

R5.A.12/R5.B.10 Modélisations mathématiques

Séquence 1 : Mariages stables

Thibault Godin, Lucie Naert

IUT de Vannes

29 août 2024

Modalités de cours

- ▶ 1 cours de 45 min en amphitheatre toutes les deux semaines
- ▶ 2 séances de TD d'1h30 en Python sur Jupyter Notebook

Pendant le cours : présentation d'un problème de modélisation ainsi qu'un ou plusieurs algorithmes permettant de trouver une solution à ce problème.

Pendant les séances de TD : QCM (sur la semaine N-1) + implémentation des algorithmes vus en cours et analyse des résultats.

Thèmes abordés en TD

- ▶ Semaine 1 : Mariages stables avec Gale-Shapley
- ▶ Semaine 2 : Mariages stables équitables avec Selkow
- ▶ Semaine 3 : Flots et Affectations avec Edmonds-Karp
- ▶ Semaine 4 : Initiations aux valeurs propres
- ▶ Semaine 5 : Classement des pages Web avec PageRank
- ▶ Semaine 6 : Clustering spectral et découpes de vaches
- ▶ Semaine 7 : Compression et débruitage d'images avec SVD
- ▶ Semaine 8 : Évaluation avec sujet surprise

(Ce programme est susceptible de changer légèrement)

Évaluation

La note finale sera calculée en faisant la moyenne pondérée de deux types d'évaluation :

- ▶ 5 à 6 QCM sur le cours et les séances de TD de la semaine précédente.
- ▶ En semaine 43 : une évaluation de 3h sur ordinateur sur la modélisation d'un problème donné.

Pour l'évaluation finale, vous travaillerez en binôme. Un problème vous sera communiqué en début de séance. Sur un Jupyter Notebook, vous devrez proposer 1) une modélisation de ce problème ainsi qu'un ou plusieurs algorithmes pour résoudre ce problème et 2) une analyse des résultats.

Plan

Modalités de cours et d'évaluation

Séquence n° 1 : Problème des mariages stables

Énoncé du problème

Application et exemples

Algorithme de Gale–Shapley

Algorithme de Selkow

Séquence n° 1 : Problème des mariages stables

Le problème des mariages stables consiste à trouver, étant donné n éléments d'une première population (P_1) et n éléments d'une deuxième population (P_2), et leurs listes de préférences, une façon stable de les appairer.

Une situation est dite **instable** s'il y a au moins un élément de P_1 et un élément de P_2 qui préféreraient se mettre en couple plutôt que de rester avec leurs partenaires actuels.

Plan

Modalités de cours et d'évaluation

Séquence n° 1 : Problème des mariages stables

Énoncé du problème

Application et exemples

Algorithme de Gale–Shapley

Algorithme de Selkow

Application à Parcoursup

Le problème des mariages stables est énoncé pour la première fois en 1962 avec le problème de l'affectation des étudiants aux diverses formations post-bac (Source : [Wikipedia](#)).

Étudions plus précisément cet exemple...

