

# Intelligence artificielle et Jeux (LU3IN025)

**Intervenants :** Nawal Benabbou, Aurélie Beynier,  
Nicolas Bredèche, Hugo Martin, Nicolas Maudet, Parham Shams

Licence Informatique - Sorbonne Université

2021-2022



## Déroulement de l'UE

- 11 semaines de cours (avec un décalage cours-TD)
- Le cours est divisé en 3 parties (4, 4, puis 3 semaines)
- Parties 1 et 2 : 1h45 TD + 1h45 TP
- Partie 3 : 2×1h45 TP

## Contrôle des connaissances

- Examen final : 40%.
- Contrôle Continu : 60%, composé de :
  - 1 note d'interro (QCM) comptant pour 10%.
  - 1 note de mini-projet sur la partie 2, comptant pour 25%.
  - et 1 note de mini-projet sur la partie 3, comptant pour 25%.

Documents : sur Moodle.

## Partie 1 : Décision collective et optimisation (4 semaines)

- Principe de stabilité, équité, efficacité, manipulabilité.
- Modélisation et méthodes de résolution.

## Partie 2 : Méthodes de l'IA (4 semaines)

- Recherche heuristiques dans des graphes.
- Algorithmes de jeux.
- Apprentissage par renforcement.

## Partie 3 : Décision collective et optimisation (3 semaines)

- Dynamiques des systèmes multi-agents.
- Émergence de comportements collectifs.
- Comportements réactifs et passage sur des robots réels.

# Fil conducteur : partage de ressources entre agents

## Un problème de décision collective

- Quelle modélisation ?
- Quels objectifs ?
- Quelles méthodes ?

## Point de vue ?

- Vue centralisée (optimisation, choix social, aide à la décision)
- Vue décentralisée (comportement des agents, stratégie individuelle/collective, jeux)

## Mise en situation

- Aspects temporels
- Spatialisation

**Des thématiques du master ANDROIDE** (AgeNts Distribués, Robotique, Recherche Opérationnelle, Interaction, DEcision)

# Chapitre 1 de la Partie 1 : “Le problème du mariage stable”

## Problème d'affectation des internes aux hôpitaux (années 50, États-Unis)

### Difficultés :

- Compétition entre hôpitaux (RDV pris pour signature de contrat 2 ans avant la fin des études...)
- Complexité de mise en œuvre à “grande échelle” (délai de réponse très court pour les candidats, seulement 12h)
- Affectation obtenue potentiellement mauvaise (vision locale, intérêt individuel VS collectif)

### Autres exemples

- Affectation des élèves/étudiants dans les écoles/universités publiques (Admission Post-Bac, Parcoursup, Affelnet...)
- Appariement donneur-receveur d'organes (mécanisme d'allocation de reins proposé par l'équipe d'Alvin Roth...)

**Nécessité** : la mise en œuvre de procédures coordonnées/centralisées.

Que doit-on avoir en entrée de la procédure ?

Des informations sur :

# Affectation des internes aux hôpitaux

Que doit-on avoir en entrée de la procédure ?

Des informations sur :

- les préférences des internes sur les hôpitaux,
- les préférences des hôpitaux sur les internes.

→ on demande aux internes de fournir une liste ordonnée des hôpitaux, et aux hôpitaux de faire la même chose sur les internes.

Étant données ces listes ordonnées, on souhaite utiliser une procédure réalisant une “bonne” affectation des internes aux hôpitaux.

Mise en œuvre :

# Affectation des internes aux hôpitaux

## Que doit-on avoir en entrée de la procédure ?

Des informations sur :

- les préférences des internes sur les hôpitaux,
- les préférences des hôpitaux sur les internes.

→ on demande aux internes de fournir une liste ordonnée des hôpitaux, et aux hôpitaux de faire la même chose sur les internes.

Étant données ces listes ordonnées, on souhaite utiliser une procédure réalisant une “bonne” affectation des internes aux hôpitaux.

## Mise en œuvre :

- Quel algorithme utiliser ?
- Quel(s) sont les objectif(s) à atteindre ?
- Y a-t-il des situations à éviter ?



