

TD Algo2 – session 11 – Couplage dans un graphe biparti

14 avril 2025

Objectifs d'apprentissage :

- trouver des exemples présentant des couplages particuliers ;
- mettre en pratique des techniques de preuve pour la validité ou la complexité d'algorithmes ;
- concevoir des algorithmes relatifs au problème des mariages stables.

Les 5 premiers exercices sont essentiels.

Exercice 1 : Stabilité

Est-il possible d'avoir un couplage qui ne soit pas stable avec 2 femmes et 2 hommes ? Si oui, fournir un exemple, sinon, donner un argument.

Cette situation se rencontre lorsque tous le monde est en accord sur son partenaire de prédilection mais qu'on apparie chaque personne avec son mauvais partenaire.

Femmes	Choix 1	Choix 2	Hommes	Choix 1	Choix 2
A	C	D	C	A	B
B	D	C	D	B	A

Le couplage A-D et B-C n'est pas stable.

Exercice 2 : Premier choix

Fournir un exemple d'un couplage stable où il n'existe pas de paire (m, w) où m classe w en premier et w classe m en premier.

Femmes	Choix 1	Choix 2	Hommes	Choix 1	Choix 2
A	C	D	C	B	A
B	D	C	D	A	B

Le couplage A-C et B-D est stable mais les hommes ont leur second choix.

Exercice 3 : Riches et pauvres

Considérons une ville avec n hommes et n femmes particulièrement vénaux et cherchant à se marier. Chaque homme a une liste de préférences qui classe toutes les femmes, et chaque femme a une liste de préférences qui classe tous les hommes. L'ensemble des $2n$ personnes est divisé en deux catégories de personnes : les riches et les pauvres. Supposons que pour un certain nombre k entre 1 et n , il y a k hommes riches et k femmes riches ; il y a donc $n - k$ hommes pauvres et $n - k$ femmes pauvres. Tout le monde préférerait épouser n'importe quelle personne riche plutôt qu'une pauvre. Formellement, chaque liste de préférences a la propriété de classer chaque personne riche du sexe opposé plus haut que chaque personne pauvre du sexe opposé : ses k premières entrées sont les personnes riches (du sexe opposé) dans un certain ordre, et ses $n - k$ suivantes sont les personnes pauvres (du sexe opposé) dans un certain ordre. Montrer que dans tout couplage stable, les riches se marient ensemble et les pauvres aussi.

Une façon naturelle de commencer à réfléchir à ce problème est de supposer que l'affirmation est fausse et d'essayer d'obtenir une contradiction. Si l'affirmation était fausse, qu'est-ce que cela signifierait ? Il existerait un appariement stable M dans lequel un homme riche m serait marié à une femme pauvre w .

Voyons maintenant à quoi ressemblent les autres paires de M . Il y a k hommes riches et k femmes riches. Se pourrait-il que toutes les femmes riches soient mariées à un homme riche dans cet appariement M ? Non : l'un des hommes riches (à savoir, m) est déjà marié à une femme pauvre, ce qui ne laisse que $k - 1$ autres hommes riches. Ainsi, même s'ils étaient tous mariés à des femmes riches, il resterait toujours une femme riche mariée à un homme pauvre.

Soit w une telle femme riche, qui est mariée à un homme pauvre. Il est maintenant facile d'identifier une instabilité dans M : considérons la paire (m, w) . Chacune est riche, mais est mariée à un partenaire pauvre. Ainsi, chacun de m et w préfère l'autre à son partenaire actuel, et donc (m, w) est une instabilité. Ceci contredit notre hypothèse que M est stable, et conclut donc la preuve.

Exercice 4 : *National Resident Matching Program*

Le NRMP consiste à assigner des étudiants en médecine à des hôpitaux aux États-Unis. Ce problème diffère du scénario du problème des mariages stables de deux manières. Premièrement, un hôpital peut être apparié à plus d'un étudiant, de sorte que l'hôpital h prend $r_h \geq 1$ étudiant(s). Deuxièmement, le nombre d'étudiants peut ne pas être égal au nombre d'hôpitaux. Décrire comment modifier l'algorithme de Gale-Shapley pour répondre aux exigences du NRMP.

Si un hôpital h_i a des places disponibles, h_i propose une place à l'étudiant suivant s_j sur sa liste de préférences si s_j est libre alors s_j accepte l'offre sinon (s_j est déjà engagé auprès d'un hôpital h_k) si s_j préfère h_k à h_i alors s_j reste engagé auprès de h_k sinon s_j s'engage auprès de h_i le nombre de places disponibles à h_k augmente d'une unité et le nombre de places disponibles à h_i diminue d'une unité.

Exercice 5 : Quadratique

Décrire comment implémenter l'algorithme de Gale-Shapley en temps $O(V^2)$.

