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

Université Abderahmane Mira de Béjaia
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de fin d'étude

En vue de l'obtention de diplôme de Master en Mathématiques Appliquées
Option : Modélisation Mathématique et Évaluation des Performances des Réseaux

Thème

La C_n-valuation des Graphes

Présenté par :

- M^{elle} BENLARBI Hayat
- M^{r} KELLOU Slimane

Soutenue publiquement le 04/07/2019 devant le jury composé de :

Président	:	M^{r} TALEM.DJ	Université de Béjaïa
Promoteur	:	M^{r} KABYL.K	Université de Béjaïa
Examinateur	:	M^{r} BAY.A	Université de Béjaïa
Examinateur	:	M^{r} SOUFIT.M	Université de Béjaïa

Promotion 2019

* Remerciement *

Nous remercions le bon Dieu qui nous a orienté au chemin du savoir et les portes de la science.

Nous tenons tout d'abord à remercier Mr.KABYL KAMAL, pour l'honneur qu'il nous a fait en acceptant de nous encadrer. Ces conseils précieux ont permis une bonne orientation dans la réalisation de ce modeste travail.

Nous tenons également à remercier les membres du jury : Mr TALEM.DJ, Mr BAY.A et Mr SOUFIT.M pour l'honneur qu'il nous ont fait en acceptant de juger ce travail, et d'avoir consacrer leurs temps pour sa lecture.

Nous tenons à exprimer notre profonde gratitude à l'ensemble du corps enseignant qui a contribué à notre formation.

Nous voudrions accorder une place d'honneur dans nos remerciements à nos famille, plus particulièrement, nos parent et nos sœurs et frères. leur soutien et encouragement n'a jamais failli au cours de ce long cursus universitaire.

∗ Dédicaces ∗

Je dédie ce modeste travail :

À mes parents. Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à tous mes frères et mes sœurs, mes nièces SARA et SIRIN, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude.

BENLARBI HAYAT

Je dédie ce modeste travail à :

À mes parents .Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

Aux personnes dont j'ai bien aimé la présence dans ce jour, à tous mes frères, ma sœur, et ma belle sœur, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

À toute ma famille, et mes amis.

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

KELLOU SLIMANE

Table des matières

Table des figures

Introduction Générale

La théorie des graphes constitue aujourd'hui un corps de connaissances très important, historiquement elle est née en 1736 avec la communication d'Euler $(1707-1783)$ dans laquelle il proposait une solution au célèbre problème des ponts de Königsberg $(Euler, 1736)$. " Les habitants de Königsberg se demandaient s'il était possible, en partant d'un quartier quelconque de la ville, de traverser tous les ponts sans passer deux fois par le même et de revenir à leur point de départ. " Euler démontra que ce problème n'a pas de solution. Le problème des ponts de Königsberg est identique à celui consistant à tracer une figure géométrique sans lever le crayon et sans repasser plusieurs fois sur un même trait.

Les graphes constituent ainsi des outils de modélisation importants et très utilisés en informatique. On peut ainsi ramener un problème concret et complexe à un modèle mathématique plus clairement posé qui consiste en l'étude de sommets et arêtes (ou arcs). Un graphe $G = (V, E)$ est essentiellement défini par une relation binaire $E \in V \times V$ sur un ensemble V le plus souvent fini.

Les domaines d'application sont assez vastes et variés. Des modélisations de problèmes par des graphes existent autant dans des domaines scientifiques et techniques comme la chimie, les mathématiques, la physique, l'informatique mais aussi dans l'industrie où ils servent souvent dans l'aide à la décision et la planification de projets. L'utilisation des graphes est très connue dans des domaines comme la géographie dans les cas par exemple de coloration de graphes, mais aussi en la chimie moléculaire.

Le but de ces études est de proposer des structures (en gardant celle de l'hypercube comme structure de base) ou des méthodes, pour trouver un plongement optimal de graphe dans un hypercube.

L'objectif de notre travail est de montrer que certaines graphes sont C_n-valué(plongeable dans un hypercube de dimension n). On a réparti notre mémoire en cinq chapitres :

Dans le premier chapitre on va présenté des différentes définitions et notations essentielles sur la théorie des graphes, il comporte les concepts de base nécessaires et dont nous nous servons dans ce document.

