

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A.Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



MÈMOIRE

En vue d'obtention du diplôme Master en Recherche Opérationnelle
Option : Modélisation Mathématique et Technique d'Aide à la Décision

Thème

Optimisation dans les Graphes Probabilistes

Présenté par :

M^{elle} Laidi Noria

M^{elle} Medjdoub Lynda

Devant le jury composé de :

Président *M^r* Khimoum Nouredine

Promoteur *M^r* Kabyle Kamal

Examineurs *M^{lle}* Ziane Yasmina

M^r Kassa Farouk Ali

Université de Béjaïa, 2016

Remerciements

Au terme de ce travail, il est agréable d'adresser nos remerciements à Allah le tout puissant de nous avoir donné le courage de finir ce modeste travail.

Nous souhaitons tout d'abord remercier notre promoteur, Monsieur, KABYL Kamal pour nous avoir suivi qui nous a encadré avec patience durant la réalisation de ce travail de fin d'études. Ses conseils nous ont été bien utiles, notamment pour la rédaction de ce mémoire.

Nous remercions vivement les membres de jury :Mr N. Khimoum pour l'honneur qu'il nous a fait en acceptant de présider le jury et à Melle Y. Ziane et Mr F. Kassa pour avoir accepté d'examiner notre travail.

Nos sincère gratitude va vers tous ceux qu'ont prêté mains fortes, et qu'ont participé de loin ou de près à la réalisation de ce travail.

Noria
Lynda

Dédicace

Je dédie ce modeste travail aux :

La personne devant laquelle tous les mots de l'univers sont incapables d'exprimer l'affection et l'amour que j'éprouve envers elle, à l'être qui m'est le plus cher, à ma douce mère. Puisse ce travail être la récompense de tes soutiens moraux et sacrifices. Mère, si tu savais combien je t'aime.

Mon cher père qui a payé plus de vingt deux années d'amour et de sacrifices que tu t'es imposé pour assurer mon bien être et mon éducation et le prix de ma façon de penser.

Père, je te remercie d'avoir fait de moi une femme.

Ma chère grande mère, que le Dieu te protège et te garde pour nous.

Mes frères : Nadjim et Mounir, mes adorables sœurs : Lilia, et la petite Rania

A mes amis qui restent à mes côtés au milieu des temps de sable et en basse mer en particulier Lamia, Ouided , Mehdi et Nabil.

A toute la famille LAIDI ET KHELFAOUI.

A celle avec qui j'ai réalisé cet humble travail Lynda et sa famille.

A celui qui m'a appris à aimer cette science.

A celui qui m'a appris à faire des maths et R.O.

Noria

Dédicace

Je dédie ce modeste travail aux :

Personnes les plus chères à mes yeux, mes parents.

Mon frère Farouk .

Ma chère sœur Mounia , son mari Youcef et son fils Adem.

Ma chère sœur Nassima et son mari Nouradin.

A toute la famille MEDJDOUB.

Mes amis et tous les gens qui m'aiment.

Ma chère binôme Noria.

Tous ceux qui sont proches de mon cœur et dont je n'ai pas cité le nom.

Lynda

Table des matières

Introduction générale	2
1 Notations et notions de bases	3
Notations et notions de bases	3
1.1 Graphes	3
1.1.1 Définitions et terminologies	4
1.2 Chaînes, Chemins, Cycles et Circuits	8
1.2.1 Chaînes et cycles	8
1.2.2 Chemins et circuits	9
1.3 Connexité et forte connexité dans les graphes	10
1.3.1 Connexité	10
1.3.2 Forte connexité	11
1.4 Distance dans les graphes	13
1.5 Opération sur les graphes	14
1.5.1 Somme Cartésienne de deux graphes	14
1.5.2 Produit Cartésien de deux graphes	15
1.5.3 Morphisme de graphe	15
1.5.4 Isomorphisme de graphe	15
1.5.5 Subdivision dans les graphes	16
1.6 Représentations matricielles	17
1.6.1 Matrice d'adjacence	17

1.6.2	Matrice d'incidence	18
1.7	Complexité des algorithmes	19
1.8	Fermeture transitive d'un graphe	19
1.8.1	La transitivité	19
1.8.2	Caractérisation de la fermeture transitive d'un graphe	20
1.9	Quelques graphes particuliers	24
1.9.1	Graphe complet	24
1.9.2	Graphe biparti	24
1.9.3	Graphe acyclique	25
1.9.4	Hypercube	25
1.9.5	Arbre	26
1.9.6	Arborescence	26
2	Graphes Probabilistes et Chaînes de Markov	28
	Graphes Probabilistes et Chaînes de Markov	28
2.1	Les graphes probabilistes	28
2.2	Matrice de transition	29
2.2.1	État probabiliste	29
2.2.2	État stable	30
2.3	Chaînes de Markov	31
2.3.1	Quelques rappels	31
2.3.2	Classification des états	32
2.3.3	Lien entre chaînes de Markov et théorie des graphes	33
2.4	Modèles des graphes probabilistes	35
2.4.1	Graphes statiques stochastiques	35
2.4.2	Graphes dynamiques stochastique	35
2.4.3	Réseaux bayésiens	36
2.4.4	Processus de décision markoviens	38
2.4.5	Jeux probabilistes et déterministes	38

3	Quelques Algorithmes d'optimisation dans les graphes probabilistes	40
	Quelques Algorithmes d'optimisation dans les graphes probabilistes	40
3.1	Quelques problèmes dans les graphes probabilistes	40
3.1.1	Problème de l'arbre couvrant de poids minimum	40
3.1.2	Problème du plus long chemin	43
3.1.3	Problème d'affectation	45
3.1.4	Problème du flot	47
4	Partie d'application	49
	Partie d'application	49
4.1	Problème de l'arbre couvrant de poids minimum	49
4.2	Problème de plus long chemin	57
4.3	Problème d'affectation optimale de programmes de télévision :	60
4.4	Application sur MATLAB	63
	Bibliographie	1

Table des figures

1.1	Deux représentations graphiques d'un même graphe	4
1.2	Les arêtes e et f n'ont pas de point commun	4
1.3	Graphe non orienté	5
1.4	Graphe orienté	5
1.5	Sous-graphes	6
1.6	Graphes partiels	6
1.7	Graphe Complémentaire	6
1.8	Graphe orienté	8
1.9	Chaîne	9
1.10	Cycle	9
1.11	Chemin	10
1.12	Circuit	10
1.13	Graphe ayant trois composantes connexes	11
1.14	Composantes fortement connexes	11
1.15	Somme cartésienne de deux graphes	14
1.16	Produit cartésien de deux graphes	15
1.17	Le cas ou le graphe est orienté	17
1.18	Le cas ou le graphe est non orienté	18
1.19	Graphe Complet	24
1.20	Graphe biparti Complet	24

1.21	Graphe acyclique	25
1.22	Q_0, Q_1, Q_2 et Q_3	25
1.23	Arbre	26
1.24	Arborescence	26
1.25	Mots reconnus par un graphe étiqueté	27
1.26	Graphe est valué	27
2.1	Graphe probabiliste	29
2.2	Graphes statiques stochastiques	36
2.3	Graphes dynamiques stochastique	37
2.4	Réseaux Bayésiens	38
2.5	Réseaux markoviens	38

Liste des tableaux

1.1	Listes de prédécesseurs et successeurs	8
4.1	Tranche horaire par tranches d'âge	60
4.2	Proposition une réparation de programmes de télévision	63

Introduction générale

D e manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques, . . .

La théorie des graphes est utilisée dans un grand nombre de disciplines (mathématiques, physique, économie, etc.). Les recherches en théorie des graphes sont essentiellement menées par des informaticiens, du fait de l'importance des aspects algorithmiques (recherche de solutions). Elle constitue l'un des instruments les plus courants et les plus efficaces pour résoudre des problèmes discrets posés en Recherche Opérationnelle (RO).

Il existe plusieurs types de graphe, dans notre travail nous allons étudier les graphes Probabilistes qui sont des graphes orientés et pondérés, dans lequel chacune des arêtes est représentée par un nombre noté (C_{ij}) borné entre 0 et 1.[2]

L'étude présentée dans ce document porte essentiellement sur l'optimisation dans les graphes probabilistes.

Ce mémoire est constitué d'une introduction générale, de quatre chapitres, d'une conclusion et d'une bibliographie

- Dans le premier chapitre, nous allons rappeler brièvement quelques notions sur la théorie des graphes et certaines définitions essentielles.
- Dans le deuxième chapitre, en introduisant des concepts de chaîne de Markov et le lien avec la théorie des graphes, comme nous décrirons précisément les différents modèles des graphes probabiliste.
- Dans le troisième chapitre, en présentant quelques problèmes d'optimisation et techniques de résolutions pour chaque problème.
- Enfin dans le quatrième chapitre on traite quelques problèmes réels après avoir les modéliser en utilisant des algorithmes d'optimisation.

1

Notations et notions de bases

L'objet de ce chapitre est de présenter brièvement, certaines définitions de bases ou concepts généraux sur les éléments de la théorie des graphes que nous allons utiliser par la suite.

1.1 Graphes

On appelle graphe $G = (X, E)$ la donnée d'un ensemble de points $X = \{x_1, x_2, \dots, x_n\}$ appelés sommets du graphe G et n le nombre de sommets de ce graphe défini par le cardinal de X . Si $(n = |X|)$ on dit que le graphe G est d'ordre n d'une partie de E symétrique $((i, j) \in E \iff (j, i) \in E)$ dont les éléments sont appelés arêtes.[15]

En présence d'une arête $e = (i, j)$ qui peut être notée simplement ij , on dit que i et j sont les extrémités de e , que e est incidente en i et en j , et que j est un successeur ou voisin de i (et vice versa).

- On dit qu'un graphe est sans boucle si E ne contient pas d'arête de la forme (i, i) , c'est-à-dire joignant un sommet à lui même.

- Un graphe ne possédant pas de boucle ni d'arêtes parallèles (deux arêtes distinctes joignant la même paire de sommets) est appelé graphe simple.

Graphiquement, les sommets peuvent être représentés par des points et l'arête $e = (i, j)$ par un trait reliant i à j . On notera que la disposition des points et la longueur des traits n'a aucune importance. Seule l'incidence des différentes arêtes et sommets compte. Les deux graphes de la Figure 1.1 sont identiques.

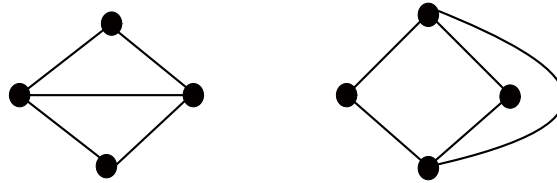


FIGURE 1.1 – Deux représentations graphiques d'un même graphe

Dans le tracé graphique d'un graphe, deux arêtes peuvent sembler avoir une intersection en un point qui n'est pas un sommet. C'est le cas ; par exemple, des arêtes e et f du graphe de la Figure 1.2. De telles arêtes peuvent être vues comme étant placées dans des plans différents et n'ayant donc aucun point commun.

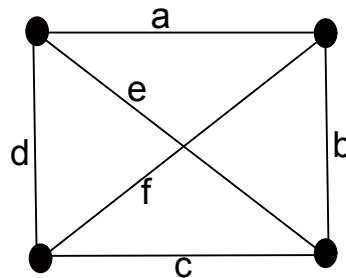


FIGURE 1.2 – Les arêtes e et f n'ont pas de point commun

1.1.1 Définitions et terminologies

Graphes non orientés

On dit que G est un graphe non orienté, si la précision de sens de lien (i, j) et la distinction entre extrémité initiale et extrémité terminale ne jouent aucun rôle. On appelle tout élément $(i, j) \in E$ une arête, qui est représentée graphiquement par un segment sans flèche liant les deux nœuds i et j . La Figure 1.3 montre un graphe non orienté.

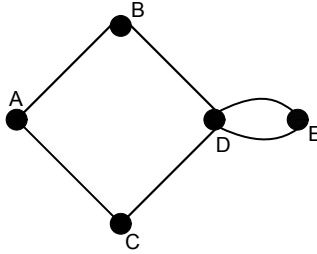


FIGURE 1.3 – Graphe non orienté

Graphes orientés

On dit que G est un graphe orienté s'il y a une distinction entre les liens (i, j) et (j, i) , c'est-à-dire $(i, j) \neq (j, i)$. Dans ce cas le lien est appelé un arc. On représente $e = (i, j)$ graphiquement par une flèche qui part de i pour joindre j qui sera la pointe de cette flèche. Dans ce cas, j sera appelé un successeur de i (ou i est un prédécesseur de j) et chaque sommet peut avoir plusieurs successeurs et plusieurs prédécesseurs. On appelle une boucle tout arc (i, i) dont ses extrémités se coïncident. La Figure 1.4 montre un graphe orienté.

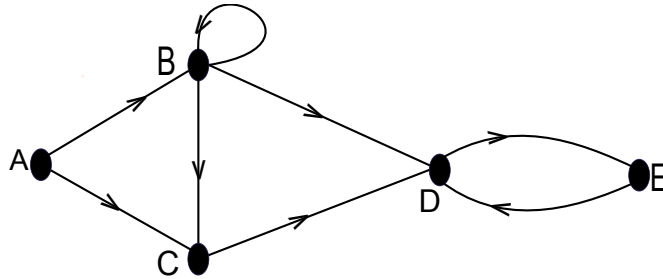


FIGURE 1.4 – Graphe orienté

Sous-graphes et graphes partiels

Soit $G = (X, E, f)$ un graphe et $Y \subset X$, le sous-graphe de G engendré par Y est le graphe dont :

- L'ensemble des sommets est Y ;
- L'ensemble des arêtes (arcs) sont les arêtes de G joignant deux sommets de Y ;
- La fonction d'incidence est la restriction de f aux arêtes (arcs) de E .

La Figure 1.5 montre un sous-graphes.

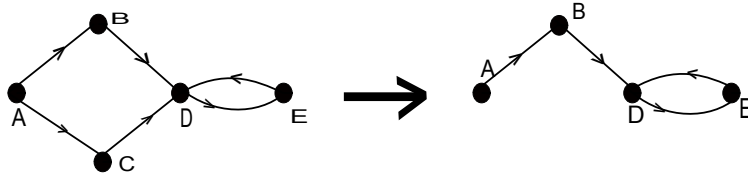


FIGURE 1.5 – Sous-graphes

Soit $G = (X, E, f)$ un graphe et $F \subset E$, le graphe partiel de G engendré par F est le graphe dont :

- L'ensemble des sommets est X ;
- L'ensemble des arêtes (arcs) est F ;
- La fonction d'incidence est la restriction de f à F .

La Figure 1.6 montre un graphes partiels.

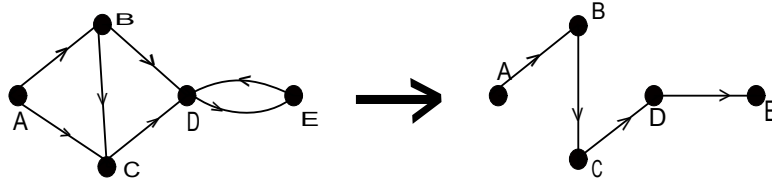


FIGURE 1.6 – Graphes partiels

Graphe complémentaire

Soit $G = (X, E)$ un graphe, $\overline{G} = (X, \overline{E})$ est le graphe complémentaire de G , si deux sommets adjacents dans G ne le sont pas dans $\overline{G}$, et inversement.

Si on considère le sous-graphe E de G comprenant les arêtes $\{AB, AE, BC, CD, DF, EF\}$. Alors le graphe complémentaire de E , noté $\overline{E}$, est le graphe comprenant les sommets $\{A, B, C, D, E, F\}$ et l'ensemble d'arêtes $\{AC, AD, AF, BD, BE, BF, CE, CF, DE\}$.

La Figure 1.7 montre un graphes Complémentaire.

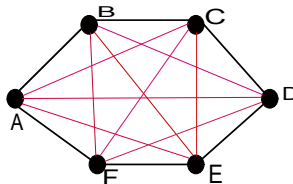


FIGURE 1.7 – Graphe Complémentaire

Degré d'un graphe

Dans le cas d'un graphe orienté, le degré sortant (semi-degré intérieur) d'un sommet x est le nombre d'arcs qui partent de x ; noté $d^+(x)$, et son degré entrant (semi-degré extérieure), noté $d^-(x)$, est le nombre d'arcs arrivant au sommet x . On a la relation :

$$\sum d^+ = \sum d^- = |E|$$

Dans le cas d'un graphe non orienté, le degré est le nombre d'arêtes rattachées au sommet x (les boucles comptent pour 2). On a la relation :

$$\sum d = 2|E|$$

Listes de successeurs et prédécesseurs

Pour chaque sommet i du graphe, on définit :

La liste de ses successeurs W^+ : liste des sommets y tel que l'arête (x, y) existe dans le graphe.

La liste de ses prédécesseurs W^- : liste des sommets y tel que l'arête (y, x) existe dans le graphe.

Un sommet sans prédécesseur est appelé une source.

Un sommet sans successeur est appelé un puits.