Dans une itération, toutes les opérations sont trivialement constante sauf : “**sinon si** m préfère w à w' avec qui il est déjà engagé **alors**”. Il faut préparer une structure intermédiaire pour savoir en temps constant si un homme m préfère la femme w à w' : $Rang[m, w]$ qui donne le rang de w parmi les préférences de m .

GALE-SHAPLEY($MPref, WPref, n$)

$Libre \leftarrow \emptyset$
pour $i = 1$ **à** n **faire**
 ENFILE($Libre, i$)
 $Prochain[i] \leftarrow 1$
 $Actuel[i] \leftarrow 0$
 pour $j = 1$ **à** n **faire**
 $Rang[i, MPref[i, j]] \leftarrow j$
tant que $Libre \neq \emptyset$ **faire**
 $w \leftarrow$ DEFILE($Libre$)
 $m \leftarrow WPref[w, Prochain[w]]$
 $Prochain[w] \leftarrow Prochain[w] + 1$
 $w' \leftarrow Actuel[m]$
 si $w' = 0$ **alors**
 $Actuel[m] \leftarrow w$
 sinon si $Rang[m, w] < Rang[m, w']$ **alors**
 ENFILE($Libre, w'$)
 $Actuel[m] \leftarrow w$
 sinon
 ENFILE($Libre, w$)
retourner $Actuel$

Exercice 6 : Colocataires stables

Le problème des colocataires stables est similaire au problème des mariages stables, sauf que le graphe est un graphe complet, non biparti, avec un nombre pair de sommets. Chaque sommet représente une personne, et chaque personne classe toutes les autres personnes. Les définitions du couplage stable s'étend de manière naturelle : un couplage est stable s'il n'y a pas deux personnes qui se préfèrent mutuellement plutôt que leur partenaire actuel. Par exemple, considérons quatre personnes (Wendy, Xenia, Yolanda et Zelda) avec les listes de préférences suivantes :

Colocataire	Choix 1	Choix 2	Choix 3
Wendy	Xenia	Yolanda	Zelda
Xenia	Wendy	Zelda	Yolanda
Yolanda	Wendy	Zelda	Xenia
Zelda	Xenia	Yolanda	Wendy

Le couplage suivant est stable : Wendy et Xenia ; Yolanda et Zelda.

Contrairement au problème des mariages stables, le problème des colocataires stables peut avoir des entrées pour lesquelles il n'existe pas de couplage stable. Trouver une telle entrée et expliquer pourquoi il n'existe pas de couplage stable.

Colocataire	Choix 1	Choix 2	Choix 3
1	2	3	4
2	3	4	1
3	4	2	1
4	2	3	1

Trois choix pour 1 : 1-2 pas stable à cause de 2-4, 1-3 à cause de 2-3, 1-4 à cause de 3-4.

Exercice 7 : Télévision

Il existe de nombreux autres contextes dans lesquels nous pouvons poser des questions liées à un certain type de principe de "stabilité". En voici un, qui concerne la concurrence entre deux entreprises. Supposons que nous ayons deux chaînes de télévision, que nous appellerons A et B. Il y a n créneaux de programmation aux heures de grande écoute et chaque chaîne a n émissions de télévision. Chaque

chaîne souhaite élaborer une grille de programmation – une affectation de chaque émission à un créneau distinct – afin d’attirer le plus grand nombre de parts de marché possible.

Voici comment nous déterminons les performances des deux chaînes l’une par rapport à l’autre, compte tenu de leurs grilles. Chaque émission a une cote fixe, basée sur le nombre de personnes qui l’ont regardée l’année dernière ; nous supposons qu’aucune émission n’a exactement la même cote. Une chaîne gagne un créneau horaire donné si l’émission qu’elle programme dans ce créneau a un taux d’audience supérieur à celui de l’émission que l’autre chaîne programme dans ce même créneau. L’objectif de chaque chaîne est de gagner autant de créneaux horaires que possible.

Supposons qu’au cours de la première semaine de la saison d’automne, la chaîne A dévoile une grille S et la chaîne B une grille T . Sur la base de cette paire de grilles, chaque chaîne gagne certains créneaux horaires, conformément à la règle ci-dessus. Nous dirons que la paire de grilles (S, T) est stable si aucun réseau ne peut modifier unilatéralement sa propre grille et gagner plus de créneaux horaires. En d’autres termes, il n’existe pas de grille S' tel que le réseau A gagne plus de créneaux horaires avec la paire (S', T) qu’avec la paire (S, T) ; et symétriquement, il n’existe pas de grille T' tel que le réseau B gagne plus de créneaux horaires avec la paire (S, T') qu’avec la paire (S, T) .