Le deuxième chapitre est consacré à la présentation de l'hypercube et ses différentes caractéristique.

Dans le Troisième chapitre, nous donnerons une définition et différentes caractéris-

tique du plongement, ainsi la notion de C_n-valuation. Et par la suit, nous donnons quelques graphes avec les conditions nécessaires et/ou suffisantes pour que ces derniers soient C_n-valués dans l'hypercube.

Au Quatrième chapitre, nous introduisons des nouvelles classes d'arbres binaires. Nous commençons par la classe d'arbre T_n, puis la classe d'arbre $A^k_{n,2^p}$,et la classe d'arbre AR_n.

Le dernier chapitre a la présentation d'un problème concret modélisé sous forme d'un graphe hypercube pour lequel nous avons donné l'arbre de poids minimum en utilisant l'algorithme Kruskal, en suite nous avons implémenté sur le MATLAB l'algorithme de Kruskal.

1

Quelques notions élémentaires

Introduction

Dans ce chapitre nous donnerons des définitions et des propriétés sur les graphes qui nous seront utiles tout au long de ce manuscrit.

1.1 Concepts de base

1.1.1 Graphes

Définition* 1.1.1** *[1] Un* ***graphe $G = (V, E)$ *est constitué d'un ensemble fini* $V = \{v_1, v_2, ..., v_n\}$ *appelé sommets avec* $|V| = n$ *et d'une famille* $E = \{e_1, e_2, ..., e_n\}$ *de paires distinctes de* V *appelées arêtes (arcs) avec* $|E| = m$ *. La Figure.1.1 montre un exemple de graphe.*

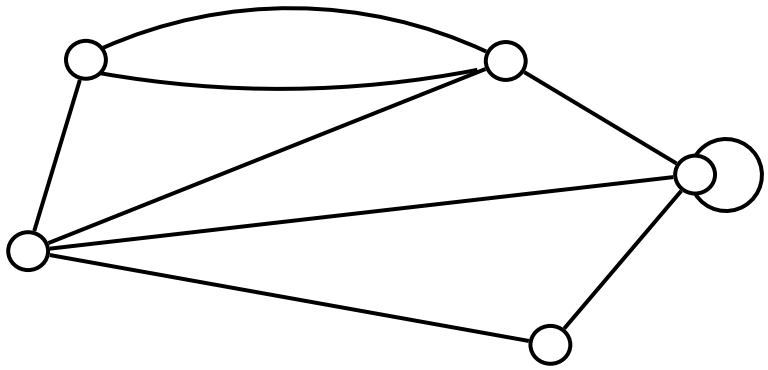

FIGURE 1.1 – Un graphe

Le nombre de sommets du graphe G est appelé **ordre** de G qu'on note O(G).

Le **degré** d'un sommet v : c'est le nombre d'arcs ayant v comme extrémité initiale ou terminale, on le note $d_G(v)$

1.1.1.1 Graphes non orientés

[2] Un graphe fini $G = (V, E)$ est défini par l'ensemble fini $V = \{v_1, v_2, ..., v_n\}$ dont les éléments sont appelés **sommets**, et par l'ensemble fini $E = \{e_1, e_2, ..., e_m\}$ dont les éléments sont appelés **arêtes**. Une arête e de l'ensemble E est définie par une paire non ordonnée de sommets, appelés les extrémités de e. Si l'arête e relie les sommets 1 et 2, on dira que ces sommets sont **adjacents**, ou **incidents** avec e, ou bien que l'arête e est incidente avec les sommets 1 et 2 .
La Figure .1.2 montre un graphe non orienté.

