

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi Ouzou



Faculté des sciences
Département de mathématiques

Mémoire

En vue de l'obtention du Diplôme de Master en Mathématiques
Option : Recherche Opérationnelle

Thème

Problème des mariages stables

Réalisé par :
Benchabane Karima
Kedir Cylia

Devant le jury composé de :

M ^r . M. Aouane	Président
M ^r . K. Kasdi	Examineur
M ^r . B. Sadi	Rapporteur

Promotion 2019-2020

Remerciements

Nous remercions Dieu, le miséricordieux d'avoir donné la force, la volonté, la santé et la patience pour accomplir ce travail.

Nous tenons à exprimer nos profondes gratitude et nos sincères remerciements à notre encadreur, M^r Sadi Bachir, pour ses conseils précieux, ses orientations judicieuses et ses critiques pertinentes.

Nos remerciements vont aussi aux membres de jury pour avoir accepté de prendre part au jury de soutenance et pour l'intérêt qu'ils portent à ce mémoire.

Nos plus vifs et chaleureux remerciements vont également à tous les enseignants qui nous ont suivi durant tout notre cursus depuis l'école fondamentale jusqu'à aujourd'hui.

Pour terminer, nous remercions nos frères, sœurs et nos très chers parents pour leur encouragement incessant durant toutes les étapes de notre vie.

Dédicaces

Je tiens très respectueusement à dédier ce modeste travail à :

Mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un honorable modèle de labeur et de persévérance, qu'ils trouvent ici l'expression de toute ma profonde reconnaissance et ma parfaite considération.

Mes très chers frères. Sans oublier tous mes amis(es), en particulier Yamina et Taous, ainsi que tous ceux qui nous ont apporté leurs soutiens pour élaborer ce travail.

Karima

Dédicaces

En signe de respect et de reconnaissance, je dédie ce modeste travail :

A ma très chère mère, qui m'a tout donnée, qui m'a soutenue par ces prières son amour et sa tendresse, qui a été toujours présente à mes cotées à tous les moments qui ont marqués ma vie et continue de l'être pour faire mon bonheur.

A mon très cher père, pour ses sacrifices et ses conseils, qui m'a encouragé à aller de l'avant tout au long de mes études.

A mon cher frère Yacine et mes chères sœurs Kahina, Sabrina et Ines, qui sont ma joie de vivre.

A mon binôme Karima pour sa vive compassion à ma réussite et surtout pour sa compréhension et sa patience.

A mes très chères Melida et Amira : en témoignage d'un amour spécial et de soutien permanent au cours des études, pour leur attachement et leurs chaleureux encouragements.

A tous mes amis et plus particulièrement les plus intimes, en témoignage des moments inoubliables, des sentiments sincères et des liens solides qui nous unissent.

A toute la promotion Recherche opérationnelle et tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail .

Cylia

Table des matières

Introduction générale	i
1 Définitions et concepts de base	1
1.1 Introduction	1
1.2 Définitions et notations	1
1.2.1 Graphe non orienté	1
1.2.2 Graphe orienté	2
1.2.3 Chaîne-cycle	3
1.2.4 Chemin-circuit	4
1.2.5 Noyau d'un graphe	4
1.3 Représentation matricielle d'un graphe	5
1.3.1 Matrice d'adjacence	5
1.3.2 Matrice d'incidence	5
1.3.3 Liste d'adjacence	6
1.4 Quelques graphes particuliers	7
1.4.1 Graphe simple	7
1.4.2 Graphe partiel et sous-graphe :	7
1.4.3 Graphe complet	8
1.4.4 Graphe complémentaire	8
1.4.5 Graphe connexe	9
1.4.6 Graphe régulier	9

1.4.7	Graphe biparti	9
1.4.8	Arbre	10
1.4.9	Graphe valué	10
1.4.10	Flot et réseau	10
1.4.11	Graphe résiduel	11
1.5	Couplage	11
1.6	Mise en ordre d'un graphe	12
1.7	Notions d'ordre dans les graphes	13
1.7.1	Relation de couverture	14
1.7.2	Diagramme	14
1.7.3	Chaîne et antichaîne	15
2	Mariage bi-latéral	17
2.1	Introduction	17
2.2	Préliminaires	18
2.3	Problème représentatif	20
2.4	Stabilité et existence des mariages stables	22
2.4.1	Algorithme de Gale-Shapley	22
2.4.2	Déroulement de l'algorithme sur un exemple	25
2.5	Autres mariages stables	26
2.5.1	Listes restreintes	26
2.5.2	Rotation	27
2.5.3	Mariage stable équitable	32
2.6	Le graphe mariage	33
2.6.1	Définitions	34
2.7	Graphe biparti et mariage stable	36
2.7.1	Algorithme pour la résolution du problème des mariages stables	37
2.7.2	Méthode des graphes résiduels	38
3	Problème des colocataires stables	41
3.1	Introduction	41

3.2	Définitions et préliminaires	42
3.3	Algorithme de résolution du problème des colocataires stables	44
3.3.1	Algorithme d'Irving	44
3.4	Enumération de tous les couplages stables	48
3.4.1	L'arbre de décision de l'algorithme d'Irving	48
3.4.2	La méthode d'énumération duale	53
3.5	La variante 3D du problème des colocataires stables	54
3.6	Le problème des luminaires stables	56
3.6.1	Préliminaires	56
3.6.2	Algorithme de résolution du problème SF	57
4	De la théorie à la pratique	60
4.1	Introduction	60
4.2	Modèles d'appariement de Shapley	61
4.2.1	Appariement avec transferts	61
4.2.2	Allocation des biens indivisibles	61
4.3	Exemples d'application	62
4.3.1	Transplantation de reins	62
4.3.2	Admission dans les écoles primaires, collèges, lycées et universités	66
4.4	Réalisation pratique	68
	conclusion générale	74
	Bibliographie	75

Introduction générale

La théorie des graphes constitue une branche de connaissances très importante, elle représente un outil appréciable qui a pris une large part au sein d'un ensemble très vaste de méthodes et techniques généralement regroupées sous l'appellation *Recherche opérationnelle* ou *mathématiques discrètes*. Elle doit son origine, au *XVIII^e* siècle, à *Leonhard Euler* dans son étude du problème des ponts de Königsberg.

Depuis le début du *XX^e* siècle, elle a connu un développement intense grâce aux travaux de König, Kuratowski, Menger, Caylay, et plus récemment, Claude Berge, Edrös et Harary. La théorie des graphes s'est développée dans diverses applications telles que, la chimie, la sociologie, l'économie, les réseaux de communication, les problèmes d'ordonnement des tâches et les problèmes d'affectation.

L'utilité de la théorie des graphes s'avère par le développement d'une modélisation qui ramène une grande variété de problèmes concrets à l'étude d'une structure particulière et simplificatrice constituée de points et de lignes sur une surface, qui déterminent en leur ensemble *un graphe*. Plus généralement, un graphe est une structure simple et efficace qui permet de représenter, modéliser et résoudre des problèmes de différents domaines.

Le contenu de ce travail repose sur l'étude d'un problème bien connu en informatique (Knuth, 1971) et en économie (Roth et Oliveira Sotomayor, 1990), qui est celui des mariages stables. Il a été présenté en 1962 par deux mathématiciens économistes, David Gale et Lloyd Shapley. Le mécanisme fondé sur la théorie des mariages a tenu un rôle prépondérant dans la résolution des problèmes d'appariement sous préférences. D'abord, nous verrons que ce dernier représente un problème d'affectation qui peut être résolu en sa forme clas-

sique en utilisant une classe particulière des graphes, qui est la classe des graphes bipartis, ainsi qu'une famille d'algorithmes dédiée à sa résolution. Ensuite, on fait une analyse plus profonde en présentant quelques généralisations et applications de ce problème.

Organisation du travail :

Notre travail commence par une introduction générale suivie de quatre chapitres.

Le premier chapitre est consacré à la définition des notions de bases concernant les graphes, afin de faciliter la compréhension du contenu de ce travail.

Au deuxième chapitre, on traite le problème des mariages stables, en donnant quelques notions et définitions relatives à ce thème, puis on passe à l'analyse de l'algorithme de Gale et Shapley, et la présentation de la notion des rotations, ainsi que l'utilisation des graphes mariages pour sa résolution, et on finit avec l'algorithme de Ford Fulkerson.

Au troisième chapitre, une variante du problème des mariages stables, appelée problème des colocataires stables est traitée.

Le quatrième chapitre, consiste en la présentation de quelques applications concrètes de la théorie des mariages qui montre sa grande utilité et efficacité colossale en pratique.

On finit par une conclusion générale récapitulative du travail.

Définitions et concepts de base

1.1 Introduction

Les graphes constituent un outil mathématique simple et puissant qui permet de modéliser et résoudre une grande variété de problèmes dans divers domaines. Dans ce chapitre, nous présentons quelques définitions fondamentales concernant les graphes qui seront utiles et nécessaires pour une bonne compréhension des chapitres suivants de ce mémoire (les concepts et les définitions sont tirés de [1] [3] [10]).

1.2 Définitions et notations

1.2.1 Graphe non orienté

Un graphe G est la donnée du couple (X, E) , où $X = \{ x_1, x_2, \dots, x_n \}$ est l'ensemble fini non vide des sommets de G , et $E = \{ e_1, e_2, \dots, e_m \}$ est l'ensemble fini de paires de sommets, appelé l'ensemble des arêtes de G . On le note $G = (X, E)$, ou également $G = (X(G), E(G))$.

Le nombre de sommets $|X|$ du graphe $G = (X, E)$ s'appelle l'ordre de G , son nombre d'arête $|E|$ définit sa taille.

Lorsque $e = xy \subset E$, on dit que e est l'arête de G d'extrémités x et y . Les sommets x et y sont alors adjacents dans G et incidents à l'arête e .

On note par $\Gamma(x) = \{y \in X / xy \in E\}$ l'ensemble des sommets adjacents (voisins) au sommet x , et $d(x) = |\Gamma(x)|$ son degré si G est simple. Deux arêtes avec les mêmes extrémités sont dites parallèles.

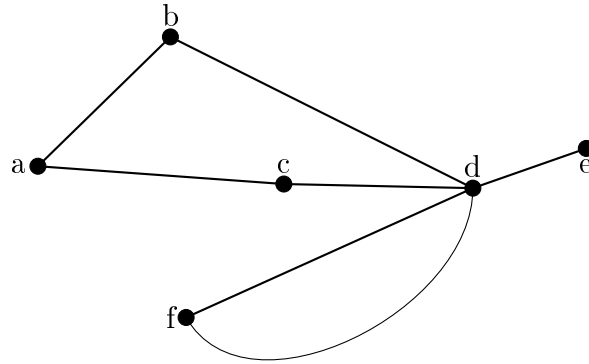


FIGURE 1.1 – Exemple d'un graphe non orienté

1.2.2 Graphe orienté

Un digraphe (ou graphe orienté) $G = (X, U)$ est le couple constitué d'un ensemble fini non vide $X = \{x_1, x_2, \dots, x_n\}$ de sommets, et d'un ensemble fini $U = \{\mu_1, \mu_2, \dots, \mu_m\}$ d'arcs. Un arc $\mu \in U$ est défini par une paire ordonnée de sommets. On le note $\mu = (x, y)$. Les sommets x et y représentent les extrémités de μ , où x est l'extrémité initiale et y est l'extrémité terminale de μ . On dit alors que y est le successeur de x et x est le prédécesseur de y .

On note par $\Gamma^+(x) = \{y / (x, y) \in U\}$ l'ensemble des successeurs de x , et

$\Gamma^-(x) = \{y / (y, x) \in U\}$ son ensemble des prédécesseurs.

La réunion de $\Gamma^-(x)$ et $\Gamma^+(x)$ définit l'ensemble des voisins de x , noté $\Gamma(x)$

$$\Gamma(x) = \Gamma^-(x) \cup \Gamma^+(x)$$

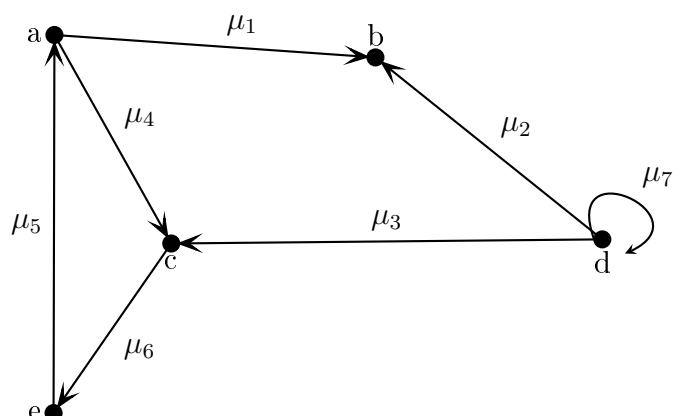


FIGURE 1.2 – Exemple d'un graphe orienté

Degré d'un graphe

C'est le degré maximum de ses sommets.

Une boucle

C'est une arête (resp.arc) reliant un sommet à lui même.

1.2.3 Chaîne-cycle

Une chaîne de $G = (X, E)$ est une séquence finie et alternée de sommets et d'arêtes, on la note par

$$C = \{x_1, e_1, x_2, e_2, \dots, x_{k-1}, e_{k-1}, x_k\}$$

Les sommets x_1 et x_k sont appelés les extrémités de la chaîne. Le nombre d'arêtes dans la chaîne définit sa longueur.

Lorsque les extrémités de C sont confondues, on dit que la chaîne est fermée et on l'appelle un cycle (le graphe sans cycle est dit acyclique).

1.2.4 Chemin-circuit

Un chemin de $G = (X, U)$ est une séquence finie ayant alternativement des sommets et des arcs, on le note par

$$C = \{x_1, \mu_1, x_2, \mu_2, \dots, x_{k-1}, \mu_{k-1}, x_k\}$$

Les sommets x_{k-1} et x_k sont les extrémités respectivement initiale et terminale de μ_{k-1} . x_1 et x_k sont les extrémités du chemin C .

Le chemin C est un circuit si x_1 et x_k sont confondues.

Remarques

- Un C . dont les sommets x_i sont distincts est dit *élémentaire* ;
 - Un C . qui ne passe pas plus d'une fois par une même arête (ou arc) est dit *simple*
- C . signifie (chaîne-cycle-chemin-circuit).

1.2.5 Noyau d'un graphe

Un noyau N d'un graphe orienté est formé d'un sous ensemble de ses sommets vérifiant :

1. $\forall x \in N, \Gamma_G^+(x) \cap N = \emptyset$
2. $\forall x \in X \setminus N, \Gamma_G^+(x) \cap N \neq \emptyset$

Ces conditions signifient que N est un stable : les sommets de N n'ont aucun arc les joignant deux à deux, de plus, quelque soit x sommet en dehors de N , x a au moins un successeur dans N , et N est dit alors absorbant.

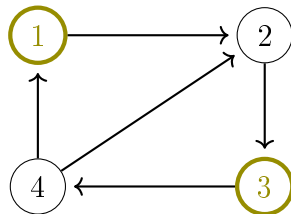


FIGURE 1.3 – $G = (X, U)$ graphe orienté, les sommets 1 et 3 sont des éléments du noyau de G

1.3 Représentation matricielle d'un graphe

Soit $G = (X, U)$ un graphe orienté avec $|X| = n$ et $|U| = m$.

1.3.1 Matrice d'adjacence

La matrice d'adjacence de G est une matrice carrée d'ordre n . On la note par $A(G) = (a_{i,j})_{i,j=1,\dots,n}$, où (i, j) désigne l'intersection de la ligne i et la colonne j . Les lignes et les colonnes de $A(G)$ représentent les sommets de G , telles que

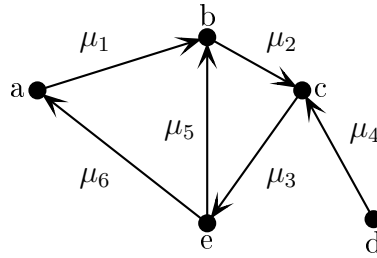
$$a_{ij} = \begin{cases} 1 & \text{si } (x_i, x_j) \in U \\ 0 & \text{si non.} \end{cases}$$

1.3.2 Matrice d'incidence

La matrice d'incidence de G est une matrice de taille $(n \times m)$, où ses lignes et ses colonnes représentent respectivement les sommets et les arcs du graphe G .

Soit $M(G) = (I_{ij})_{\substack{i=1,\dots,n \\ j=1,\dots,m}}$ cette matrice, telle que

$$I_{ij} = \begin{cases} +1 & \text{si } x_i \text{ est l'extrémité initiale de } \mu_j \\ -1 & \text{si } x_i \text{ est l'extrémité terminale de } \mu_j \\ 0 & \text{si non.} \end{cases}$$



$$A(G) = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$M(G) = \begin{matrix} & \begin{matrix} \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_6 \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} +1 & 0 & 0 & 0 & 0 & -1 \\ -1 & +1 & 0 & 0 & -1 & 0 \\ 0 & -1 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & -1 & 0 & +1 & +1 \end{pmatrix} \end{matrix}$$

FIGURE 1.4 – Graphe orienté, sa matrice d’adjacence $A(G)$ et sa matrice d’incidence $M(G)$

1.3.3 Liste d’adjacence

Une représentation par liste d’adjacence d’un graphe orienté sert à construire une liste d’adjacence des arcs issus d’un sommet donné. Cette structure est formée alors de deux champs : un sommet et une liste chaînée des sommets adjacents à celui-ci.

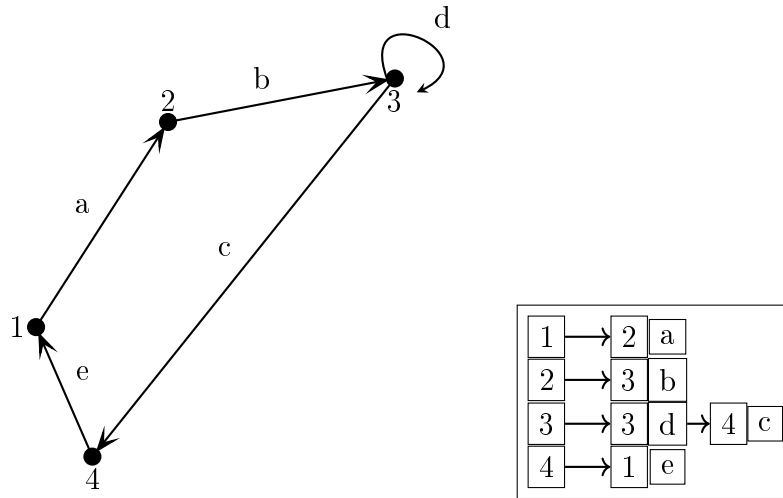


FIGURE 1.5 – Graphe orienté et sa liste d'adjacence

1.4 Quelques graphes particuliers

1.4.1 Graphe simple

Un graphe $G = (X, E)$ est simple s'il ne contient ni de boucle ni d'arêtes parallèles. Si, entre deux sommets, il existe plusieurs arêtes ou une arête qui relie un sommet à lui-même, le graphe est alors un *multigraphe*.