Voisinage d'un sommet

Si $xy \in E$, on dit que x est voisin de y et vice versa. On note $N(x)$ l'ensemble de tous les voisins de x :

$$N(x) = \{y \in X / xy \in E\}$$

De même on définit un voisinage fermé comme suit :

$$\overline{N(x)} = \{y \in X / xy \in E\} \cup \{x\}$$

Dans le cas orienté; comme pour la Figure 1.4, illustrée précédemment, on distingue entre l'ensemble des successeurs et l'ensemble des prédécesseurs qui sont donnés comme suit :

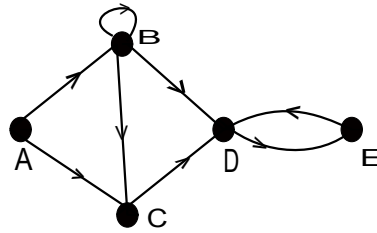


FIGURE 1.8 – Graphe orienté

Sommet	A	B	C	D	E
Les successeurs $W^+(x)$	B et C	C et D	D	E	D
Les prédécesseurs $W^-(x)$	/	A et B	A, B	B, C et E	D

TABLE 1.1 – Listes de prédécesseurs et successeurs

Relations d'adjacence

Dans un graphe G , une arête e reliant un sommet x à un sommet y est noté par XY et dans ce cas on dit :

✓ x et y sont adjacents ;

✓ x et y sont les extrémités de e , e incidente à x et y .

deux arêtes sont dites adjacentes si elles sont incidentes à un même sommet.

1.2 Chaînes, Chemins, Cycles et Circuits

1.2.1 Chaînes et cycles

- Une chaîne joignant les sommets i et j dans le graphe $G = (X, E)$ est une séquence d'arêtes de G telle que :

✓ La première (resp. dernière) arête de la séquence est adjacent à i (resp. j) par de ses extrémités est au second (resp. avant dernier) par son autre extrémité ;

✓ Chaque arête intermédiaire de la séquence est adjacente au précédent par une de ses extrémités et au suivant par l'autre extrémité.

Une chaîne est dit simple si la séquence d'arête qui la constitue ne comporte pas plusieurs fois le même élément.

Une chaîne est dit élémentaire si les sommets de G sont adjacents à deux arêtes de la chaîne aux plus. La Figure 1.9 montre une chaîne.

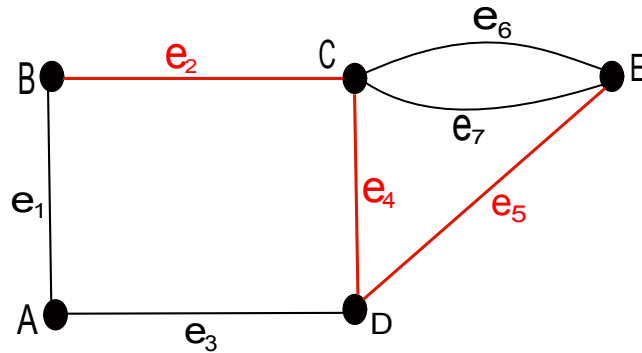


FIGURE 1.9 – Chaîne

• Un cycle est une séquence circulaire d'arêtes toutes distinctes telle que chaque arête de la séquence soit adjacente au précédente par une de ses extrémités et au suivante par l'autre longueur de chaîne et de cycle. La Figure 1.10 montre un cycle.

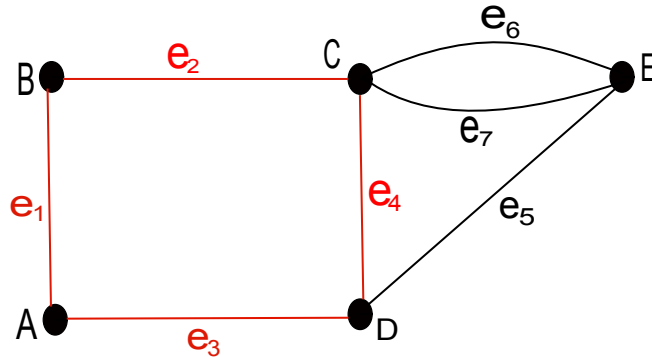


FIGURE 1.10 – Cycle

1.2.2 Chemins et circuits

• Un chemin joignant le sommet i au sommet j dans le graphe $G = (X, E)$ est une séquence d'arcs de G telle que :

- L'extrémité initiale du premier arc de la séquence est i ;
- L'extrémité de chacun des autres arcs coïncide avec l'extrémité terminale de l'arc précédent ;
- L'extrémité terminale du dernier arc de la séquence est j .

Un chemin est dit simple si la séquence d'arcs qui le constitue ne comporte pas plusieurs fois le même élément, Un chemin est dit élémentaire si les sommets de G sont adjacents à deux arcs du chemin au plus. La Figure 1.11 montre un chemin.

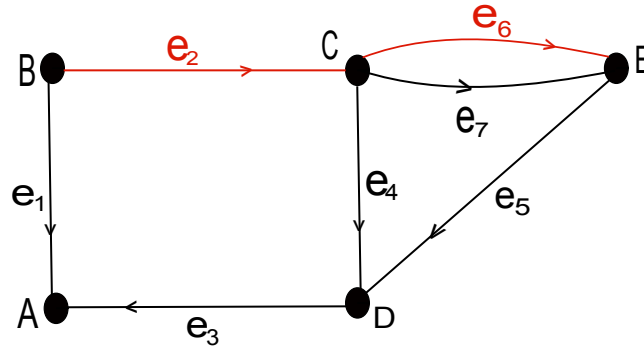


FIGURE 1.11 – Chemin

• Un circuit est une séquence circulaire d'arcs tous distincts telle que chaque arc de la séquence soit adjacent à l'arc précédent par son extrémité initiale et à l'arc suivant par son extrémité terminale. La Figure 1.12 montre un circuit

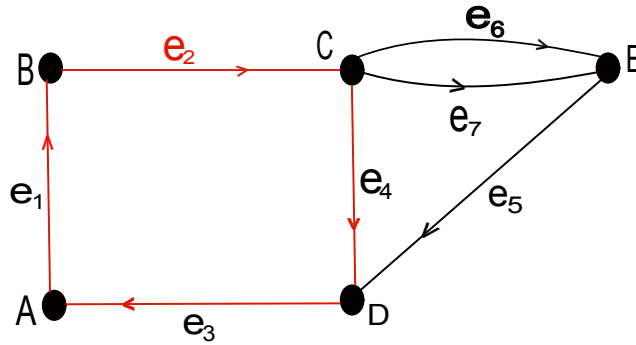


FIGURE 1.12 – Circuit

1.3 Connexité et forte connexité dans les graphes

1.3.1 Connexité

Un graphe $G = (X, E)$ est connexe si $\forall i, j \in X$, il existe une chaîne entre i et j .

• **Composantes connexes :** On appelle composante connexe le sous-ensemble de sommets tels qu'il existe une chaîne entre deux sommets quelconques. Un graphe est connexe s'il comporte une composante connexe maximale et une seule. Chaque composante connexe est un graphe connexe.

La Figure 1.13 montre un graphe ayant trois composantes connexes.

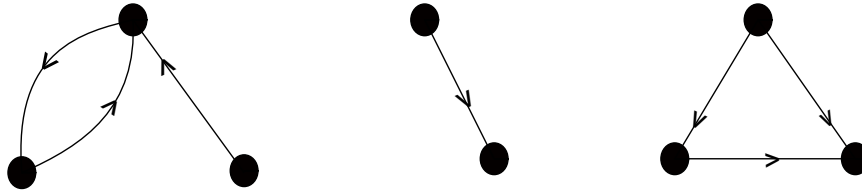


FIGURE 1.13 – Graphe ayant trois composantes connexes

- Un graphe ayant une seule composante connexe est appelé graphe connexe .

1.3.2 Forte connexité

Un graphe $G = (X, E)$ est fortement connexe si $i, j \in X$, il existe un chemin de i à j et un autre chemine de j à i .

• **Composante fortement connexe (cfc)** : est un sous-ensemble de sommets tel qu'il existe un chemin entre deux sommets quelconques. Une (cfc) maximale (cfcm) est un ensemble maximal de (cfc). Les différentes (cfcm) définissent une partition de X . Un graphe est fortement connexe s'il comporte une seule (cfcm). La Figure 1.14 montre deux composantes fortement connexe.

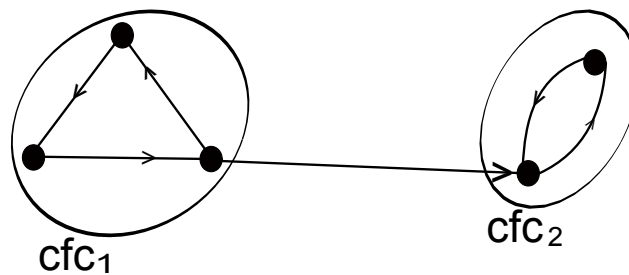


FIGURE 1.14 – Composantes fortement connexes

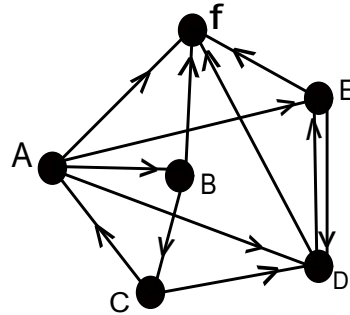
- **Algorithme de recherche composante fortement connexe (cfc)** :

Soit $G = (X, E)$ un graphe orienté.

1. Initialisation $k = 0$, $w = x$.
2. Choisir un sommet de w et le marque d'un signe ($\mp$).
3. Marquer tous les successeurs directs et indirects d'un signe (+).
4. Marquer tous les prédécesseurs directs et indirects d'un signe (-).

5. Poser $k = k + 1$, et c_k l'ensemble des sommets marquer d'un (+) et (-).
6. Poser $w = w - c_k$, et effacer toutes les marques.
7. Tester si $w = \emptyset$:
 Si oui, terminer ;
 Si non, aller en(2).
8. Le nombre de composantes fortement connexes est k et c_k sont les composantes fortement connexes de G .

Soit le graphe $G = (X, E)$ suivant :

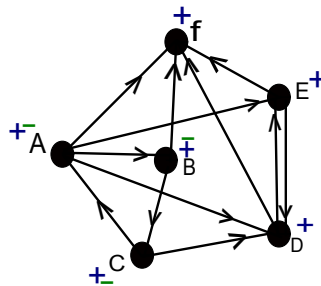


G est connexe. Il y a donc une seule composante connexe ; l'ensemble de tous ses sommets.

Nous allons d'abord trouver la composante fortement connexe qui contient le sommet A .

Rappel de l'algorithme :

- marquer par + et - le sommet A ;
- marquer par + chaque successeur non marqué + d'un sommet marqué + ;
- marquer par - chaque prédécesseur non marqué - d'un sommet marqué - ;
- quand on ne peut plus marquer, les sommets marqués + et - constituent la composante fortement connexe contenant A .



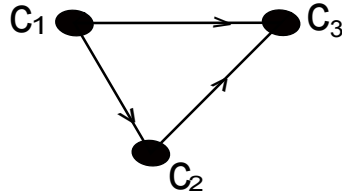
La composante fortement connexe qui contient A est $C_1 = \{A, B, C\}$.

On retire les sommets A , B et C et les arcs qui leur sont adjacents, puis on recommence

par exemple avec le sommet D .

On trouve que la composante fortement connexe qui contient D est $C_2 = \{D, E\}$. On retire ces sommets. Il n'y a plus que le sommet F . Ainsi, la troisième composante fortement connexe est $C_3 = \{F\}$.

Voici le graphe réduit de G :



1.4 Distance dans les graphes

- **Distance** : La distance entre deux sommets d'un graphe connexe (ou entre 2 sommets d'une même composante connexe d'un graphe non connexe) est le nombre minimum d'arcs (on dit aussi la longueur) d'une chaîne allant de l'un à l'autre. La distance entre x et y est notée par $d(x, y)$.

- **L'Excentricité** : Soit un graphe simple non orienté $G = (X, E)$, $\forall x \in X$ l'excentricité de x est :

$$exc(u) = \max\{d(x, y) : x, y \in X\}$$

Autrement dit :

- l'excentricité de x désigne la distance qui sépare x du sommet le plus éloigné de x dans G ;
- l'excentricité d'un sommet est la distance maximum de ce sommet aux autres sommets.

- **Centre** : On appelle centre d'un graphe, le sommet d'écartement minimal, (le centre n'est pas nécessairement unique). Par exemple, dans une clique, tous les sommets sont "centrés"

- **Diamètre** : Le diamètre d'un graphe est la plus grande distance entre deux

sommets quelconques de ce graphe. Le diamètre de G est :

$$D = \max\{exc(x) : x \in X\}$$

• **Rayon** : On appelle rayon d'un graphe G , noté $\rho(G)$, l'écartement d'un centre de G . Autrement dit : le rayon d'un graphe est le minimum des excentricités des différents sommets.

1.5 Opération sur les graphes

1.5.1 Somme Cartésienne de deux graphes

On appelle somme Cartésienne de deux graphes $G = (X(G), E(G))$ et $H = (X(H), E(H))$; notée GH , le graphe dont l'ensemble des sommets est le produit cartésien $X(G) \times X(H)$ et/ou deux sommets (x, x') et (y, y') sont adjacents si et seulement si l'une des propriétés suivantes est vérifiées :

$$\begin{aligned} x = y \text{ et } x'y' \in E(H); \\ \text{ou} \\ xy \in E(G) \text{ et } x' = y'. \end{aligned}$$

La Figure 1.15 montre la somme cartésienne de deux graphes :

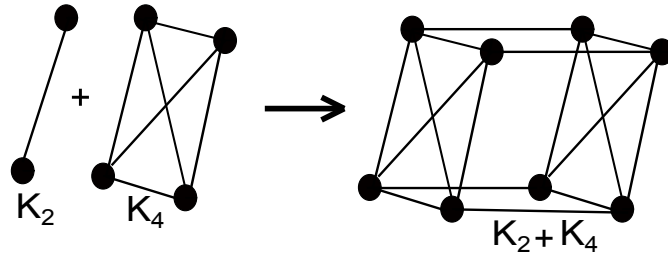


FIGURE 1.15 – Somme cartésienne de deux graphes

On note que le nombre de sommets dans $G \times H$ est $|X(G)| \cdot |X(H)|$, et que le nombre d'arêtes est : $|X(G)| \cdot |E(H)| + |X(H)| \cdot |E(G)|$.

1.5.2 Produit Cartésien de deux graphes

Le produit Cartésien de deux graphes $G = (X(G), E(G))$ et $H = (X(H), E(H))$ est le graphe note $G \times H$ ou $X(G \times H) = X(G) \times X(H)$. Deux sommets (x, y) et (x', y') sont adjacents si et seulement si $xx' \in E(G)$ et $yy' \in E(H)$.

La Figure 1.16 montre produit cartésien de deux graphes.

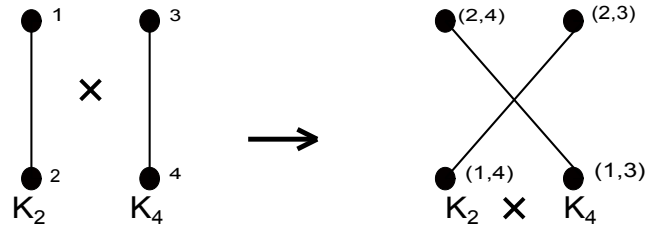
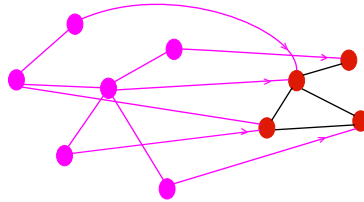


FIGURE 1.16 – Produit cartésien de deux graphes

1.5.3 Morphisme de graphe

Soient $G = (X, E)$ et $G' = (X', E')$ deux graphes. Un morphisme de graphes de G dans G' est une bijection $f : X \rightarrow X'$ telle que, $\forall (x_1, x_2) \in E \Rightarrow (f(x_1), f(x_2)) \in E'$. Plus simplement, f est un morphisme de graphes si l'image de toute arête de G est une arête de G' . S'il y a un morphisme de G dans G' , on dit que G “se projette” dans G' .

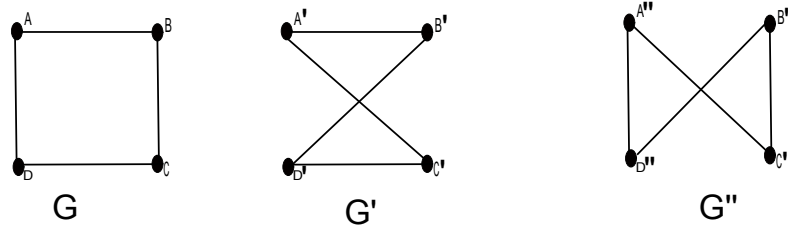
La Figure suivant donne un morphisme de graphe.



1.5.4 Isomorphisme de graphe

Soient $G = (X, E)$ et $G' = (X', E')$ deux graphes. Un isomorphisme de graphes de G dans G' est une bijection $f : X \rightarrow X'$ telle que, pour tous sommets x_1 et x_2 de G , x_1 et x_2 sont adjacents dans G si, et seulement si, $f(x_1)$ et $f(x_2)$ sont adjacents dans G' .

On considère les trois graphes G , G' et G'' de la Figure suivante :



Ces trois graphes sont isomorphes. Des isomorphismes sont donnés par :

$$f : \begin{cases} A \longrightarrow A' \\ B \longrightarrow B' \\ C \longrightarrow D' \\ D \longrightarrow C' \end{cases}$$

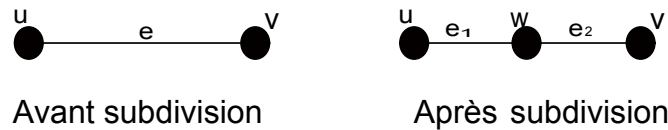
$$f' : \begin{cases} A' \longrightarrow A'' \\ B' \longrightarrow C'' \\ C' \longrightarrow D'' \\ D' \longrightarrow B'' \end{cases}$$

1.5.5 Subdivision dans les graphes

Subdivision d'une arête

La subdivision d'une arête uv conduit à un graphe contenant un nouveau sommet w et où l'on a remplacé l'arête uv par deux nouvelles arêtes, uw et wv .