Exemple de mariages

a,b,c,e,d ①

a,c,b,e,d ②

c,a,e,b,d ③

a,b,c,d,e ④

a,b,c,d,e ⑤

Ⓐ 1,3,5,4,2

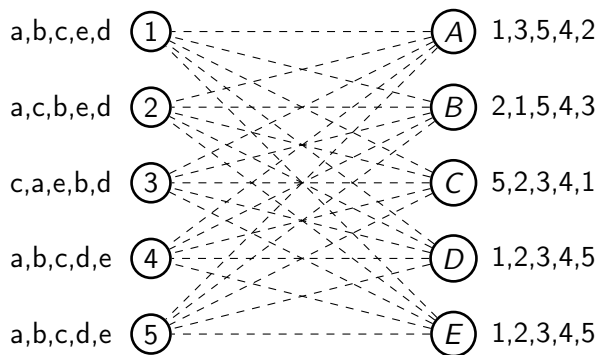
Ⓑ 2,1,5,4,3

Ⓒ 5,2,3,4,1

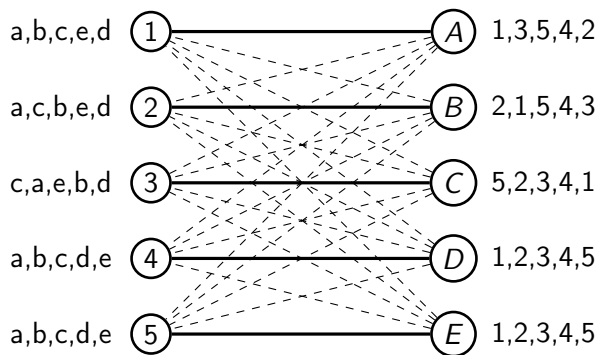
Ⓓ 1,2,3,4,5

Ⓔ 1,2,3,4,5

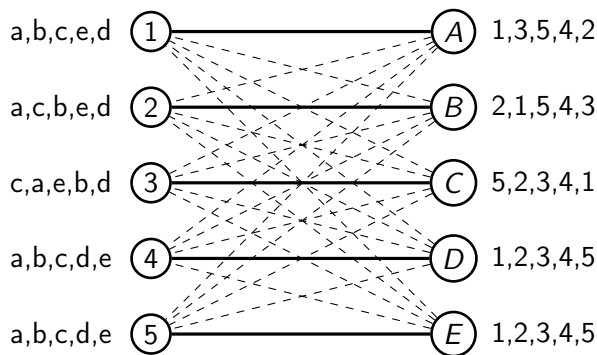
Exemple de mariages



Exemple de mariages

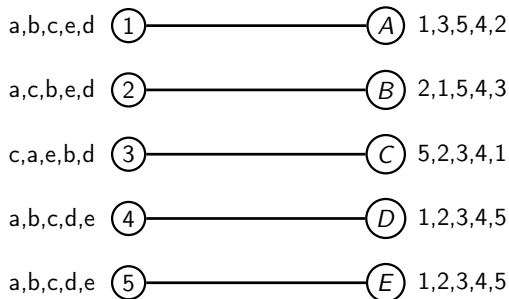


Exemple de mariages



Pas satisfaisant : les préférences ne sont pas utilisées, certains mariages sont instables

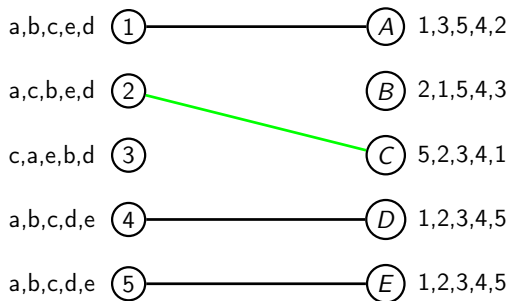
Mariages instables



couple instable (2, C) car

- ▶ 2 préfère C à B
- ▶ C préfère 2 à 3

Mariages instables



couple instable (2, C) car

- ▶ 2 préfère C à B
- ▶ C préfère 2 à 3

On améliorerait la proposition
en prenant l'arête (2, C)

Plan

Modalités de cours et d'évaluation

Séquence n° 1 : Problème des mariages stables

Énoncé du problème

Application et exemples

Algorithme de Gale–Shapley

Algorithme de Selkow

Gale–Shapley

L'algorithme de Gale-Shapley permet de résoudre le problème des mariages stables : il délivre une solution où tous les individus sont dans des mariages stables.

Initialisation de l'algorithme

On considère une population P_1 d'étudiants et une population P_2 de formations.

Entrées : Préférences de P_1 et P_2

Sortie : Liste de mariages stables

Initialisation : Personne n'est marié

Déroulé de l'algorithme

- ▶ Pour chaque étudiant e non marié :
 - ▶ $k = 1$
 - ▶ si le k^{ieme} choix de la liste de préférence de e est non marié \rightsquigarrow mariage avec e
 - ▶ sinon on demande au k^{ieme} choix son avis
 - ▶ si couple instable \rightsquigarrow mariage avec e
 - ▶ sinon $k \leftarrow k + 1$