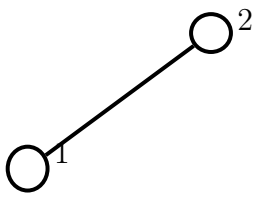

FIGURE 1.2 – Graphes non orientés

1.1.1.2 Graphes orientés

En donnant un sens aux arêtes d'un graphe, on obtient un digraphe (ou graphe orienté).Un digraphe fini $G = (V, E)$ est défini par l'ensemble fini $V = \{v_1, v_2, ..., v_n\}$ dont les éléments sont appelés **sommets**, et par l'ensemble fini $E = \{e_1, e_2, ..., e_m\}$ dont les éléments sont appelés **arcs**. Un arc e de l'ensemble E est défini par une paire ordonnée de sommets. Lorsque $e = (1, 2)$, on dit que l'arc e va de 1 à 2. On dit aussi que 1 est l'extrémité initiale et 2 l'extrémité finale de e.
La Figure .1.3 montre un graphe orienté.

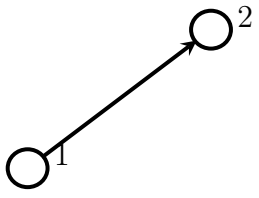

FIGURE 1.3 – Graphes orientés

1.1.2 Chaînes, cycles et chemins, circuits

▷ Une **chaîne** dans G, est une suite de sommets $(v_1, v_2, \ldots, v_k)$ tels que pour $i = 1...k-1$, $v_i v_{i+1}$ est une arête.

Une chaîne dont les sommets de départ et de fin sont les mêmes est appelée chaîne fermée ou **cycle**.

▷ Un **chemin** est une suite de sommets $(v_1, v_2, \ldots, v_k)$ tels que pour $i = 1...k-1$, (v_i, v_{i+1}) est une arc.
Un chemin fermé est dit **circuit**.

▷ On dira que la chaîne relie le premier sommet de la suite au dernier sommet. En plus, on dira que la chaîne a pour **longueur** le nombre d'arêtes qui la composent.

▷ On appelle **distance** entre deux sommets la langueur de la plus petite chaîne qui les relie.

▷ Une **chaîne est élémentaire** si chaque sommet y apparaît au plus une fois.

▷ Une **chaîne est simple si** chaque arête apparaît au plus une fois.

▷ Une **chaîne hamiltonienne** (resp. cycle hamiltonien) est une chaîne (resp. cycle) passant une fois et une seule par chacun des sommets de G.

▷ Un graphe G = (V, E) possède un **cycle hamiltonien** est dit graphe hamiltonien.

▷ une **Chaine Eulérienne** (resp. **cycle hamiltonien**), une chaine qui n'utilise qu'une et une seule fois toutes ses arêtes.

1.1.3 Graphes et sous-graphes connexes

***Définition* 1.1.2** *Un graphe non orienté est connexe si chaque sommet est accessible à partir de n'importe quel autre . Autrement dit, si pour tout couple de sommets distincts $(v_i, v_j) \in V^2$, il existe une chaîne entre sv_i et v_j.*

***Exemple* 1.1.1** *La Figure .1.4 montre un graphe et sous-graphe connexe :*

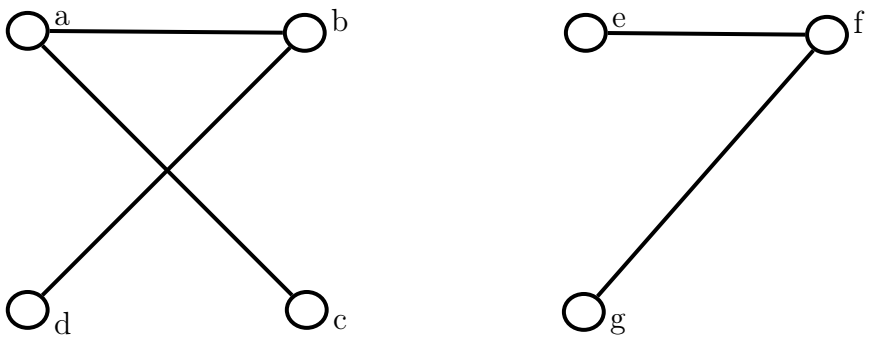

FIGURE 1.4 – Graphe et sous-graphe connexe

Le graphe n'est pas connexe, car il n'existe pas de chaîne entre les sommets a et e. En revanche, le sous-graphe induit par les sommets $\{a, b, c, d\}$ est connexe.