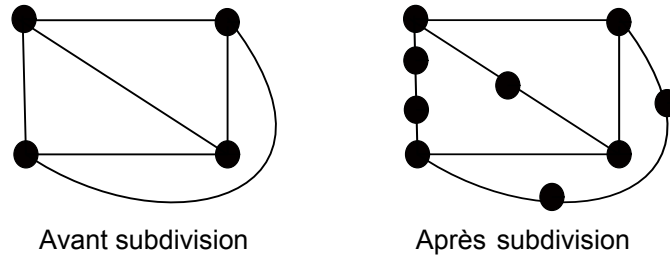
La Figure suivant montre une arête avant et après la subdivision.



Subdivision d'un graphe

G' est une subdivision de G si G' est obtenu à partir de G en rajoutant des sommets sur des arêtes.

La Figure suivant montre un graphe avant et après la subdivision.



1.6 Représentations matricielles

1.6.1 Matrice d'adjacence

Les outils classiques d'algèbre linéaire peuvent également être utilisés pour coder les graphes. La première idée consiste à considérer chaque arc comme un lien entre deux sommets.

Considérons un graphe $G = (X, E)$ comportant n sommets. La matrice d'adjacence de G est égale à la matrice $U = (u_{ij})$ de dimension $n \times n$ telle que :

$$U_{ij} = \begin{cases} 1, & \text{si } (i, j) \in E, \\ 0, & \text{sinon.} \end{cases} \quad (\text{c'est-à-dire } (i, j) \text{ est une arête});$$

Une telle matrice, ne contenant que des $\prec 0 \succ$ et des $\prec 1 \succ$ est appelée; de manière générale, une Matrice Booléenne, voir les deux exemples illustrés dans les Figures 1.17, 1.18 ci-dessous :

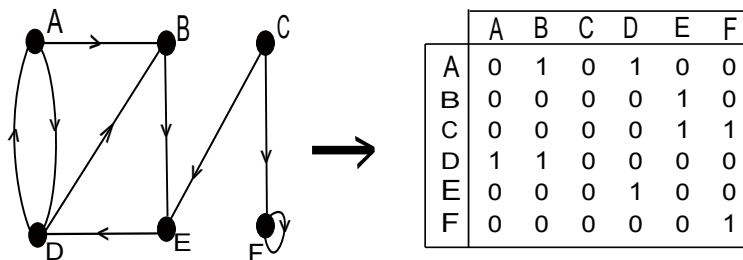


FIGURE 1.17 – Le cas où le graphe est orienté

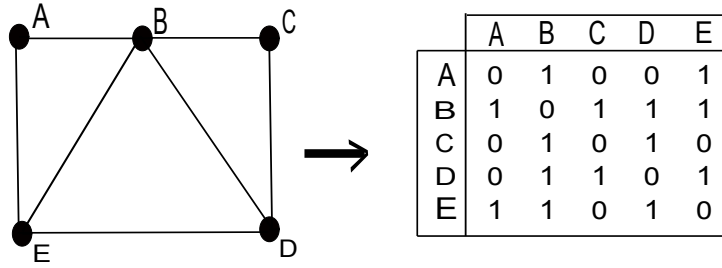


FIGURE 1.18 – Le cas où le graphe est non orienté

Proposition : Soit G un graphe d'ordre n , orienté ou non. Soit $k \in \mathbb{N}$, M la matrice d'adjacence de G . On pose :

$$M^k = \begin{pmatrix} m_{1,1}^k & \cdot & \cdot & \cdot & m_{1,n}^k \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ m_{n,1}^k & \cdot & \cdot & \cdot & m_{n,n}^k \end{pmatrix}$$

Alors $m_{i,j}^k$ est le nombre de parcours (d'arcs dans le cas orienté) de longueur k , d'extrémités initiale x_i et terminale x_j .

1.6.2 Matrice d'incidence

La seconde idée permettant une représentation matricielle d'un graphe qu'exploite la relation d'incidence entre arêtes et sommets. Considérons un graphe orienté sans boucle $G = (X, E)$ comportant n sommets $x_1, \dots, x_n$ et m arêtes $e_1, \dots, e_m$. On appelle matrice d'incidence (aux arcs) de G la matrice $M = (m_{ij})$ de dimension $n \times m$ telle que :

$$m_{ij} = \begin{cases} 1, & \text{Si } x_i \text{ est l'extrémité initiale de } e_j; \\ -1, & \text{Si } x_i \text{ est l'extrémité terminale de } e_j; \\ 0, & \text{Si } x_i \text{ n'est pas une extrémité de } e_j. \end{cases}$$

Pour un graphe non orienté sans boucle, la matrice d'incidence (aux arêtes) est définie par :

$$m_{ij} = \begin{cases} 1, & \text{Si } x_i \text{ est une extrémité de } e_j; \\ 0, & \text{Si non.} \end{cases}$$

1.7 Complexité des algorithmes

On appelle complexité en temps d'un algorithme le nombre d'instructions élémentaires mises en œuvre dans cet algorithme afin de résoudre un problème donné. Une instruction élémentaire sera une affectation, une comparaison, une opération algébrique, la lecture et l'écriture, etc.

Mais comme le décompte précis de toutes les instructions d'un programme risque d'être assez pénible, et qu'entre deux exécutions du même algorithme avec un jeu de paramètres différents, le nombre d'instructions exécutées peut changer, on se contentera, en général, d'apprécier un ordre de grandeur de ce nombre d'instructions. C'est ce qu'on désigne sous le nom de complexité de l'algorithme.

Donc pour mesurer la complexité temporelle d'un algorithme, on s'intéresse plutôt aux opérations les plus coûteuses :

- Racine carrée, Log, Exp, Addition réelle, etc.
- Comparaisons dans le cas des tris, etc.

On dit que : $f(n) = O(g(n))$, avec ($f(n)$ est de complexité $g(n)$), chaque fois qu'il existe k et n_0 tels que : $n > n_0 \Rightarrow f(n) < k.g(n)$.

1.8 Fermeture transitive d'un graphe

Étant donné un graphe $G = (X, E)$, construire le graphe $G^+ = (X, E^+)$ tel que $(x, y) \in E^+$, il existe un chemin de G allant de x à y . G^+ s'appelle la fermeture transitive de G .

1.8.1 La transitivité

Un graphe est transitif si la relation binaire qui lui est associée est transitive. En d'autres termes : $\forall x, y, z \in X : (x, y) \in E \wedge (y, z) \in E \Rightarrow (x, z) \in E$.

Proposition : Un graphe G est transitif si et seulement si tout chemin de longueur deux est sous-tendu par un arc, c'est-à-dire : $E^2 \subset E$.

Corollaire : Un graphe G est transitif si et seulement si tout chemin est sous-tendu par un arc, c'est-à-dire : $\forall k \geq 2 : E^k \subset E$

L'interprétation "concrète" de ce corollaire est la suivante : dans un graphe transitif, la notion de chemin se confond avec la notion d'arc.

Nous allons maintenant aborder les caractérisations et les algorithmes de calcul de la fermeture transitive. Le terme de fermeture transitive provient du fait que, d'une part, G^+ est transitif (par construction!), et d'autre part, G^+ est le plus petit graphe transitif supérieur ou égal à G (au sens de la relation $\leq$). Plus précisément :

- G^+ est transitif;
- $G \leq G^+$;
- Si H est transitif et $G \leq H$, alors $G^+ \leq H$.

Intuitivement, on obtient G^+ à partir de G en rajoutant "juste ce qu'il faut" d'arcs pour rendre le graphe transitif.

1.8.2 Caractérisation de la fermeture transitive d'un graphe

Proposition : $G^+ = G \cup G^{[2]} \cup \dots \cup G^{[k]} \cup \dots$

La proposition suivante montre que le problème se réduit à l'existence des chemins élémentaires.

Proposition : Les deux propositions suivantes sont équivalentes :

- i) Dans le graphe G , il existe un chemin de x à y ;
- ii) Dans le graphe G , il existe un chemin élémentaire de x à y .

Corollaire : Soit $G = (X, E)$ et $n = |X|$. On a :

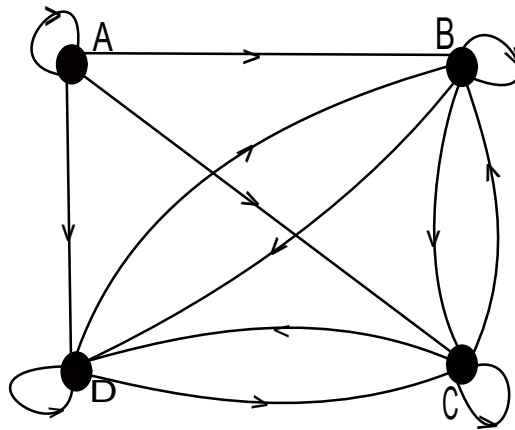
$$G^+ = G \cup G^{[2]} \cup \dots \cup G^{[n]}$$

Exemple 1 :

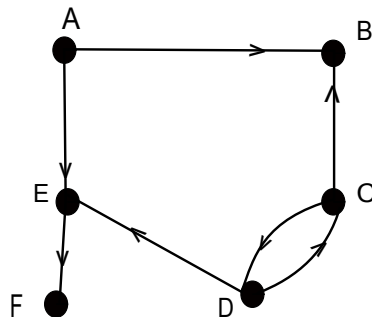
On considère le graphe $G = \{(A; A), (A; B), (A; C), (A; D), (B; D), (D; C), (C; B)\}$ défini sur l'ensemble $E = \{A, B, C, D\}$.

On obtient G^+ en complétant G par les 6 arcs $(B; B), (B; C), (C; C), (C; D), (D; B), (D; D)$.

$G^+ = \{(A; A), (A; B), (A; C), (A; D), (B; D), (D; C), (C; B), (B; B), (B; C), (C; C), (C; D), (D; B), (D; D)\}$
Le graphique ci-dessous donne la représentation sagittale de G et celle de G^+ .

**Exemple 2 : 'Calcul de fermeture transitive avec les matrices'**

Voici le graphe pour lequel on se propose de calculer la fermeture transitive en calculant les puissances successives des matrices.



La matrice d'adjacence associée à ce graphe est la suivante :

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

On calcule successivement M^2 , M^3 , M^4 et M^5 (il y a 6 sommets).

$$M^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Les chemins de longueur 2 que l'on ajoute comme arêtes sont $(1; 6)$, $(3; 3)$, $(3; 4)$, $(5; 2)$, $(5; 5)$, $(5; 6)$.

$$M^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Le chemin de longueur 3 que l'on ajoute comme arête est $(3; 6)$, les arêtes $(3; 2)$, $(3; 5)$, $(5; 3)$, $(5; 4)$ sont déjà dans le graphe.

$$M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

On n'ajoute aucun chemin de longueur 4 comme arête, les arêtes (3; 3), (3; 4), (5; 2), (5; 5), (5; 6) sont déjà dans le graphe.

$$M^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

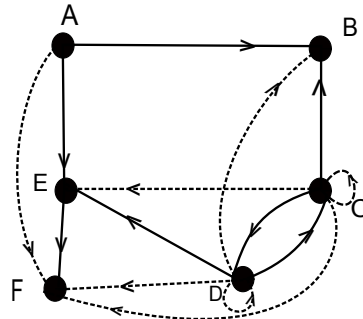
On n'ajoute aucun chemin de longueur 5 comme arête, les arêtes (3; 2), (3; 5), (3; 6), (5; 3), (5; 4) sont déjà dans le graphe.

La disjonction de ces matrices, qui représente la matrice de la fermeture transitive, est :

$$M \oplus M^2 \oplus M^3 \oplus M^4 \oplus M^5 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

La somme booléenne $\oplus$ consiste à remplacer dans la somme classique tous les réels supérieurs ou égaux à 1 par 1.

Le graphe obtenu par fermeture est alors le suivant :



1.9 Quelques graphes particuliers

1.9.1 Graphe complet

Un graphe $G = (X, E)$ est dit complet si pour toute paire de sommets (x, y) , il existe au moins un arc de la forme (x, y) ou (y, x) . Un graphe simple complet d'ordre n est noté K_n . Un sous-ensemble de sommets $C \subset X$ tel que deux sommets quelconques de C sont reliés par une arête est appelé une clique. La Figure 1.19 donne un graphe complet.

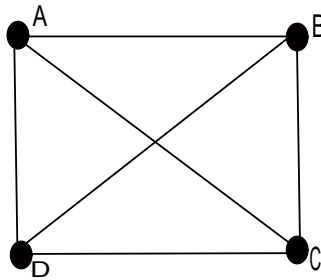


FIGURE 1.19 – Graphe Complet

1.9.2 Graphe biparti

Un graphe $G = (X, E)$ est dit biparti s'il existe une partition de l'ensemble des sommets de G en deux sous-ensemble X_1 et X_2 tel que toute arête de G a une extrémité dans X_1 et l'autre extrémité dans X_2 . Si de plus $|X_1| = |X_2|$, G est dit biparti équilibré (Balance). Un graphe biparti $G = (X_1 \cup X_2, E)$ est complet si tous sommets de X_1 est adjacent à tout sommet de X_2 . Un tel graphe est noté $K_{p,q}$ avec $p = |X_1|$, $q = |X_2|$. La Figure 1.20 donne un graphe biparti complet

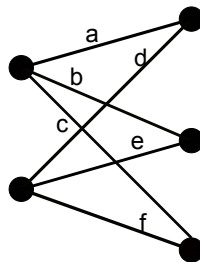


FIGURE 1.20 – Graphe biparti Complet

1.9.3 Graphe acyclique

Dans le cas des graphes non orientés, le terme acyclique décrit un graphe sans cycle, alors que si l'on traite des graphes orientés, on appelle acyclique tout graphe sans circuit. La Figure 1.21 donne un graphe acyclique.

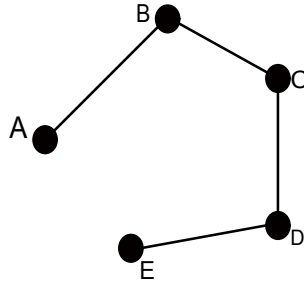


FIGURE 1.21 – Graphe acyclique

1.9.4 Hypercube

L'hypercube de dimension n , noté Q_n est le graphe dont l'ensemble des sommets est formé des n -uplets binaires, et deux sommets sont adjacents si et seulement s'ils diffèrent par exactement une seule composante (coordonnée).

Notons que $Q_0 = K_1$, $Q_1 = K_2$ et d'une manière générale, Q_n peut être défini récursivement en utilisant la somme Cartésienne de Q_{n-1} avec K_2 . [3]

Il est clair que pour tout $n \geq 2$, Q_n est isomorphe à $\underbrace{K_2 \square K_2 \square \dots \square K_2}_{n \text{ fois}}$. La Figure 1.22 montre les premiers hypercubes.

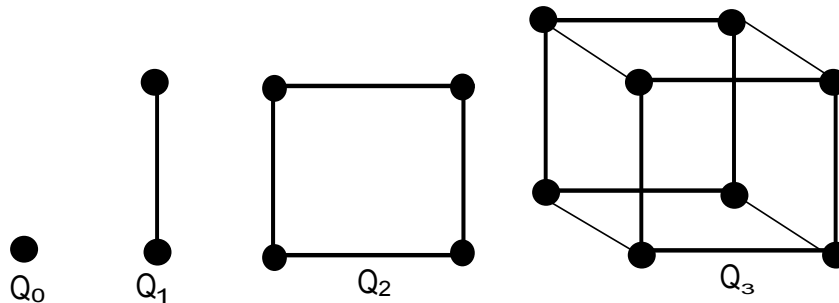


FIGURE 1.22 – Q_0 , Q_1 , Q_2 et Q_3

1.9.5 Arbre

Un arbre est un graphe connexe sans cycles.

Un graphe sans cycle qui n'est pas connexe est appelé une forêt (chaque composante connexe est un arbre).

Par définition même, un arbre est donc un graphe simple. On constate également que $G = (X, E)$ est un arbre si et seulement s'il existe une chaîne et une seule entre deux sommets quelconques. La Figure 1.23 donne un arbre.

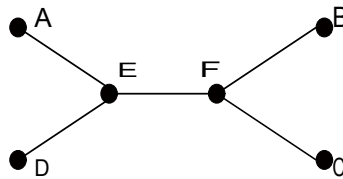


FIGURE 1.23 – Arbre

1.9.6 Arborescence

Un graphe G est une arborescence s'il existe un sommet R appelé racine de G tel que, pour tout sommet S de G , il existe un chemin et un seul de R vers S .

La notion d'arborescence couvrante se définit comme celle d'arbre couvrant, mais elle est plus délicate car il faut trouver une racine (qui n'existe pas toujours).

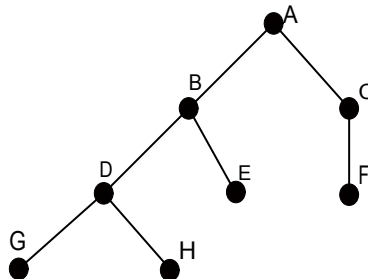


FIGURE 1.24 – Arborescence

Dans le cas où chacune des arêtes d'un graphe G est affecté par une étiquette (lettre, mot, nombre, symbole, code, ...), on dira que G est un graphe étiqueté. Comme cas particuliers de ces graphes nous avons les graphes pondérés et les graphes valeurs définis comme suit :

- On considère un graphe étiqueté orienté, où un sommet est marqué « Début » et un autre « Fin ». On dit qu'un mot est reconnu par ce graphe si les lettres qui le composent correspondent, dans l'ordre, aux étiquettes des arêtes composant une chaîne orientée partant du sommet « Début » et arrivant au sommet « Fin ». On considère le graphe étiqueté de la Figure 1.25 .