1.4.2 Graphe partiel et sous-graphe :

On appelle graphe *partiel* d'un graphe $G = (X, U)$, tout graphe $H = (X, V)$, avec $V \subseteq U$.

Le *sous-graphe* de $G = (X, U)$ est le graphe $G_A = (A, U')$, avec $A \subseteq X$ et $U' = \{\mu = (x, y) \in U / x \in A \text{ et } y \in A\}$

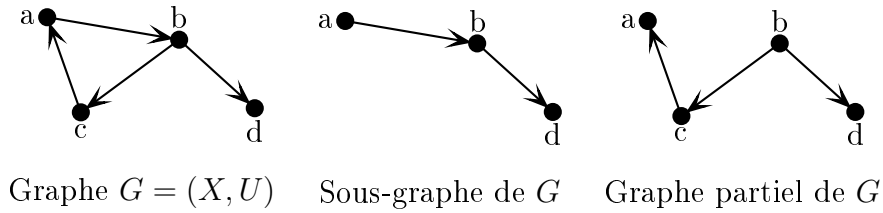
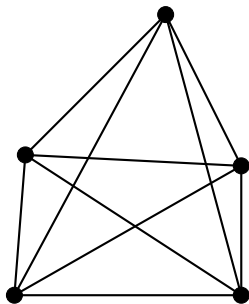


FIGURE 1.6 –

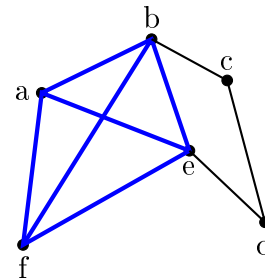
1.4.3 Graphe complet

Un graphe $G = (X, E)$ est complet si tous ses sommets sont deux à deux adjacents, i.e $\forall x, y \in X, \exists e = xy \in E$. Un graphe complet d'ordre n est noté K_n .

Un sous graphe d'un graphe complet est appelé *clique*, elle est maximale si elle n'est pas contenue dans une autre. On note le cardinal d'une plus grande clique de G par $\omega(G)$



Graphe complet



$C = \{ a, b, e, f \}$ clique maximale de G

FIGURE 1.7 –

1.4.4 Graphe complémentaire

Le graphe complémentaire d'un graphe $G = (X, E(G))$ est le graphe \bar{G} défini par $X(G) = X(\bar{G})$, et $\forall x, y \in X(\bar{G}), xy \in E(\bar{G})$ si et seulement si $xy \notin E(G)$.

Une clique dans $G = (X, E(G))$ est un *stable* dans $\bar{G} = (X, E(\bar{G}))$, autrement $S \subset X$ est *stable* de G si le sous-graphe engendré par S est vide d'arête.

1.4.5 Graphe connexe

Un graphe $G = (X, E)$ est connexe si $\forall x, y \in X$, il existe une chaîne entre x et y . Sur l'ensemble X des sommets de G , on construit ainsi une relation d'équivalence. Une classe d'équivalence de G correspond à une composante connexe, qui représente également un sous-graphe de G qui est connexe. Un graphe non connexe se décompose en au moins deux composantes connexes.

1.4.6 Graphe régulier

Un graphe $G = (X, E)$ est *régulier* de degré k , si $\forall x \in X, d(x)=k$. Ce graphe est dit *k-régulier*.

1.4.7 Graphe biparti

Un graphe $G = (X, E)$ est biparti s'il existe une partition de ses sommets en deux ensembles X_1 et X_2 de façon à ce que toute arête ait une extrémité dans X_1 et une extrémité dans X_2 , et deux éléments de X_1 (resp. X_2) sont deux à deux non adjacents, une telle partition (X_1, X_2) est appelée *bipartition* du graphe. Nous désignons ce graphe biparti par $G = (X_1, X_2, E)$ [ou également par $G = (X_1 \cup X_2, E)$].

Si tout sommet de X_1 est relié à tout sommet de X_2 , alors G est *biparti complet*.

Un graphe biparti complet avec $|X_1| = p$ et $|X_2| = q$ est noté $K_{p,q}$.

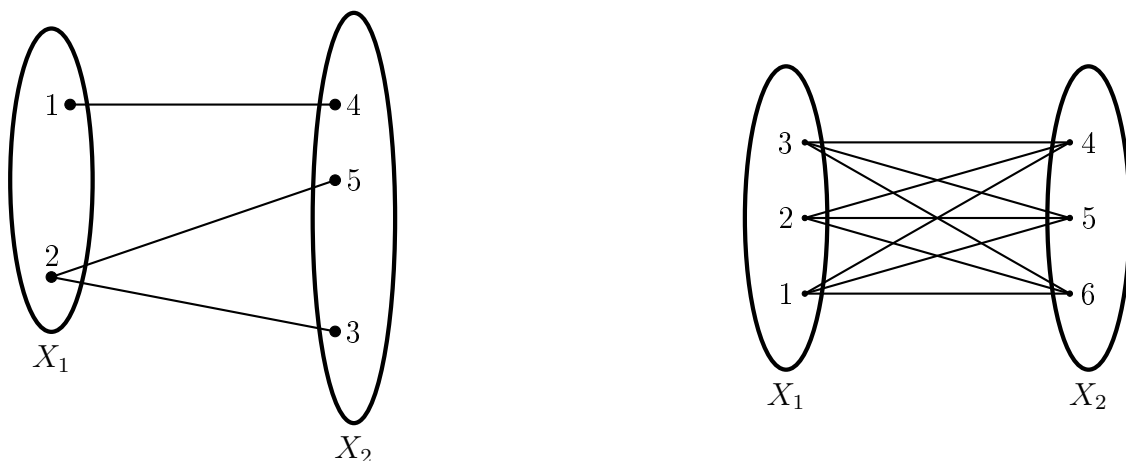


FIGURE 1.8 – Graphe biparti (non complet), et graphe biparti complet $K_{3,3}$

1.4.8 Arbre

Un graphe $G = (X, E)$ est *un arbre* s'il est connexe et sans cycle (acyclique).

Un sommet de degré 1 dans un arbre est appelé *feuille*, ou sommet pendant.

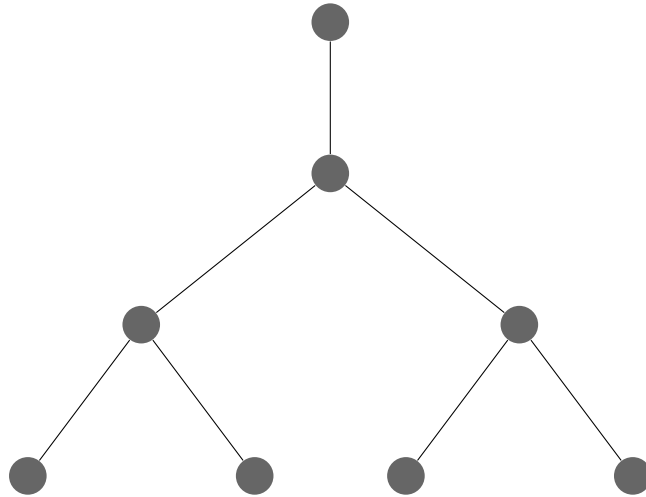


FIGURE 1.9 – Arbre

Dans un digraphe $G = (X, U)$, $r \in X$ est appelé *racine* si $\forall x \in X$, il existe un chemin de r à x . Un arbre enraciné en r est une orientation d'un arbre qui possède r comme racine.

Une *branche* de $x \in X$ est un chemin partant de x et allant jusqu'à une feuille.

1.4.9 Graphe valué

Un graphe orienté $G = (X, U)$ (resp. un graphe non orienté $G = (X, E)$) est dit graphe valué s'il est muni d'une fonction $\gamma : U \rightarrow \mathbb{R}$ (resp. $\gamma : E \rightarrow \mathbb{R}$).

La fonction γ est souvent appelée fonction *coût*.

1.4.10 Flot et réseau

Un *réseau* est un graphe valué $G = (X, U, \gamma)$ dans lequel il y a un unique sommet s , appelé *source* tel que $d^-(s) = 0$, et un unique sommet p , appelé *puits* tel que $d^+(p) = 0$.

Pour $\mu \in U$, $\gamma(\mu)$ représente la *capacité* de l'arc μ .

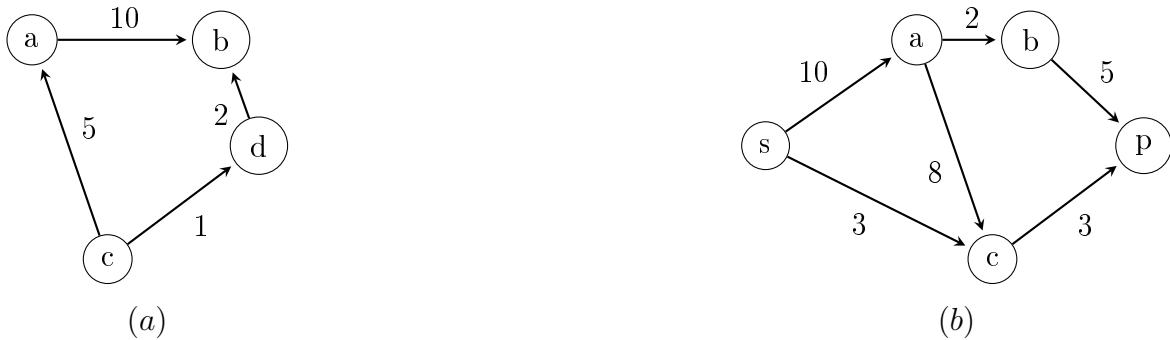


FIGURE 1.10 – (a) Graphe valué, (b) Réseau

Un *flot* f pour un réseau $G = (X, U, \gamma)$, est une fonction $f : U \rightarrow \mathbb{R}$ telle que :

1. $\forall \mu \in U, 0 \leq f(\mu) \leq \gamma(\mu)$ [$f(\mu)$ est appelé *flux*];
2. La somme des flux qui sortent de s est égale à la somme des flux qui entrent en p ;
3. En chaque sommet intermédiaire, la somme des flux qui entrent est égale à la somme des flux qui sortent.

Un flot qui vérifie 1), 2) et 3) est dit *flot réalisable*.

1.4.11 Graphe résiduel

Un graphe résiduel est composé de l'ensemble des sommets X du réseau $G = (X, U)$, et pour chaque arc $\mu \in U$, il existe

- un arc μ de capacité $\gamma(\mu) - f(\mu)$ si $\gamma(\mu) - f(\mu) > 0$, appelé la capacité résiduelle de l'arc d'origine;
- un arc retour μ de capacité $f(\mu)$ si $f(\mu) > 0$.

1.5 Couplage

Soit le graphe $G = (X, E)$.

Un *couplage* M de G est un sous-graphe partiel régulier de degré 1; en d'autres termes, M est un ensemble d'arêtes deux à deux non-adjacentes. On dit que M *sature* un sommet x si $d_M(x) = 1$, et que M *sature* $S \subseteq M$, si $\forall x \in S, M$ *sature* x .

Dans le cas où $S = X$, M est dit *couplage parfait*.

- M est un couplage maximum si $|M| \geq |C|, \forall C$ couplage de G ;
- M est maximal si $M \cup \{e\}; e \in E \setminus M$, n'est pas un couplage.

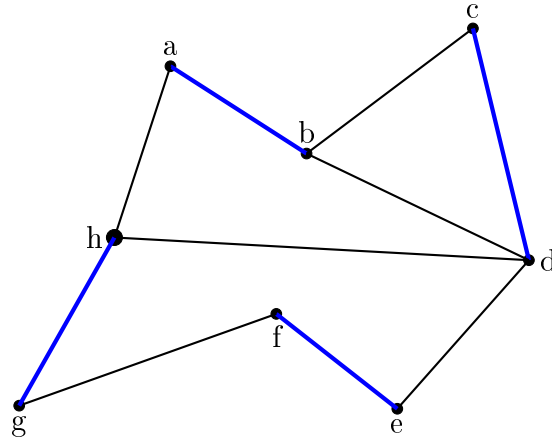


FIGURE 1.11 – $\{ab, cd, ef, gh\}$ est un couplage parfait

1.6 Mise en ordre d'un graphe

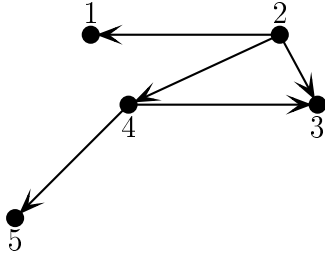
Soit $G = (X, U)$ un graphe sans circuit. On appelle *dictionnaire* de G , le couple $(x, \Gamma^-(x)), \forall x \in X$.

Ordonner le graphe G consiste à le représenter dans le plan de sorte que tous ses arcs aient le même sens d'orientation. Cela se fait en exploitant le dictionnaire de G .

On organise l'ensemble des sommets de G en *niveaux* en utilisant son dictionnaire, comme suit :

- Niveau 0 : $N_0 = \{x \in X / \Gamma^-(x) = \emptyset\}$
- Niveau 1 : $N_1 = \{x \in X / \Gamma^-(x) \subseteq N_0\}$
- Niveau 2 : $N_2 = \{x \in X / \Gamma^-(x) \subseteq N_0 \cup N_1\}$
- ⋮
- Niveau k : $N_k = \{x \in X / \Gamma^-(x) \subseteq \bigcup_{i=0}^{k-1} N_i\}$

Exemple



x	$\Gamma^-(x)$
1	2
2	/
3	2, 4
4	2
5	4

$$N_0 = \{2\}, N_1 = \{1, 4\}, N_3 = \{3, 5\}$$

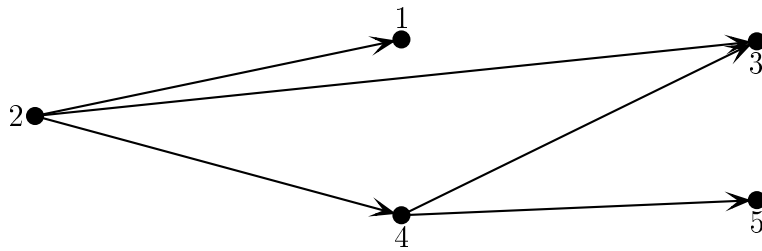


FIGURE 1.12 – Graphe ordonné

1.7 Notions d'ordre dans les graphes

Définition

Un ensemble ordonné est un couple $P = (X, \preceq_p)$ constitué d'un ensemble X ($X \neq \emptyset$) et d'une relation binaire sur les éléments de X , notée \preceq_p qui vérifie :

1. La réflexivité : $x \preceq_p x, \forall x \in X$
 2. L'antisymétrie : $x \preceq_p y$ et $y \preceq_p x \implies x = y, \forall x, y \in X$
 3. La transitivité : $x \preceq_p y$ et $y \preceq_p z \implies x \preceq_p z, \forall x, y, z \in X$
- \preceq_p , vérifiant 1) 2) et 3), est une relation d'ordre.

Deux éléments x et y sont dits *comparables* si $x \preceq_p y$ ou $y \preceq_p x$.

Dans le cas inverse, x et y seront dits *incomparables* ; ils sont notés $x \parallel y$, le couple (x, y) est alors appelé *un saut*.

Si $\forall x, y \in X$, $x \preceq_p y$ ou $y \preceq_p x$, on dit que la relation d'ordre est *totale* et P est un *ensemble totalement ordonné* (linéairement ordonné).

Dans le cas où P a au moins un couple d'éléments incomparables, la relation d'ordre est alors *partielle*, et l'ensemble P est dit un *ensemble partiellement ordonné* ou *poset* (partially ordered set).

Si $x \preceq_p y$ et $x \neq y$, la relation d'ordre est dite *stricte*, elle est irreflexive et transitive. On la note \prec_p .

1.7.1 Relation de couverture

Soit P un ensemble ordonné et $x, y \in P$.

On dit que x est *couvert* par y (ou y *couvre* x), et on écrit $x < y$, s'il n'existe pas un élément $z \in P$ tel que $x \preceq_p z \preceq_p y$. Le couple $(x, y) \in X^2$ est appelé *couple de couverture* ou *arc de couverture*. Dans ce cas x et y sont dits *directement comparables*.

La relation de comparabilité est *indirecte* s'il existe une séquence finie de relations de couverture $x = x_0 < x_1 < x_2 < \dots < x_n = y$.

1.7.2 Diagramme

Soit P un ensemble ordonné. Il est possible de représenter P par une configuration constituée d'un ensemble de points qui correspondent aux éléments de P et d'un ensemble d'arêtes.

On appelle cette configuration *diagramme*. Elle est construite comme suit :

1. On associe à chaque sommet $x \in X$, $p(x)$, sa position dans le plan ;
2. A chaque paire de couverture $x < y$ dans P , on aura $p(x) < p(y)$ et on trace une arête joignant $p(x)$ à $p(y)$, telle que la position de x dans le plan est en dessous de celle de y .

Le *diagramme de Hasse* d'un poset est un graphe dont les sommets sont les éléments de P et les arêtes sont les relations de couvertures dans P .

1.7.3 Chaîne et antichaîne

Un ensemble ordonné dont les éléments sont totalement ordonnés s'appelle *une chaîne*, elle est *maximale* si elle n'est pas contenue dans une autre.

Si, maintenant, les éléments de l'ensemble ordonné sont deux à deux non comparables alors ce dernier forme *une antichaîne*. Une antichaîne est maximale si elle n'est pas contenue dans une autre.

Exemple 1.1. Soit $P=(X, \preceq_p)$ tel que, $X = \{1, 2, 3, 4, 5, 6, 7\}$;

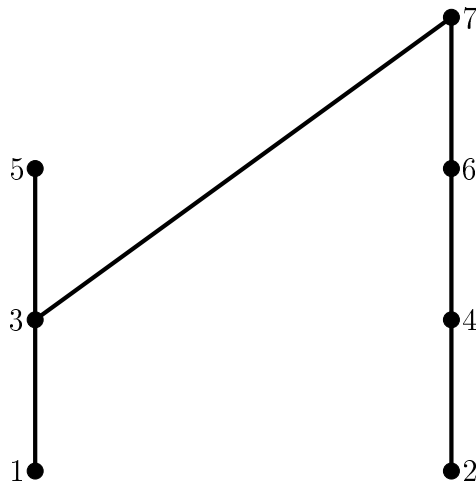


FIGURE 1.13 – diagramme

$(1,2), (1,4), (1,6), (2,3), (2,5), (3,4), (3,6), (5,6), (5,7)$ sont les sauts de P .