# Quelques exemples

Quels objectifs ?

Faire en sorte que :

Des situations à éviter ?

Il ne faudrait pas que :

# Quelques exemples

## Quels objectifs ?

Faire en sorte que :

- globalement tout le monde soit content (objectif **utilitariste**).
- personne ne soit trop mécontent (objectif **égalitariste**).
- il n'existe pas d'allocation permettant de rendre tout le monde plus content (objectif **Pareto-optimalité**).

## Des situations à éviter ?

Il ne faudrait pas que :

# Quelques exemples

## Quels objectifs ?

Faire en sorte que :

- globalement tout le monde soit content (objectif **utilitariste**).
- personne ne soit trop mécontent (objectif **égalitariste**).
- il n'existe pas d'allocation permettant de rendre tout le monde plus content (objectif **Pareto-optimalité**).

## Des situations à éviter ?

Il ne faudrait pas que :

- certains hôpitaux/internes soient favoriser (**neutralité/anonymité**).
- deux internes  $I_1$  et  $I_2$  soient respectivement affectés aux hôpitaux  $H_1$  et  $H_2$  alors que  $I_1$  préfère  $H_2$  et que  $H_2$  préfère  $I_1$  (**stabilité**).

→ Tout d'abord, on va se concentrer sur cette dernière notion : la stabilité.

# Un problème simplifié : le mariage stable

(cas où il y a exactement une place par hôpital)

**En entrée :**  $n$  hommes et  $n$  femmes classent les membres du sexe opposé du “meilleur” au “pire”.

## Quelques définitions :

- **Mariage/couplage parfait** : tous les hommes sont affectés à une femme différente.
- **Paire instable** : une paire  $(h, f)$  est dite “instable” si l’homme  $h$  et la femme  $f$  préfèrent être ensemble plutôt qu’être avec le conjoint qui leur a été affecté.
- **Mariage stable** : affectation homme-femme sans paire instable.

## Problème du mariage stable :

Étant données les listes de préférences de  $n$  hommes et de  $n$  femmes, déterminer un mariage parfait et stable (s’il en existe...).

# Un petit exemple

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

Questions-réponses :

- Est-ce que le mariage X-C, Y-B, Z-A est stable ?
- Même question pour le mariage X-A, Y-B, Z-C ?
- Plus généralement, existe-t-il toujours un mariage stable ?

# Un petit exemple

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

Questions-réponses :

- Est-ce que le mariage X-C, Y-B, Z-A est stable ?  
Non, (B,X) forme une paire instable.
- Même question pour le mariage X-A, Y-B, Z-C ?
- Plus généralement, existe-t-il toujours un mariage stable ?

# Un petit exemple

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

Questions-réponses :

- Est-ce que le mariage X-C, Y-B, Z-A est stable ?  
Non, (B,X) forme une paire instable.
- Même question pour le mariage X-A, Y-B, Z-C ?  
Oui, il n'y aucune paire instable.
- Plus généralement, existe-t-il toujours un mariage stable ?

# Un petit exemple

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

Questions-réponses :

- Est-ce que le mariage X-C, Y-B, Z-A est stable ?  
Non, (B,X) forme une paire instable.
- Même question pour le mariage X-A, Y-B, Z-C ?  
Oui, il n'y aucune paire instable.
- Plus généralement, existe-t-il toujours un mariage stable ?  
Oui, on peut toujours construire un mariage stable avec l'algorithme de Gale-Shapley "Proposer et Rejeter" (cf. transparent suivant).



# Algorithme de Gale-Shapley “Proposer et Rejeter”

[Gale-Shapley 1962] : Méthode intuitive qui construit un mariage parfait stable.

---

## Algorithm 1: “Proposer et Rejeter”

---

**Entrées** ↓ les listes ordonnées de  $n$  hommes et de  $n$  femmes.

**Sortie** ↑ : un mariage parfait et stable.

Initialiser chaque personne comme libre.