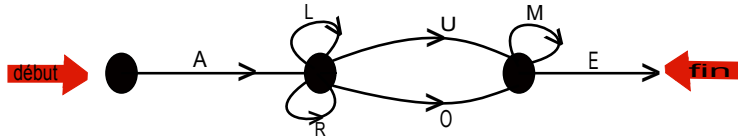


FIGURE 1.25 – Mots reconnus par un graphe étiqueté

- Un graphe pondéré est un couple (G, f) où $G = (X, E)$ est un graphe et $f : E \rightarrow R$ est une application. Pour toute arête $e \in E$, $f(e)$ est appelé poids de e . On dira que G est valué si de plus f est à valeurs strictement positives. On considère le graphe valué de la Figure 1.26 .

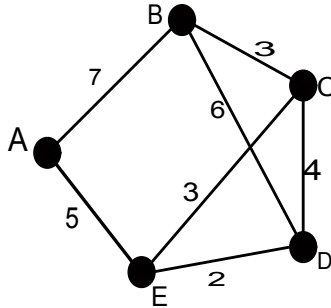


FIGURE 1.26 – Graphe est valué

2

Graphes Probabilistes et Chaînes de Markov

L'objet de ce chapitre est de donner aux étudiants une première approche de la modélisation mathématique en utilisant dans un premier temps des notions de base des graphes probabilités, puis en introduisant des concepts de chaîne de Markov et le lien avec la théorie des graphes, finalement nous donnons quelque modèles des graphes probabiliste.

2.1 Les graphes probabilistes

Un graphe probabiliste est un graphe orienté pondéré dans lequel la somme des poids des arêtes issues de chaque sommet est égale à 1. Les graphes probabilistes sont utilisés pour modéliser l'évolution d'un système pouvant changer aléatoirement d'état, tels que :

- ✓ les sommets du graphe sont les états possibles du système ;
- ✓ Le poids d'une arête orientée issue du sommet i et d'extrémité j est la probabilité conditionnelle de la réalisation de l'événement j à l'étape $n + 1$ sachant que l'événement i est réalisé à l'étape n .

La Figure 2.1 montre un graphe probabiliste.

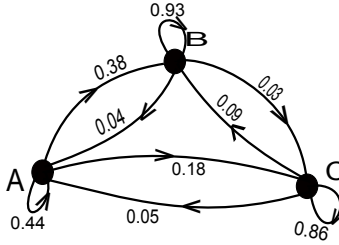


FIGURE 2.1 – Graphe probabiliste

2.2 Matrice de transition

La matrice de transition associée à un graphe probabiliste d'ordre p est la matrice carrée

$M = (m_{i,j})$ d'ordre p telle que, pour tout entier i et j vérifiant $1 \leq i \leq p$ et $1 \leq j \leq p$, $m_{i,j}$ est égal au poids de l'arête orientée d'origine le sommet i et d'extrémité le sommet j si cette arête existe, et est égal à 0 sinon.

Tous les coefficients sont positifs ou nuls, et pour chaque ligne la somme des coefficients est égale à 1. Cette matrice décrit le passage d'un état au suivant. Le coefficient $m_{i,j}$ est la probabilité conditionnelle d'être dans l'état j à l'instant $n + 1$ sachant que l'on est dans l'état i à l'instant n . On considère un exemple, de la matrice de transition de graphe précédant :

$$M_{ij} = \begin{pmatrix} 0,44 & 0,38 & 0,18 \\ 0,04 & 0,93 & 0,03 \\ 0,05 & 0,09 & 0,86 \end{pmatrix}$$

2.2.1 État probabiliste

Un état probabiliste est une loi de probabilité sur l'ensemble des états possibles. Cette loi est représentée par une matrice ligne telle que la somme des termes est égale à 1.

Propriétés :

- ✓ Soit M la matrice de transition d'un graphe probabiliste, P_0 l'état probabiliste initial, P_n l'état probabiliste à l'étape n et P_{n+1} l'état probabiliste à l'étape $n + 1$.
On a ; pour tout entier n : $P_{n+1} = P_n \times M$.
- ✓ Soit M la matrice de transition d'un graphe probabiliste ayant 2 sommets, P_0 la matrice ligne décrivant l'état probabiliste initial et P_n la matrice ligne décrivant

l'état probabiliste à l'étape n . On a, pour tout entier n : $P_n = P_0 \times M^n$.

On admet que ces propriétés restent vraies pour les graphes probabilistes ayant trois sommets ou plus.

Exemple :

Soit la matrice de transition :

$$M = \begin{pmatrix} 0,6 & 0,4 \\ 0,2 & 0,8 \end{pmatrix}$$

✓ L'état probabiliste initial est donc $P_0 = (0,1 \ 0,9)$.

✓ On a donc, par exemple, $P_1 = P_0 \times M = (0,24 \ 0,76)$.

✓ On a aussi, par exemple, $P_2 = P_0 \times M^2 = (0,296 \ 0,704)$.

2.2.2 État stable

On dit qu'un état probabiliste P est stable s'il reste le même dans la répétition de l'expérience décrite par le graphe probabiliste de matrice de transition M , c'est-à-dire :

$$P \times M = P$$

Pour tout graphe probabiliste d'ordre 2 dont la matrice de transition M ne comporte pas de zéro, l'état probabiliste P_n converge toujours vers l'état stable P vérifiant $P \times M = P$, indépendamment de l'état probabiliste initial P_0 l'état P est appelé état stable du système.

Exemple : Soit la matrice de transition :

$$M = \begin{pmatrix} 0,7 & 0,3 \\ 0,2 & 0,8 \end{pmatrix}$$

✓ L'état initial $P_0 = (0,4 \ 0,6)$.

✓ On a donc, par exemple, $P_1 = P_0 \times M = (0,4 \ 0,6)$.

✓ On a aussi, par exemple, $P_2 = P_0 \times M^2 = (0,4 \ 0,6)$.

De proche en proche que, $P_n = P_0$ pour tout entier naturel n . L'état décrit par la matrice P_0 est donc un état stable.

2.3 Chaînes de Markov

2.3.1 Quelques rappels

Définition. Un Processus Stochastique est un quadruplet $(\Omega, F, \mathbb{P}, (X_i)_{i \in I})$ où Ω est un espace de probabilité, F sa σ -algèbre, $\mathbb{P}$ sa mesure de probabilité et $(X_i)_{i \in I}$ une collection de variables aléatoires de (Ω, F) vers $(E, \mathcal{C})$ (où E est un ensemble quelconque et $\mathcal{C}$ sa σ -algèbre) et I est un ensemble quelconque.[1]

Une propriété intéressante des processus stochastiques est celle de Markov, énoncée ci-dessous.

Définition. Un processus stochastique $(X_t)_{t \in I}$ est une chaîne de Markov ou (Markovien) si : $\forall t_1 < t_2 < \dots < t_n, t_i \in I$ et $\forall i_1 \dots i_n \in E$

$$P(X_{t_n} = i_n | X_{t_{n-1}} = i_{n-1}, \dots, X_{t_1} = i_1) = P(X_{t_n} = i_n | X_{t_{n-1}} = i_{n-1})$$

Définition. Soit $(X_t)_{t \in I}$ une chaîne de Markov. Les expressions

$$p_{i,j}(s, t) = P(X_t = j | X_s = i), \quad 0 \leq s < t, \quad i, j \in E$$

sont les probabilités de transition de la chaîne. Sous forme matricielle nous avons

$$\mathbb{P}(s, t) = \begin{pmatrix} & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ \cdot & \cdot & \cdot & p_{i,j}(s, t) & \cdot & \cdot & \cdot \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \end{pmatrix}$$

qu'on appelle matrice de transition.

- La matrice de transition d'une chaîne de Markov est stochastique (i.e $\sum_j p_{i,j}(s, t) = 1$).

$$\begin{aligned}
\sum_j p_{i,j}(s, t) &= \sum P(X_t = j | X_s = i) \\
&= \sum_j \frac{P(X_t = j, X_s = i)}{P(X_s = i)} \\
&= \frac{1}{P(X_s = i)} \sum_j P(X_t = j, X_s = i) = 1
\end{aligned}$$

Nous allons considérer dans ce séminaire des chaînes de Markov dites homogènes (ou stationnaires), c'est-à-dire que

$$P(X_t = j | X_s = i) = P(X_{t-s} = j | X_0 = i).$$

Pour une chaîne de Markov donnée, l'équation de Chapman-Kolmogorov

$$p_{i,j}(s, t) = \sum_{k \in E} p_{i,k}(s, u) p_{k,j}(u, t), \quad s < u < t.$$

nous permet de montrer que (dans le cas homogène)

$$\mathbb{P}(s + t) = \mathbb{P}(s)\mathbb{P}(t).$$

En connaissant notre loi initiale ainsi que notre matrice de transition, nous pouvons déterminer la probabilité de se trouver dans un état après un temps t par

$$\pi(t) = \mathbb{P}^t \pi(0), \quad \mathbb{P} := \mathbb{P}(1)$$

Dans le cas de la modélisation, il est intéressant de regarder le cas où notre chaîne ne changera plus son comportement à l'infini. Ainsi, nous introduisons la notion de "stationnarité".

Définition. Soit $(X_t)_{t \in I}$ une chaîne de Markov de matrice de transition $\mathbb{P}$ et π une mesure de probabilité sur E . On dit que π est stationnaire si :

$$\pi \mathbb{P} = \pi$$

2.3.2 Classification des états

Les états d'une chaîne de Markov se classifient en fonction de la probabilité qu'a la chaîne d'atteindre les uns à partir des autres.

- **Accessible** : j est accessible depuis i si il existe un chemin dans le diagramme de transition, partant de i et arrivant en j . Ceci se traduit également en termes des probabilités de transition en m pas et des probabilités de premier passage.

- **Communiquent** : On dit que deux états i et j communiquent si chacun est accessible depuis l'autre.

⇒ La relation de communication est symétrique et transitive, mais elle n'est pas nécessairement réflexive (quand la chaîne quitte un état i elle peut ne jamais revenir).

- **Irréductible** : On appelle classe irréductible tout sous ensemble d'états, maximal au sens de l'inclusion, composé d'états qui communiquent deux à deux.

⇒ Si tous les états de E communiquent deux à deux, E tout entier est la seule classe irréductible. On dit alors que la chaîne est irréductible.

- **Périodique** : L'état i est dit périodique de période $k > 1$ si tous les entiers m tels que $p_{ii}^{(m)} > 0$ sont multiples de k . Un état qui n'admet pas de période est dit apériodique.

⇒ Si i est périodique de période k et communique avec j , on démontre que j est également de période k . Les classes irréductibles périodiques constituent un cas particulier que l'on ne rencontre pas dans les applications.

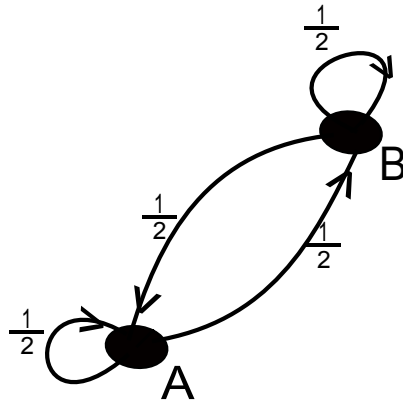
2.3.3 Lien entre chaînes de Markov et théorie des graphes

Dans le cas d'une chaîne de Markov où l'ensemble E (appelé ensemble des états) est fini, on peut représenter la chaîne sous forme graphique. En effet, les sommets représentent les états de la chaîne et les arrêtes représentent les probabilités de transition. Ainsi, en connaissant la matrice de transition, on peut déterminer le graphe associé à la chaîne.

Exemple : Si nous avons $E = \{A, B\}$, que la probabilité de passer de A à B (ou de B à A) est $\frac{1}{2}$ et la probabilité de rester en A ou B est aussi $\frac{1}{2}$, la matrice de transition sera :

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

et le graphe associé



Définition : On dit qu'une chaîne est irréductible si son graphe associé ne possède qu'une composante connexe.

Nous tentons ici d'amener le lecteur à la découverte d'un lien entre le graphe d'une chaîne de Markov et sa loi stationnaire.

Définition : On dit que T_i est le temps de premier passage dans l'état i si

$$T_i = \inf\{n \geq 1, X_n = i\}.$$

Le temps moyen de premier retour en i (si on est parti de l'état i) est défini comme

$$\mathbb{E}_i(T_i).$$

On dira qu'une chaîne est positivement récurrente si pour tous les états i on a la propriété suivante

$$\mathbb{E}_i(T_i) < \infty.$$

Avec cela, nous pouvons désormais énoncer un théorème primordial dans la suite de ce

séminaire .

Théorème : Une chaîne de Markov irréductible est positivement récurrente si et seulement si elle possède une unique loi stationnaire

Cela signifie que pour une chaîne de Markov possédant une composante connexe dans son graphe, il y a une loi stationnaire unique.

2.4 Modèles des graphes probabilistes

2.4.1 Graphes statiques stochastiques

Soit $G = (X, E)$ un graphe orienté et pondéré. G est appelé graphe statique stochastique si le poids d'un arc $(i, j) \in E$ est défini par une distribution de probabilité discrète. Le poids de l'arc (i, j) est défini de la manière suivante :

$$P_{(ij)} = \begin{cases} C_1 & P_1 \\ C_2 & P_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ C_m & P_m \end{cases}$$

Avec $c_1, c_2, \dots, c_m \in R^+$ les poids possible pour l'arc (i, j) et $p_1, p_2, \dots, p_m$ les probabilités associées aux poids. Ces probabilités doivent vérifier la condition suivante :

$$\sum_{k=1}^m P_k = 1$$

On considère l'exemple de la Figure 2.2 montre un graphes statiques stochastiques :

2.4.2 Graphes dynamiques stochastique

Soit $G = (X, E)$ un graphe orienté et pondéré. G est appelé graphe dynamique stochastique si le poids d'un arc $(i, j) \in E$ est défini par une distribution de probabilité discrète et que cette dernier dépende du temps, d'où le poids de l'arc (i, j) est défini par

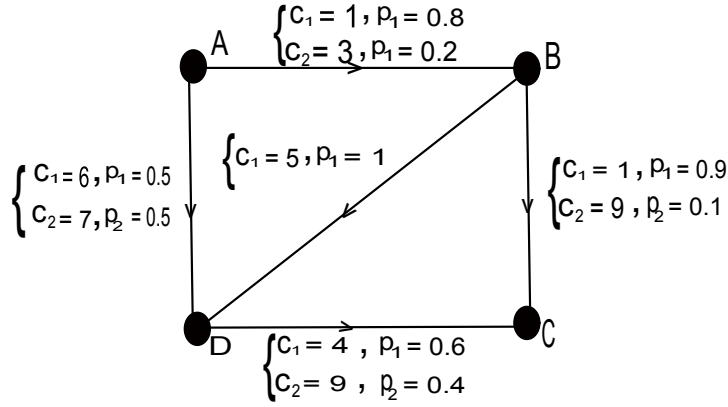


FIGURE 2.2 – Graphes statiques stochastiques

la fonction $p_{(i,j)}^t$ avec t la date de départ de i . [10]

La fonction $p_{(i,j)}^t$ est déterminée à chaque instant t , par une distribution de probabilité de la forme :

$$P_{(ij)}^t = \begin{cases} C_1^t & P_1^t \\ C_2^t & P_2^t \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ C_m^t & P_m^t \end{cases}$$

Avec $C_1^t, C_2^t, \dots, C_m^t \in \mathbb{R}$, les poids possibles pour l'arc (i, j) à l'instant t et $P_1^t, P_2^t, \dots, P_m^t$ les probabilistes associées aux poids à l'instant t . La condition suivante sur les probabilités P_k^t doit être vérifiée :

$$\sum_{k=1}^m P_k^t = 1$$

On considère l'exemple de la Figure 2.3 montre un graphes dynamiques stochastique :

2.4.3 Réseaux bayésiens

Soit $G(X, E)$, un graphe acyclique orienté, où X est l'ensemble de nœuds, et E celui des arcs. Ce graphe peut être statique, ou dynamique. Dans ce deuxième cas on parlera de réseau bayésien dynamique. Ces graphes ont la particularité d'évoluer au cours du temps, des arcs peuvent être ajoutés entre chaque pas de temps. Le coût de l'inférence probabiliste sera donc plus important dans ce genre de réseau.

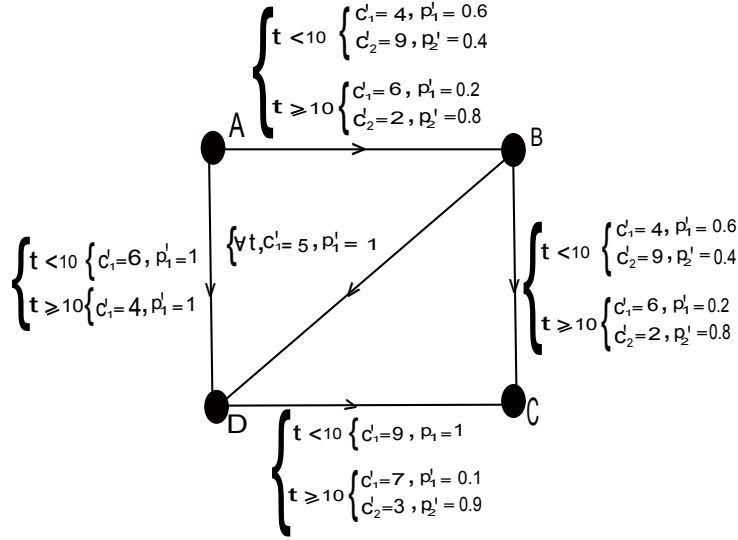


FIGURE 2.3 – Graphes dynamiques stochastique

Soit $X = \{x_i; x_i \in X\}$ un ensemble de variables aléatoires. Chaque variable x_i correspond à un nœud X du graphe. Pour chaque nœud $x \in X$, on définit π_i , l'ensemble de ses parents dans le graphe, désigne l'ensemble des valeurs observées pour les parents de x .