$[1\ 3\ 5]$ $[1\ 3\ 7]$ sont des chaînes maximales.

$[1\ 3]$ est une chaîne qui est contenue dans une autre chaîne qui est maximale.

$[2\ 4\ 6\ 7]$ est une chaîne maximum.

$[1\ 2]$, $[2\ 5]$, $[1\ 4]$, $[5\ 6]$ sont des antichânes maximales.

$(1,3)$, $(3,5)$, $(2,4)$, $(4,6)$, $(6,7)$, $(3,7)$ sont des couples de couverture de P .

Mariage bi-latéral

2.1 Introduction

On considère un groupe d'individus composé de n hommes et de n femmes en âge de se marier (mariage monogame) qui désirent se mettre en couple chacun avec son conjoint préféré. Néanmoins, ces individus n'ont pas forcément les mêmes préférences, d'où la difficulté de les appairer de manière à ce que tout le monde soit satisfait et que les couples formés ne risquent pas de se séparer. Le but est alors de trouver une façon stable pour les mettre en couple de telle sorte que personne n'ait envie d'échanger son partenaire. Cette situation décrite ci-dessus est connue sous le nom de problème de mariage stable.

En 1962, deux mathématiciens économistes David Gale et Lloyd Shapley ont prouvé qu'il existe toujours une solution stable quelque soit les préférences des individus. Ils ont également construit un algorithme appelé algorithme de Gale-Shapley permettant de trouver cette solution qui représente un couplage stable.

Dans ce chapitre, on va présenter, détailler et écrire formellement des informations sur le problème de mariage stable. On commence d'abord par définir quelques notions nécessaires et utiles pour une bonne compréhension.

2.2 Préliminaires

Considérons un exemple de mariage stable avec n hommes et n femmes.

Soient :

$H = \{h_1, h_2, \dots, h_n\}$ l'ensemble des n hommes ;

$F = \{f_1, f_2, \dots, f_n\}$ l'ensemble des n femmes.

On cherche à marier ses membres afin d'obtenir des mariages stables et parfaits.

A chaque individu, on associe un ordre total sur l'ensemble du sexe opposé.

Cet ordre représente ses préférences qui sont données sous forme d'une liste ordonnée. On le note par $\succeq_{p \in H \cup F}$.

Pour tout homme h , on note $P(h)$, sa *liste des préférences* sur l'ensemble $F \cup \{h\}$, qui peut être représentée sous la forme suivante :

$$P(h) = f_1, f_2, [f_3, f_5], h, \dots, f_n$$

Le premier choix de l'homme h est d'être marié à la femme f_1 , qu'il préfère à son deuxième choix qui est d'être marié à la femme f_2 , qu'il préfère aussi à son troisième choix qui est *indifférent* entre f_3 et f_5 (un individu est indifférent entre deux autres s'il peut choisir l'un ou l'autre), et qu'il préfère rester célibataire plutôt que d'épouser quelqu'un d'autre [12].

De même, chaque femme f dans F a une liste de préférences, $P(f)$, sur l'ensemble $H \cup \{f\}$, qui peut être sous forme identique à celle présentée par l'homme h (cela en remplaçant h_i par f_i), ou sous une forme simple sans l'indifférence :

$$P(f) = h_3, h_2, \dots, h_n$$

On note P l'ensemble des listes des préférences de tous les individus, avec

$$P = \{P(h_1), P(h_2), \dots, P(h_n), P(f_1), P(f_2), \dots, P(f_n)\}$$

On écrit

- $f \succ_h f'$ pour signifier que h préfère f à f' , et $f \succeq_h f'$ pour signifier que h préfère f au moins autant que f' .
- $h \succ_f h'$ pour signifier que f préfère h à h' , et $h \succeq_f h'$ pour signifier que f préfère h au moins autant que h' .

Un(e) homme/femme acceptable

- Une femme f est *acceptable* pour l'homme h si : $f \succeq_h h$.
- Un homme h est *acceptable* pour la femme f si : $h \succeq_f f$.

Une paire (h, f) est dite *acceptable* si $f \in P(h)$ et $h \in P(f)$, si non elle est dite *inacceptable*.

Meilleur(e) partenaire

- Pour une femme f : c'est l'*unique homme acceptable* de f , noté $h^* \in H$ tel que : $h^* \succ_f h, \forall h \in H$.
- Pour un homme h : c'est l'*unique femme acceptable* de h , notée $f^* \in F$, telle que : $f^* \succ_h f, \forall f \in F$.

Pire partenaire

- C'est l'*unique femme acceptable* de h , notée f_* telle que : $f_* \prec_h f, \forall f \in F$.
- C'est l'*unique homme possible* de f , noté h_* tel que : $h_* \prec_f h, \forall h \in H$.

Femme ou homme possible

On dit qu'une femme, ou un homme est possible pour l'individu de l'autre sexe, s'il existe au moins un couplage stable pour lequel cette femme ou cet homme sont en couple. Si non on dit que chacun est impossible pour l'autre.

Un couplage M

Un couplage M est une application injective entre H et F , en d'autres termes, c'est l'ensemble des paires acceptables homme-femme, tel que chaque homme et chaque femme appartient au plus à une seule paire de M .

$(h, f) \in M$ signifie que l'homme h et la femme f sont partenaires dans M , formellement $M(h) = f$ et $M(f) = h$.

Un couplage parfait : M est parfait si c'est une bijection entre H et F : chaque homme et chaque femme apparaissent dans exactement une paire de M .

Couple instable (paire bloquante)

Une paire $(h, f) \notin M$ est dite couple instable, si elle vérifie les conditions suivantes :

1. h et f ne sont pas partenaires dans M ;
2. h préfère f à $M(h)$: $f \succ_h M(h)$;
3. f préfère h à $M(f)$: $h \succ_f M(f)$.

Couplage stable : Un couplage M est stable s'il ne contient pas de couple instable (dans le cas contraire ce couplage est dit instable).

Remarque

Pour la suite de notre travail on suppose que les préférences sont strictes, et on note $\succ_{p \in H \cup F}$.

2.3 Problème représentatif

Soit $\delta = (H, F, (\succ_p)_{p \in H \cup F})$ une instance du problème de mariage stable. On appelle graphe associé à δ le graphe biparti complet $G = (H, F, E)$ noté $K_{n,n}$ avec la bipartition H et F , où ;

$$H = \{h_1, h_2, \dots, h_n\}$$

$$F = \{f_1, f_2, \dots, f_n\}$$

$$E = \{(h_i, f_j) / i = 1 \dots n \text{ et } j = 1 \dots n\}$$
 l'ensemble des arêtes de G .

Le but est d'associer, deux à deux, les sommets de chacune des parties de G en tenant compte des préférences des sommets.

Soient $L=(l_{ij})$ et $R=(r_{ij})$ deux matrices carrées d'ordre n dont les lignes et les colonnes représentent des permutations des nombres de 1 à n .

Notre objectif est d'utiliser ces deux matrices pour ordonner les individus de H et F dans le but de trouver un couplage stable parfait de G tel que, un individu h_i ordonne tous les éléments de F , comme suit ;

la femme f_j est classée l_{ij} sur la liste de h_i dans la matrice L . De même pour les éléments de H qui se classent ainsi dans la matrice R .

$$\mathbf{L} = \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 \\ \begin{matrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{matrix} & \begin{pmatrix} 1 & 2 & 5 & 7 & 6 & 8 & 4 & 3 \\ 4 & 1 & 7 & 2 & 3 & 6 & 5 & 8 \\ 3 & 1 & 4 & 5 & 2 & 6 & 7 & 8 \\ 4 & 3 & 5 & 2 & 1 & 6 & 8 & 7 \\ 3 & 2 & 1 & 5 & 7 & 8 & 6 & 4 \\ 2 & 8 & 6 & 7 & 3 & 5 & 4 & 1 \\ 4 & 8 & 5 & 6 & 3 & 2 & 1 & 7 \\ 3 & 2 & 4 & 5 & 6 & 7 & 1 & 8 \end{pmatrix} \end{matrix} \quad \mathbf{R} = \begin{matrix} & h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 & h_8 \\ \begin{matrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{matrix} & \begin{pmatrix} 5 & 6 & 1 & 3 & 4 & 2 & 7 & 8 \\ 2 & 6 & 5 & 7 & 3 & 1 & 8 & 4 \\ 8 & 1 & 2 & 6 & 7 & 4 & 3 & 5 \\ 1 & 7 & 2 & 6 & 5 & 4 & 8 & 3 \\ 2 & 5 & 3 & 7 & 1 & 8 & 6 & 4 \\ 7 & 3 & 6 & 2 & 4 & 5 & 8 & 1 \\ 2 & 5 & 3 & 8 & 1 & 4 & 7 & 6 \\ 3 & 1 & 7 & 2 & 4 & 6 & 8 & 7 \end{pmatrix} \end{matrix}$$

FIGURE 2.1 – Représentation d'un exemple du problème des mariages stables par les matrices des préférences

2.4 Stabilité et existence des mariages stables

2.4.1 Algorithme de Gale-Shapley

Théorème d'existence d'un couplage stable [6]

Quelque soit la configuration des préférences P , il existe toujours une solution stable (couplage stable) pour P (Gale et Shapley 1962).

Preuve :

La preuve de ce théorème est constructive, elle repose sur un algorithme itératif construit par les deux mathématiciens, *David Gale* et *Lloyd Shapley*.

Nous appelons cet algorithme '*algorithme de Gale-Shapley*', il fonctionne par *la procédure d'acceptation différée*

Principe de la procédure d'acceptation différée

Au début de l'algorithme, tous les individus sont célibataires.

La procédure se fait par étapes :

1. Chaque individu révèle ses préférences et les classe du préféré au moins préféré selon un ordre strict.
2. Pour commencer, chaque homme non engagé propose un mariage à la première femme de sa liste de préférences des femmes acceptables.
3. La femme qui reçoit la proposition
 - soit accepte temporairement la proposition la plus préférée parmi celles qu'elle a reçues, et rejette les autres ;
 - soit rejette définitivement toute proposition si elle a déjà reçu celle de son meilleur partenaire.
4. Chaque homme dont la proposition n'est pas rejetée est maintenu engagé.
5. Chaque homme dont la proposition est rejetée propose à nouveau un mariage à la femme qu'il préfère parmi celles qui ne l'ont pas encore rejeté, sans prendre en considération si elle est déjà en couple

Le processus se termine lorsque les acceptations temporaires deviennent définitives, c'est-à-dire tous les individus sont engagés.

Algorithme 1 mariage stable (orienté homme)

Entrée

H := l'ensemble des hommes

F := l'ensemble des femmes

P := l'ensemble des préférences des individus

Sortie M := mariage stable parfait

tant que il existe $h \in H$ qui n'a pas encore proposé de mariage **faire**

f := la femme suivante sur la liste des préférences de h pour laquelle Il n'a pas encore proposé

h propose à f

si f est libre **alors**

f accepte la proposition de h

si non

h' := l'homme avec lequel f est engagée

fin si

si f préfère h' à h **alors**

f refuse la proposition de h

si non

f accepte la proposition de h

fin si

$M := \{(h, f) \in H \times F / f \text{ accepte la proposition de } h\}$

fin tant que

Proposition 2.1. (Gale et Shapley 1962) L'algorithme de la recherche des mariages stables est de complexité $o(n^2)$.

Preuve 2.1. Au pire des cas, chaque homme propose un mariage à chacune des femmes, ce qui constitue au total n^2 propositions possibles, et chaque proposition est de complexité $o(1)$. Par conséquent, la complexité de cet algorithme est en $o(n^2)$.

Proposition 2.2. Deux implications fondamentales de cette séquence de propositions sont :

- (i) si h propose un mariage à f , alors il n'y a pas de couplage stable dans lequel h a une meilleure femme que f .
- (ii) si f reçoit une proposition de la part de h , alors il n'y a pas de couplage stable dans lequel f a un pire partenaire que h .

Définition 2.1. On dit qu'un couplage M est optimal pour les hommes (ou H -optimal), si $\forall h \in H$, h est apparié avec son partenaire possible le plus préféré $f \in F$.

D'autre part, le couplage M est moins optimal pour les femmes si $\forall f \in F$, f est relié avec le partenaire possible le moins préféré $h \in H$.

Théorème 2.1. (Gale-Shapley 1962) Le couplage M obtenu par la procédure d'acceptation différée est un couplage stable parfait.

Preuve 2.2. Montrons que M est stable : on suppose que M n'est pas stable, c'est à dire qu'il existe un couple instable qu'on note $(h, f) \notin M$, avec $h \succ_f M(f)$ et $f \succ_h M(h)$. L'homme h préfère la femme f à son partenaire respectif dans M , alors il aurait demandé f à un moment antérieur avant de proposer un mariage à $M(h)$. Si h a demandé un choix moins préféré que f , c'est parce qu'elle lui a déjà rejeté, or on a supposé que f préfère h à $M(f)$ (contradiction). Donc M est stable.

Montrons que M est parfait : supposons qu'il existe un homme h non engagé, ce qui signifie que toute femme $f \in F$ lui est refusée. Une femme qui refuse h doit être déjà en couple, alors toutes les femmes sont engagées, ce qui ne peut être possible que si tous les hommes sont engagés, donc h est engagé. Ce qui contredit notre supposition.

Propriété 2.1. (Gale-Shapley 1962) L'algorithme de Gale-Shapley construit l'unique couplage stable optimal pour les hommes.

Preuve 2.3. *On suppose, par l'absurde, que le couplage obtenu par la procédure d'acceptation différée n'est pas optimal pour les hommes. Alors il existe au moins un homme rejeté par sa femme possible préférée, soit h l'homme rejeté par la femme f . Comme f a refusé h , elle doit être déjà apparié avec un homme h' qu'elle préfère à h . Puisque f est une femme possible à h , alors il existe un couplage stable M qui contient le couple (h, f) . Dans ce couplage, f préfère h' à $M(f) = h$, et, h' préfère f à $M(h')$, ce qui contredit la stabilité de M .*

Propriété 2.2. *(Gale-Shapley 1962) L'algorithme de Gale-Shapley construit l'unique couplage stable le moins optimal pour les femmes.*

Preuve 2.4. *Supposons, par contradiction, que le couplage M obtenu par la procédure d'acceptation différée est optimal pour les hommes, mais pas moins optimal pour les femmes. Alors il existe une femme f appariée avec un homme h , et un homme h' possible pour f , tel que $h \succ_f h'$. Comme h' est possible pour f , alors il existe un autre couplage M' dans lequel h' et f sont en couple, or f préfère h à h' , de plus, puisque M est un couplage optimal pour les hommes, h doit strictement préférer f à $M'(h)$ (i.e. ; h et f se préfèrent). Donc le couplage M' ne peut pas être stable, ce qui est une contradiction.*

Théorème 2.2. *Si les hommes sont les proposants sous l'algorithme d'acceptation différée, alors ce dernier est non manipulable par les hommes.*

Preuve 2.5. *L'algorithme de Gale Shapley assure le choix optimal pour les hommes (les proposants), alors s'ils manipulent leurs choix ils ne seront affectés qu'à des femmes moins préférées que celles qu'ils peuvent détenir s'ils révèlent les vraies préférences.*

2.4.2 Déroulement de l'algorithme sur un exemple

On reprend l'exemple de la figure 2.1 - qui comporte un ensemble de 8 hommes et 8 femmes avec leurs préférences.

Au départ tous les individus sont célibataires, alors les hommes commencent à proposer des mariages aux femmes :

1. L'homme h_1 propose un mariage à sa meilleure femme f_1 . Comme f_1 est libre, alors elle accepte la proposition de h_1 .
2. L'homme h_2 propose un mariage à sa meilleure femme f_2 . f_2 est libre alors elle accepte cette proposition ;
3. L'homme h_3 propose un mariage à la femme f_2 , or f_2 est déjà en couple avec l'homme h_2 , donc la femme f_2 est devant deux choix. Pour choisir, on considère la liste des préférences de la femme f_2 . On remarque que $h_3 \succ_{f_2} h_2$, alors la femme f_2 préfère se mettre avec l'homme h_3 que de rester avec l'homme h_2 , par conséquent h_3 sera apparié à la femme f_2 et h_2 sera libre ;
4. L'homme h_2 propose à son choix suivant qui est la femme f_4 ;
5. On poursuit la même procédure jusqu'à trouver le couplage stable parfait qu'on cherche.

Le couplage stable parfait obtenu par l'algorithme de Gale-Shapley sur cet exemple est :

$$(1,1), (2,4), (3,2), (4,5), (5,3), (6,8), (7,6), (8,7)$$

Il est optimal pour les hommes et moins optimal pour les femmes.

2.5 Autres mariages stables

2.5.1 Listes restreintes

D'après la *proposition 2.2*, on doit explicitement retirer h de la liste des préférences de f , et f de celle de h , si et seulement si f reçoit une proposition de quelqu'un qu'elle préfère à h . Soit h_i l'homme engagé à f_j , alors f_j retire tout homme h_k classé moins préféré que h_i de sa liste des préférences, également tout individu h_k retire f_j de sa liste des préférences.

On appelle les listes qui en résultent *les listes restreintes*

Exemple 2.1. *Après l'application de la procédure d'acceptation différée sur l'exemple précédent, on aura les listes restreintes suivantes :*

Par souci de simplification, on suppose que les individus sont numérotés de 1 à 8.