**Tant que** il existe un homme libre qui n’a pas proposé à toutes les femmes :

    Choisir un tel homme  $h$ .

    Soit  $f$  la 1<sup>ère</sup> femme dans la liste de  $h$  qui n’a pas reçu de proposition de  $h$ .

**Si**  $f$  est libre **alors** :

        Considérer  $h$  et  $f$  comme fiancés.

**Sinon si**  $f$  préfère  $h$  à son fiancé  $h'$  **alors** :

        Considérer  $h$  et  $f$  comme fiancés.

        Considérer  $h'$  comme libre.

**Sinon** :

$f$  rejette la proposition de  $h$ .

---

# Algorithme de Gale-Shapley “Proposer et Rejeter”

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

# Algorithme de Gale-Shapley “Proposer et Rejeter”

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

On obtient la suite de propositions suivante :

Proposition	Décision	Mariage

# Algorithme de Gale-Shapley “Proposer et Rejeter”

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

### Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

### Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

On obtient la suite de propositions suivante :

Proposition	Décision	Mariage
$Z : A$	A accepte	$(Z-A)$

# Algorithme de Gale-Shapley “Proposer et Rejeter”

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

On obtient la suite de propositions suivante :

Proposition	Décision	Mariage
Z :A	A accepte	(Z-A)
Y :B	B accepte	(Z-A, Y-B)

# Algorithme de Gale-Shapley “Proposer et Rejeter”

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

On obtient la suite de propositions suivante :

Proposition	Décision	Mariage
Z :A	A accepte	(Z-A)
Y :B	B accepte	(Z-A, Y-B)
X :A	A accepte	(X-A, Y-B)

# Algorithme de Gale-Shapley “Proposer et Rejeter”

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

On obtient la suite de propositions suivante :

Proposition	Décision	Mariage
Z : A	A accepte	(Z-A)
Y : B	B accepte	(Z-A, Y-B)
X : A	A accepte	(X-A, Y-B)
Z : B	B refuse	(X-A, Y-B)

# Algorithme de Gale-Shapley "Proposer et Rejeter"

## Exercice 1 : Application

Appliquer l'algorithme de GS sur le même exemple, en considérant l'ordre Z-Y-X sur les hommes.

Préférences des hommes :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

Préférences des femmes :

	1	2	3
Amy	Xavier	Yohan	Zach
Bea	Xavier	Yohan	Zach
Claire	Yohan	Xavier	Zach

On obtient la suite de propositions suivante :

Proposition	Décision	Mariage
Z : A	A accepte	(Z-A)
Y : B	B accepte	(Z-A, Y-B)
X : A	A accepte	(X-A, Y-B)
Z : B	B refuse	(X-A, Y-B)
Z : C	C accepte	(X-A, Y-B, Z-C)

On obtient le mariage X-A, Y-B, Z-C.



# Terminaison et validité de l'algorithme

## Exercice 2 : Terminaison

Montrer que l'algorithme de GS se termine.

# Terminaison et validité de l'algorithme

## Exercice 2 : Terminaison

Montrer que l'algorithme de GS se termine. Preuve : à chaque itération, une proposition est faite. Il y a donc au plus  $n^2$  itérations.

# Terminaison et validité de l'algorithme

## Exercice 2 : Terminaison

Montrer que l'algorithme de GS se termine. Preuve : à chaque itération, une proposition est faite. Il y a donc au plus  $n^2$  itérations.

## Exercice 3 : Validité

Montrer que :

- ① l'algorithme retourne un mariage parfait (tout le monde est fiancé),
- ② et que le mariage est stable.

# Terminaison et validité de l'algorithme

## Exercice 2 : Terminaison

Montrer que l'algorithme de GS se termine. Preuve : à chaque itération, une proposition est faite. Il y a donc au plus  $n^2$  itérations.

## Exercice 3 : Validité

Montrer que :

- ① l'algorithme retourne un mariage parfait (tout le monde est fiancé),
- ② et que le mariage est stable.