Soit : P , l'ensemble de probabilités conditionnelles tels que :

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(x_i))$$

Exemple : L'exemple suivant représente un réseau bayésien contenant cinq variables. Il décrit par les saisons de l'année (X_1) si la pluie tombe (X_2), si l'arrosage est en marche (X_3), si le chemin est mouillé (X_4) et si le chemin est glissant (X_5). Toutes ces variables sont binaires. L'absence d'arc allant de X_1 à X_4 signifie que la saison n'influe pas directement sur l'état mouillé ou non du chemin. Ainsi le graphe indique que la probabilité jointe des cinq variables se décompose de la façon suivante :

$$P(x_1, \dots, x_5) = P(x_1) \times P(x_2 | x_1) \times P(x_3 | x_1) \times P(x_4 | x_2, x_3) \times P(x_5 | x_4)$$

Et la Figure 2.4 représente un réseau bayésien.

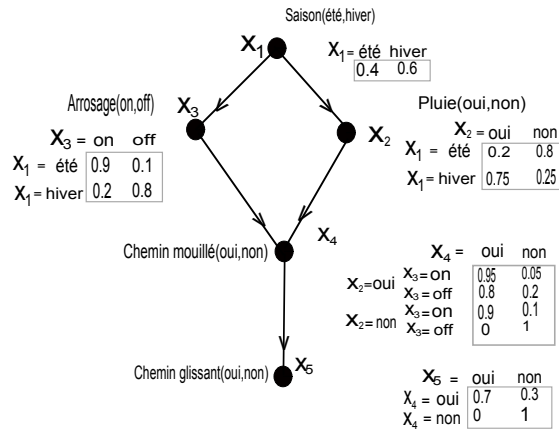


FIGURE 2.4 – Réseaux Bayésiens

2.4.4 Processus de décision markoviens

Un processus de décision markovien (PDM) est un graphe orienté valué auquel on ajoute des labels (ou actions) sur les arcs. En outre, chaque arc porte une probabilité telle que, pour un sommet du graphe et une action donnés, la somme des probabilités quittant ce sommet via cette action soit égale à 1.

Exemple : Soit en RER, soit à vélo. S'il prend le RER, celui-ci peut être fonctionnel, et atteindre Cachan en 40 mn comme à l'ordinaire avec une probabilité de 0,6, ou bien être en panne avec une probabilité 0,4, et n'être arrivé qu'à la moitié du trajet (à Gentilly) après 40 mn de trajet, pour ensuite atteindre Cachan en à nouveau 40 mn avec une probabilité 1. S'il opte pour le vélo, il arrivera de façon certaine (avec une probabilité 1) à Cachan en 50 mn.

Et la Figure 2.5 représente un réseaux markoviens.

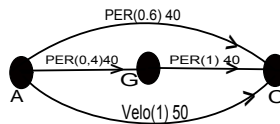


FIGURE 2.5 – Réseaux markoviens

2.4.5 Jeux probabilistes et déterministes

L'apparition du hasard est fréquente dans les jeux. Le déterminisme est l'appellation de ce qui ne dépend pas du hasard. En théorie des probabilités, le déterminisme est un

cas dégénère de hasard il coïncide avec des distributions de type spin.

Un jeu est dit probabiliste dès que l'une des données fonctions d'utilités ou stratégies est aléatoire. Cette définition se formalise au moyen d'espace probabilisé. Mettons plutôt en avant les points suivants :

- ✓ Les joueurs n'ont pas toujours accès aux mêmes stratégies car elles sont accessibles uniquement en fonction des résultats du hasard ;
- ✓ Les gains dépendant des stratégies accessibles, on n'a pas forcément besoin de probabilisé, dès que les stratégies sont aléatoires ;
- ✓ Réciproquement, on n'a pas besoin de probabilisé les stratégies si les gains le sont déjà (pensez la roulette) ;
- ✓ Sous forme extensive, les ensembles d'actions peuvent être probabilisés. En fait, dès qu'un paramètre du jeu peut être probabilisé, on aura affaire à un jeu probabiliste. Il faut cependant éviter des pièges.

3

Quelques Algorithmes d'optimisation dans les graphes probabilistes

Les graphes permettent de manipuler plus facilement des objets et leurs relations avec une représentation graphique naturelle.

Dans ce chapitre nous allons rappeler brièvement quelques problèmes d'optimisation pour les quelles nous donnons quelque algorithmes de résolutions pour chaque problème.

Parmi les problèmes nous citons : Problème de l'arbre couvrant de poids minimum, Problème du plus long chemin, Problème d'affectation, Problèmes de flots.[4]

3.1 Quelques problèmes dans les graphes probabilistes

3.1.1 Problème de l'arbre couvrant de poids minimum

Le problème de l'arbre couvrant, consiste à étudier le problème de l'arbre couvrant de poids minimum, qui joue un rôle important dans l'optimisation combinatoire et il

figure parmi les problèmes les plus étudiés dans la littérature. Pour avoir un aperçu sur ses propriétés et les algorithmes qui permettent de le résoudre.

Cet algorithme peut être utilisé dans le cas où on considère un graphe $G = (X, E)$, tel que chaque arc $(i, j) \in E$ a une probabilité de présence p_{ij} (les probabilités sont indépendantes). Le problème consiste à trouver un sous graphe qui est un arbre couvrant, dont la somme des poids des arcs est minimale.

Considérons le problème qui consiste à relier n villes par un réseau câble de la manière la plus économique possible. On suppose connue la longueur $l_{ij} = l(a_{ij})$ la longueur de câble nécessaire pour relier les villes i et j . Le réseau doit évidemment être connexe et il ne doit pas admettre de cycles pour être de coût minimal ; c'est donc un arbre et ce doit être l'arbre maximum le plus économique.

L'objectif du problème est alors de trouver une structure de réseau a priori d'un coût espéré minimum.

1. Algorithme de Kruskal :

L'algorithme de *KRUSKAL* permet de déterminer l'arbre couvrant de poids minimal dans un graphe pondéré positivement.

L'ensemble des arêtes gardées forme par construction un arbre couvrant. De plus on a gardé les plus petites arêtes nécessaires à la couverture du graphe : c'est l'arbre couvrant minimal.

L'algorithme de *kruskal* est l'utilisé dans la théorie des graphes pour construire une arborescence de poids minimal.

1.1 Principe de l'algorithme de Kruskal :

L'idée de l'algorithme de *KRUSKAL* est tout d'abord de numéroter les arcs par ordre des poids croissants. En suite de construire progressivement l'arbre A en rajoutant dans leur ordre, les arcs un par un. Un arc est ajouté seulement si son adjonction à A ne détermine pas de cycle, c'est à dire si A ne prend pas sa notion d'arbre sinon on passe à

l'arc suivant dans l'ordre de la numérotation.

1.2 Énoncé de l'algorithme de *Kruskal* :

✓ **Données** : Soit $G = (X, E, P)$ un graphe probabiliste,

(0). Initialisation : Numéroté les arcs de G dans l'ordre des poids croissants :

$P(u_1) \leq P(u_2) \leq P(u_m)$, soit $W = \emptyset$, $i = 1$;

(1). Si $(X, W \cup u_i)$ contient un cycle aller en (3) sinon aller en (2) ;

(2). On pose $W = W \cup u_i$, aller en (3) ;

(3). Si $i = m$ termine.

$A = (X, W)$ est l'arbre de poids minimum $P(W) = \sum P(u_i)$, pour $u_i \in W$.

Sinon, $i = i + 1$ aller en (1).

✓ **Le critère d'arrêt** : L'algorithme s'arrête lorsque le nombre d'arcs retenus est égal à $n - 1$.

✓ **Résultat** : Trouver un arbre couvrant de poids minimum de G .

2. Algorithme de PRIM

L'algorithme de *Prim*, est un algorithme qui permet de trouver un arbre couvrant minimal dans un graphe connexe valué et non orienté. En d'autres termes, cet algorithme trouve un sous-ensemble d'arêtes formant un arbre sur l'ensemble des sommets du graphe initial, et tel que la somme des poids de ces arêtes soit minimale. Si le graphe n'est pas connexe, alors l'algorithme ne déterminera l'arbre couvrant minimal que d'une composante connexe du graphe. Il a été conçu en 1957 par Robert C. Prim.

2.1 Principe de l'algorithme de *PRIM* :

L'algorithme consiste à choisir arbitrairement un sommet et à faire croître un arbre à partir de ce sommet. Chaque augmentation se fait de la manière la plus économique possible.

2.2 Énoncé de l'Algorithme de *PRIM* :

✓ **Données** : Soit $G = (X, E, P)$ un graphe probabiliste,

Soit I l'ensemble de nœuds déjà inclus dans l'arbre et NI l'ensemble de nœuds non encore inclus,

(0). Initialisation : $I = \emptyset$, NI = l'ensemble de tous les nœuds ;

(1). Mettre le nœud de départ dans I ;

(2). Si l'arbre est connexe alors Aller au (5) ;

(3). Trouver un nœud de NI dont la distance à un nœud quelconque a de I est minimal.

On relie ce nœud a au b et faire $I = I \cup b$, $NI = NI - b$;

(4). Aller au (3) ;

(5). Fin.

✓ **Le critère d'arrêt** : L'algorithme s'arrête lorsque le nombre de sommets retenus est égal à n .

✓ **Résultat** : Trouver un arbre couvrant de poids minimum de G .

3.1.2 Problème du plus long chemin

Le problème de la recherche du plus long chemin dans un graphe se rencontre dans de nombreuses applications. Lorsqu'un chemin existe entre deux sommets dans un graphe, l'être humain se pose rapidement la question non seulement de trouver un tel chemin, mais bien souvent il est intéressé par le plus long chemin possible entre ces deux sommets.

✓ Un chemin joignant (reliant) un sommet x_n à x_y est dit de longueur maximal s'il maximise cette longueur $l(u)$ dans l'ensemble de tous les chemins " U " joignant x_n à x_y ,

✓ La longueur d'un tel chemin est appelée distance maximale de x_n à x_y selon le cas. Par abus de langage, et bien que d'unicité ne soit pas nécessairement réalisée, on parle par fois "de plus long (court) chemin" de x_n à x_y ,

On peut s'intéresser à la recherche d'un plus long chemin dans un graphe :

1. entre deux sommets donnés ;
2. d'un sommet à tous les autres ;
3. entre tous les couples de sommets.

Il existe de nombreux algorithmes permettant l'obtention d'une solution. Nous nous contenterons simplement ici de décrire les algorithmes les plus classiques, sans explorer toutes les situations.

Dans beaucoup de situations, les longueurs sont positives. On utilisera alors l'algorithme de Moore-Dijkstra pour calculer le plus long chemin d'un sommet (arbitrairement le sommet numéro 1) à tous les autres.

1. Algorithme de Dijkstra :

L'algorithme de *Dijkstra* permet uniquement l'obtention un chemin de longueur maximale (une arborescence de plus long distance) sur un Réseau $R = (X, E, d)$ et sous la condition que les longueurs " C_{ij} " soient toutes non négatives, ie : $(l(u) \geq 0, \forall x \in U)$. Il présente de grands avantages de rapidité dans son application.

1.1. Principe de l'algorithme :

- ✓ L'idée de l'algorithme de *dijkstra* est de calculer de proche en proche l'arborescence des plus longs distances, issue du sommet $(x = s)$ à un sommet donné $(y = l)$,
- ✓ Une particularité de cet Algorithme est que les distances (longueurs) s'introduisent dans l'ordre croissant.

1.2. Énoncé de l'algorithme de Dijkstra :

Numérotons les sommets du graphe de 1 à n . Cet algorithme calcule le plus court chemin du sommet 1 à tous les sommets du graphe (il donnera donc la première ligne de la matrice de poids maximum). On construit un vecteur $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ ayant n composantes tel que $\pi(i)$ soit égal à la longueur du plus long chemin allant de 1 au sommet i . On initialise ce vecteur à $(c_{1,i})$, c'est-à-dire à la première ligne de la matrice des poids du graphe. On considère ensuite deux ensembles de sommets, S initialisé à $\{1\}$ et $\bar{S}$ son complémentaire dans X , ensemble des sommets du graphe. À chaque pas de l'algorithme, on ajoute à S des sommets jusqu'à ce que $S = X$ de telle sorte que le vecteur π donne à chaque étape le coût maximal des chemins de 1 aux sommets de S . Les étapes de cet algorithme sont comme suit :

✓ **Données :** Soit $G = (X, E, P)$ un graphe probabiliste,

Première itération : $\pi = (c_{1,i})$ pour $i \in \{1, 2, \dots, n\}$

$S = \{1\}, \bar{S} = \{2, 3, \dots, n\}$.

Deuxième itération : Tant que $\bar{S} \neq \emptyset$,

- Choisir i dans $\bar{S}$ tel que $\pi(i)$ est maximum ;
- Retirer i de $\bar{S}$ et l'ajouter à S ;

- Pour chaque successeur j de i dans $\bar{S}$:

$$\pi(j) = \max(\pi(j), \pi(i) + (i, j)).$$

✓ **Le critère d'arrêt** : L'algorithme s'arrête lorsque le nombre d'arcs retenus est égal à $n - 1$. ✓ **Résultat** : Trouver un arborescence de plus long distance de G .

3.1.3 Problème d'affectation

On considère les ensembles suivants : $P = \{p_1, p_2, \dots, p_n\}$ ensemble de personnes et $T = \{t_1, t_2, \dots, t_m\}$ ensemble de tâches il s'agit de répartir les différentes tâches entre les différentes personnes de façon optimale en tenant compte des préférences de chaque personnes aux différentes tâches, allant de la satisfaction générale à la satisfaction individuelle. la préférence de chaque personne p_i à la tâche t_j est représentée par un nombre note a_{ij} . qui peut correspondre a un temps, un rendement, un coût, un profit, etc.

1. La Méthode Hongroise

L'algorithme hongrois ou méthode hongroise (appelé aussi algorithme de Kuhn) est un algorithme d'optimisation combinatoire, qui résout le problème d'affectation en temps polynomial.

1.2 Principe de La Méthode Hongroise :

La recherche d'une affectation optimale (maximale ou minimale) par la méthode Hongroise est basée sur la notion de zéros indépendants dans une matrice carrée.

2.2 Énoncé de l'algorithme :

Soit $A = (a_{ij})$ une matrice d'affectation de type (n, m) .

- **Initialisation** : Soit $A = a_{ij}$ une matrice de type (n, m) .

1. On repère le plus grand élément de la matrice d'affectation A , soit M ce nombre, on lui retranche tous les éléments de la matrice A . On obtient une nouvelle matrice $B = (b_{ij})$ de type (n, m) telle que :

$$B_{ij} = \begin{cases} M - a_{ij}, & \text{si } a_{ij} \neq \infty, \\ \infty, & \text{si } a_{ij} = \infty. \end{cases}$$

2. On a deux possibilités, sont les suivantes

$$\begin{cases} Si \quad n = m, & \text{aller vers (3)}; \\ Si \quad n \neq m, & \text{On complète la matrice B par des lignes ou des colonnes de zéros, puis aller vers(3).} \end{cases}$$

3. On applique l'algorithme de minimisation à la matrice B .

• **Première étape :**

1. On fait apparaître au moins un zéro sur chaque ligne et chaque colonne de la matrice d'affectation, en retranchant de chaque ligne, le plus petit élément, on fait de même pour chaque colonne,
2. On choisi sur chaque ligne un zéro en l'encadrant, et on barre les zéros qui se trouvent sur la même ligne et la même colonne,
3. Si on a obtenu n zéros indépendants, alors l'affectation est optimale. Tout zéro encadre correspond a la valeur $\sum a_{ij}$ dans la matrice d'affectation, Alors La personne p_i est affectée à la tâche T_j , et la valeur de l'affectation minimale est égale à la somme des a_{ij} de la matrice de départ V (aff), avec $i = \{1, \dots, n\}$ et $j = \{1, \dots, m\}$. Sinon, on passe a la deuxième étape.

• **Deuxième étape :**

1. Marquer toutes les lignes ne contenant pas de zéro encadre,
2. Marquer toutes les colonnes contenant un zéro barrée sur une ligne marquée,
3. Marquer toutes les lignes contenant un zéro encadre sur une colonne marquée,
4. Revenir a (2) et reprendre le même procède jusqu'a ne pouvoir plus marquer ni de lignes ni de colonnes,
5. On barre toutes les lignes non marquées et toutes les colonnes marquées,
6. On considère l'élément non barres de la matrice, soit m le plus petit élément d'entre eux,
 - A chaque élément non barre de la matrice, en retranche l'élément m ,
 - A chaque élément barre deux fois de la matrice, on rajoute l'élément m ,
 - On laisse inchangé les éléments barres une seule fois dans la matrice.
7. On intègre les nouveaux éléments calcules dans A , on obtient ainsi une nouvelle matrice,
8. On choisit de nouveaux zéros pour les lignes ne contenant pas de zéro encadre,

9. Si on a n zéros indépendant allé vers (3) de la première étape sinon aller vers (2) de la première étape.

3.1.4 Problème du flot

Dans un graphe orienté G , un flot est l'affectation d'une valeur réelle à chaque arc de G , représentant une quantité transportée sur cet arc, de telle sorte que, en chaque sommet, la somme des flots entrants soit égale à la somme des flots sortants (loi de Kirchhoff : conservation des flux en chaque sommet).