Application pas à pas

b,a,c,e,d ①

a,c,b,e,d ②

c,a,e,b,d ③

c,b,e,d,a ④

a,b,c,d,e ⑤

Ⓐ 3,1,5,4,2

Ⓑ 2,1,5,4,3

Ⓒ 5,2,3,4,1

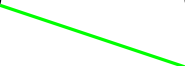
Ⓓ 2,1,5,4,3

Ⓔ 3,2,5,4,1

Initialisation

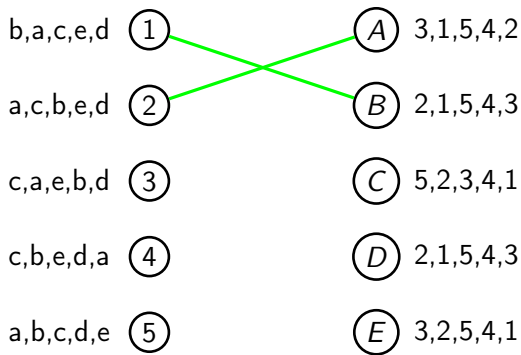
Application pas à pas

b,a,c,e,d	①	Ⓐ	3,1,5,4,2
a,c,b,e,d	②	Ⓑ	2,1,5,4,3
c,a,e,b,d	③	Ⓒ	5,2,3,4,1
c,b,e,d,a	④	Ⓓ	2,1,5,4,3
a,b,c,d,e	⑤	Ⓔ	3,2,5,4,1



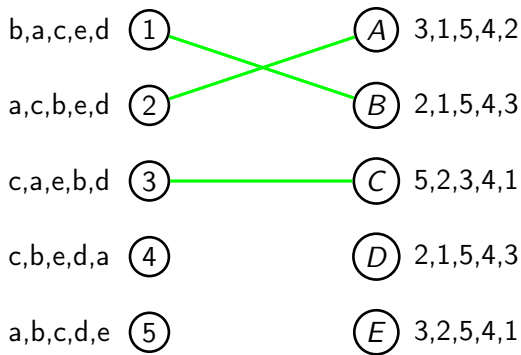
Le premier choix de 1, B , est non marié : on forme donc le couple $(1, B)$

Application pas à pas



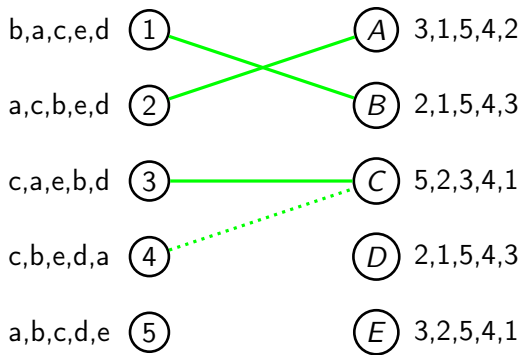
Le premier choix de 2, A, est non marié : on forme donc le couple (2, A)

Application pas à pas



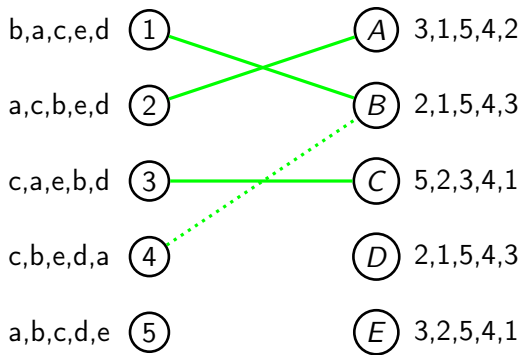
Le premier choix de 3, C, est non marié : on forme donc le couple (3, C)

Application pas à pas



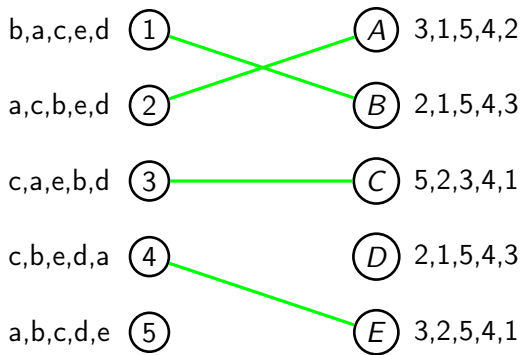
Le premier choix de 4, C, est marié à 3. On regarde donc la stabilité du couple : 4 est **moins bien** placé que 3 dans la liste des préférences de C, on conserve donc le couple initial. On passe au deuxième choix de 4.