Preuve 1 : supposons qu'il existe un homme  $h$  non marié. Il existe alors une femme  $f$  non mariée. Comme une femme reste mariée à partir de la 1<sup>ère</sup> proposition reçue, alors  $f$  n'a jamais reçue de proposition. Pourtant,  $h$  est célibataire à la fin de la procédure, donc  $h$  a forcément proposé à toutes les femmes (contradiction).

# Terminaison et validité de l'algorithme

## Exercice 2 : Terminaison

Montrer que l'algorithme de GS se termine. Preuve : à chaque itération, une proposition est faite. Il y a donc au plus  $n^2$  itérations.

## Exercice 3 : Validité

Montrer que :

- ① l'algorithme retourne un mariage parfait (tout le monde est fiancé),
- ② et que le mariage est stable.

Preuve 1 : supposons qu'il existe un homme  $h$  non marié. Il existe alors une femme  $f$  non mariée. Comme une femme reste mariée à partir de la 1<sup>ère</sup> proposition reçue, alors  $f$  n'a jamais reçue de proposition. Pourtant,  $h$  est célibataire à la fin de la procédure, donc  $h$  a forcément proposé à toutes les femmes (contradiction).

Preuve 2 : Supposons qu'il existe  $h-f$  et  $h'-f'$  dans l'allocation finale et que la paire  $(h, f')$  est instable. Si  $h$  n'a jamais proposé à  $f'$ , alors  $h$  préfère  $f$  à  $f'$  et donc  $(h, f')$  n'est pas instable (contradiction). Si  $h$  a proposé à  $f'$ , alors  $f'$  a rejeté  $h$  pour un meilleur partenaire, donc  $f'$  préfère  $h'$  à  $h$  (contradiction).

# Comprendre la solution retournée par l'algorithme de GS

## Questions :

- Peut-il exister plusieurs mariages parfaits stables ?
- Si oui, lequel est retourné par l'algorithme de GS ?

# Comprendre la solution retournée par l'algorithme de GS

## Questions :

- Peut-il exister plusieurs mariages parfaits stables ?
- Si oui, lequel est retourné par l'algorithme de GS ?

Un exemple :

	1	2	3
Xavier	Amy	Bea	Claire
Yohan	Bea	Amy	Claire
Zach	Amy	Bea	Claire

	1	2	3
Amy	Yohan	Xavier	Zach
Bea	Xavier	Yohan	Zach
Claire	Xavier	Yohan	Zach

## Réponse sur l'exemple :

- L'algorithme de GS retourne toujours le mariage X-A, Y-B, Z-C.
- Il existe pourtant un autre mariage parfait stable : X-B, Y-A, Z-C.

## Définition : partenaire valide

Un homme (resp. une femme) est un “partenaire valide” pour une femme (resp. un homme) si et seulement s’il existe un mariage parfait et stable dans lequel ils sont mariés.

## Définition : homme-optimalité

Un mariage parfait est dit “homme-optimal” si tous les hommes sont mariés avec leur meilleure partenaire valide.

## Définition : femme-optimalité

Un mariage parfait est dit “femme-optimal” si toutes les femmes sont mariées avec leur meilleur partenaire valide.



# Dissymétrie apparente hommes/femmes avec l'algorithme de GS

Question :

D'après vous, est-ce que l'algorithme GS favorise un sexe ? Si oui, lequel ?

Question :

D'après vous, est-ce que l'algorithme GS favorise un sexe ? Si oui, lequel ?

**Proposition** : L'algorithme de GS conduit à un mariage homme-optimal.

## Question :

D'après vous, est-ce que l'algorithme GS favorise un sexe ? Si oui, lequel ?

**Proposition :** L'algorithme de GS conduit à un mariage homme-optimal.

**Preuve :** Soit  $S$  un mariage parfait et stable obtenu par l'algorithme de GS. Par l'absurde, supposons qu'au moins un homme n'est pas marié avec sa meilleure partenaire valide. Notons  $h_1$  le premier homme rejeté par sa meilleure partenaire valide, appelée  $f_1$ . Notons  $h_2$  l'homme avec qui est mariée  $f_1$  quand elle rejette  $h_1$ . Comme  $f_1$  est une partenaire valide de  $h_1$ , alors il existe un mariage parfait et stable, noté  $S_1$ , où  $h_1$  et  $f_1$  sont mariés. Notons  $f_2$  la femme mariée à  $h_2$  dans le mariage  $S_1$ . Montrons que  $(h_2, f_1)$  est instable dans  $S_1$ .