Parmi les problèmes les plus classique, on peut citer celui de la recherche d'un flot maximal. On se donne une capacité maximale sur chaque arc qui sera une borne supérieure du flot autorisé sur cet arc. Le problème du flot maximal consiste à déterminer un flot dont la valeur en un certain lieu est maximale. On peut, de plus se donner un poids de transport d'une unité de flot sur chaque arc et chercher le flot maximal de poids minimal.

1.1 Principe d'un réseau de transport :

Le réseau de transport donné est de déterminer un flot de valeur maximale ainsi que les flots le long de chaque arc. Il arrive fréquemment également que l'on doive considérer des réseaux avec des capacités localisées non seulement sur les arêtes mais également sur les sommets. C'est notamment le cas pour les réseaux téléphoniques pour lesquels la limite de capacité est autant due aux lignes qu'aux centraux.

1.2 Recherche d'un flot maximal “ algorithme de Ford et Fulkerson” :

La valeur maximal d'un flot allant de S ou V sur le réseau $R = (S, A, V)$ est égal à la capacité d'une (S, V) coupe $\Omega^+(s)$ de capacité minimum de s et v .

- **étape 1 :** Initialisation

- Position d'un flot réalisable ($x = 0 = f$).
- Marquer la source s d'un signe $(s, +\infty)$.
- Tous les autres sommets sont non marqués et non examinés.

- **étape 2 :** Examen d'un sommet

- Pour tout sommet j tel que $a = (i, j) \in A$ avec :

$$\begin{cases} f(a) < v(a); \\ i \text{ marqué}, j \text{ non marqué.} \end{cases}$$

faire marqué j d'un signe $\{+i, \min[s, v(a) - f(a)]\}$

- Pour tout sommet j tel que $a = (j, i) \in A$ avec :

$$\begin{cases} f(a) > 0; \\ i \text{ marqué}, j \text{ non marqué.} \end{cases}$$

faire marqué j d'un signe $\{-i, \min[s, f(a)]\}$,

- Le sommet j maintenant examiner.

• **étape 3** : Test d'optimalité du flot couvrant

- Si le point p est marqué aller en l'étape (4)
- Si non voir
 - S'il existe d'autre sommet marquée et non examinée. retourner à l'étape (2).
 - Si non il n'existe plus de sommets marquée et non examinée.
- Terminer le flot courant est maximum.

• **étape 4** : Actualisation des flots

- Identifier la chaîne augmentent joignant la source s au point p .
- Actualiser le flot courant le long de la chaîne augmentent.
- Effacer toute marque et retourner à l'étape (1).

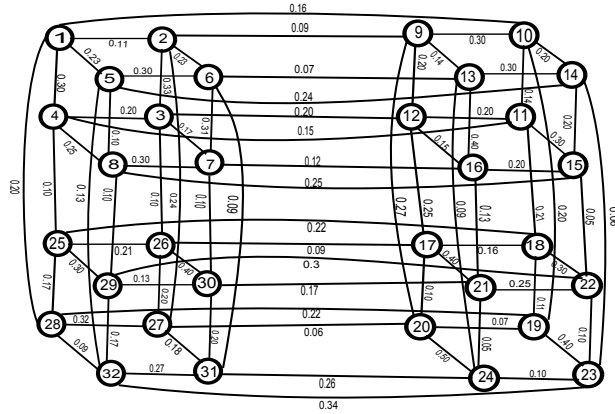
4

Partie d'application

4.1 Problème de l'arbre couvrant de poids minimum

Considérons 2^n personnes qui se communiquent entre elles dans un réseau tels que la probabilité qu'un message confidentiel entre la personne i et la personne j soit intercepté par une personne étrangère est p_{ij} .

Trouver une communication qui minimise les probabilité d'interception, car ce dernier ne possède pas de cycle pour ce faire nous allons utiliser : l'algorithme de Kruskal et Prim. Cette situation est représenté par le graphe suivant :



i	61	62	63	64	65	66	67	68	69	70
e_i	(24,31)	(31,32)	(1,4)	(6,5)	(7,8)	(9,10)	(14,13)	(15,11)	(18,22)	(22,29)
$p(e_i)$	0.26	0.27	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30

i	71	72	73	74	75	76	77	78	79	80
e_i	(25,29)	(6,7)	(27,28)	(2,3)	(23,32)	(13,16)	(17,21)	(23,19)	(26,30)	(20,24)
$p(e_i)$	0.30	0.31	0.32	0.33	0.34	0.40	0.40	0.40	0.40	0.50

Soit : $W = \emptyset$; $i = 1$; $m = 80$.

1^{re} Itération :

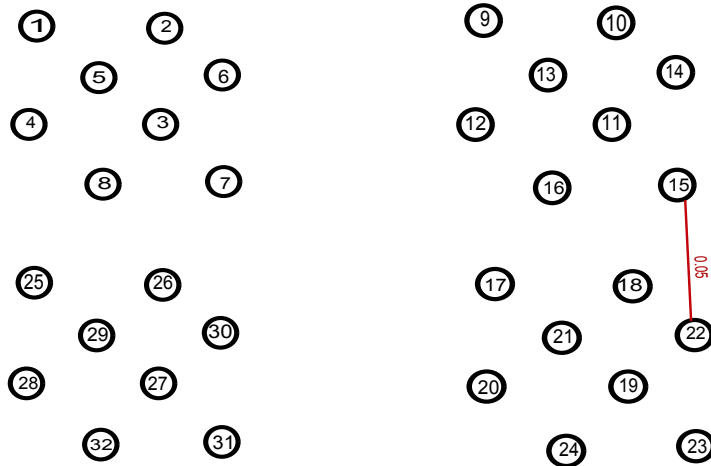
On a : $e_1 = (15, 22)$;

Soit : $W = W \cup \{e_1\} = W \cup \{(15, 22)\}$;

Le graphe $G = (X, W)$ ne contient pas de cycle, on pose alors :

$W = \{(15, 22)\}$, et $|W| = 1$.

On a : $i \neq m$, alors on pose $i = i + 1 = 2$.



2^{me} Itération :

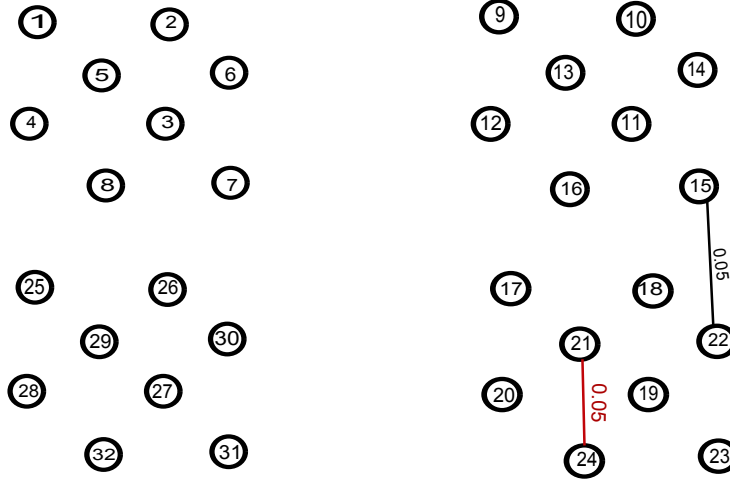
On a : $e_2 = (21, 24)$;

Soit : $W = W \cup \{e_2\}$;

Le graphe $G = (X, W)$ ne contient pas de cycle, on pose alors :

$W = \{(15, 22), (21, 24)\}$, et $|W| = 2$.

On a : $i \neq m$, alors on pose $i = i + 1 = 3$.



3^{me} Itération :

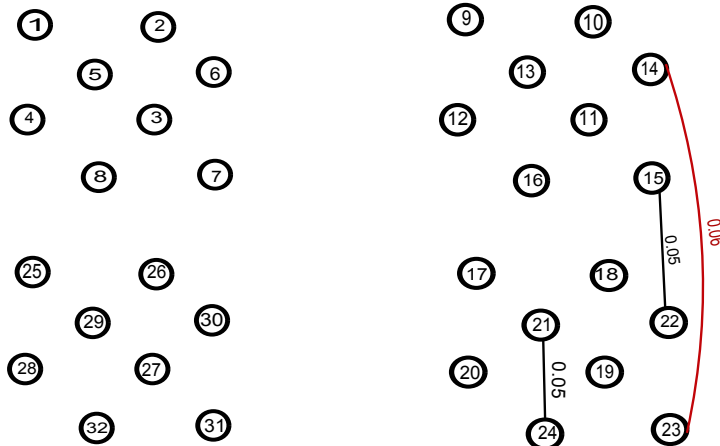
On a : $e_3 = (14, 23)$;

Soit : $W = W \cup \{e_3\}$;

Le graphe $G = (X, W)$ ne contient pas de cycle, on pose alors :

$W = \{(15, 22), (21, 24), (14, 23)\}$, et $|W| = 3$.

On a : $i \neq m$, alors on pose $i = i + 1 = 4$.



4^{me} Itération :

On a : $e_4 = (20, 27)$;

Soit : $W = W \cup \{e_4\}$;

Le graphe $G = (X, W)$ ne contient pas de cycle, on pose alors :

$W = \{(15, 22), (21, 24), (14, 23), (20, 27)\}$, et $|W| = 4$.

On a : $i \neq m$, alors on pose $i = i + 1 = 5$.

32^{me} Itération :

On a : $e_{32} = (7, 3)$;

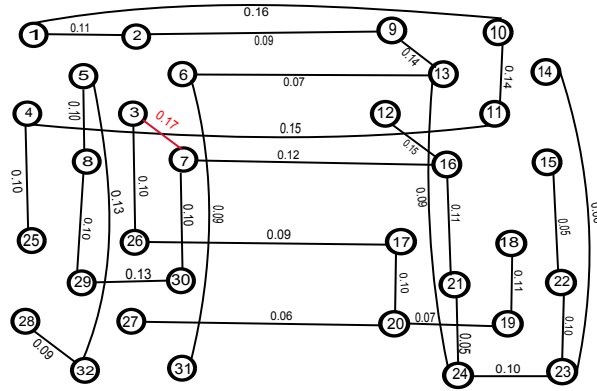
Soit : $W = W \cup \{e_{32}\}$;

Le graphe $G = (X, W)$ ne contient pas de cycle, on pose alors :

$W = \{(15, 22), (21, 24), (14, 23), (20, 27), (6, 13), (19, 20), (2, 9), (6, 31), (13, 24), (17, 26), (28, 32), (7, 30), (3, 26), (5, 8), (4, 25), (8, 29), (17, 20), (22, 23), (23, 24), (1, 2), (18, 19), (7, 16), (5, 32), (16, 21), (30, 29), (9, 13), (10, 11), (4, 11), (12, 16), (1, 10), (7, 3)\}$, et $|W| = 31$.

On a : $i \neq m$, alors on pose $i = i + 1 = 33$.

terminer. $|W| = n - 1 = 32 - 1 = 31$;



L'arbre de poids (probabilites) minimum est représenté par le graphe $A = (X; W)$.
La probabilité minimale totale est :

$$P(W) = 1 - \prod_{i=1}^{32} (1 - P_{ij})$$

$$P(W) = 1 - [(1 - p(15, 22)) \times (1 - p(21, 24)) \times (1 - p(14, 23)) \times (1 - p(20, 27)) \times (1 - p(6, 13)) \times (1 - p(19, 20)) \times (1 - p(2, 9)) \times (1 - p(6, 31)) \times (1 - p(13, 24)) \times (1 - p(17, 26)) \times (1 - p(28, 32))$$

$$\begin{aligned} & \times (1 - p(7, 30)) \times (1 - p(3, 26)) \times (1 - p(5, 8)) \times (1 - p(4, 25)) \times (1 - p(8, 29)) \times (1 - p(17, 20)) \\ & \times (1 - p(22, 23)) \times (1 - p(23, 24)) \times (1 - p(1, 2)) \times (1 - p(18, 19)) \times (1 - p(7, 16)) \times (1 - p(5, 32)) \\ & \times (1 - p(16, 21)) \times (1 - p(30, 29)) \times (1 - p(9, 13)) \times (1 - p(10, 11)) \times (1 - p(4, 11)) \times (1 - p(12, 16)) \\ & \times (1 - p(1, 10)) \times (1 - p(7, 3))] \end{aligned}$$

$$\begin{aligned} P(W) = 1 - & [(1 - 0.05) \times (1 - 0.05) \times (1 - 0.06) \times (1 - 0.06) \times (1 - 0.07) \times (1 - 0.07) \times (1 - 0.09) \\ & \times (1 - 0.09) \times (1 - 0.09) \times (1 - 0.09) \times (1 - 0.10) \times (1 - 0.10) \times (1 - 0.10) \\ & \times (1 - 0.10) \times (1 - 0.10) \times (1 - 0.10) \times (1 - 0.10) \times (1 - 0.11) \times (1 - 0.11) \\ & \times (1 - 0.12) \times (1 - 0.13) \times (1 - 0.13) \times (1 - 0.13) \times (1 - 0.14) \times (1 - 0.14) \times (1 - 0.15) \\ & \times (1 - 0.15) \times (1 - 0.16) \times (1 - 0.17)] \end{aligned}$$

$$P(W) = 0.9683.$$

B) Appliquon de l'algorithme PRIM

Initialisation

$$I = \emptyset.$$

$$NI = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32\}.$$

1^{re} Itération

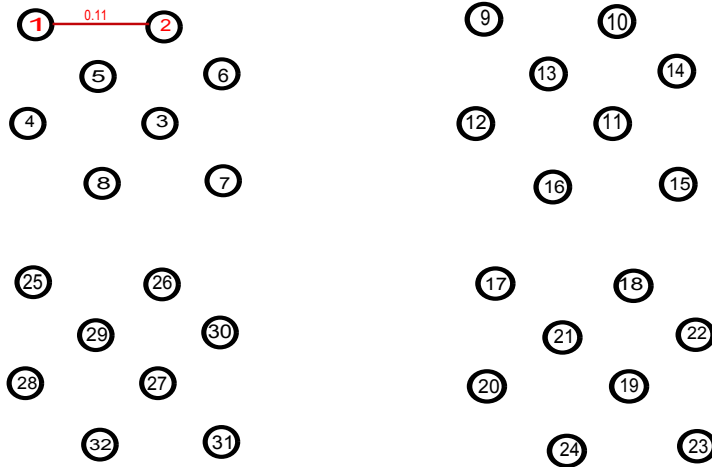
On prend le sommet 1 comme un sommet de départ.

$$\min\{P(1, 2), P(1, 4), P(1, 5), P(1, 28), P(1, 10)\} = \{P(1, 2)\};$$

Donc on ajoute l'arrête $e_1 = (1, 2)$ au graphe ;

$$\text{Soit : } I = I \cup \{2\} = \{1, 2\};$$

$$NI = \{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32\}.$$



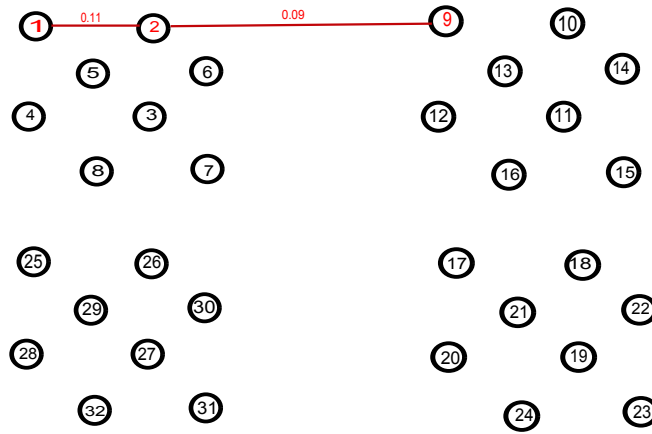
2^{me} Itération

$$\min\{P(1, 4), P(1, 5), P(1, 28), P(1, 10), P(2, 3), P(2, 6), P(2, 9), P(2, 27)\} = \{P(2, 9)\};$$

Donc on ajoute l'arrête $e_2 = (2, 9)$ au graphe ;

$$\text{Soit : } I = I \cup \{9\} = \{1, 2, 9\};$$

$$NI = \{3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32\}.$$

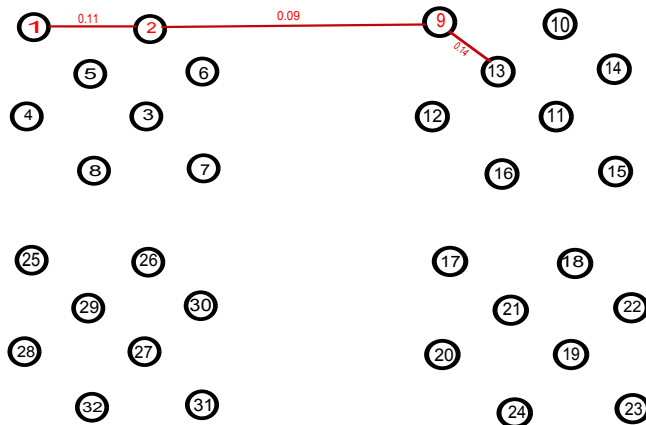
**3^{me} Itération**

$$\min\{P(1, 4), P(1, 5), P(1, 28), P(1, 10), P(2, 3), P(2, 6), P(2, 27), P(9, 10), P(9, 12), P(9, 13), P(9, 20)\} = \{P(9, 13)\};$$