Application pas à pas



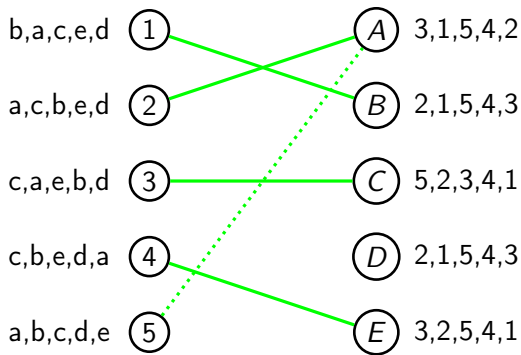
Le deuxième choix de 4, B , est marié à 1. On regarde donc la stabilité du couple : 4 est **moins bien** placé que 1 dans la liste des préférences de B , on conserve donc le couple initial. On passe au troisième choix de 4.

Application pas à pas



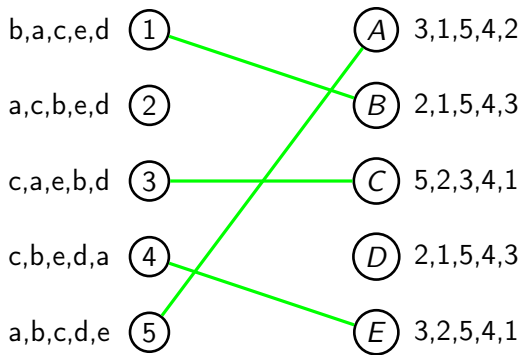
Le troisième choix de 4, E , est non marié : on forme donc le couple $(4, E)$

Application pas à pas



Le premier choix de 5, A, est marié à 2. On regarde donc la stabilité du couple : 5 est **mieux** placé que 2 dans la liste des préférences de A, on casse donc le couple initial et on le remplace par le couple (5, A)

Application pas à pas



2 se retrouve seul, on passe à son deuxième choix...

A vous de terminer...

b,a,c,e,d ①

a,c,b,e,d ②

c,a,e,b,d ③

c,b,e,d,a ④

a,b,c,d,e ⑤

Ⓐ 3,1,5,4,2

Ⓑ 2,1,5,4,3

Ⓒ 5,2,3,4,1

Ⓓ 2,1,5,4,3

Ⓔ 3,2,5,4,1

Que remarquez-vous ?

Qualité de la solution

Une solution stable existe toujours... mais ne satisfait pas forcément les même personnes

A : YXZ	B : ZYX	C : XZY
X : BAC	Y : CBA	Z : ACB

Ici : 3 solutions stables

- ▶ "optimale pour les étudiants" : AY, BZ, CX (les étudiants proposent)
- ▶ "second choix" : AX, BY, CZ (sex-equal stable matching, NP—dur)
- ▶ "optimale pour les universités" AZ, BX, CY (les universités proposent)

(ref : D. F. Manlove, Algorithmics of Matching Under Preferences)

Plan

Modalités de cours et d'évaluation

Séquence n° 1 : Problème des mariages stables

Énoncé du problème

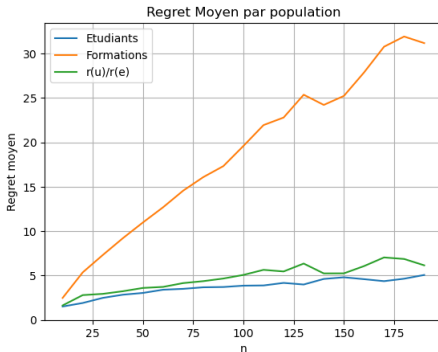
Application et exemples

Algorithme de Gale–Shapley

Algorithme de Selkow

Motivation

L'algorithme de Gale-Shapley permet de donner une solution au problème des mariages stables entre les individus de deux populations P_1 et P_2 tout en avantageant l'une ou l'autre des parties.



Existe t-il un algorithme permettant d'obtenir des mariages toujours stables mais plus équitables ?

Principe de l'algorithme de Selkow

L'algorithme de Selkow a pour but de minimiser le regret maximal, c'est à dire le regret de l'individu (étudiant ou formation) le plus malheureux.

Definition 1

On appelle regret d'un étudiant (ou plus généralement, d'un élément de P_1) le rang, parmi ses préférences, de la formation (ou plus généralement, de l'élément de P_2) avec qui il est apparié. On définit de même le regret d'une formation.