$h_1 : 1\ 2\ 8\ 7\ 5\ 4\ 6$	$f_1 : 3\ 6\ 4\ 5\ 1$
$h_2 : 4\ 5\ 7\ 6\ 3\ 8$	$f_2 : 6\ 1\ 5\ 8\ 3$
$h_3 : 2\ 5\ 1\ 3\ 4\ 6\ 7$	$f_3 : 2\ 3\ 7\ 6\ 8\ 4\ 5$
$h_4 : 5\ 4\ 1\ 3\ 6\ 8$	$f_4 : 1\ 3\ 8\ 6\ 5\ 4\ 2$
$h_5 : 3\ 2\ 1\ 8\ 4\ 7\ 5\ 6$	$f_5 : 5\ 1\ 3\ 8\ 2\ 7\ 4$
$h_6 : 8\ 1\ 7\ 6\ 3\ 4\ 2$	$f_6 : 8\ 4\ 2\ 5\ 6\ 3\ 1\ 7$
$h_7 : 6\ 5\ 3$	$f_7 : 5\ 1\ 3\ 6\ 2\ 8$
$h_8 : 7\ 2\ 3\ 4\ 5\ 6\ 8$	$f_8 : 2\ 4\ 1\ 5\ 8\ 6$
<i>listes restreintes des hommes</i>	<i>listes restreintes des femmes</i>

Propriété 2.3. *si f n'apparaît pas dans la liste restreinte de h , alors il n'y a pas de couplage stable dans lequel h et f sont partenaires.*

Propriété 2.4. *f figure sur la liste restreinte de h si et seulement si h figure sur celle de f , et est le premier sur la liste restreinte de h si et seulement si h est le dernier sur celle de f .*

2.5.2 Rotation

On aborde maintenant la notion des rotations, introduite par *Irving* et *Leather* (1986), qui permet efficacement de trouver et énumérer différents mariages stables en démarrant d'un mariage stable H -optimal (celui obtenu par Gale-Shapley).

Définition 2.2. *Soit M un couplage stable ;*

On appelle une M -rotation, la séquence ordonnée de paires de M , notée ρ , avec

$$\rho = \{(h_1, f_1), \dots, (h_r, f_r)\}, r > 1$$

Pour tout $1 \leq i \leq r$

- $(h_i, f_i) \in M$.
- f_i est la meilleure femme de h_i , et $f_{(i+1) \bmod r}$ est la deuxième femme qui succède à f_i dans la liste des préférences de h_i , et qui le préfère à son conjoint respectif.

ρ est une rotation si elle est M -rotation.

Dans cette séquence ordonnée de paires, le couplage stable M/ρ est obtenu par l'élimination de ρ du couplage M en faisant correspondre h_i à $f_{(i+1) \bmod r}$ plutôt que son conjoint respectif f_i .

Dans ce cas, on dit que (h_i, f_i) est éliminée par ρ et $(h_i, f_{(i+1) \bmod r})$ est produite par ρ . ρ est dite *exposée* dans M .

On note par R l'ensemble de toutes les rotations.

Propriété 2.5. *Une rotation améliore le couplage pour les femmes quant aux hommes c'est le contraire.*

Propriété 2.6. [14] *Si les premières entrées dans les listes restreintes ne spécifient pas la solution optimale féminine, alors au moins une rotation doit être exposée.*

Théorème 2.3. [14] *Chaque couple stable est exactement dans une seule rotation, et évidemment chaque couple qui appartient à une rotation est stable, sauf pour les couples stables qui sont dans le mariage optimal des femmes dès le départ, i.e., ils n'appartiennent à aucune rotation.*

Exemple 2.2. *Dans notre exemple de taille huit de la figure 2.1-, il existe huit rotations exposées qu'on représente dans le tableau suivant avec les couplages stables qui en résultent à chaque élimination d'une rotation.*

	couplage stable	rotation
M_1	(1,1) (2,4) (3,2) (4,5) (5,3) (6,8) (7,6) (8,7)	$\rho_1 = (2,4) (4,5)$
$M_2 = M_1 \setminus \rho_1$	(1,1) (2,5) (3,2) (4,4) (5,3) (6,8) (7,6) (8,7)	$\rho_2 = (2,5) (3,2) (8,7)$
$M_3 = M_2 \setminus \rho_2$	(1,1) (2,7) (3,5) (4,4) (5,3) (6,8) (7,6) (8,2)	$\rho_3 = (5,3) (8,2)$
$M_4 = M_3 \setminus \rho_3$	(1,1) (2,7) (3,5) (4,4) (5,2) (6,8) (7,6) (8,3)	$\rho_4 = (1,1) (5,2)$
$M_5 = M_4 \setminus \rho_4$	(1,2) (2,7) (3,5) (4,4) (5,1) (6,8) (7,6) (8,3)	$\rho_5 = (5,1) (6,8)$
$M_6 = M_5 \setminus \rho_5$	(1,2) (2,7) (3,5) (4,4) (5,8) (6,1) (7,6) (8,3)	$\rho_6 = (4,4) (7,6) (8,3)$
$M_7 = M_6 \setminus \rho_6$	(1,2) (2,7) (3,5) (4,6) (5,8) (6,1) (7,3) (8,4)	$\rho_7 = (2,7) (5,8)$
$M_8 = M_7 \setminus \rho_7$	(1,2) (2,8) (3,5) (4,6) (5,7) (6,1) (7,3) (8,4)	$\rho_8 = (1,2) (3,5) (6,1)$
$M_9 = M_8 \setminus \rho_8$	(1,5) (2,8) (3,1) (4,6) (5,7) (6,2) (7,3) (8,4)	

FIGURE 2.2 –

A la fin du processus d'élimination des rotations, on aboutit au couplage stable parfait optimal pour les femmes : (1,5) (2,8) (3,1) (4,6) (5,7) (6,2) (7,3) (8,4)

Représentation graphique des rotations

Définitions

Une rotation ρ' précède une autre rotation ρ si ρ' est éliminée dans la séquence d'élimination qui commence par M_0 et se termine en un couplage stable dans lequel ρ est exposé. On dit alors que ρ' est *prédécesseur* de ρ , et ρ est *successeur* de ρ' .

De même, ρ' est *prédécesseur immédiat* de ρ si :

$$\nexists \rho'' / \rho' \prec \rho'' \prec \rho$$

Notons aussi que la relation de précédence est transitive :

$$\rho'' \prec \rho' \wedge \rho' \prec \rho \Rightarrow \rho'' \prec \rho$$

\prec définit une relation d'ordre sur l'ensemble des éléments de R . Soit

1. $N^-(\rho)$: l'ensemble des prédécesseurs immédiats de ρ .
2. $N^+(\rho)$: l'ensemble des successeurs immédiats de ρ .

On représente l'ensemble de toutes les rotations par un graphe orienté $G = (X, U)$, où $X = R = \{\rho_1, \rho_2, \dots, \rho_m\}$, et $U = \{(\rho_i, \rho_j) / i, j = 1 \dots m\}$ l'ensemble des arcs de G qui sont orientés comme suit :

Si ρ' est prédécesseur immédiat de ρ alors, il existe un arc orienté de ρ' vers ρ .

Formellement, pour $(h, f) \in \rho$ et $(h', f') \in \rho'$;

- Si $f = f'$ et $h \succ_f h'$, alors $\rho' \rho \in U$.
- Si $h = h'$ et $f' \succ_h f$, alors $\rho \rho' \in U$

Si $N^-(\rho) = 0$, ρ est dit sommet source.

Si $N^+(\rho) = 0$, ρ est dit sommet de réception.

En considérant l'ensemble des rotations muni de ses relations de précédence, il est possible de les représenter en utilisant le diagramme de Hasse du poset des rotations.

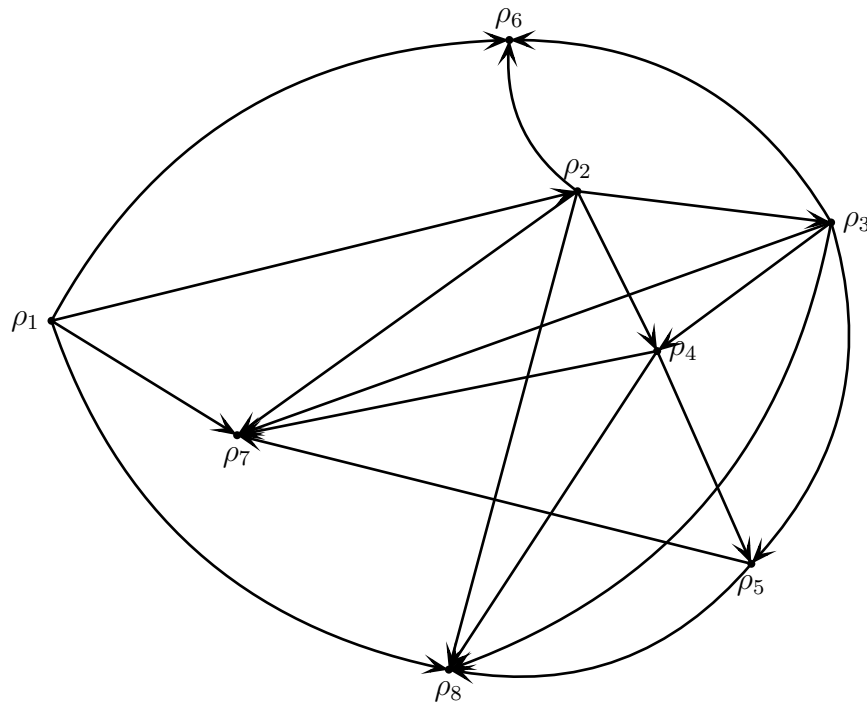


FIGURE 2.3 – La représentation de l'ensemble R des rotations par le graphe orienté $G=(X,U)$

Proposition 2.3. (Irving Leather et Gusfield, 1987) le digraphe des rotations peut être construit en $o(n^2)$ complexité temporelle [18].

Le poset des rotations

L'ensemble ordonné (R, \prec) se représente graphiquement par le diagramme de Hasse comme suit :

1. On commence d'abord par définir le dictionnaire de G ;
2. Ensuite, on passe à la décomposition de $G = (X, U)$ en niveaux ;
3. Enfin, on dessine le diagramme.

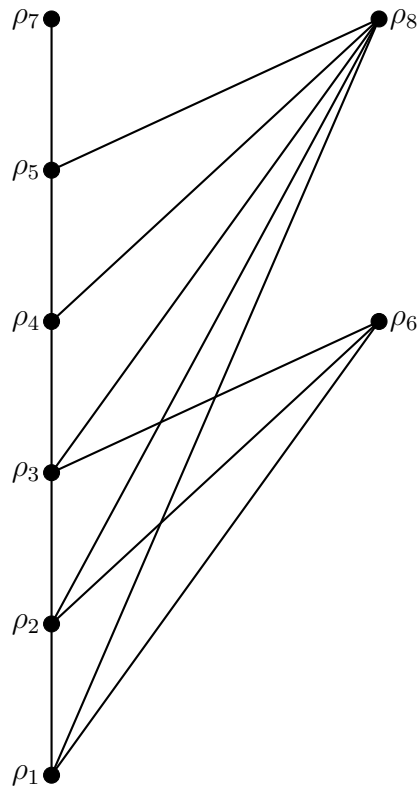


FIGURE 2.4 – Digraphe de Hasse du poset des rotations

Chaînes maximums

$[\rho_1 \rho_2 \rho_3 \rho_4 \rho_5 \rho_7]$

$[\rho_1 \rho_2 \rho_3 \rho_4 \rho_5 \rho_8]$

Chaînes maximales

[$\rho_1 \rho_6$]

[$\rho_1 \rho_2 \rho_8$]

Les chaînes maximums/maximales contiennent des paires stables distinctes possédant des individus en commun qui sont soit directement, ou indirectement comparables.

Antichaîne maximale

[$\rho_4 \rho_6$]

[$\rho_5 \rho_6$]

Antichaîne maximum

[$\rho_6 \rho_7 \rho_8$]

Les antichaînes maximums/maximales contiennent des paires stables distinctes, dont les éléments sont tous distincts (incomparables).

L'antichaîne maximum [$\rho_6 \rho_7 \rho_8$] est égale à [(1,2) (2,7) (3,5) (4,4) (5,8) (6,1) (7,6) (8,3)], elle correspond à un couplage stable parfait.

2.5.3 Mariage stable équitable

On a vu, dans ce qui précède, que l'algorithme de Gale-Shapley permet de trouver un mariage stable qui garantit l'optimalité à l'un des deux sexes, quant à l'autre, il lui assure le choix le moins optimal, ce qui n'est pas juste.

Notre but est alors, de trouver un mariage stable optimal qui ne favorise ni hommes, ni femmes. Un tel mariage est dit *mariage stable équitable*.

Définition 2.3. *Pour un couplage stable M d'une instance I du problème des mariages stables, on définit les notations suivantes :*

H_M (resp. F_M) l'ensemble des hommes de H (resp. l'ensemble des femmes de F) qui sont engagé(e)s dans M .

$C^H(M)$ le coût du couplage M pour les hommes, tel que

$$C^H(M) = \sum_{h_i \in H_M} p(h_i, M(h_i))$$

où $p(h_i, M(h_i))$ est la position de $M(h_i)$ dans la liste des préférences de l'homme h_i

$C^F(M)$ le coût du couplage M pour les femmes, tel que

$$C^F(M) = \sum_{f_i \in F_M} p(M(f_i), f_i)$$

où $p(M(f_i), f_i)$ est la position de $M(f_i)$ dans la liste des préférences de f_i .

$C(M)$ le coût du couplage M , tel que

$$C(M) = C^H(M) + C^F(M)$$

Définition 2.4. M est le mariage stable équitable si,

$C(M)$ est le coût minimum de tout mariage stable de l'instance I .

2.6 Le graphe mariage

On s'intéresse maintenant à une nouvelle représentation du problème des mariages stables impliquant des graphes.

Définition 2.5. Soit $\delta = (H, F, (\succ_p)_{p \in H \cup F})$, une instance du problème des mariages stables .

Le graphe mariage $G=(X, U)$ associé à δ est défini comme suit :

- (i) Pour toute paire acceptable (\mathbf{h}, \mathbf{f}) on lui associe un sommet $\mathbf{x}_{(\mathbf{h}, \mathbf{f})}$ dans le graphe G .
- (ii) Il existe un arc $(\mathbf{x}_{(\mathbf{h}, \mathbf{f})}, \mathbf{x}_{(\mathbf{h}', \mathbf{f}')}) \in \mathbf{U}$ si et seulement si :
 $(\mathbf{h} = \mathbf{h}'$ et $\mathbf{f}' \succ_{\mathbf{h}} \mathbf{f})$ ou $(\mathbf{f} = \mathbf{f}'$ et $\mathbf{h}' \succ_{\mathbf{f}} \mathbf{h})$

Les arcs expriment les ordres de préférences.

Exemple

Soit l'exemple suivant d'un problème de mariage stable composé de 3 hommes et de 3 femmes :

$$\begin{array}{ll} h_1 : 1 & 3 & 2 & f_1 : 3 & 2 & 1 \\ h_2 : 2 & 1 & 3 & f_2 : 1 & 3 & 2 \\ h_3 : 3 & 2 & 1 & f_3 : 2 & 1 & 3 \end{array}$$

Avec la situation décrite dans l'article [9], on obtient le graphe suivant :

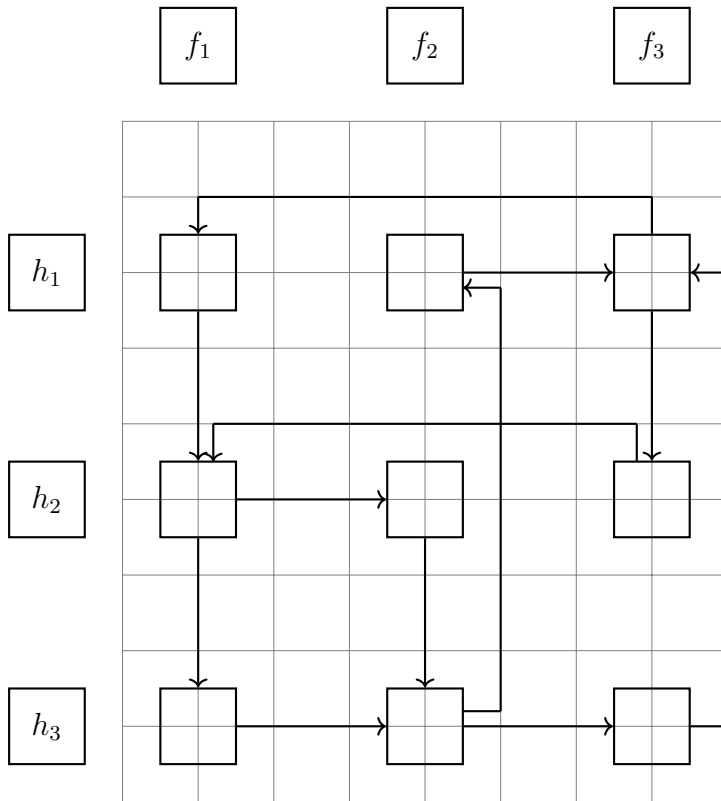


FIGURE 2.5 – Exemple d'un graphe mariage

2.6.1 Définitions

Définition 2.6. $x_{(h,f)}$ est un homme maximal, en d'autres mots, la paire acceptable correspondante à $x_{(h,f)}$ dans un couplage M est meilleure pour h si et seulement si, un arc de la forme $(x_{(h,f)}, x_{(h,f')})$ n'existe pas.

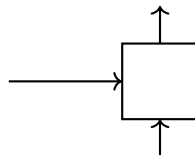


FIGURE 2.6 – Homme-max

Définition 2.7. $x_{(h,f)}$ est une femme maximale si et seulement si, il n'existe pas d'arc de la forme $(x_{(h,f)}, x_{(h',f)})$

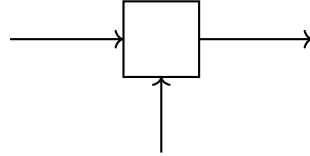


FIGURE 2.7 – Femme-max

On donne dans la figure qui suit les différentes interprétations des représentations des sommets et arcs dans le graphe mariage :

homme-min	femme-min	homme-min ou femme-max	homme-max ou femme-min

FIGURE 2.8 – Signification des sommets d'un graphe mariage

Théorème 2.4. (Ratier) [9] Un couplage M est stable si et seulement si ses sommets correspondants dans le graphe mariage $G = (X, U)$ forme un noyau de G .

Preuve 2.6. Notons R l'ensemble des sommets correspondants à un mariage stable M dans le graphe mariage $G = (X, U)$.

On sait que deux sommets appartiennent à un arc de G si et seulement si, un individu est présent dans les deux couples, alors M n'est pas un mariage que lorsque la condition suivante est vérifiée : $\forall x \in R, \Gamma_G^+(x) \cap R = \emptyset$ (1)

De plus, l'ensemble des couples qui correspondent aux successeurs de $x_{(h,f)}$ sont (h, f') et (h', f) avec $f' \succ_h f$ et $h' \succ_f h$. Donc l'ensemble X qui vérifie $\forall x \in X, \Gamma_G^+(x) \cap R \neq \emptyset$ (2) est équivalent à dire que M est stable. (1) et (2) forment la définition d'un noyau du graphe G , ceci donc montre le théorème.

2.7 Graphe biparti et mariage stable

Le problème des mariages stables peut être modélisé sous forme d'une classe particulière de graphes qui est la classe des graphes bipartis.