- Comme la femme  $f_1$  a rejeté  $h_1$  pour  $h_2$ , alors  $f_1$  préfère  $h_2$  à  $h_1$ .
- Comme  $h_1$  est le premier homme rejeté par sa meilleure partenaire valide (qui est  $f_1$ ), et que  $h_2$  est marié à  $f_1$  à ce moment là, alors  $h_2$  a forcément proposé à  $f_1$  avant  $f_2$ . En effet, dans le cas contraire, cela voudrait dire que  $f_2$  aurait rejeté  $h_2$  avant qu'il ne se marie avec  $f_1$ , et comme  $f_2$  est une partenaire valide de  $h_2$  (car mariés dans  $S_1$ ), alors cela contredirait le fait que  $h_1$  soit le premier homme rejeté par sa meilleure partenaire valide. Donc  $h_2$  préfère  $f_1$  à  $f_2$ .

$\Rightarrow (h_2, f_1)$  forme une paire instable pour  $S_1$  (contradiction).

## Conséquences :

- Toutes les exécutions de l'algorithme GS mènent au même mariage, indépendamment de l'ordre dans lequel on a considéré les hommes/propositions.
- L'algorithme de GS est donc neutre/anonyme.

## Conséquences :

- Toutes les exécutions de l'algorithme GS mènent au même mariage, indépendamment de l'ordre dans lequel on a considéré les hommes/propositions.
- L'algorithme de GS est donc neutre/anonyme.

## Et qu'en pensent les femmes ?

**Proposition :** Dans le mariage retourné par l'algorithme de GS, toutes les femmes sont mariées avec leur pire partenaire valide (on parle de mariage "femme-pessimal").

**Preuve :** En exercice à faire à la maison.

# Une extension : le problème des colocataires

## Mariages

Étant donnés  $n$  hommes et  $n$  femmes, chacun ayant classé les membres de l'autre sexe par ordre de préférence, former  $n$  couples de sorte à obtenir un mariage avec de bonnes propriétés (la stabilité par exemple).

## Colocataires

Étant données  $2n$  personnes, chacune ayant classé les  $2n - 1$  autres personnes par ordre de préférence, former  $n$  paires de sorte que le mariage obtenu vérifie de bonnes propriétés.

## Stabilité (colocataires)

Pour le problème des colocataires :

- Une paire  $(x, y)$  est dite “instable” pour un mariage si les personnes  $x$  et  $y$  préfèrent être ensemble plutôt qu’être avec la personne donnée par le mariage.
- Un mariage est dit “stable” s’il n’existe aucune paire instable.

# Problème des colocataires : Stabilité

Rappel : stabilité pour le problème des mariages stables

Il existe toujours un mariage parfait stable, et l'algorithme de Gale-Shapley permet de trouver un tel mariage en temps polynomial.

Questions : stabilité pour le problème des colocataires

Pour ce problème, existe-t-il toujours un mariage parfait stable ? Si oui, peut-on en trouver un efficacement ? Si non, donner un contre exemple.

# Problème des colocataires : Stabilité

Rappel : stabilité pour le problème des mariages stables

Il existe toujours un mariage parfait stable, et l'algorithme de Gale-Shapley permet de trouver un tel mariage en temps polynomial.

Questions : stabilité pour le problème des colocataires

Pour ce problème, existe-t-il toujours un mariage parfait stable ? Si oui, peut-on en trouver un efficacement ? Si non, donner un contre exemple.

