie
stable_marriage students_first	liste des étudiants, liste des établissements, préférences des étudiants, préférences des établissements	Exécuter le mariage stable ne donnant la priorité aux étudiants
stable_marriage schools_first	liste des étudiants, liste des établissements, préférences des étudiants, préférences des établissements	Exécuter le mariage stable en donnant la priorité aux établissements
calculate_satisfaction students_first	préférences des étudiants, préférences des établissements, affectations des étudiants priorité étudiants, affectation des établissements priorité étudiants	Mesurer la satisfaction des étudiants et des établissements, ainsi que la satisfaction moyenne pour la version de l'algorithme privilégiant les étudiants
calculate_satisfaction schools_first	préférences des étudiants, préférences des établissements, affectations des étudiants priorité établissements, affectation des établissements priorité établissements	Mesurer la satisfaction des étudiants et des établissements, ainsi que la satisfaction moyenne pour la version de l'algorithme privilégiant les établissements

TABLE 1 – Description des méthodes

4.3 Interface :

L'interface réalisée est sous forme d'une page web afin de fournir une plateforme accessible et conviviale pour les utilisateurs.

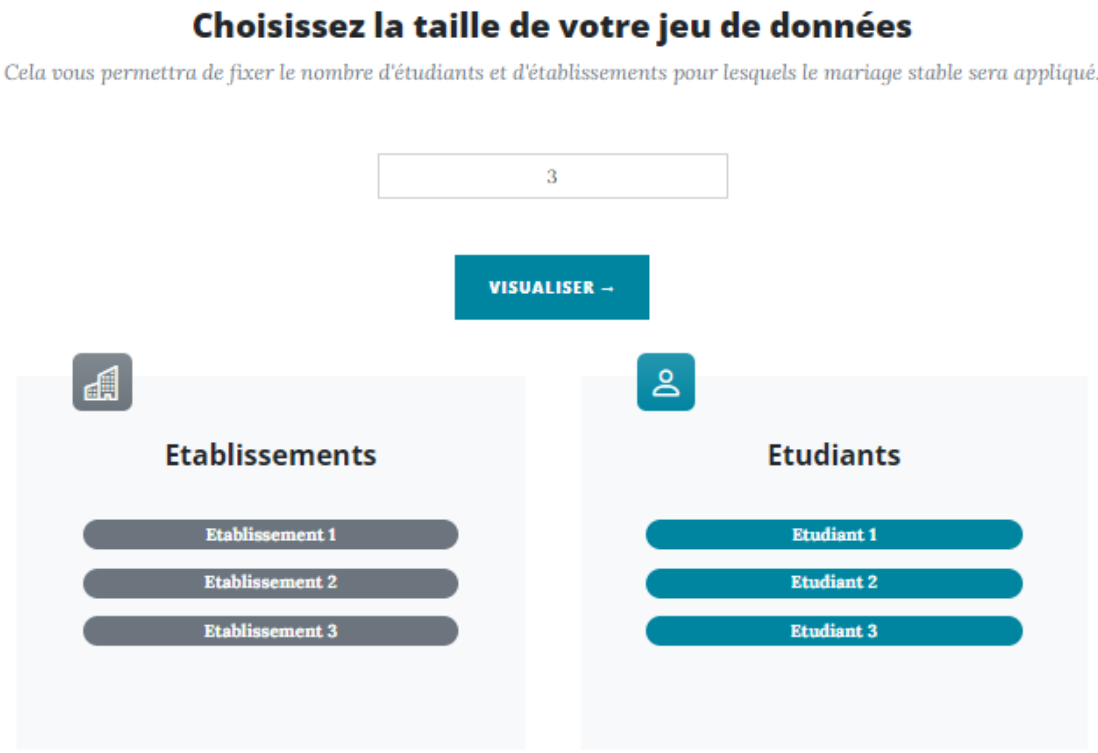

FIGURE 3 – Génération du jeu de données

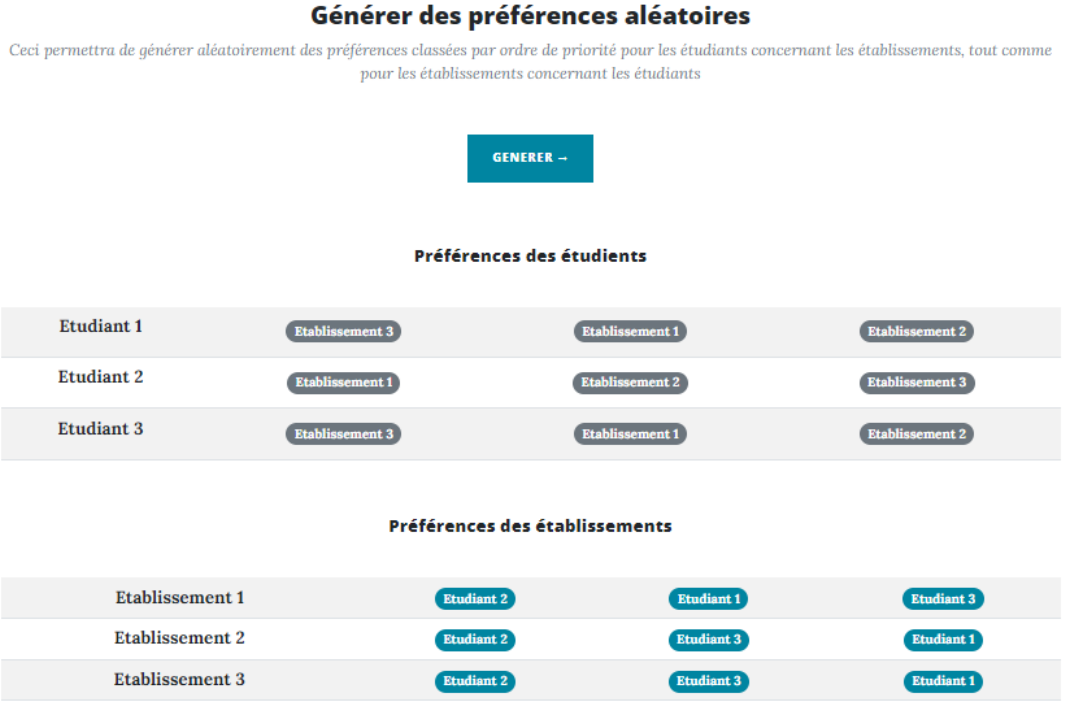

FIGURE 4 – Génération des préférences aléatoires.