Comme on l'a mentionné précédemment, à chaque instance I d'un problème de mariage stable, on associe un graphe biparti $G = (H \cup F, E)$ dont H et F est une bipartition qui représente respectivement l'ensemble des hommes et femmes.

Maximiser le nombre de couples stables pour une instance I du problème des mariages stables revient désormais à résoudre un problème de couplage maximum dans le graphe biparti $G = (H \cup F, E)$.

- On représente dans ce qui suit un théorème qui a été à l'origine de la théorie des couplages. Il est formulé et démontré par le mathématicien *Philip Hall* en 1935 :

Théorème de Hall

[4] Un graphe biparti $G = (X_1 \cup X_2, E)$ admet un couplage M de G saturant X_1 si et seulement si

$$|V(A)| \geq |A|, \forall A \subset X_1$$

où $V(A) \subseteq X_2$ est l'ensemble des sommets voisins de A .

Corollaire

[4] Un graphe biparti $G = (X_1 \cup X_2, E)$ admet un couplage parfait si et seulement si $|X_1| = |X_2|$ et $|V(A)| \geq |A|, \forall A \subset X_1$.

Corollaire de König (1914 1916)

Tout graphe biparti simple régulier admet un couplage parfait.

- En transposant ce théorème dans le langage hommes et femmes, on obtient une réponse au problème de mariage stable.

Lemme des mariages

Il est possible de marier un groupe de n hommes chacun avec une femme compatible si et seulement si, chaque sous ensemble de k hommes est compatible avec au moins k femmes, pour tout $k = 1 \dots n$.

Preuve du corollaire de König

Soit G un graphe simple biparti k -régulier avec la bipartition (X_1, X_2) .

Comme G est simple et k -régulier alors, entre deux sommets $x \in X_1$ et $y \in X_2$ il existe au plus une arête, de plus $\forall x \in X_1, d(x) = k$ et $\forall y \in X_2, d(y) = k$, alors le nombre d'arêtes de G est à la fois $k |X_1|$ et $k |X_2|$ donc $|X_1| = |X_2|$.

Soit maintenant $A \subseteq X_1$ et un sous-graphe $H = (A \cup (V(A)), E(H))$. Si on compte le nombre d'arêtes dans H on aura $k |A| = |E(H)| \leq k |V(A)|$ car, évidemment les arêtes incidentes à A sont incluses dans l'ensemble des arêtes incidentes à son voisinage, ceci implique $|A| \leq |V(A)|$.

Par le théorème de Hall, il existe donc un couplage saturant X . Comme $|X_1| = |X_2|$ le couplage est également parfait.

2.7.1 Algorithme pour la résolution du problème des mariages stables

L'algorithme de Ford et Fulkerson peut être utilisé pour résoudre le problème de couplage maximum dans G .

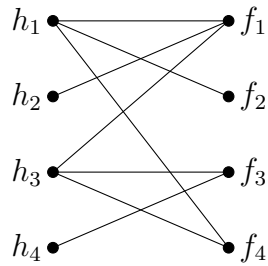
La première étape est de construire un réseau à flots en ajoutant deux sommets virtuels ; s : la source (entrée), et p : le puits (sortie), au graphe biparti $G = (H, F, E)$, puis relier s à tout sommet de H et p à tout sommet de F en orientant tous les arcs de gauche à droite.

De plus, on pose $c(\mu) = 1$ pour tout arc, i.e : toutes les capacités des arcs valent 1.

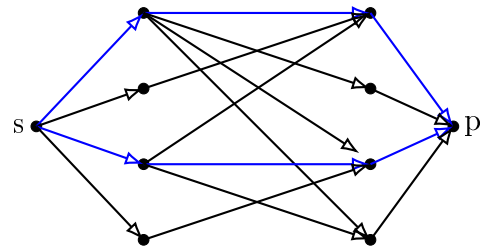
On note $R = (X, U, C)$ notre graphe réseau à flots, où $X = H \cup F \cup \{s, p\}$.

L'ensemble des arêtes transportant une unité de H à F détermine un couplage sur G .

Ford et Fulkerson (1956) proposent d'abord de construire des chemins améliorants de la source au puits, un à un, comme le représente la figure suivante. On suppose que les listes des préférences ne sont pas forcément complètes pour tous les individus.



Exemple d'un SMP



Réseau à flot avec un flot valide

FIGURE 2.9 –

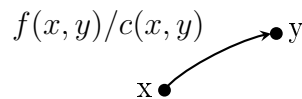
On obtient un flot valide qui transporte un flux de valeur 1 au niveau de deux arêtes de H à F qui déterminent un couplage sur G .

Remarquons que ce couplage rend d'autres mariages impossibles tels que deux hommes et deux femmes soient libres.

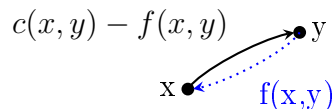
Ford et Fulkerson permettent alors d'annuler ce mariage en autorisant le parcours des chemins inverses par la méthode des graphes résiduels (graphes d'écart).

2.7.2 Méthode des graphes résiduels

Soit C un chemin de s vers p et (x,y) un arc de C de capacité $c(x,y)$. Soit $f(x,y)$ le flux qui circule dans l'arc (x,y) .



Dans le graphe résiduel, on dessine l'arc (x,y) avec une capacité égale à $c(x,y) - f(x,y)$ et un arc de retour (y,x) d'une capacité égale à $f(x,y)$



La méthode des graphes résiduels consiste à dessiner à chaque itération un graphe résiduel puis à trouver dans ce graphe *un chemin améliorant* de s vers p . Un chemin améliorant est un chemin de s vers p sur lequel il est possible d'augmenter la valeur du flot.

S'il y a un chemin améliorant de la source au puits dans le graphe d'écart alors on dessine un arc retour sur chaque chemin améliorant (les arcs de capacité 0 dans le graphe résiduel sont supprimés par souci de simplification mais ils existent toujours).

S'il n'y a pas de chemin améliorant dans le graphe résiduel alors le flot obtenu est maximal.

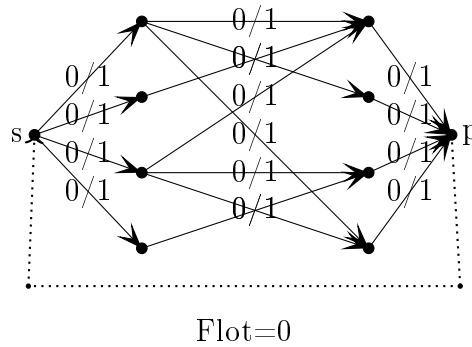


FIGURE 2.10 – Graphe résiduel $GR(0)$

Dans $GR(0)$, il existe des chemins améliorants, alors le flot courant n'est pas maximal. Les arcs retour et les arcs de gauche à droite valent tous 1. Par souci de simplification, on dessine notre réseau sans valuation.

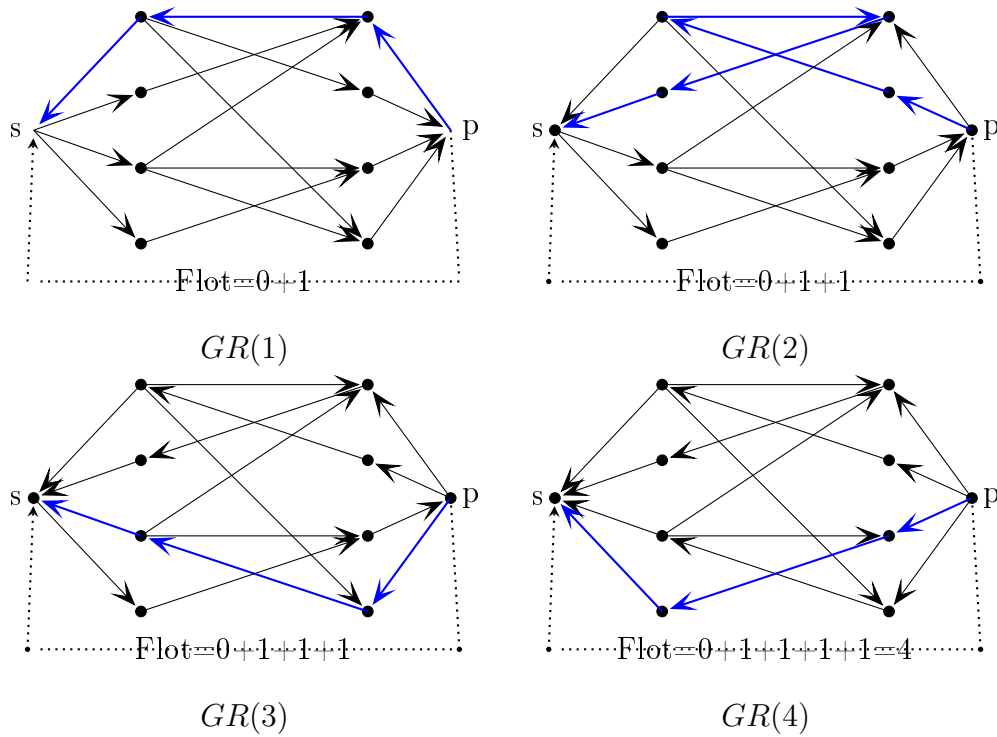


FIGURE 2.11 –

Dans $GR(4)$, il n'y a plus de chemin améliorant de s vers p , donc ce flot est maximal, et l'arrangement des mariages correspondant à ce graphe résiduel est optimal, on aura alors résolu le problème des mariages.

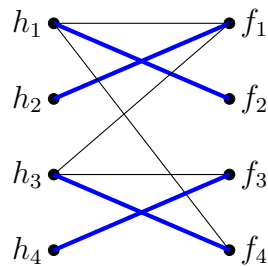


FIGURE 2.12 – Le couplage parfait

Remarque

Cette technique ne gère pas la notion des préférences.

Problème des colocataires stables

3.1 Introduction

On abordera dans ce chapitre une variante du problème des mariages stables dans les graphes non-bipartis, connue sous le nom de *problème des colocataires stables* **SRP** (stable roommates problem). Il consiste à chercher, étant donné un groupe d'individus, une partition en paires stables des colocataires.

Contrairement à ce que nous avons présenté dans le chapitre des couplages bilatéraux, où Gale et Shapley ont montré que le problème des mariages stables admet au moins un appariement stable comme solution, le problème des colocataires stables, peut admettre des situations pour lesquelles aucun couplage stable n'est possible.

Un résultat dû à Robert Irving (1985) indique qu'il est possible de déterminer en temps polynomial si une instance du *SRP* admet un couplage stable, et si oui, de trouver un tel couplage. La recherche d'une solution est donc basée sur un algorithme itératif présenté par Irving et composé de deux phases, et qui porte le nom *d'algorithme d'Irving*.

Nous donnons dans ce qui suit des définitions et généralisations du *SRP* ainsi qu'une analyse de ce dernier dans le but de le résoudre et trouver enfin une ou plusieurs solutions ou encore affirmer qu'il n'existe plus de solution.

3.2 Définitions et préliminaires

Définition 1

Le problème des colocataires stables consiste en la formation de couples disjoints à partir d'un ensemble de $2n$ personnes. Formellement, il est défini par le couple $(A, (\succ_{a_i})_{a_i \in A})$, où $A = \{a_1, a_2, \dots, a_{2n}\}$ est un ensemble fini d'individus de cardinalité égale à $2n$, et $(\succ_{a_i})_{a_i \in A}$ est une relation totale, transitive, antisymétrique et stricte qui représente les préférences des individus $a_i \in A, i = 1, \dots, 2n$. Chaque personne $a_i \in A$ classe les $A - \{a_i\}$ autres individus selon l'ordre strict des préférences, \succ_{a_i} .

On note $P(a_i)$, la liste des préférences de l'individu a_i . Une personne a_j est *acceptable* pour a_i si $a_j \succ_{a_i} a_i$, si non elle est dite *inacceptable*.

Définition 2

Une solution d'une instance du problème des colocataires stables est un couplage M qui représente une partition de l'ensemble A en paires disjoints non ordonnées.

Il représente également une bijection $M : A \rightarrow A$, telle que

$M = \{(a_i, M(a_i)) \in A \times A\}$, avec ;

- $\forall a_i \in A, M(a_i) \neq a_i$;
- $M(a_i) = a_j$ si $M(a_j) = a_i$.

Un couplage M est complet si la partition de $2n$ personnes est égale à n paires disjoints de colocataires telles que, pour chaque paire $(a_i, a_j) \in M$; $a_i \in P(a_j)$ et $a_j \in P(a_i)$.

Définition 3

Une paire $(a_i, a_j) \notin M$ est *bloquante* si $a_j \succ_{a_i} M(a_i)$ et $a_i \succ_{a_j} M(a_j)$.

Un couplage M est stable s'il ne contient pas de paire bloquante.

Exemple

Considérons quatre personnes a_1, a_2, a_3 et a_4 avec leurs listes des préférences suivantes :

$$P(a_1) = a_2, a_3, a_4$$

$$P(a_2) = a_3, a_1, a_4$$

$$P(a_3) = a_1, a_2, a_4$$

$$P(a_4) = \text{arbitraire}$$

Les couplages correspondants dans cet exemple sont :

- $M_1 = \{(a_1, a_3), (a_2, a_4)\}$
- $M_2 = \{(a_1, a_2), (a_3, a_4)\}$
- $M_3 = \{(a_1, a_4), (a_2, a_3)\}$

$(a_1, a_2), (a_2, a_3)$ et (a_1, a_3) sont des paires bloquantes pour M_1, M_2 et M_3 respectivement.

Aucun mariage ne sera stable, puisque tout couplage doit associer une personne à l'individu a_4 (le pire partenaire de a_1, a_2 et a_3), et que cette personne pourra trouver une autre qu'elle préfère à a_4 , et qui le préfère également.

Un problème est dit *résoluble* si l'ensemble des couplages stables n'est pas vide, si non il n'est pas résoluble (cas de l'exemple donné). On dit qu'une personne a_i préfère le couplage stable M à M' si et seulement si, a_i préfère son partenaire dans M que dans M' , i.e. $M(a_i) \succ_{a_i} M'(a_i)$.

On dit aussi que l'individu a_i est *indifférent* entre M et M' s'il est en couple avec la même personne dans les deux couplages stables, on note $M \sim_{a_i} M'$.

Définition 4

Soit $I = (A, (\succ_{a_i})_{a_i \in A})$, une instance du problème des colocataires stables. Notons $G = (X, E)$ le graphe associé à I , où G est un graphe complet, noté K_n dont son ensemble de sommets représente les individus de I , et toutes les paires acceptables sont des arêtes de G . Il s'agit d'associer deux à deux les éléments de l'ensemble A afin de former un couplage stable et parfait. Un couplage parfait dans un graphe ayant un nombre paire $(2n)$ de sommets est un ensemble de n arêtes du graphe, deux à deux disjointes et telles que chaque sommet du graphe est incident à exactement une arête du couplage.

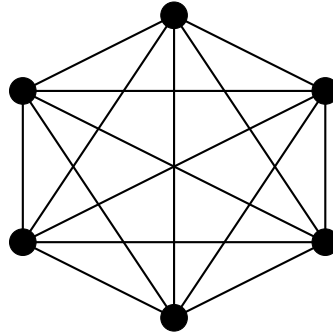


FIGURE 3.1 – Un exemple d’un graphe complet représentant un problème des colocataires stables. Les sommets représentent les personnes. Les arêtes désignent les préférences.

3.3 Algorithme de résolution du problème des colocataires stables

3.3.1 Algorithme d’Irving

L’algorithme d’Irving représente une solution du problème des colocataires stables. Son principe est de supprimer successivement les entrées des listes des préférences jusqu’à ce que chaque personne n’ait qu’une seule entrée sur sa liste, ou bien, jusqu’à ce qu’une personne n’ait plus d’entrée. Le premier cas nous affirme que le problème admet un couplage stable, quant au deuxième, il nous assure qu’il n’existe plus de couplage.

Cet algorithme est composé essentiellement de deux phases, dont la première est basée sur une séquence de propositions de la part de chaque personne aux autres, elle est similaire à la procédure d’acceptations différées, sauf que dans ce cas les propositions et les acceptations sont faites simultanément et par tous les individus. Dans la deuxième phase, une nouvelle réduction dans les listes des préférences se fait en se basant sur la notion de rotation. Durant l’exécution de cet algorithme, l’ensemble des listes des préférences de tous les individus est appelé *une table*. Présentons alors cet algorithme.

Phase 1

Algorithme 2 phase 1 de l'algorithme d'Irving

tant que \exists au moins une personne libre a_i dont sa liste n'est pas vide **faire**

a_j : meilleure personne de a_i

a_i propose à a_j

si a_j est libre **alors**

a_j accepte le proposition de a_i

si non

$a_k :=$ la personne qui est déjà acceptée par a_j

fin si

si a_j préfère a_k à a_i **alors**

a_j rejette a_i

si non

a_j accepte a_i

 pour toute personne $a_k \prec_{a_j} a_i$: retirer a_k de la liste de a_j et a_j de la liste de a_k

fin si

si il existe une liste vide **alors**

 pas de couplage stable

fin si

si chaque personne est semi-engagée **alors**

 aller à la phase 2

fin si

fin tant que

Principe de la phase 1

A cette phase, tous les individus sont initialement libres, et chacun dispose d'une liste des préférences. Une personne a_i propose à la première personne a_j sur sa liste des préférences, qui, soit accepte temporairement cette proposition en rejetant simultanément toute proposition moins préférée que celle qu'elle détient actuellement, soit la rejette immédiate-

ment, si elle a déjà reçu et accepté une proposition d'une autre personne classée meilleure dans sa liste des préférences. En poursuivant ce processus d'acceptations et de refus, chaque individu propose aux autres dans l'ordre dans lequel ils apparaissent sur sa liste des préférences. Il s'arrête lorsqu'il reçoit une promesse de prise en considération de la proposition, et tout rejet ultérieur l'amène à poursuivre sa séquence de propositions.

Cette phase de l'algorithme se terminera dans deux cas distincts :

- Soit, avec une personne ayant une liste des préférences vide (cette personne est rejetée par tous les autres individus) auquel cas l'algorithme renvoie que l'instance du problème des colocataires n'a pas de couplage stable.
- Ou bien, avec chaque personne détenant une proposition et ayant une liste des préférences qui contient au moins deux participants. A ce niveau, on dit que chaque personne est *semi-engagée*.

Notons qu'une personne a_i est *semi-engagée* à sa meilleure femme a_j , si et seulement si a_i est la dernière sur la liste des préférences de a_j . Si une personne n'est pas semi-engagée, on dit qu'elle est *libre*.