Algorithme de Selkow

Entrées : Préférences de P_1 et P_2

Sortie : Liste de mariages équitables

Initialisation : Personne n'est marié

Algorithme de Selkow

Tant qu'un individu n'est pas marié :

- ▶ Faire un Gale-Shapley **sur les individus non mariés**
- ▶ Suite à ces propositions de mariages, calculer les bornes supérieures et inférieures des regrets de chaque individu du mariage
- ▶ Pour chaque individu :
 - ▶ si la borne supérieure est égale à la borne inférieure : le mariage est considéré comme *non perfectible*. Il est conservé.
 - ▶ sinon, le mariage est perfectible. Il est annulé.
- ▶ S'il reste des individus non mariés :
 - ▶ recherche de l'individu e avec le regret maximal parmi les individus célibataires
 - ▶ Interdire la possibilité de refaire ce mariage (on retire e de la liste de préférences de son partenaire et/ou on retire le partenaire de la liste de préférences de e)

Bornes supérieures et inférieures

L'algorithme de Selkow se base sur une propriété intéressante de Gale-Shapley pour fonctionner : le mariage résultant de Gale-Shapley est le mariage **le plus avantageux** pour les éléments de P_1 (qui choisissent selon leurs préférences) et **le moins avantageux** pour les éléments de P_2 (qui subissent les choix des partenaires).

Bornes supérieures et inférieures

Ainsi, après un mariage de type Gale-Shapley, le regret des éléments de P_1 sera donc minimal (**borne inférieure des regrets de P_1**) et celui des éléments de P_2 maximal (**borne supérieure des regrets de P_2**).

Évidemment, si l'on exécute Gale-Shapley après avoir échangé P_1 et P_2 , nous aurons donc une deuxième proposition de mariage qui avantage P_2 ! Le calcul des regrets sur ce nouveau mariage nous donnera la **borne inférieure des regrets de P_2** et la **borne supérieure des regrets de P_1** .

Mariages perfectibles/non perfectibles

Il suffit ensuite de comparer la borne inférieure et supérieure de chaque individu pour savoir s'il est dans un mariage **perfectible** (bornes différentes) ou **non** (les bornes sont égales : un meilleur mariage ne pourra pas être proposé).

Application sur un exemple

Initialisation : personne n'est marié.

0,2,5,1,4,3 ①

① 0,5,4,1,3,2

5,0,1,3,4,2 ②

② 5,0,4,3,1,2

2,5,1,3,0,4 ③

③ 0,1,3,4,5,2

3,4,5,0,1,2 ④

④ 1,4,5,0,2,3

0,2,5,1,3,4 ⑤

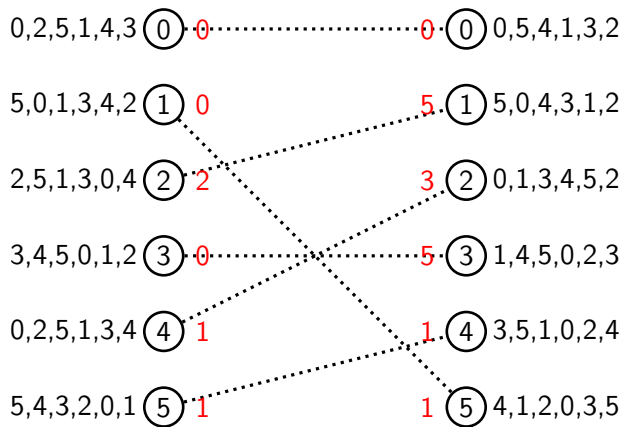
⑤ 3,5,1,0,2,4

5,4,3,2,0,1 ⑥

⑥ 4,1,2,0,3,5

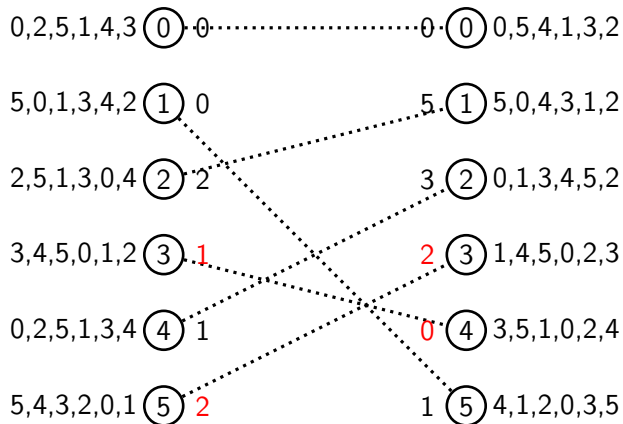
Mariages finaux : aucun

Gale-Shapley (P_1, P_2)