L’analogue de la question de Gale et Shapley pour ce type de stabilité est le suivant : pour tout ensemble d’émissions télévisées et d’indices d’écoute, existe-t-il toujours une paire de grilles stable ? Résoudre cette question en faisant l’une des deux choses suivantes : (a) donner un algorithme qui, pour tout ensemble d’émissions télévisées et d’évaluations associées, produit une paire stable d’horaires ; ou (b) donner un exemple d’un ensemble d’émissions télévisées et d’évaluations associées pour lequel il n’y a pas de paire stable d’horaires.

Il n’y a pas toujours une paire stable de grilles. Supposons que le réseau A ait deux émissions $\{a_1, a_2\}$ avec des cotes d’écoute de 20 et 40 ; et que le réseau B ait deux émissions $\{d_1, d_2\}$ avec des cotes d’écoute de 10 et 30.

Chaque réseau peut révéler l’une des deux grilles. Si, dans la paire résultante, a_1 est associé à d_1 , la chaîne B voudra inverser l’ordre des émissions dans sa grille (afin de gagner un créneau au lieu d’aucun). Si, dans la paire résultante, a_2 est associé à d_2 , alors le réseau A voudra intervertir l’ordre des émissions dans sa grille (afin de gagner deux créneaux au lieu d’un).

Exercice 8 : Compagnie maritime

Peripatetic Shipping Lines, Inc. est une compagnie maritime qui possède n navires et dessert n ports. Chacun de ses navires possède un programme qui indique, pour chaque jour du mois, le port qu’il visite actuellement ou s’il est en mer. (On peut supposer que le “mois” compte ici m jours, pour un certain $m > n$.) Chaque navire visite chaque port pendant exactement un jour au cours du mois. Pour des raisons de sécurité, PSL Inc. a l’exigence stricte suivante :

(†) Deux navires ne peuvent pas se trouver dans le même port le même jour.

La société souhaite effectuer des travaux de maintenance sur tous les navires ce mois-ci, selon le schéma suivant. Elle veut tronquer le programme de chaque navire : pour chaque navire S_i , il y aura un jour où il arrivera dans le port prévu et y restera simplement pour le reste du mois (pour la maintenance). Cela signifie que S_i ne visitera pas les ports restants de son programme (s’il y en a) ce mois-là, mais ce n’est pas grave. Ainsi, le programme tronqué de S_i consistera simplement en son programme original jusqu’à un certain jour spécifié où il se trouve dans un port P ; le reste du programme tronqué lui permet simplement de rester dans le port P .

La question que vous pose l’entreprise est la suivante : étant donné les horaires de chaque navire, trouver une troncature de chacun d’entre eux de manière à ce que la condition (†) continue de s’appliquer : deux navires ne sont jamais dans le même port le même jour.

Montrer qu’un tel ensemble de troncatures peut toujours être trouvé, et donner un algorithme pour les trouver.

Exemple. Supposons que nous ayons deux navires et deux ports, et que le “mois” ait quatre jours. Supposons que le programme du premier navire soit le suivant

port P_1 ; en mer ; port P_2 ; en mer

et que le programme du second navire soit

en mer ; port P_1 ; en mer ; port P_2

La (seule) façon de choisir les troncatures serait que le premier navire reste au port P_2 à partir du jour 3 et que le deuxième navire reste au port P_1 à partir du jour 2.

Pour chaque horaire, nous devons choisir un port d'escale : le port dans lequel le navire passera le reste du mois. Implicitement, ces ports d'escale définiront des troncatures des horaires. Nous dirons qu'une affectation de navires à des ports d'escale est acceptable si les troncatures qui en résultent satisfont aux conditions du problème - en particulier, à la condition ($\dagger$). (Notons qu'en raison de la condition ($\dagger$), chaque navire doit avoir un port d'arrêt distinct dans toute affectation acceptable).

Nous établissons un problème de mariage stable impliquant des navires et des ports. Chaque navire classe chaque port dans l'ordre chronologique de ses visites. Chaque port classe chaque navire dans l'ordre chronologique inverse de ses visites. Il ne nous reste plus qu'à démontrer :

Fait : une correspondance stable entre les navires et les ports définit une affectation acceptable des ports d'arrêt.

Preuve. Si l'affectation n'est pas acceptable, alors elle viole la condition ($\dagger$). C'est-à-dire qu'un navire S_i passe par le port P_k alors que le navire S_j s'y est déjà arrêté. Mais dans ce cas, en vertu de notre relation de préférence ci-dessus, le navire S_i "préfère" P_k à son port d'arrêt réel, et le port P_k "préfère" le navire S_i au navire S_j . Cela contredit l'hypothèse selon laquelle nous avons choisi une correspondance stable entre les navires et les ports.