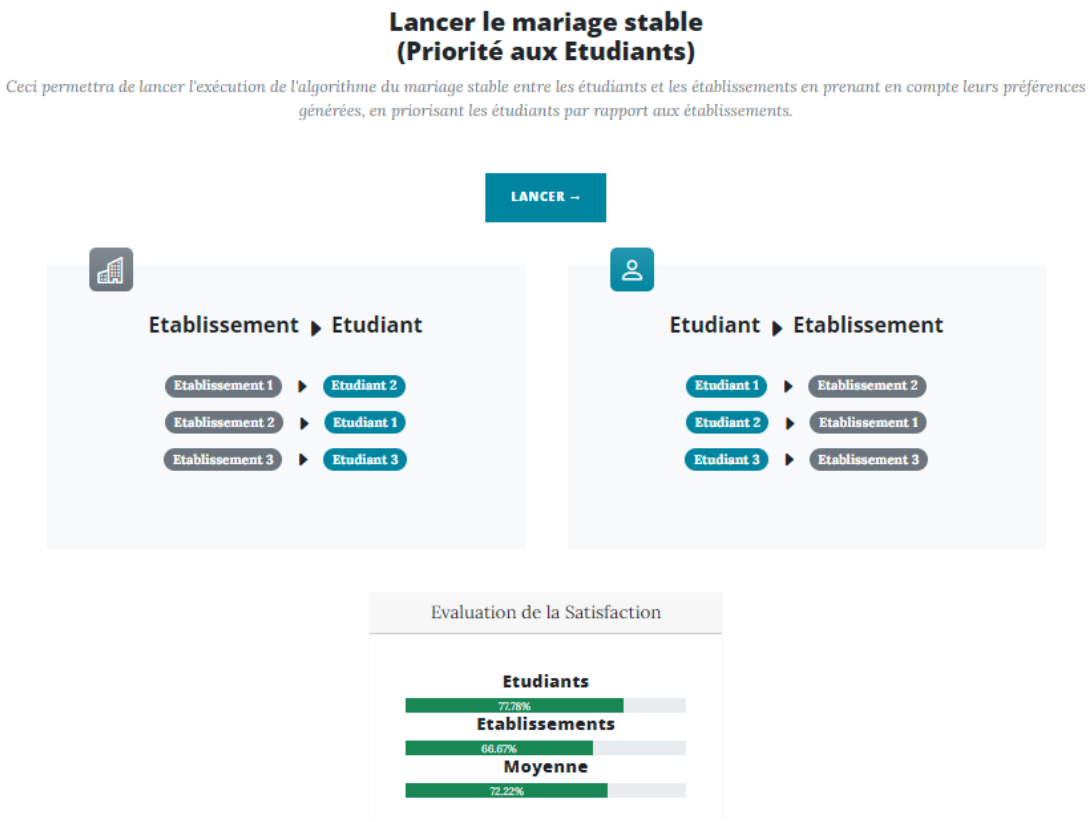

FIGURE 5 – Exécution du mariage stable en donnant la priorité aux étudiants

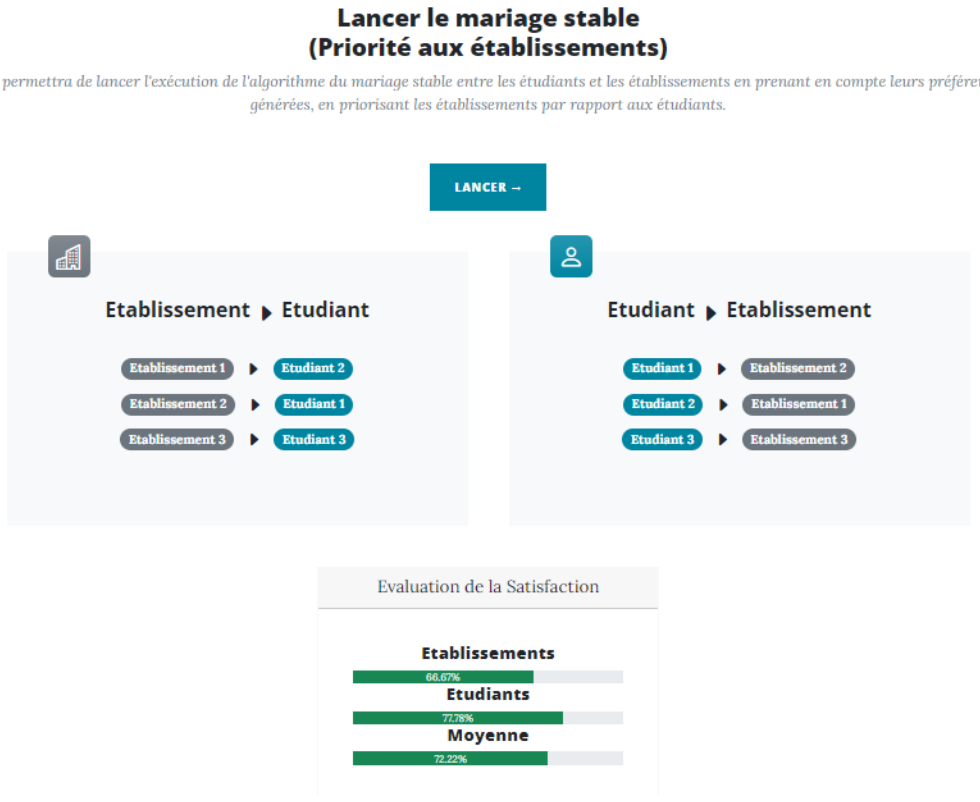

FIGURE 6 – Exécution du mariage stable en donnant la priorité aux établissements

FIGURE 7 – Comparaison des deux versions en termes de satisfaction

5 Evaluation :

5.1 Génération des jeux de données et Tests

Pour l'évaluation du système, nous avons varier n (nombre d'étudiants et d'établissements) et lancer automatiquement le processus pour chaque valeur de n, enregistrant les scores de satisfaction pour les étudiants et les établissements pour chacune des deux versions de l'algorithme. Les résultats sont exportés dans un fichier nommé results.csv.

Les tests sont effectués pour déterminer si la satisfaction obtenue par la méthode proposée est aléatoire ou si elle est liée à l'ordre de priorité ou au paramètre n.

5.2 Analyse des résultats

Nous obtenons des résultats de satisfaction très élevés à la fois pour les établissements et pour les étudiants, démontrant ainsi la présence de partenaires stables. Pour chaque ensemble de données, nous constatons une satisfaction plus élevée des étudiants lorsque nous prenons d'abord en considération leurs préférences, et une satisfaction plus élevée des établissements lorsque nous considérons en premier lieu leurs préférences. Cela démontre la cohérence du système et le respect des préférences de chaque partie impliquée.