L'ensemble des listes des préférences à la fin de cette phase est appelé *la table de la phase 1*, on la note \mathbf{T}_0 . L'algorithme dans ce cas passe à la deuxième phase de son exécution.

Phase 2

Algorithme 3 Phase 2 de l'algorithme d'Irving

$T := T_0$

tant que \exists des listes avec plus d'une seule entrée et aucune liste T est vide **faire**

Trouver et éliminer une rotation ρ

si chaque personne a exactement une entrée sur sa liste **alors**

T est le couplage stable

fin si

si il existe une liste vide **alors**

il n'existe pas de couplage stable

fin si

fin tant que

Principe de la phase 2

La clé d'élimination dans cette phase est la recherche des rotations.

Une rotation ρ exposée dans la table T , sera éliminée.

Définissons alors ce qu'est une rotation :

Une rotation dans une table stable est *une séquence cyclique de paires ordonnées d'individus*, noté R , telles que :

$$\mathbf{R} = (\mathbf{a}_1, \mathbf{b}_1), (\mathbf{a}_2, \mathbf{b}_2), \dots, (\mathbf{a}_k, \mathbf{b}_k) \in \mathbf{A} \times \mathbf{A}$$

où $a_1 = a_{k+1}$ et $b_1 = b_{k+1}$

- Les a_i sont tous distincts, $i = 1 \dots k$;
- b_i est le premier dans la liste des préférences de a_i ;
- a_i est le dernier sur la liste des préférences de b_i .

Propriété 3.1. [13]

- b_{i+1} est la seconde personne sur la liste des préférences de a_i ;
- a_{i+1} est la dernière personne sur la liste des préférences de b_{i+1} ;
- b_{i+1} est la première personne sur la liste des préférences de a_{i+1} .

On peut également écrire $R = (A, H, S)$, où H (*head*) est l'ensemble des individus qui sont classés sur les premières entrées des éléments de A , et S (*second*) est l'ensemble des deuxièmes entrées de A , avec l'ordre correspondant.

A chaque fois qu'une rotation R est exposée dans la table stable T , T est réduite à T / ρ .

$$\begin{array}{lll} \mathbf{a}_i [\mathbf{b}_i \mathbf{b}_{i+1} \dots] & \text{devient} & \mathbf{a}_i [\mathbf{b}_{i+1} \dots] \\ \mathbf{b}_i [\dots \mathbf{a}_{i-1} \mathbf{a}_i] & \text{devient} & \mathbf{b}_i [\dots \mathbf{a}_{i-1}] \end{array}$$

L'ordre d'élimination n'est pas important, car les rotations ne sont éliminées de la table stable que lorsqu'elles sont exposées.

Lemme 3.1. *si T est une table dont aucune liste de préférences est vide, avec au moins une personne ayant plus d'une seule entrée sur sa liste, alors il existe une rotation exposée dans T .*

Preuve 3.1. Soit a_i une personne ayant au moins deux entrées sur sa liste des préférences, dont b_i et b_{i+1} sont respectivement la meilleure et la seconde personne sur la liste de a_i . Puisque toutes les meilleures entrées sur les listes des préférences sont une permutation sur l'ensemble des individus, alors il existe sûrement une personne a_j dont b_{i+1} est sa meilleure entrée. Si on suppose que a_j a seulement une seule entrée sur sa liste qui est b_{i+1} , alors b_{i+1} a une seule entrée sur sa liste, or b_{i+1} est la seconde sur la liste de a_i , alors a_i doit être aussi sur la liste de b_{i+1} , alors b_{i+1} et a_j doivent avoir au moins deux entrées sur leurs listes des préférences. En répétant ce raisonnement, on trouve une rotation.

- Si la table stable qui en résulte possède une liste vide alors aucun couplage stable n'existe.

Si non, toutes les listes ont exactement un seul individu, l'algorithme renvoie donc un couplage stable et parfait.

Proposition 3.1. Irving (1985) montre que son algorithme est de complexité $o(n^2)$, dont chacune des phases 1 et 2 sont en $o(n^2)$ complexité algorithmique [17].

3.4 Enumération de tous les couplages stables

3.4.1 L'arbre de décision de l'algorithme d'Irving

Nous nous intéressons maintenant à l'énumération de tous les couplages stables pour une instance I du problème des colocataires stables.

On construit **un arbre de décision** $D = (X, E)$ où, X représente l'ensemble des sommets de D tels que chaque sommet représente une table, dont la **racine** de l'arbre est la table stable obtenue de la première phase de l'algorithme d'Irving, et E représente l'ensemble des arêtes de D telles qu'une arête existe entre deux sommets (tables) lorsqu'il y a une rotation R exposée dans une des deux tables et qui produit l'autre après l'élimination de R de cette dernière.

Dans l'arbre D , la phase 2 de l'algorithme d'Irving est exécutée de toutes les manières possibles dans le but de générer tous les couplages stables qu'une instance I du SRP peut avoir comme solution. On note $D(x)$ le sous-arbre de D de racine x .

Définition 3.1. Soit $R = (A, H, S)$ une rotation dans D . On définit R^d le **dual** de R par le triplet (S, A, A^r) où, S et A ont le même ordre dans R^d que celui dans R . Notons de plus que $(R^d)^d = R$. On appelle alors R et R^d **une paire duale** de rotations. Si une rotation n'admet pas de dual, alors elle est appelée **singleton**.

Définition 3.2. L'ensemble des listes des préférences d'une table T contenant plus d'une personne est appelé **la partie active de T**

Proposition 3.2. (Gusfield 1988) [5] Deux chaînes de D avec le même ensemble de rotations produisent le même couplage stable.

Proposition 3.3. (Gusfield 1988) [5] Tout chemin de la racine à la feuille dans l'arbre D contient toutes les rotations singletons et exactement une rotation d'une paire duale de rotations.

Exemple 3.1. Un exemple du problème des colocataires stables composé de 8 individus est le suivant :

1	3	8	6	2	5	7	4
2	5	1	4	7	3	8	6
3	6	2	4	5	7	1	8
4	1	6	3	7	2	5	8
5	1	4	7	3	6	8	2
6	2	7	8	1	5	4	3
7	1	8	3	4	5	2	6
8	3	2	4	5	6	1	7

FIGURE 3.2 – Les individus avec leurs listes des préférences

On applique l'algorithme d'Irving à cet exemple afin de trouver un couplage stable.

On commence d'abord par la phase 1 qui consiste en une séquence de propositions d'acceptations et de rejets jusqu'à l'obtention la table stable suivante :

1	3	8	6	2	5		
2	5	1	4	7	3		
3	2	4	5	7	1		
4	6	3	7	2	5	8	
5	1	4	7	3	6	8	2
6	7	8	1	5	4		
7	8	3	4	5	2	6	
8	4	5	6	1	7		

FIGURE 3.3 – La table stable T

A la deuxième phase de l'algorithme, on cherche à trouver des rotations. Chaque rotation exposée dans une table sera éliminée de cette dernière jusqu'à arriver enfin à un couplage stable.

1	8	6	2	5			
2	5	1	4	7	3		
3	2	4	5	7			
4	6	3	7	2	5	8	
5	1	4	7	3	6	8	2
6	7	8	1	5	4		
7	3	4	5	2	6		
8	4	5	6	1			

FIGURE 3.4 – La table $T1$ après élimination de $R_1 = (1,3), (7,8)$ de T

1	8	6	2				
2	1	4	7	3			
3	2	4	5	7			
4	6	3	7	2	5		
5	4	7	3	6	8		
6	7	8	1	5	4		
7	3	4	5	2	6		
8	5	6	1				

FIGURE 3.5 – La table $T2$ après élimination de $R_2 = (2,5), (5,1), (8,4)$ de la table $T1$

1	6	2
2	1	3
3	2	4
4	3	7
5	7	8
6	8	1
7	4	5
8	5	6

FIGURE 3.6 – La table $T3$ après élimination de R_3 de la table $T2$

$$R_3 = (1,8), (4,6), (5,4), (6,7), (7,3)$$

1	2
2	1
3	4
4	3
5	7
6	8
7	5
8	6

FIGURE 3.7 – La table $T4$ après élimination de R_4 de la table $T3$

$$R_4 = (1,6), (3,2), (7,4), (8,5)$$

Le couplage stable parfait obtenu par l'algorithme d'Irving est donc :

$$(1,2) (3,4) (5,7) (6,8)$$

- *On remarque que dans la table $T3$ on peut également trouver une autre rotation $R_5 = (2,1), (4,3), (5,7), (6,8)$, et que son élimination produit la table suivante :*

1	6
2	3
3	2
4	7
5	8
6	1
7	4
8	5

FIGURE 3.8 – La table $T5$ après élimination de R_5 de la table $T3$

- *Construisons alors l'arbre de décision de l'algorithme d'Irving pour notre exemple :*

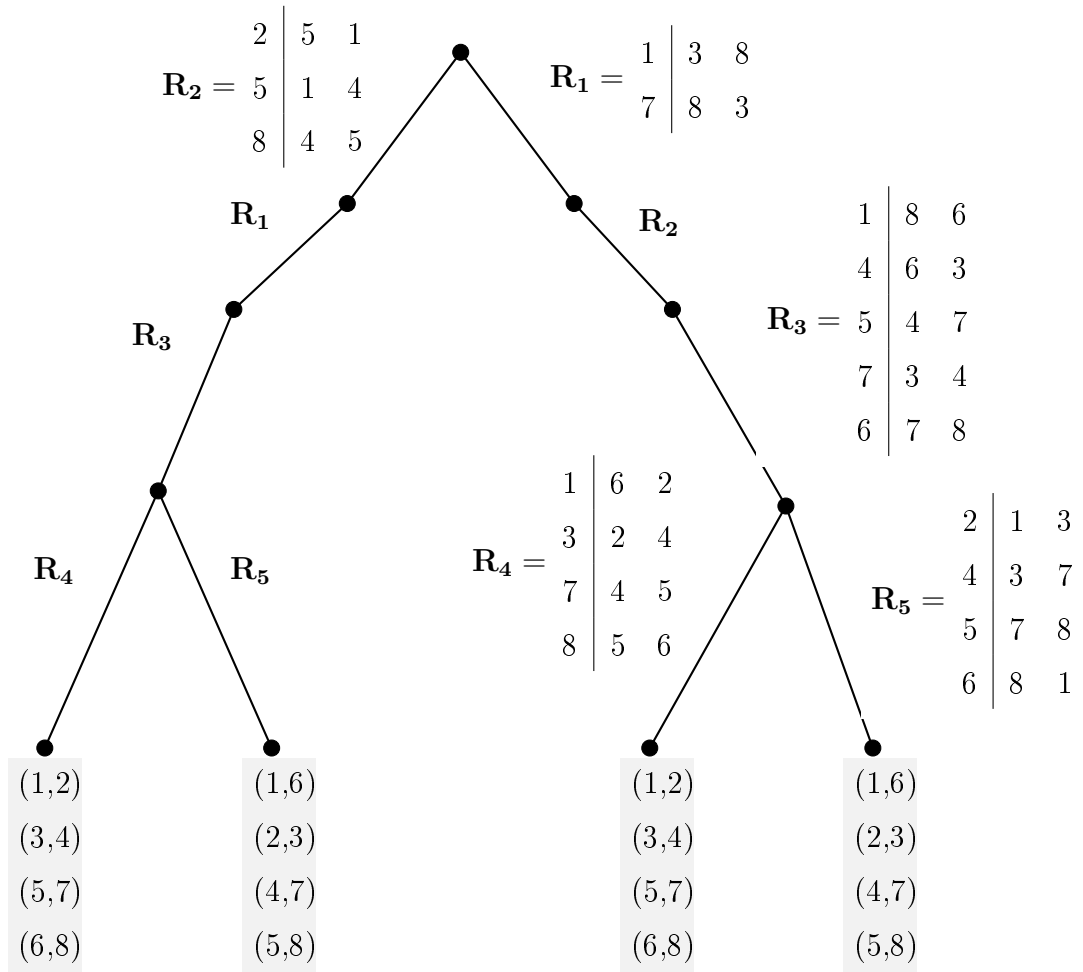


FIGURE 3.9 – Arbre de décision D de l’algorithme d’Irving appliqué à l’exemple précédent.

L’ensemble de toutes les rotations qui existent dans cet exemple est :

- $R_1 = (1,3), (7,8)$
- $R_2 = (2,5), (5,1), (8,4)$
- $R_3 = (1,8), (4,6), (5,4), (6,7), (7,3)$
- $R_4 = (1,6), (3,2), (7,4), (8,5)$
- $R_5 = (2,1), (4,3), (5,7), (6,8)$

- Les rotations R_4 et R_5 sont duales

$$\mathbf{R}_4 = \begin{array}{c|cc} \mathbf{A} & \mathbf{H} & \mathbf{S} \\ \hline 1 & 6 & 2 \\ 3 & 2 & 4 \\ 7 & 4 & 5 \\ 8 & 5 & 6 \end{array} \qquad \mathbf{R}_5 = \begin{array}{c|cc} \mathbf{S} & \mathbf{A} & \mathbf{H}^r \\ \hline 2 & 1 & 3 \\ 4 & 3 & 7 \\ 5 & 7 & 8 \\ 6 & 8 & 1 \end{array}$$

- Les rotations R_1, R_2 et R_3 sont des singletons (ne sont pas duales)
- Tous les chemins de l'arbre D sont de longueur égale à 4.
- L'ensemble de toutes les solutions stables et parfaites possibles pour notre exemple des colocataires stables est :

1. $\mathbf{A}=(1,6) , (2,3) , (4,7) , (5,8)$

2. $\mathbf{B}=(1,2) , (3,4) , (5,7) , (6,8)$

3.4.2 La méthode d'énumération duale

Contrairement à la méthode d'énumération naïve de tous les couplages stables par l'arbre de décision, la méthode *d'énumération duale* génère chaque solution exactement une fois. On construit un arbre binaire B . Comme dans l'arbre D , la racine de B est un sommet qui représente la table stable de la phase 1. Lorsqu'un sommet x n'est pas une feuille, une rotation est exposée dans la table représentée par x , et une arête étiquetée par R joint x à la table stable obtenue après élimination de R dans x , notée y . On appelle cette arête, *arête de gauche*. Egalement une arête d'extrémité x si elle existe est étiquetée par un ensemble de rotations, on l'appelle *arête de droite*. Si la rotation R est singleton, alors l'arête de droite n'existe pas, si non elle sera étiquetée par $\pi^*(R^d)$. Gusfield (1988) montre que l'ensemble des feuilles après la construction de cet arbre est l'ensemble des affectations stables sans répétition [5].

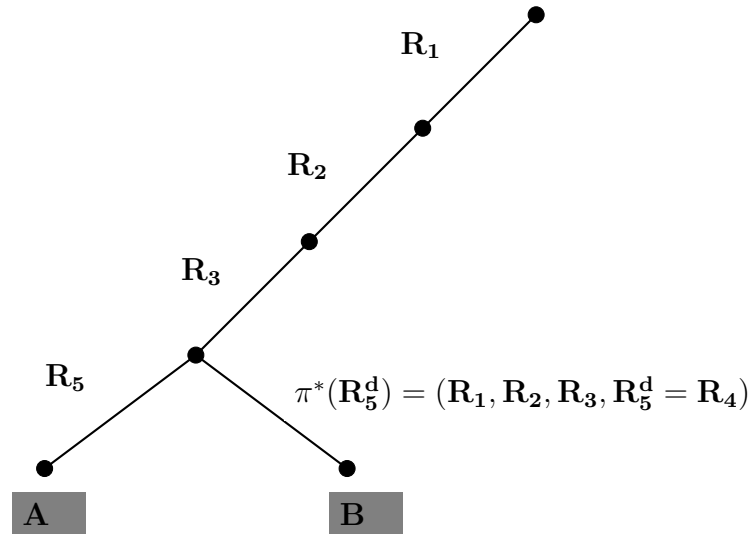


FIGURE 3.10 – L’arbre B obtenu par la méthode d’énumération duale.

3.5 La variante 3D du problème des colocataires stables

Nous aborderons dans cette section, un problème de recherche qui consiste en la partition d’un ensemble composé de $3N$ personnes en triplets disjoints, dont chaque personne de cet ensemble classe les $3N - 1$ autres personnes dans un ordre total des préférences. Ce problème présenté est une extension du problème des colocataires stables, il est appelé *problème des colocataires stables en trois dimensions* ou simplement 3D-SR.

Une solution du problème 3D-SR est un couplage stable M constitué de N triplets.

Supposons que M contient les trois triplets suivants ; (p_1, p'_1, p''_1) , (p_2, p'_2, p''_2) et (p_3, p'_3, p''_3) , on dit qu’un triplet $(p_1, p_2, p_3) \notin M$ est *bloquant* pour M si les conditions suivantes sont vérifiées :

- $p_2 \succ_{p_1} p'_1$ et $p_3 \succ_{p_1} p''_1$, **ou** $p_3 \succ_{p_1} p'_1$ et $p_2 \succ_{p_1} p''_1$;
- $p_1 \succ_{p_2} p'_2$ et $p_3 \succ_{p_2} p''_2$, **ou** $p_3 \succ_{p_2} p'_2$ et $p_1 \succ_{p_2} p''_2$;
- $p_1 \succ_{p_3} p'_3$ et $p_2 \succ_{p_3} p''_3$, **ou** $p_2 \succ_{p_3} p'_3$ et $p_1 \succ_{p_3} p''_3$;

Si M ne contient pas de triplet bloquant, alors M est un couplage *stable*, si non il est *instable*.

Théorème 3.2. *Le 3D-SR est un problème NP-complet.*

Preuve 3.2. *On montre que le problème 3D-SR est NP-complet à l'aide d'une transformation polynomiale au problème de partition en triangles, qui est un problème qui appartient à la classe des problèmes NP-complets.*

Partition en triangles

Le problème de partition en triangles consiste à chercher étant donné un graphe non orienté $G = (X, E)$ tel que, $|X| = 3q$ ($q \in \mathbb{N}$), une partition de X en q triangles, notés T_1, T_2, \dots, T_q , qui satisfont :

- Chaque T_i contient exactement trois sommets;
- Pour tout $\mathbf{T}_i = (\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}''_1)$, chacune des arêtes $\mathbf{x}_1 \mathbf{x}'_1$, $\mathbf{x}_1 \mathbf{x}''_1$, $\mathbf{x}'_1 \mathbf{x}''_1$ est dans E .