Donc on ajoute l'arrête $e_3 = (9, 13)$ au graphe ;

$$\text{Soit : } I = I \cup \{13\} = \{1, 2, 9, 13\};$$

$$NI = \{3, 4, 5, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32\}.$$



$$\min\{P(1, 4), P(1, 5), P(1, 28), P(1, 10), P(2, 3), P(2, 6), P(2, 27), P(9, 10), P(9, 12), P(9, 20), P(13, 6), P(13, 14), P(13, 16), P(13, 24)\} = \{P(13, 6)\};$$

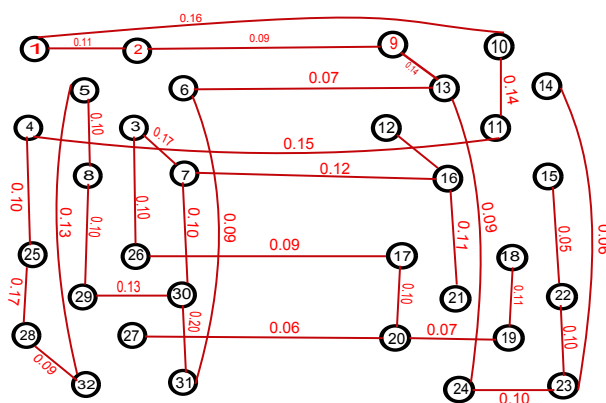
Soit : $I = I \cup \{6\} = \{1, 2, 9, 13, 6\}$;

$$NI = \{3, 4, 5, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32\}.$$

$$\begin{aligned} & \min\{P(1, 4), P(1, 5), P(1, 28), P(2, 3), P(2, 6), P(2, 27), P(9, 10), P(9, 12), P(9, 20), P(13, 14), P(13, 16), \\ & P(6, 5), P(6, 7), P(6, 31), P(24, 20), P(24, 21), P(24, 32), P(31, 27), P(31, 30), P(31, 32), P(23, 19), P(23, 32), \\ & P(14, 5), P(14, 10), P(14, 13), P(14, 15), P(22, 18), P(22, 21), P(22, 29), P(15, 11), P(15, 16), P(15, 8), P(10, \\ & P(10, 14), P(10, 19), P(11, 12), P(11, 15), P(11, 18), P(4, 3), P(4, 8), P(25, 18), P(25, 26), P(25, 29), P(28, 19), \\ & P(28, 27), P(32, 29), P(32, 31), P(5, 6), P(8, 7), P(30, 21), P(30, 26), P(30, 31), P(21, 17), P(12, 3), P(17, 18), \\ & P(19, 18)\} = \{P(19, 18)\}; \end{aligned}$$

Soit : $I = I \cup \{18\} = \{1, 2, 9, 13, 6, 24, 31, 23, 14, 22, 15, 10, 11, 4, 25, 28, 32, 5, 8, 29, 30, 7, 16, 21, 12, 3, 26, 17, 20, 27, 19, 18\}$;

$$NI = \{\emptyset\}.$$



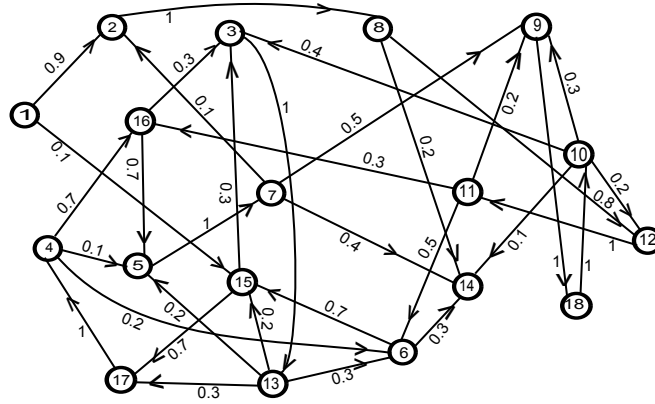
La probabilité minimale totale est :

$$1 - \prod_1^{32}(1 - P_{ij}) = 0.9683.$$

4.2 Problème de plus long chemin

A) Application de l'algorithme de Dijkstra

Dans une région en proie à un conflit ou à une catastrophe naturelle, un convoi doit être acheminé d'une ville s à une ville t . Le réseau routier de la région est donné par un graphe valué $G = (X, A, P)$. X désigne l'ensemble des villes et A l'ensemble des tronçons de route entre villes. Pour tout arc (i, j) , la valuation $p(i, j)$ est la probabilité de parcourir sans dommage la route de i à j . On souhaite déterminer un itinéraire maximisant la probabilité pour le convoi d'arriver sans dommage en t , c'est-à-dire un chemin de fiabilité maximale.



Initialisation :

$S = 1$, $\pi(1) = 1$, $\bar{S} = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$.

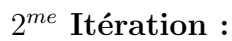
Les successeurs de 1 sont 2 et 15.

1^{re} Itération :

$i = 2$ car $\pi(2) = \max\{0.9, 0.1\} = 0.9$, $\pi(2) = 0.9$, $S = \{1, 2\}$;

$\bar{S} = \{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}$.

Successeur de 2 est 8.


$$\overline{S} = \{3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\}.$$

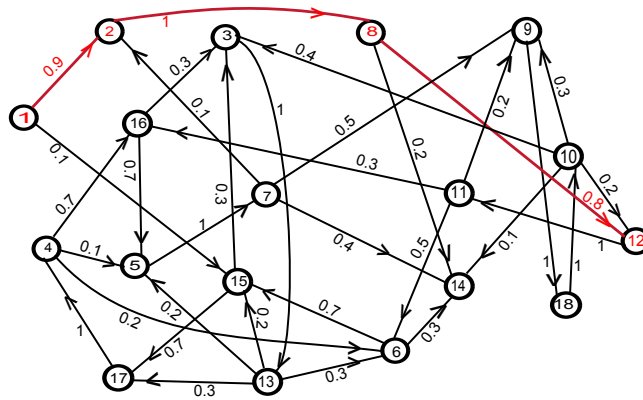
The figure shows a directed graph with 18 nodes. Node 1 is the source and node 18 is the sink. The edges and their capacities are as follows:

- 1 → 2: 0.9
- 1 → 4: 0.7
- 2 → 3: 0.3
- 2 → 7: 0.1
- 3 → 7: 1
- 3 → 9: 0.4
- 4 → 5: 0.1
- 4 → 17: 1
- 5 → 15: 0.2
- 5 → 16: 0.7
- 6 → 13: 0.3
- 6 → 14: 0.5
- 7 → 15: 0.3
- 7 → 11: 0.5
- 8 → 9: 0.9
- 8 → 11: 0.3
- 9 → 10: 0.3
- 9 → 14: 0.2
- 10 → 12: 0.8
- 10 → 18: 1
- 11 → 12: 0.7
- 11 → 14: 0.1
- 12 → 18: 1
- 13 → 15: 0.2
- 13 → 17: 0.3
- 14 → 15: 0.7
- 15 → 17: 0.1
- 16 → 7: 1
- 17 → 13: 0.3

3^{me} Itération :

$$\overline{S} = \{3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 18\}.$$

Successeur de 12 est 11 .



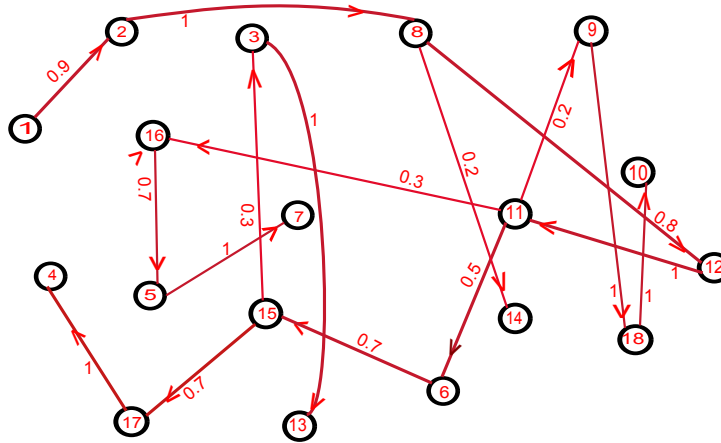
4^{me} Itération :

$i = 11$, car $\pi(11) = \pi(12) * \gamma(12, 11) = 0.72 * 1 = 0.72$, Donc $\pi(11) = 0.72$, $S = \{1, 2, 8, 12, 11\}$;

$\bar{S} = \{3, 4, 5, 6, 7, 9, 10, 13, 14, 15, 16, 17, 18\}$.

Les successeurs de 11 sont 6, 9 et 16.

17^{me} Itération :



Ages \ Horaires	10h - 11h30	11h30 - 13h	13h - 15h	15h - 16h	16h - 18h	18h - 21h	Après 21h
Moins de 10 ans	0.70	0	0.15	0.05	0	0	0.10
10 à 15 ans	0.50	0	0.30	0	0	0	0.20
15 à 20 ans	0.55	0	0	0.25	0	0	0.20
20 à 25 ans	0.40	0.10	0	0	0.30	0.20	0
25 à 40 ans	0.15	0	0.20	0.30	0.10	0	0.25
40 à 60 ans	0.80	0	0	0.10	0	0.10	0
Plus de 60 ans	0.30	0.15	0	0.25	0.10	0.20	0

TABLE 4.1 – Tranche horaire par tranches d'âge

4.3 Problème d'affectation optimale de programmes de télévision :

1.1 Enonce du problème : Une chaîne de télévision décide de revoir les programmes offerts aux téléspectateurs. Elle a effectué une enquête auprès de la population, sur un échantillon représentatif de mille personnes.

La question posée était de classer les probabilités des programmes diffusés durant différentes tranches horaires.

Les horaires sélectionnés étaient les suivants : 10h à 11h30 ; 11h30 à 13h ; 13h à 15h ; 15h à 16h ; 16h à 18h ; 18h à 21h ; après 21h.

La classification des probabilités des programmes diffusés se notait :

$$\begin{cases} 0 = & \text{Pas intéressé ou bien indisponible ;} \\ \frac{1}{2} = & \text{Intéressé ;} \\ 1 = & \text{Très intéressé.} \end{cases}$$

Après avoir effectué la moyenne des classements par catégorie d'âges, le bilan de l'enquête est indiqué dans le tableau suivant.

Trouver l'affectation optimale des programmes par tranches horaires et par tranche d'âge, l'objectif étant de maximiser la satisfaction de chacune des catégories de téléspectateurs.

1.2 Solution du problème : On détermine des catégories optimale des programmes, par tranche horaire et par tranche d'âges.

Remarque : La méthode hongroise vue précédemment permet de minimiser des coûts de réparation. Or dans notre cas, il s'agit d'obtenir une maximisation de la satisfaction des téléspectateurs.

Il faut donc modifier la matrice de départ en retranchant toutes les valeurs à la valeur maximale du tableau 1 ; ce qui conduit aux tableaux suivants :

0.7	0	0.15	0.05	0	0	0.1
0.50	0	0.30	0	0	0	0.20
0.55	0	0	0.25	0	0	0.20
0.40	0.10	0	0	0.30	0.20	0
0.15	0	0.20	0.30	0.10	0	0.25
0.80	0	0	0.10	0	0.10	0
0.30	0.15	0	0.25	0.10	0.20	0

0.10	0.80	0.65	0.75	0.80	0.80	0.70
0.30	0.80	0.50	0.80	0.80	0.80	0.60
0.25	0.80	0.80	0.55	0.80	0.80	0.60
0.40	0.70	0.80	0.80	0.50	0.60	0.80
0.65	0.80	0.60	0.50	0.70	0.80	0.55
0	0.80	0.80	0.70	0.80	0.70	0.80
0.50	0.65	0.80	0.55	0.70	0.60	0.80

$$a'_{ij} = |\max(a_{ij}) - a_{ij}|$$

$$\min = 0.10$$

0	0.70	0.55	0.65	0.70	0.70	0.60
0.20	0.70	0.40	0.70	0.70	0.70	0.50
0.15	0.70	0.70	0.45	0.70	0.70	0.50
0.30	0.60	0.70	0.70	0.40	0.50	0.70
0.50	0.70	0.50	0.40	0.60	0.70	0.45
0 ^x	0.80	0.80	0.70	0.80	0.70	0.80
0.40	0.55	0.70	0.45	0.60	0.50	0.70

$$\min = 0.40$$

0	0.30	0.15	0.25	0.30	0.30	0.20
0.20	0.30	0	0.30	0.30	0.30	0.10
0.15	0.30	0.30	0.05	0.30	0.30	0.10
0.30	0.20	0.30	0.30	0	0.10	0.30
0.50	0.30	0.10	0	0.20	0.30	0.05
0 ^x	0.40	0.40	0.30	0.40	0.30	0.40
0.40	0.15	0.30	0.05	0.20	0.10	0.30

$$\min = 0.05$$

0	0.25	0.10	0.20	0.25	0.25	0.15
0.25	0.30	0	0.30	0.30	0.30	0.10
0.15	0.25	0.25	0	0.25	0.25	0.05
0.35	0.20	0.30	0.30	0	0.10	0.30
0.55	0.30	0.10	0 [×]	0.20	0.30	0.05
0 [×]	0.35	0.35	0.25	0.35	0.25	0.35
0.40	0.10	0.25	0 [×]	0.15	0.05	0.25

min = 0.05

0	0.20	0.05	0.20	0.20	0.20	0.10
0.30	0.30	0	0.35	0.30	0.30	0.10
0.15	0.20	0.20	0	0.20	0.20	0 [×]
0.40	0.20	0.30	0.35	0	0.10	0.30
0.55	0.25	0.05	0 [×]	0.15	0.25	0
0 [×]	0.30	0.30	0.25	0.30	0.20	0.30
0.40	0.05	0.20	0 [×]	0.10	0	0.20

min = 0.05

0	0.20	0.05	0.20	0.20	0.20	0.10
0.30	0.30	0	0.35	0.30	0.30	0.10
0.15	0.20	0.20	0	0.20	0.20	0 [×]
0.40	0.20	0.30	0.35	0	0.10	0.30
0.55	0.25	0.05	0 [×]	0.15	0.25	0
0 [×]	0.30	0.30	0.25	0.30	0.20	0.30
0.40	0.05	0.20	0 [×]	0.10	0	0.20

min = 0.05

0	0.15	0	0.20	0.15	0.15	0.10
0.35	0.30	0 [×]	0.40	0.30	0.30	0.15
0.15	0.15	0.15	0	0.15	0.15	0 [×]
0.45	0.20	0.30	0.40	0	0.10	0.35
0.55	0.20	0 [×]	0 [×]	0.10	0.20	0
0	0.25	0.25	0.25	0.25	0.15	0.30
0.40	0.05	0.20	0.05	0.10	0	0.25

min = 0.10

0	0.05	0 [×]	0.20	0.05	0.05	0.10
0.35	0.20	0	0.40	0.20	0.20	0.15
0.15	0.05	0.15	0	0.05	0.05	0 [×]
0.50	0.20	0.40	0.50	0	0.10	0.45
0.55	0.10	0 [×]	0 [×]	0 [×]	0.10	0
0 [×]	0.15	0.25	0.25	0.15	0.05	0.30
0.50	0.05	0.30	0.15	0.10	0	0.35

min = 0.05

0 [×]	0	0 [×]	0.20	0.05	0 [×]	0.10
0.35	0.15	0	0.40	0.20	0.15	0.15
0.15	0 [×]	0.15	0	0.05	0 [×]	0 [×]
0.50	0.15	0.40	0.50	0	0.05	0.45
0.55	0.05	0 [×]	0 [×]	0 [×]	0.05	0
0	0.10	0.25	0.25	0.15	0 [×]	0.30
0.55	0.05	0.35	0.20	0.15	0	0.40

Optimum

En fonction des résultats obtenus et après études plus approfondies, une réparation des programmes comme celle qui est représentée ci-dessous peut être proposée.