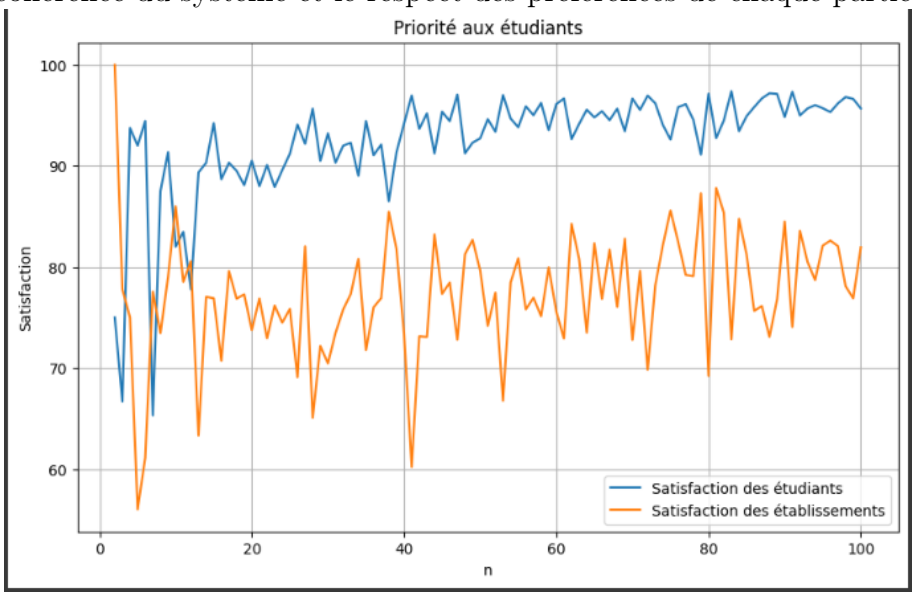

FIGURE 8 – La variation de satisfaction en fonction de la taille du jeu de données (priorité aux étudiants)

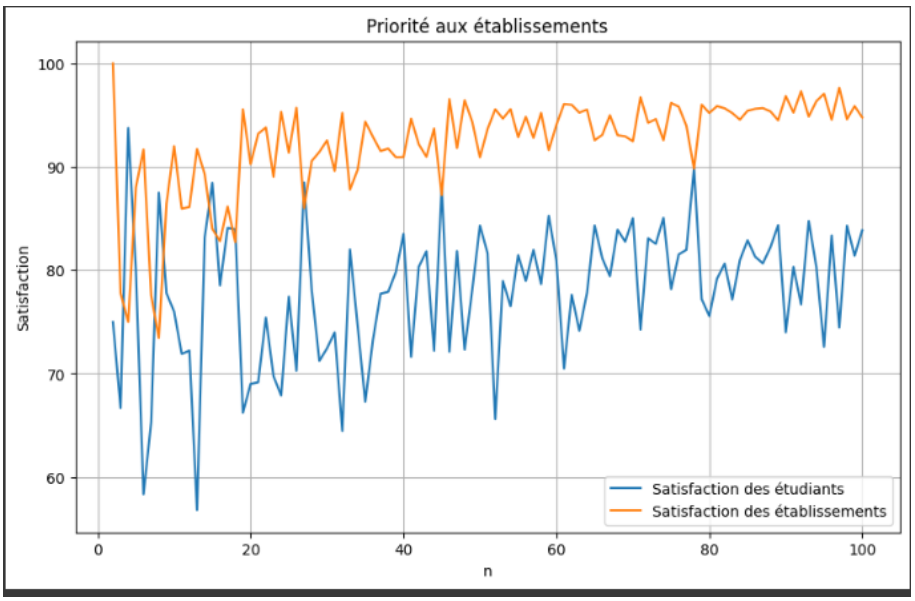

FIGURE 9 – La variation de satisfaction en fonction de la taille du jeu de données (priorité aux établissements)

On remarque également que les valeurs de satisfaction sont plus basses lorsque la taille du jeu de données est réduite. Cela s'explique par la probabilité accrue d'avoir des préférences ordonnées de manière similaire. Ainsi, on peut en déduire que la satisfaction est liée à l'ordre de préférence, à la taille du jeu de données et à la priorité accordée.

6 Conclusion :

Dans ce projet, nous avons mis en œuvre l'algorithme du mariage stable et l'avons étendu de différentes manières, notamment en accordant une importance égale aux établissements et aux étudiants.

La mesure de satisfaction choisie a donné de bons résultats dans les deux versions de l'algorithme, évitant également la manipulation des choix.

La première extension exploite la présentation avec une logique pondérée, avec deux versions distinctes : la logique possibiliste et la logique de pénalité. Elle permet à l'utilisateur d'exprimer ses préférences avec des poids, facilitant ainsi l'établissement d'un ordre total final en exploitant la règle de Crémer-Simpson.

La deuxième extension du système exploite la logique conditionnelle avec deux versions, la sémantique possibiliste et le graphe CPNet. Elle permet de prendre les préférences d'un utilisateur sous forme de préférences conditionnelles selon des critères de choix prédéfinis, ce qui est similaire à des situations réelles. En fin de compte, elle permet d'établir un ordre total à intégrer dans la logique du mariage stable.