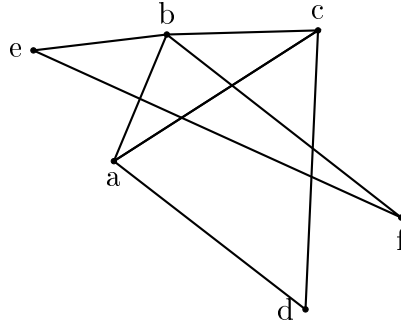


FIGURE 3.11 – G peut être partitionné en deux triangles $\{a, d, c\}$ et $\{b, e, f\}$

Le problème 3D-SR admet une solution si et seulement si le problème de partition en triangles admet une solution [11].

3.6 Le problème des luminaires stables

3.6.1 Préliminaires

Le problème des luminaires stables, *The stable fixtures problem SF*, introduit par Irving et Scott [19], est une généralisation du problème des colocataires stables où, chaque participant a une capacité fixe sur l'ensemble des autres participants, dont il doit être apparié à un nombre inférieur ou égal à cette capacité. Son nom provient d'une application possible, dans laquelle un ensemble de personnes ou équipes participent à une compétition, dont les rencontres doivent être précisées à l'avance. Chaque agent dispose d'un nombre déterminé de rencontres et peut jouer au maximum une fois.

Nous définissons formellement ce problème par :

- $X = \{ x_1, \dots, x_n \}$, l'ensemble des participants, ou simplement *joueurs* ;
- $1 \leq C_{x_i} \leq n - 1$, la capacité de chaque participant x_i pour $i = 1, \dots, n$;
- la *structure des préférences* notée P ; elle comprend la liste des préférences P_{x_i} strictement ordonnée pour chaque joueur x_i ($1 \leq i \leq n$), qui classe les $X \setminus \{ x_i \}$ participants selon ses préférences.

Un *couplage* M est un ensemble de paires acceptables telles que, pour tout i ($1 \leq i \leq n$) :

$$|\{ x_j : \{ x_i, x_j \} \in M \}| \leq C_{x_i}$$

La *taille* du couplage M définit le nombre de paires dans M .

$M(x_i) = \{ x_j : \{ x_i, x_j \} \in M \}$ est l'ensemble des partenaires de x_i dans M .

Une paire acceptable $\{ x_i, x_j \} \notin M$ est une *paire bloquante* pour le couplage M , si

- soit x_i a moins de C_{x_i} partenaires **ou**, préfère x_j à au moins un de ses partenaires dans M ;
- soit x_j a moins de C_{x_j} partenaires **ou**, préfère x_i à au moins un de ses partenaires dans M .

Un couplage qui n'a pas de paire bloquante est *stable*, si non il est *instable*.

3.6.2 Algorithme de résolution du problème SF

Phase 1 de l'algorithme

Cette phase ressemble à la première phase de l'algorithme d'Irving pour la résolution du problème des colocataires stables ; elle consiste en une séquence d'*offres* (propositions) de la part de chaque joueur. La différence principale est que les offres et les suppressions d'entrées des listes des préférences sont contrôlées par les capacités des joueurs.

On appelle *soumissionnaire*, un joueur x_i qui fait une offre à un autre joueur x_j , qui est dit *cible* de x_i .

Algorithme 4 phase 1 du problème SF

```

S := ∅
axi := nombre de joueurs pour lesquels xi a fait une offre et qui ne l'ont pas rejetée
tant que axi < min(Cxi, |Pxi|) faire
    xj := le meilleur partenaire de xi
    si xi fait une offre à xj alors
        S := S ∪ { (xi, xj) }
    fin si
    xk := le Cèmexj soumissionnaire
    xl := un successeur de xk sur Pxj
    si xj a reçu un nombre d'offres supérieur à sa capacité alors
        pour tout successeur xl de xk sur Pxj
            S := S \ { (xl, xj) }
            supprimer la paire {xl, xj} de P
    fin si
fin tant que
dxi := min(Cxi, |Pxi|)
si ∑xi∈X dxi est impaire alors
    il n'existe pas de couplage stable
fin si

```

Principe de l'algorithme

Chaque offre successive est faite par un joueur x_i ($1 \leq i \leq n$) qui a moins de cibles que sa capacité C_{x_i} , il propose d'abord à son meilleur partenaire possible, noté x_j .

Dans le cas où x_j a reçu un nombre d'offres inférieur à sa capacité, la paire $\{x_i, x_j\}$ est identifiée.

Si x_j a détenu un nombre d'offres supérieur à sa capacité, dont x_k est le $C_{x_j}^{\text{ème}}$ soumissionnaire, alors la paire $\{x_i, x_j\}$ où, x_j préfère x_k à x_i , sera supprimée. Nous entendons de la suppression de la paire $\{x_i, x_j\}$ l'élimination de x_i de P_{x_j} , et x_j de P_{x_i} .

Cette phase se termine lorsque pour tout i ($1 \leq i \leq n$), le nombre de cibles de x_i est égal au minimum entre sa capacité C_{x_i} , et son nombre d'entrées $|P_{x_i}|$ sur sa liste des préférences P_{x_i} , qu'on note par $d_{x_i} = \min(C_{x_i}, |P_{x_i}|)$, appelé *degré*.

On note par P^1 la structure des préférences qui en résulte après l'application de la phase 1 de l'algorithme, et respectivement $P_{x_i}^1$, la liste des préférences de chaque participant x_i pour tout i ($1 \leq i \leq n$).

Phase 2 de l'algorithme

Soient les notations suivantes

- \mathbf{P} : la structure des préférences ;
- $\mathbf{F_P}(\mathbf{x}_i)$: les premiers $\min(C_{x_i}, |P_{x_i}|)$ participants sur P_{x_i} , $\forall i$ ($1 \leq i \leq n$);
- $\mathbf{L_P}(\mathbf{x}_i)$: l'ensemble des participants $\{y_j : x_i \in \mathbf{F_P}(y_j)\}$;
- $\mathbf{l_P}(\mathbf{x}_i)$: le dernier participant sur P_{x_i} ;
- $\mathbf{s_P}(\mathbf{x}_i)$: le premier participant $y_j \notin \mathbf{F_P}(\mathbf{x}_i)$ ($s_P(x_i)$ est défini seulement lorsque $|P_{x_i}| > C_{x_i}$, $\forall i$ ($1 \leq i \leq n$)).

Durant l'exécution de cette phase, la structure de préférences P est caractérisée par :

- Pour tout participant x_i ($\forall i=1, \dots, n$), $|\mathbf{L_P}(\mathbf{x}_i)| = |\mathbf{F_P}(\mathbf{x}_i)|$;
- Pour tout joueur x_i ($\forall i=1, \dots, n$), $\mathbf{l_P}(\mathbf{x}_i) \in \mathbf{L_P}(\mathbf{x}_i)$;
- Une paire acceptable $\{x_i, x_j\} \notin P$ si et seulement si x_i préfère $\mathbf{l_P}(\mathbf{x}_i)$ à x_j , et x_j préfère $\mathbf{l_P}(\mathbf{x}_j)$ à x_i .

La structure des préférences P qui vérifie les propriétés précédentes est dite *stable*.

Algorithme 5 phase 2 du problème SF

$P := P^1$

tant que $\nexists P_{x_i}$ tel que $|P_{x_i}| < d_{x_i}$ **et** $\exists P_{x_i}$ tel que $|P_{x_i}| > d_{x_i}$ **faire**

trouver une rotation ρ dans P

$P = P/\rho$

si \exists des listes $P_{x_i} < d_{x_i}$ ($1 \leq i \leq n$) dans P **alors**

ce problème n'admet pas de solution

si non

retourner P, le couplage stable

fin si

fin tant que

principe de l'algorithme

Cette phase débute avec la structure des préférences obtenue de la phase 1, dont sa clé de réduction est basée sur les *rotations*.

Définition 3.3.

Une rotation ρ est une séquence ordonnée de paires $\rho = ((x_0, y_0), (x_1, y_1), \dots, (x_{r-1}, y_{r-1}))$

$r > 1$, où pour tout i ($0 \leq i \leq r - 1$) ; $\mathbf{x}_i = \mathbf{l}_P(\mathbf{y}_i)$ et $\mathbf{y}_{(i+1) \bmod r} = \mathbf{S}_P(\mathbf{x}_i)$

On dit que x_i, y_i et la paire (x_i, y_i) sont dans la rotation ρ .

Si durant l'exécution de cette phase, il existe une liste de préférences P_{x_i} qui vérifie $|P_{x_i}| < d_{x_i}$ alors pas de couplage stable, par conséquent, pas de solution pour le problème. Contrairement, dans le cas où $|P_{x_i}| > d_{x_i}$, il existe au moins une rotation exposée dans P [19].

De la théorie à la pratique

4.1 Introduction

Nous nous intéressons dans ce chapitre à l'analyse de l'importance pratique de la théorie du *matching* ou mécanisme de l'*appariement sous préférences*. Il représente un sujet crucial d'une structure mathématique profonde, et d'une algorithmique élégante.

L'importance de ce domaine de recherche a récemment été reconnue par l'attribution du prix de sciences économiques de la banque royale de Suède à la mémoire d'Alfred Nobel, en 2012, à Alvin Roth et Lloyd Shapley, pour leurs travaux sur le problème du mariage stable, qui a permis d'apporter des réponses non seulement mathématiques mais aussi pratiques à de nombreuses situations problématiques.

Le problème du mariage stable a été initié et étudié à l'origine par David Gale et Lloyd Shapley, en 1962, puis développé par le mathématicien Alvin Roth, à partir de 1980, en montrant les nombreuses relations économiques qui se fondent sur ce modèle. Nous disposons, grâce à leurs travaux, d'un outillage analytique dédié à étudier et résoudre ce que nous appellerons, les marchés régis par appariement. Nous allons considérer en ce qui suit, un ensemble de marchés où se font face deux types d'agents complémentaires, chacun possédant des préférences envers les agents du type opposé, qui peuvent échanger des biens et services afin de satisfaire leurs besoins, tel est le cas du marché du travail, marché boursier, affectation des internes en médecine aux hôpitaux, appariement efficace des donneurs d'organes et malades en tenant compte des incompatibilités, problème de

l'admission post-bac (APB), marché des locations immobilières, etc.

Nous explorons dans ce qui suit l'efficacité et l'utilité de la théorie des mariages qui s'avère en ses multiples applications pratiques, qu'on illustre par quelques exemples.

4.2 Modèles d'appariement de Shapley

4.2.1 Appariement avec transferts

Dans ce modèle bilatéral les agents sont des propriétaires et les mariages sont caractérisés par la possibilité d'effectuer des transferts monétaires.

4.2.2 Allocation des biens indivisibles

On distingue deux types d'agents qui sont tous des propriétaires : exemple des individus possédant chacun une maison, et également chaque agent a des préférences sur l'ensemble des maisons mais n'a l'usage que d'une seule (modèle de Shapley et Shubik 1972) [16].

Cependant, le marché dans ce modèle n'est pas bilatéral et les agents ne peuvent pas effectuer des transferts monétaires.

La caractéristique cruciale de ce modèle est *l'algorithme du cycle d'échange optimal - top trading cycle-* introduit par David Gale, qui se produit entre un sous-ensemble de propriétaires que l'on peut ordonner de sorte que, chaque propriétaire préfère la maison de son successeur à toutes les autres maisons qui existent. On peut dire alors que l'échange dans ce modèle se fait via un système de troc.

Algorithme du cycle d'échange optimal

- chaque agent fait une offre à l'agent possédant sa maison la plus préférée ;
- si un propriétaire n'a aucune maison qu'il préfère à la sienne, alors il fait une offre à lui-même et forme par conséquent, un cycle avec lui-même ;
- affecter chaque agent appartenant à un cycle à la maison de l'agent vers lequel il fait offre ;
- supprimer les agents et les maisons du cycle de la procédure.

On poursuit le même processus jusqu'à ce que tous les propriétaires soient affectés par un cycle.

4.3 Exemples d'application

La théorie des mariages peut être adaptée à modéliser des situations concrètes en remplaçant l'ensemble des hommes et femmes par des individus, chacun appartenant à une population donnée.

4.3.1 Transplantation de reins

La transplantation de reins est le meilleur traitement de l'insuffisance rénale en phase terminale, dont les patients vivent 10 ans de plus que ceux qui sont sous dialyse. Il existe deux moyens d'obtenir un rein pour un malade en déficience rénale. Le premier moyen est le cas d'un donneur décédé, et le deuxième est celui d'un donneur vivant. Le don de rein est particulier car souvent on accepte de donner à un proche, mais on refuserait de vendre son rein ou de le donner à une autre personne. Il arrive dans certaines situations, que pour un malade, un donneur de ses proches est disposé à lui donner un rein, mais malheureusement, son rein n'est pas compatible avec le patient. Pour cela, le programme d'échange de reins est certainement l'application la plus importante des modèles d'échange avec des biens indivisibles, où les patients souhaitent échanger leurs donneurs incompatibles afin d'obtenir un grand succès en augmentant le nombre de transplantations de reins et sauver ainsi le plus grand nombre possible de patients, de sorte que la solution soit de taille maximale sous certains critères de priorité. Le premier programme d'échange a été mis en place en Corée du Sud en 1991 et les Pays-Bas ont lancé le premier programme national en 2004 [16], dont la caractéristique la plus importante de ce système d'échange est que les longueurs des cycles d'échange soient limitées (tel est le cas des Pays-Bas qui limitent leurs cycles à quatre paires de *patients-donneurs*) suite à la contrainte que peut imposer ce système : *opérer simultanément des donneurs nécessite autant de salles, de matériel, et d'équipes chirurgicales*. Le problème d'échange des reins est unilatéral, cela est dû au fait que chaque candidat patient et son candidat donneur initial soient considérés comme

un seul élément. Une caractéristique extrême pour ce modèle d'appariement consiste en l'absence d'intervention d'une transaction financière.

L'algorithme de Gale-Shapley a tenu le rôle central auquel revient le mérite théorique sur lequel s'appuient les travaux d'Irving Roth pour la résolution de ce problème concret [8].

Selon Roth le théoricien des jeux, au marché d'appariement, les participants ne peuvent pas simplement choisir ce qu'ils veulent, mais ils doivent aussi être choisis. Il a commencé en collaboration avec Sönmez et Ünver par la modification de l'algorithme du cycle d'échange optimal présenté par Gale en 1970.

Imaginons deux paires incompatibles entre un donneur et un receveur, si chaque donneur était compatible avec le patient de l'autre paire, un échange bénéfique serait possible, et les deux patients recevraient un donneur compatible, comme l'illustrent les figures suivantes :

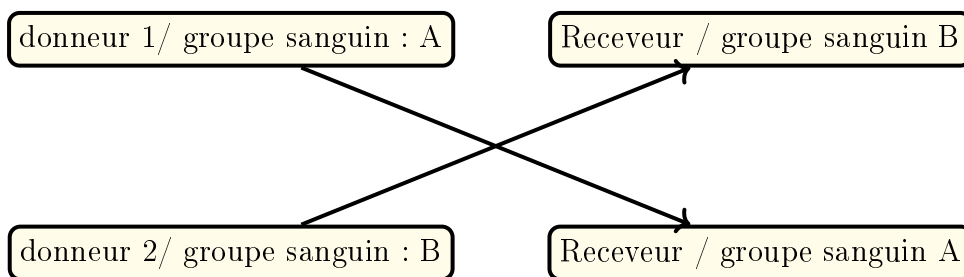


FIGURE 4.1 – Deux paires d'échange

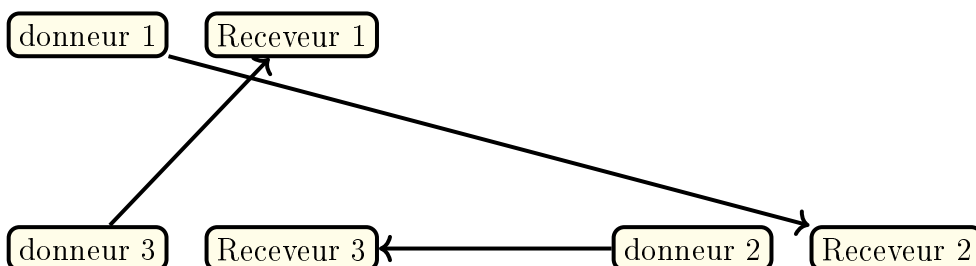


FIGURE 4.2 – Cycle de 3 paires d'échange

Définition 4.1. On peut représenter ce problème par la construction d'un graphe non orienté, dont les sommets représentent les paires incompatibles (donneur-malade), et ses arêtes, les compatibilités mutuelles entre les paires.

Le but est alors de trouver un couplage maximum dans ce graphe.

Définition 4.2. Soient le graphe $G=(X,E)$, et M un couplage ;

- Une chaîne est dite **alternée**, ou, **M -alternée**, si elle est élémentaire et ses arêtes sont alternativement dans M et dans $E \setminus M$.
- Une chaîne **augmentante** (**M -augmentante**) est une chaîne alternée de longueur impaire, dont les extrémités ne sont pas saturées par M , ie., **M -insaturées**.

Théorème 4.1. [2] (Berge) : un couplage M dans un graphe $G=(X,E)$ est maximum, si et seulement si, il n'existe pas de chaîne M -augmentante.

Algorithme 6 couplage maximum

Entrée un graphe $G=(X,E)$, un couplage M

Sortie un couplage maximum M

$p :=$ chaîne augmentante

si $p = \emptyset$ **alors**

 Retourner M

si non

 Améliorer M

fin si

Définition 4.3. La **contraction** d'une arête xy dans un graphe G , consiste à **retirer** l'arête en question et **fusionner** les deux sommets x et y . Le sommet résultant est adjacent aux sommets voisins de x et de y .

Algorithme blossom (Algorithme des fleurs et des pétales) :

Contraction du blossom

Soient $G=(X,E)$ un graphe et M un couplage de G .

Définition 4.4. *Un blossom est un cycle de longueur impaire avec $2k+1$ arêtes, dont k arêtes sont dans M .*

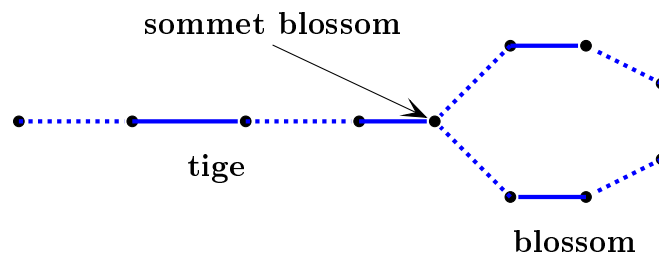


FIGURE 4.3 – Fleur

Soit G' le graphe obtenu en contractant toutes les arêtes du blossom dans G , et soit M' le couplage de G' obtenu.