Réponse : NON

	1	2	3
Amy	Yohan	Bea	Zach
Bea	Amy	Yohan	Zach
Yohan	Bea	Amy	Zach
Zach	x	x	x



# Problème des colocataires : Stabilité

Rappel : stabilité pour le problème des mariages stables

Il existe toujours un mariage parfait stable, et l'algorithme de Gale-Shapley permet de trouver un tel mariage en temps polynomial.

Questions : stabilité pour le problème des colocataires

Pour ce problème, existe-t-il toujours un mariage parfait stable ? Si oui, peut-on en trouver un efficacement ? Si non, donner un contre exemple.

Réponse : NON

	1	2	3
Amy	Yohan	Bea	Zach
Bea	Amy	Yohan	Zach
Yohan	Bea	Amy	Zach
Zach	x	x	x

Avec qui mettre Z ?

- Avec A ? non,  $(A, B)$  instable.
- Avec B ? non,  $(B, Y)$  instable.
- Avec Y ? non,  $(Y, A)$  instable.

# Problème des colocataires : Stabilité

Rappel : stabilité pour le problème des mariages stables

Il existe toujours un mariage parfait stable, et l'algorithme de Gale-Shapley permet de trouver un tel mariage en temps polynomial.

Questions : stabilité pour le problème des colocataires

Pour ce problème, existe-t-il toujours un mariage parfait stable ? Si oui, peut-on en trouver un efficacement ? Si non, donner un contre exemple.

Réponse : NON

	1	2	3
Amy	Yohan	Bea	Zach
Bea	Amy	Yohan	Zach
Yohan	Bea	Amy	Zach
Zach	x	x	x

Avec qui mettre Z ?

- Avec A ? non,  $(A, B)$  instable.
- Avec B ? non,  $(B, Y)$  instable.
- Avec Y ? non,  $(Y, A)$  instable.

**Néanmoins** quand un mariage stable existe, on peut en trouver un efficacement par un algorithme de recherche de couplage stable dans un graphe (qui sera étudié dans le master ANDROIDE).

# Retour sur le problèmes des hôpitaux et des internes

On considère :

- $n$  internes et  $m$  hôpitaux,
- chaque hôpital  $H_i$  a une capacité  $C_i$ ,
- et on suppose que  $\sum_{i=1}^m C_i = n$  (pour simplifier).

Entrées :

- Chaque interne classe les hôpitaux par ordre de préférence.
- Chaque hôpital classe les internes par ordre de préférence.

Définition : paire instable

Une paire  $(I, H)$  est instable si l'interne  $I$  préfère  $H$  à l'hôpital où il est affecté, et l'hôpital  $H$  préfère  $I$  à l'un des internes qui lui ont été affectés.

Définition : affectation stable

Une affectation des internes aux hôpitaux est dite “stable” si elle ne contient aucune paire instable.

# Retour sur le problèmes des hôpitaux et des internes

## Existence d'une affectation stable

La généralisation de l'algorithme de GS décrite ci-dessous retourne toujours une affectation des internes aux hôpitaux stable.

---

### Algorithm 2: "Proposer et Rejeter" (internes-hôpitaux)

---

**Entrées** ↓ les listes ordonnées de  $n$  internes et de  $m$  hôpitaux.

**Sortie** ↑ : un mariage parfait et stable.

Initialiser chaque interne/hôpital comme libre.

**Tant que** il existe un interne libre qui n'a pas proposé à tous les hôpitaux :

    Choisir un tel interne  $I$ .

    Soit  $H$  le 1er hôpital dans la liste de  $I$  qui n'a pas reçu de proposition de  $I$ .

**Si**  $H$  n'a pas encore atteint sa capacité max **alors** :

        Affecter  $I$  à  $H$ .

**Sinon** :

        Soit  $I'$  le dernier interne dans la liste de  $H$  qui est affecté à  $H$ .

**Si**  $H$  préfère  $I$  à  $I'$  **alors** :

            Affecter  $I$  à  $H$ .

            Considérer  $I'$  comme libre.

**Sinon** :

$H$  rejette la proposition de  $I$ .

---

→ Version "interne-optimal", on fera la version "hôpital-optimal" en TME.