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
21h	Dessins -animés						
11h30	Variétés et Jeux						
13h	Variétés rétro Histoire émission médicale	Emissions 15 ans		Animaux,culture Histoire, variétés,etc..		séries sports,variétés	
	Journal Télévisé						
15h	Feuilleton ou séries						
16h	Emissions-15ans,musique, dessins-animés... ets						Documentation
18h	Emissions pou tout,jeux et rire						Actual
	Mini-journal pour moins de 15 ans						
21h	Films	Débat télévisé	Variétés	Actual Economique	série Culture	série Variétés	Films

TABLE 4.2 – Proposition une réparation de programmes de télévision

4.4 Application sur MATLAB

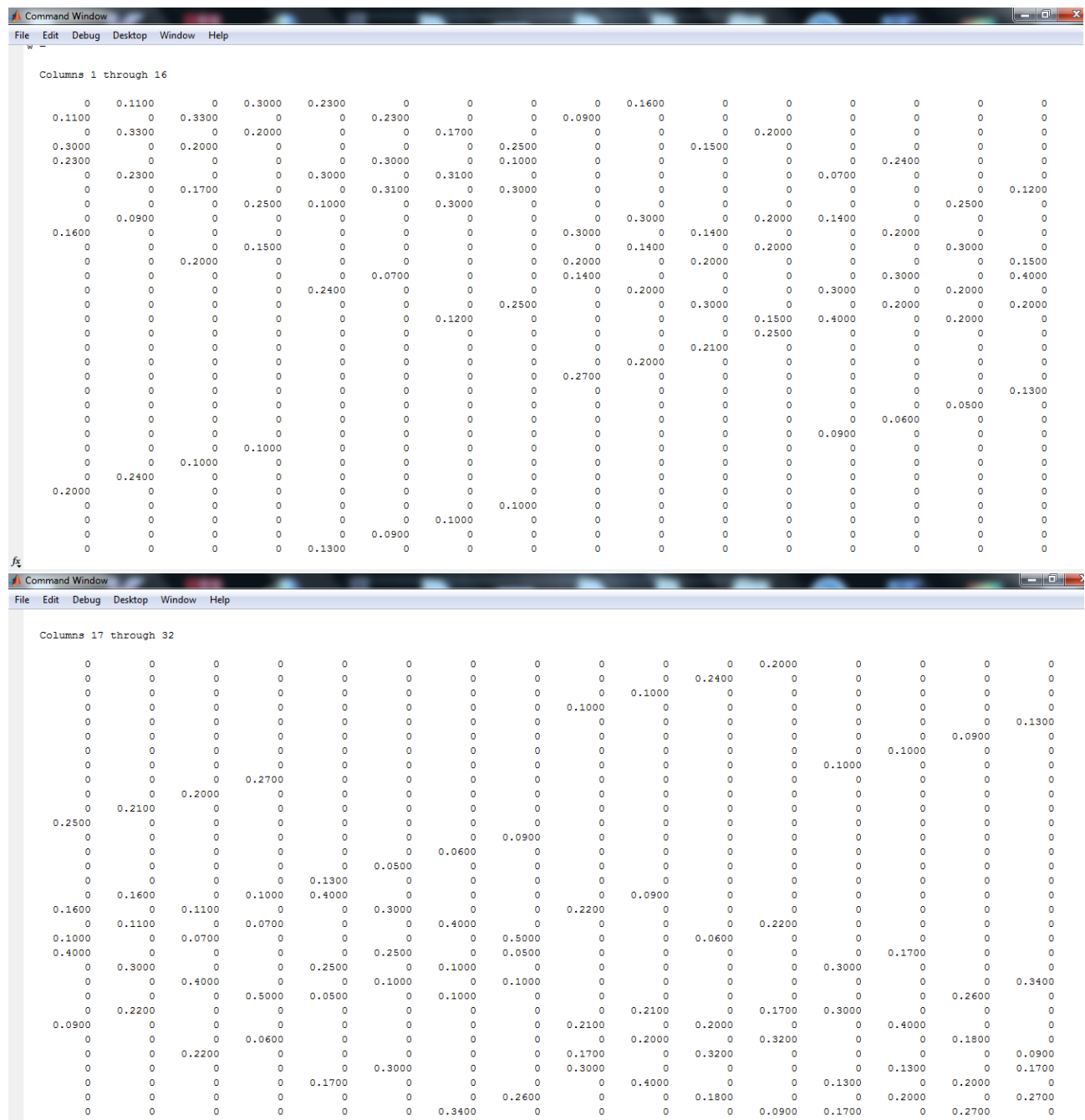
MATLAB est un logiciel commercial de calcul interactif. Il permet de réaliser des simulations numériques basées sur des algorithmes d'analyse numérique. Il peut donc être utilisé pour la résolution approchée d'équations différentielles, d'équations aux dérivées partielles ou de systèmes linéaires.

Dans toutes les représentations graphiques, le logiciel se base sur des données discrètes rangées dans des matrices ou des vecteurs colonnes.

A) Programmation de l'algorithme Kruskal sur MATLAB

Dans ce programme on a fait rentrer la matrice de transition de problème de telecommu-
nication qu'on a déjà vue et sa matrice d'adjacence comme donnés.

Cette figure montre La matrice de transitions qu'on a fait rentré :



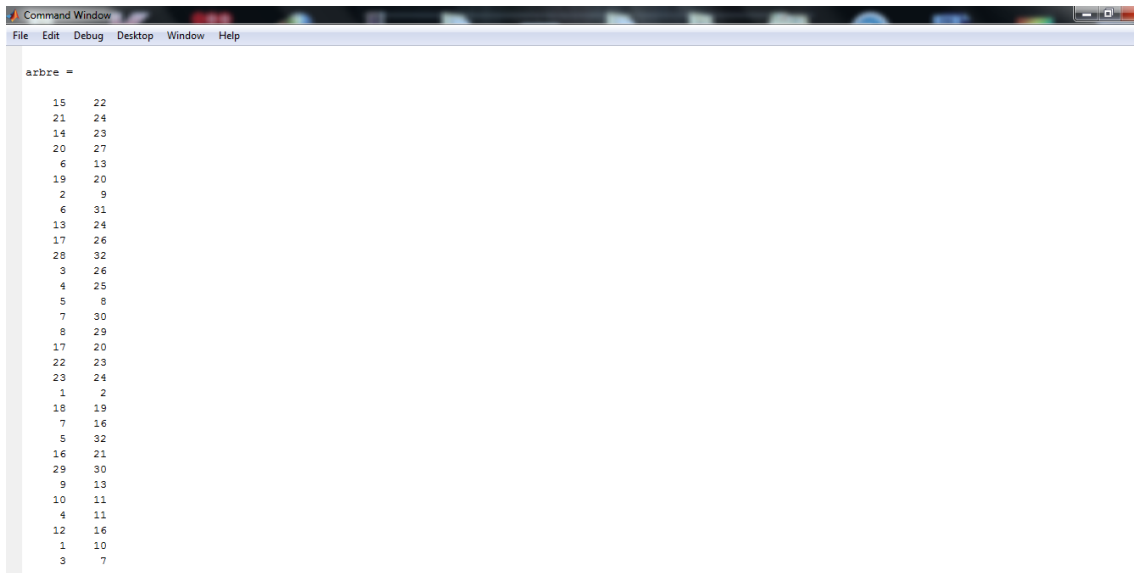
Columns 1 through 16

0	0.1100	0	0.3000	0.2300	0	0	0	0	0.1600	0	0	0	0	0	0
0.1100	0	0.3300	0	0	0.2300	0	0	0.0900	0	0	0	0	0	0	0
0	0.3300	0	0.2000	0	0	0.1700	0	0	0	0	0.2000	0	0	0	0
0.3000	0	0.2000	0	0	0	0	0.2500	0	0	0.1500	0	0	0	0	0
0.2300	0	0	0	0	0.3000	0	0.1000	0	0	0	0	0.2400	0	0	0
0	0.2300	0	0	0.3000	0	0.3100	0	0	0	0	0	0.0700	0	0	0
0	0	0.1700	0	0	0.3100	0	0.3000	0	0	0	0	0	0	0.1200	0
0	0	0	0.2500	0.1000	0	0.3000	0	0	0	0	0	0	0.2500	0	0
0	0.0900	0	0	0	0	0	0	0.3000	0	0.2000	0.1400	0	0	0	0
0.1600	0	0	0	0	0	0	0	0.3000	0	0.1400	0	0	0.2000	0	0
0	0	0	0.1500	0	0	0	0	0	0.1400	0	0.2000	0	0	0.3000	0
0	0	0.2000	0	0	0	0	0	0.2000	0	0.2000	0	0	0	0	0.1500
0	0	0	0	0	0.0700	0	0	0.1400	0	0	0	0	0.3000	0	0.4000
0	0	0	0	0.2400	0	0	0	0	0.2000	0	0	0.3000	0	0.2000	0
0	0	0	0	0	0	0	0.2500	0	0	0.3000	0	0	0.2000	0	0.2000
0	0	0	0	0	0	0.1200	0	0	0	0	0.1500	0.4000	0	0.2000	0
0	0	0	0	0	0	0	0	0	0	0	0.2500	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.2100	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.2000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0.2700	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1300
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0500	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.0600	0	0
0	0	0	0.1000	0	0	0	0	0	0	0	0	0.0900	0	0	0
0	0	0.1000	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.2400	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.2000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.1000	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.0900	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1300	0	0	0	0	0	0	0	0	0	0	0

Columns 17 through 32

0	0	0	0	0	0	0	0	0	0	0	0.2000	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.2400	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.1000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1300
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0900	0
0	0	0	0	0	0	0	0	0	0	0	0	0.1000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.1000	0	0	0
0	0	0	0.2700	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0.2000	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.2100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.2500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.0900	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.0600	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.0500	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.1300	0	0	0	0	0	0	0	0	0	0	0
0	0.1600	0	0.1000	0.4000	0	0	0	0.0900	0	0	0	0	0	0	0
0.1600	0	0.1100	0	0	0.3000	0	0	0.2200	0	0	0	0	0	0	0
0	0.1100	0	0.0700	0	0	0.4000	0	0	0	0.2200	0	0	0	0	0
0.1000	0	0.0700	0	0	0	0	0.5000	0	0.0600	0	0	0	0	0	0
0.4000	0	0	0	0	0.2500	0	0.0500	0	0	0	0	0.1700	0	0	0
0	0.3000	0	0	0.2500	0	0.1000	0	0	0	0	0.3000	0	0	0	0
0	0	0.4000	0	0	0.1000	0	0.1000	0	0	0	0	0	0	0.3400	0
0	0	0	0.5000	0.0500	0	0.1000	0	0	0	0	0	0	0.2600	0	0
0	0.2200	0	0	0	0	0	0	0	0.2100	0	0.1700	0.3000	0	0	0
0.0900	0	0	0	0	0	0	0	0.2100	0	0.2000	0	0.4000	0	0	0
0	0	0	0.0600	0	0	0	0	0	0.2000	0	0.3200	0	0	0.1800	0
0	0	0.2200	0	0	0	0	0	0.1700	0	0.3200	0	0	0	0	0.0900
0	0	0	0	0	0.3000	0	0	0.3000	0	0	0	0.1300	0	0.1700	0
0	0	0	0	0.1700	0	0	0	0.4000	0	0	0.1300	0	0.2000	0	0
0	0	0	0	0	0	0	0.2600	0	0	0.1800	0	0	0.2000	0	0.2700
0	0	0	0	0	0	0.3400	0	0	0	0.0900	0.1700	0	0	0.2700	0

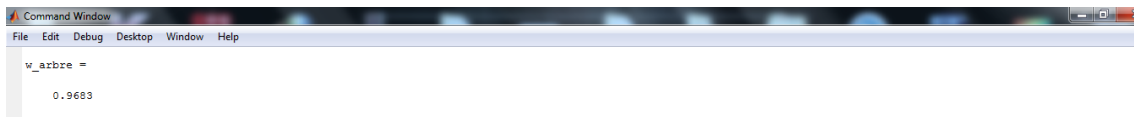
Les résultats obtenue est comme la figure suivante la montre :



```
Command Window
File Edit Debug Desktop Window Help

arbre =

15 22
21 24
14 23
20 27
6 13
19 20
2 9
6 31
13 24
17 26
28 32
3 26
4 25
5 8
7 30
8 29
17 20
22 23
23 24
1 2
18 19
7 16
5 32
16 21
29 30
9 13
10 11
4 11
12 16
1 10
3 7
```



```
Command Window
File Edit Debug Desktop Window Help

w_arbre =

0.9683
```

tel que arbre présente l'arbre couvrant minimal et w_arbre désigne son poids.

Conclusion

L'importance de la théorie des graphes vient du fait qu'elle fournit un cadre conceptuel adéquat pour l'analyse et la résolution de nombreux problèmes. Dans la vie quotidienne la plus part des problèmes réels n'est pas possible d'inclure tous les facteurs environnants, En effet pour modéliser ce type de problème en utilisant des graphes probabilistes. Ce travail présente une vision globale sur les graphes probabilistes qui englobe un sujet très intéressant qui est l'optimisation dans les graphes probabilistes.

Notre travail consiste à donner aux lecteurs quelques techniques d'optimisation dans les graphes en générale, utilisable pour résoudre les problèmes rencontrés. et l'exactitude de notre objectif consiste à résoudre des problèmes en utilisant ces techniques particulièrement dans les graphes probabilistes. pour cela on a opté la résolution de trois problèmes.

Le premier consiste à résoudre un problème de télécommunication entre trente deux personnes en prenant compte de la fiabilité de communication c'est à dire qu'il y a une probabilité qu'un message entre deux personnes soit intercepté par une personne étrangère en utilisant l'algorithme de KRUSKAL et PRIM.

Dans le deuxième on a considéré un réseau routier de la région en proie d'un conflit ou une catastrophe naturelle, le problème consiste à chercher le plus fiable chemin entre cette région et une ville donnée en utilisant l'algorithme de DIJKSTRA.

Le troisième consiste à résoudre un problème de classement optimal des programmes de télévision en satisfaisant chacune des catégories de téléspectateurs, pour cela on a utilisé la méthode d'HONGROISE.

Resumé

L'objectif de ce travail est de montrer l'utilité de la "théorie des graphes" en optimisation, et cela prenant des graphes probabilistes. Ce mémoire contribue également à montrer l'importance d'optimiser les poids pour un distributeur de problème, et cela en cherchant quelques problèmes d'optimisation pour les quelles nous donnons quelque algorithmes de résolutions pour chaque problème, suivie d'une résolution, en faisant appel à un programme réalisé sous **Matlab**.

Mots-clés : Graphe probabiliste, optimisation, Problème du plus court chemin, Problème de l'arbre couvrant de poids minimum, ...

Abstract

The objective of this work is to show the utility of the "Graph theory" in optimization, and that by taking of the probabilistic graphs. This memory also contributes to show the importance to optimize the weights for a distributor of problems, and that by seeking some problems of optimization for which we give some algorithms of resolutions for each problem, followed by a resolution, by calling on a program carried out under Matlab.

Keywords : Probabilistic graph, optimization, Problem of the more long way, Problem of the tree covering of minimum weight, ...

Bibliographie

- [1] Amour, Mémoire et Modélisation, Sous la direction du Prof. Christian Mazza, 2010.
- [2] A. HENRY-LABORD ÈRE. « Cours de recherche opérationnell », 1. E.N.P.C, Paris, 1995.p.218 (003 HEN : CDPS). 1995.
- [3] C. BERGE. « Graphes et hypergraphes. Monographies universitaires de mathématiques ». Dunod, Paris, 1970. p.XVIII-502 , (511.5 BER : CDPS). 1970.
- [4] C. PRINS, « Algorithmes de graphes, Eyrolles », Paris, 1994.
- [5] F. DROESBEKE, M. HALLIN, C. LEFEVRE , « Les graphes par l'exemple », Ellipse, 1987.
- [6] [http ://mat.gsia.cmu.edu](http://mat.gsia.cmu.edu) .
- [7] I. TOMESCU. « Problems in combinatorics and graph theory ». John Wiley et Sons, New York, 1985. tranlated from Romanian by R. A. Melter p. 335 (511.6 TOM : CDPS). 1985.
- [8] J. MICHEL HÉLARY, « Algorithmique des graphes », 2004.
- [9] J. MICHEL HÉLARY, PÉDRONO. « Recherche opérationnelle : exercices corrigés : graphes et programmation dynamique ». Hermann, Paris, 1983. p.462 ISBN 2-7056-5954-4, (003.076 HEL : CDPS). 1983.
- [10] « Les graphes Stochastique-dynamyque », la cote 003/25, Université de Béjaia.
- [11] M. N.S SWAMY K. THULASIRAMAN. « Graphs : theory and algorithms ». John Wiley, New York, 1992. p . XV-460 (511.5 THU : CDPS). 1992.
- [12] M. GONDRAN, M. MINOUX, « Graphes et algorithmes », 4e édition, Lavoisier, 2009.
- [13] M. GONDRON, M. MINOUX, « Graphes et algorithmes », Eyrolles, Paris, 1984.

- [14] M. SAKAROVITCH. « Optimisation combinatoire, Tome1 : Graphes et programmation linéaire. Coll. Enseignement des sciences ». Hermann, Paris, 1984. XIII- 249 p. ISSN 0768-0341 ; 31. ISBN 2-7056-5974-9 Br. (519.71 SAK : CDPS). 1984.
- [15] O. COGIS, C. ROBERT , « Théorie des graphes » , Vuibert, 2003.
- [16] Ph. COMPOINT. « Les Graphes en recherche opérationnelle : ensembles remarquables et algorithmes de tâtonnement ». Dunod, Paris, 1972. VIII-168 p. (511.5 COM : CDPS). 1972.
- [17] Ph. Lacomme, C. Prins, M. Sevaux, « Algorithmes de graphes », Eyrolles, 2003.
- [18] Robert 1918-1982 FAURE. Roseaux. « Exercices et problèmes résolus de recherche opérationnelle, 1 Graphes : leurs usages, leurs algorithmes ». Masson, Paris, 1983. XXIV-299 p. ISBN 2-903-60717-6 Br, (003.076 ROS (I-A) : CDPS). 1983.
- [19] S. Ammar, Ph. Leray, and L. Wehenkel. « Mélanges sous-quadratiques d'arbres de markov pour l'estimation de la densité de probabilité ». In Proceedings of 5^{mes} journées francophones de réseaux bayésiens JFRB 2010, Nantes, France, 2010.
- [20] S. Ammar, Ph. Leray, and L. Wehenkel. « Sub-quadratic markov tree mixture models for probability density estimation ». In 19th International Conference on Computational Statistics (COMPSTAT 2010), Paris, France, 2010.
- [21] S. Ammar, Ph. Leray, B. Defoumy, and L. Wehenkel. « High-dimensional probability density estimation with randomized ensembles of tree structured bayesian networks. In Proceedings of the fourth European Workshop on Probabilistic Graphical Models », Hirtshals, Denmark, 2008.
- [22] S. EVEN. « Graphes algorithmes. Ed. Computer Science Press » , 1979. p.249 (511.5 EVE : CDPS). 1979.
- [23] S. SKIENA, « Implementing Discrete Mathematics : Combinatorics and Graph Theory With Mathematica », Addison-Wesley, 1990.
- [24] Thèse de Wajdi Tekaya, « Problème d'arbre couvrant de poinds minimum probabiliste » , Université Paris dauphine.
- [25] Thèse de Aurélie Favier, « Décompositions fontionnelles et structurelles dans les modèles graphique probabiliste », Université Toulouse.