Théorème 4.2. [2] *G admet un chemin augmentant pour M , si et seulement si G' admet un couplage augmentant pour M' .*

Résultat

*L'algorithme des fleurs et des pétales permet de trouver rapidement une solution au problème de départ de **transplantation de reins** par une suite d'opérations simples*

Algorithme 7 blossom

Entrée (G, M)

Sortie couplage maximum

si chemin augmentant est trouvé **alors**

 Stop

si non

 trouver un blossom B

 contracter B et retourner G'

fin si

$p :=$ chemin augmentant (G', M')

si $p = \emptyset$ **alors**

 le couplage actuel est maximum

si non

 Eclatement (p)

fin si

4.3.2 Admission dans les écoles primaires, collèges, lycées et universités

Suite à l'augmentation des effectifs dans le secteur de l'éducation, la communauté scientifique a adopté un mécanisme fondé sur la théorie de l'appariement qui vise à l'organisation efficace de ces taux élevés.

Le problème de base de ce modèle d'appariement est d'affecter des élèves ou étudiants aux écoles primaires, collèges, lycées ou établissements d'enseignement supérieur en tenant compte à la fois de leurs préférences et des critères d'acceptation fixés par ces établissements.

Ce problème est identique à celui du mariage, dont l'algorithme d'acceptation différée assure la stabilité, l'efficacité et la sincérité pour les participants. La seule différence est la prise en considération des capacités d'accueil de chaque établissement dans le choix de garder temporairement ou rejeter un élève/étudiant à une étape donnée.

Formellement, on a :

1. Un ensemble fini d'élèves/ étudiants : $I = \{ i_1, \dots, i_n \}$;
2. Un ensemble fini d'écoles/universités : $E = \{ e_1, \dots, e_m \}$
3. Un vecteur de capacité $q = (q_{e_1}, \dots, q_{e_m})$
4. Un vecteur d'ordre de préférences strict pour les élèves/étudiants sur l'ensemble des écoles/universités : $\theta = (\succ_{i_1}, \dots, \succ_{i_n})$
5. Un vecteur d'ordre de priorités strict pour les écoles/universités concernant les élèves/étudiants $(p_{e_1}, \dots, P_{e_m})$

Affectation dans les universités

Ce modèle consiste en l'affectation des élèves de terminale aux établissements du supérieur. Il a d'abord été décrit et résolu par Gale et Shapley, puis par Roth (1985) qui lui a donné une dimension supplémentaire, en supposant que les préférences des universités soient définies sur les groupes d'étudiants plutôt que de les considérer séparément les uns à la suite des autres.

- En *Espagne*, les notes de passage ont été calculées par la procédure d'acceptation différée, qui propose aux étudiants des listes de préférences limitées et des classements d'universités basés sur les notes des étudiants et les résultats d'entrée [16].
- En *Hongrie*, environ 100.000 à 150.000 étudiants sont affectés au programme d'admission à l'enseignement supérieur hongrois, chaque année. À partir de 1996, une heuristique basée sur la procédure d'acceptation différée proposée par l'université a été utilisée, et en 2007, cette procédure a été mise en œuvre avec quelques ajustements pour tenir compte des particularités [16].
- **Le problème d'APB** [15]
Admission post-bac (APB) est le nom de la plateforme numérique publique qui permet d'affecter les lycéens aux universités en France. Dans ce mécanisme l'affectation est réalisée par un algorithme qui vise à satisfaire au mieux les préférences des candidats tout en respectant les contraintes des formations et les règles de priorité des universités. Cet algorithme inclut la méthode mathématique de Gale-Shapley pour la résolution du problème des mariages stables

Affectation dans les écoles

Le modèle d'admission dans les universités de Gale et Shapley (1962) a joué un rôle déterminant pour les procédures d'affectation des élèves dans les écoles, appelé problème de *la carte scolaire*. Ce problème est simple, il est constitué d'un ensemble d'élèves qui doivent être affectés dans des écoles, chaque école ayant un nombre limité de places. Les parents des enfants révèlent leurs vœux sur la future école de leurs enfants selon un ordre de préférences sur les écoles et chaque école a également un ordre de préférences sur les élèves. Il est supposé que ces préférences sont strictes.

- **Le problème d'AFELNET** [20]

La procédure d'acceptation des élèves dans les lycées en France '*Affelnet*' (AFFec-tation des ELèves par le NET) est un algorithme qui permet de classer les vœux d'orientation de tous les élèves afin de leur proposer une place dans un lycée. Il représente un mécanisme similaire à la procédure d'acceptation différée *école-proposant* définie par Gale et Shapley. Néanmoins, à la théorie de répartition initiale, deux caractéristiques importantes sont ajoutées, la première est l'existence d'un bonus premier vœu pour les écoles classées en premier sur la liste des préférences, et la deuxième consiste en la limite sur la mémoire de vœux qu'ils peuvent lister. *Affelnet* se fonde généralement sur trois principaux critères : géographique, scolaire et social.

4.4 Réalisation pratique

Lors de la réalisation de cette étude, nous avons été amenés à utiliser un logiciel pour l'implémentation de l'algorithme de Gale-Shapley. Nous avons pu le programmer à l'aide du langage Python.



Python est un langage de programmation multiplateforme, gratuit et open-source. Il est l'un des langages de programmation les plus intéressants du moment. Python a été créé par le programmeur hollandais Guido van Rossum en 1990 dans le but d'inventer un successeur

au langage ABC. Il tire son nom de la série anglaise 'Monty Python's Flying Circus' dont G. van Rossum est fan. Depuis sa création, le succès du langage n'a cessé de croître, jusqu'à porter parmi les dix langages de programmation les plus populaires.

Avantages de Python

- Sa syntaxe est conçue pour être très simple, elle est combinée à des types de données évolués (listes, dictionnaires,...) ce qui conduit à des programmes à la fois très compacts et très lisibles.
- Python est le langage qui possède une bibliothèque étendue avec le plus de librairies, d'outils et de sources pour faire de la Machine Learning, du Big Data et de la Data Science.
- Presque toutes les tâches effectuées en Python nécessitent moins de codage que lorsque la même tâche est effectuée dans d'autres langages.
- Le débogage sous python est plus facile.
- La simplicité du langage et les bibliothèques étendues rendent les programmeurs plus productifs que d'autres langages comme Java et C++.
- Il possède une très grande communauté d'utilisateurs.

Inconvénients de Python

- L'exécution d'un code Python se fait ligne par ligne ce qui la rend lente. Cependant, il compense sa lenteur par sa versatilité.
- Python est un langage faible pour le développement mobile.
- Couches d'accès aux bases de données sous-développées. Par conséquent, il est souvent moins appliqué dans les grandes entreprises.

L'essai sur quelques exemples a été fructueux. Ci-dessous on donnera l'essai sur l'exemple de la figure 2.1 - (chapitre 1).

Algorithme de Gale-Shapley

```
import collections

Liste_Pref_Hommes = {
    "h1" : ['f1', 'f2', 'f8', 'f7', 'f3', 'f5', 'f4', 'f6'],
    "h2" : ['f2', 'f4', 'f5', 'f1', 'f7', 'f6', 'f3', 'f8'],
    "h3" : ['f2', 'f5', 'f1', 'f3', 'f4', 'f6', 'f7', 'f8'],
    "h4" : ['f5', 'f4', 'f2', 'f1', 'f3', 'f6', 'f8', 'f7'],
    "h5" : ['f3', 'f2', 'f1', 'f8', 'f4', 'f7', 'f5', 'f6'],
    "h6" : ['f8', 'f1', 'f5', 'f7', 'f6', 'f3', 'f4', 'f2'],
    "h7" : ['f7', 'f6', 'f5', 'f1', 'f3', 'f4', 'f8', 'f2'],
    "h8" : ['f7', 'f2', 'f1', 'f3', 'f4', 'f5', 'f6', 'f8']
}

Liste_Pref_Femmes = {
    "f1" : ['h3', 'h6', 'h4', 'h5', 'h1', 'h2', 'h7', 'h8'],
    "f2" : ['h6', 'h1', 'h5', 'h8', 'h3', 'h2', 'h4', 'h7'],
    "f3" : ['h2', 'h3', 'h7', 'h6', 'h8', 'h4', 'h5', 'h1'],
    "f4" : ['h1', 'h3', 'h8', 'h6', 'h5', 'h4', 'h2', 'h7'],
    "f5" : ['h5', 'h1', 'h3', 'h8', 'h2', 'h7', 'h4', 'h6'],
    "f6" : ['h8', 'h4', 'h2', 'h5', 'h6', 'h3', 'h1', 'h7'],
    "f7" : ['h5', 'h1', 'h3', 'h6', 'h2', 'h8', 'h7', 'h4'],
    "f8" : ['h2', 'h4', 'h1', 'h5', 'h8', 'h6', 'h3', 'h7']
}

# Pour afficher les listes :
print("\n Les hommes et leur listes de préférences : \n \n", Liste_Pref_Hommes, "\n")
print("Les femmes et leur listes de préférences : \n \n" , Liste_Pref_Femmes, "\n" )
print (" \n Début de la procédure d'acceptation différée : \n ")

#Garder une trace des personnes qui pourraient se retrouver ensemble.
engagements = [ ]

# Les hommes libres qui vont proposer des mariages
hommes_libres= [ ]
```

```
print("\n La liste des hommes libres est :", hommes_libres)

def init_hommes_libres() :
    """Initialiser les tableaux des femmes et hommes pour représenter
        qu'ils sont tous initialement libres et non engagés"""
    for homme in Liste_Pref_Hommes.keys() :
        hommes_libres.append(homme)

def couplage_stable() :
    """ On commence à chercher le couplage stables
    """
    while (len(hommes_libres) > 0) :
        for homme in hommes_libres :
            begin_couplage(homme)

def begin_couplage(homme) :
    """Trouver la femme possible la plus préférée pour l'homme"""
    print("\n Pour l'homme % s" %(homme))
    for femme in Liste_Pref_Hommes[homme] :

        couple_HF=[couple for couple in engagements if femme in couple]
        if (len(couple_HF)==0) :
            #les acceptations temporaires des demandes de mariages
            engagements.append([homme, femme])
            hommes_libres.remove(homme)
            print('%s n\'est plus un homme libre, il est maintenant engagé à
                % s' %(homme, femme))
            print("La liste des engagements est maintenant :", engagements)
            break
        elif (len(couple_HF) > 0) :
            print('%s Reçoit la proposition de %s' % (femme, homme))
            #comparer le classement de l'homme avec lequel la femme est en
```

```
couple à l'homme possible qui lui propose un mariage
homme_actuel=Liste_Pref_Femmes[femme].index(couple_HF[0][0])
homme_possible=Liste_Pref_Femmes[femme].index(homme)

if (homme_actuel < homme_possible) :
    print("Le mariage de %s et %s est instable " % (femme,
    homme))
    print('Elle préfère être avec %s'%(couple_HF[0][0]))
else :
    print(' %s est plus préféré que %s'%(homme, couple_HF[0][0]))
    print('Par conséquent, %s est à nouveau libre, %s et %s
    sont temporairement en couple'%(_HF[0][0], homme, femme))

    #retirer l'homme engagé de la liste des hommes libres
    hommes_libres.remove(homme)

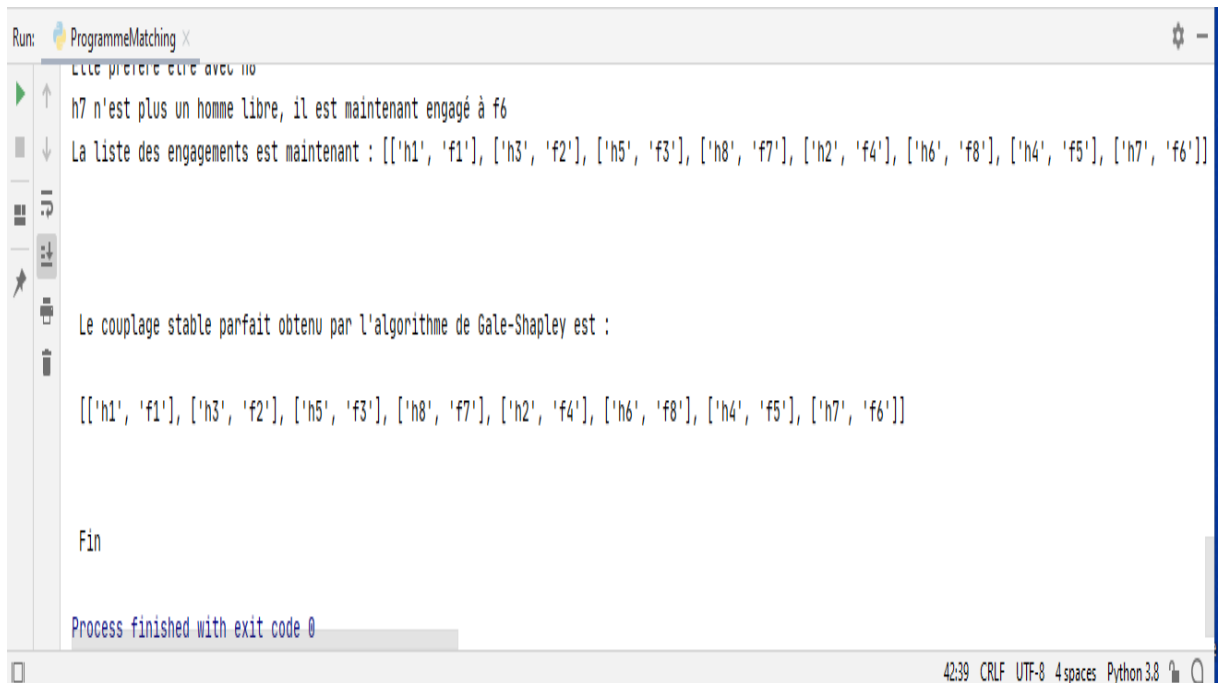
    #ajouter l'homme rejeté à la liste des hommes
    célibataires.
    hommes_libres.append(couple_HF[0][0])

    #former le nouveau couple
    couple_HF[0][0] = homme
    break

def main() :
    init_hommes_libres()
    print(hommes_libres)
    couplage_stable()
    print(" \n \n Le couplage stable parfait obtenu par l'algorithme de Gale-Shapley
    est :\n \n", engagements)
    print(" \n \n Fin")

main()
```

L'exécution de notre programme affichera :



```
Run: ProgrammeMatching x
Cette premiere etre avec no
h7 n'est plus un homme libre, il est maintenant engagé à f6
La liste des engagements est maintenant : [['h1', 'f1'], ['h3', 'f2'], ['h5', 'f3'], ['h8', 'f7'], ['h2', 'f4'], ['h6', 'f8'], ['h4', 'f5'], ['h7', 'f6']]

Le couplage stable parfait obtenu par l'algorithme de Gale-Shapley est :

[['h1', 'f1'], ['h3', 'f2'], ['h5', 'f3'], ['h8', 'f7'], ['h2', 'f4'], ['h6', 'f8'], ['h4', 'f5'], ['h7', 'f6']]

Fin

Process finished with exit code 0
```

FIGURE 4.4 – Résultat de l'exécution du programme

Conclusion générale

Ce travail consiste en l'étude du problème des mariages stables qui est une variante du problème d'affectation.

On a pu constater que les graphes constituent une méthode qui permet de modéliser, analyser et résoudre une grande variété de problèmes. Notre étude consiste à donner un certain nombre d'outils de la théorie des graphes ainsi qu'une famille d'algorithmes pour la résolution du problème des mariages stables et ses variantes.

Comme réalisation pratique, nous avons utilisé le langage de programmation Python pour l'implémentation de l'algorithme de Gale-Shapley.

Nous avons également montré l'importance pratique de ce domaine de recherche en illustrant par quelques exemples concrets.

Bibliographie

- [1] Alain Bretto, Alain Faisant, François Hennecart, *Eléments de théorie des graphes*, Springer, 2012.
- [2] Amy Shoemaker, Sagar Vate, Edmond's blossom algorithm, *CME*, 2016.
- [3] Antoine Gournay, *Théorie des graphes*, Institut de Mathématiques, Université de Neuchâtel Suisse, Septembre, 2013.
- [4] Armen S. Asratian, Tristan MJ Denley, Roland Häggkvist, *Bipartite graphs and their applications*, volume 131, Cambridge university press, 1998.
- [5] Dan Gusfield, The structure of the stable roommate problem : efficient representation and enumeration of all stable assignments, *SIAM Journal on Computing*, 1988.
- [6] Dan Gusfield, Three fast algorithms for four problems in stable marriage, *YALEU/DCS/TR-407*, July, 1985.
- [7] David Manlove, *Algorithmics of matching under preferences*, volume 2, World Scientific, 2013.
- [8] Françoise Forges, Guillaume Haeringer, Vincent Iehlé, Appariement : des modèles de Lloyd Shapley à la conception de marchés d'Alvin Roth, *Revue d'économie politique*, 2013.
- [9] Guillaume Ratier, On the stable marriage polytope, *Discrete Mathematics*,, 1996.
- [10] Jacques Labelle, *Théorie des graphes*, Outremont [Québec], Modulo, 1981.

- [11] Kazuo Iwama, Shuichi Miyazaki, Kazuya Okamoto, Stable roommates problem with triple rooms, In Proc, 10th KOREA-JAPAN joint workshop on algorithms and computation (WAAC 2007), pages 105–112, 2007.
- [12] Marilda A., Oliveira Sotomayor, Alvin E. Roth, Two Sided Matching A Study in Game Theoretic Modeling and Analysis, Cambridge University Press, 1992.
- [13] Michael St.Jules, Stable Matchings, November 16, 2013.
- [14] Mohamed Siala, Barry O’Sullivan, Rotation-based formulation for stable matching, In International Conference on Principles and Practice of Constraint Programming, pages 262–277, Springer, 2017.
- [15] Olivier Beaud, Francois Vatin, Orientation et réussite des étudiants, Commentaire, (3),687–698, 2018.
- [16] Péter Biró, Applications of matching models under preferences, 2017.
- [17] Robert W Irving, An efficient algorithm for the “stable roommates” problem, Journal of Algorithms, 1985.
- [18] Robert W. Irving, Paul Leather, Dan Gusfield, An efficient algorithm for the “optimal” stable marriage, Journal of the ACM (JACM), 1987.
- [19] Robert W Irving and Sandy Scott, The stable fixtures problem a many to many extension of stable roommates, *Discrete Applied Mathematics*, 2007.
- [20] Victor Hiller, Olivier Tercieux, Choix d’écoles en france : une évaluation de la procédure Affelnet, 2012.