Borne inf(P_1) : [0, 0, 2, 0, 1, 1] - Borne sup(P_2) : [0, 5, 3, 5, 1, 1]

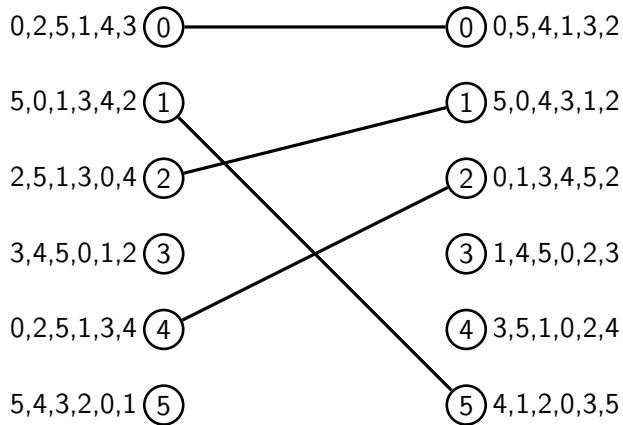
Gale-Shapley (P_2, P_1)



Borne inf(P_1) : [0, 0, 2, 0, 1, 1] - Borne sup(P_2) : [0, 5, 3, 5, 1, 1]

Borne sup(P_1) : [0, 0, 2, 1, 1, 2] - Borne inf(P_2) : [0, 5, 3, 2, 0, 1]

Conservation des mariages non perfectibles



Mariages finaux (mariages non perfectibles) : (0,0), (1,5), (2,1) et (4,2)

Regret maximal

①

②

③

borne = 1 ④

⑤

borne = 2 ⑥

①

②

③

④ borne = 5

⑤ borne = 1

⑥

Recherche du regret maximal dans les bornes supérieures des individus non mariés

Réduction du problème

3,4,5,0,1,2 (3)

(3) 1,4,5,0,2,3

(4) 3,5,1,0,2,4

5,4,3,2,0,1 (5)

Réduction du problème

3,4 ③

③ 5,3

④ 3,5

4,3 ⑤

- ▶ on ne considère que les individus non mariés

Réduction du problème

4 ③

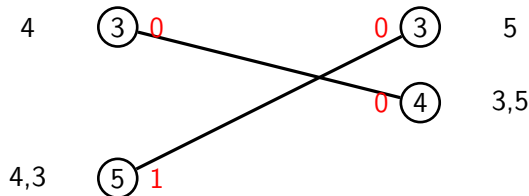
③ 5

④ 3,5

4,3 ⑤

- ▶ on ne considère que les individus non mariés
- ▶ on retire la possibilité de faire le mariage qui apporte le regret max

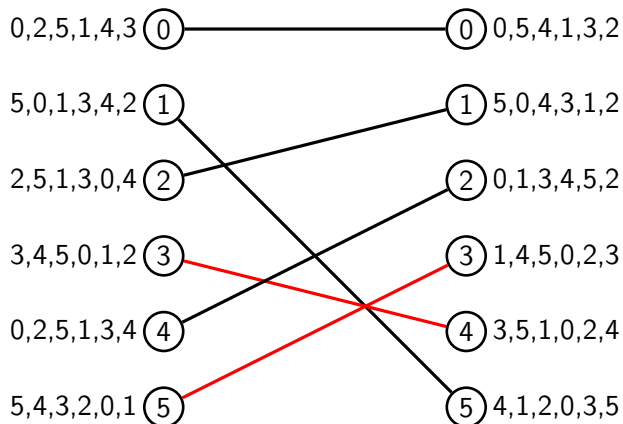
Réitération des étapes précédentes sur le sous-problème



Mariages finaux : (0,0), (1,5), (2,1), (3,5), (4,2) et (5, 3)

Tout le monde est marié : fin de l'algorithme.

Résultat



Regret maximal moyen