***Définition* 1.1.3** *Une composante connexe d'un graphe non-orienté G est un sous-graphe G' de G qui est connexe et maximal (c'est-à-dire qu'aucun autre sous-graphe connexe de G ne contient G').*

Un graphe est dit connexe si et seulement s'il admet une unique composante connexe.

Par exemple, le graphe de la Figure 1.4 est composé de 2 composantes connexes : la première est le sous-graphe induit par les sommets $\{a, b, c, d\}$, et la seconde est le sous-graphe induit par les sommets $\{e, f, g\}$.

1.2 Représentation Matricielle

1.2.1 Matrice d'adjacence

Soit $G = (V, E)$ un graphe, avec $|V| = n$. On appelle **matrice d'adjacence**, la $n \times n$ matrice booléenne suivante :

$$M_{ij} \begin{cases} 1 \text{ si il existe au moins une arête de } v_i \text{ vers } v_j \\ 0 \text{ sinon} \end{cases}$$

1.2.2 Matrice d'incidence

Soit $G = (V, E)$ un graphe non orienté, avec $|V| = n$ et $|E| = m$. On appelle **matrice d'incidence** sommets-arêtes, la matrice définie comme suit :

$$M_{ij} \begin{cases} 1 \text{ si } v_i \text{ est une extrémité de l'arc } e_i \\ 0 \text{ sinon} \end{cases}$$

Considérons un graphe orienté sans boucle $G = (V, E)$ comportant n sommets $\{v_1, ..., v_n\}$ et m arrêtes $\{e_1, ..., e_m\}$. On appelle matrice d'incidence (aux arcs) de G la matrice $M = M_{ij}$ de dimension $(n \times m)$ telle que :

$$M_{ij} = \begin{cases} 1 & \text{si } x_i \text{ est l'extrémité initiale de l'arc } e_j \\ -1 & \text{si } x_i \text{ est l'extrémité terminale de l'arc } e_j \\ 0 & \text{si } x_i \text{ n'est pas une extrémité de l'arc } e_j \end{cases}$$

1.3 Opérations sur les graphes

1.3.1 Somme cartésienne de deux graphes

Soient $G = (V(G), E(G))$ et $H = (V(H), E(H))$ deux graphes. On appelle somme Cartésienne de deux graphes G et H, notée $G \oplus H$. le graphe dont l'ensemble des sommets est $V(G) \times V(H)$ et où deux sommets (u, u') et (v, v') sont adjacents si et seulement si l'une des propriétés suivantes est vérifiée :

▷ $u = v$ et $u'v' \in E(H)$
ou
▷ $uv \in E(G)$ et $u' = v'$

La Figure.1.5 montre la somme Cartésienne de deux graphes.

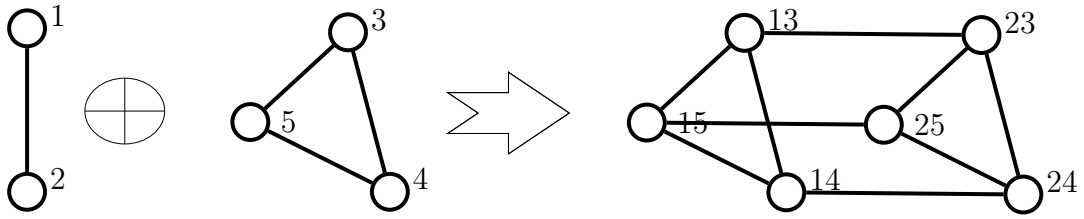

FIGURE 1.5 – Somme Cartésienne de deux graphes

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

1.3.2 Produit cartésien de deux graphes

Soient $G = (V(G), E(G))$ et $H = (V(H), E(H))$ deux graphes, on définit le graphe $G \bigotimes H = (S, T)$ appelé produit cartésien de G et H, tel que $S = V(G) \times V(H)$ et

$$T = \{e = (v, u)(v', u')/(v, u) \sim (v', u') \text{ ssi } v = v' \text{ et } u \sim u'\}.$$

La Figure.1.6 montre le produit Cartésien de deux graphes

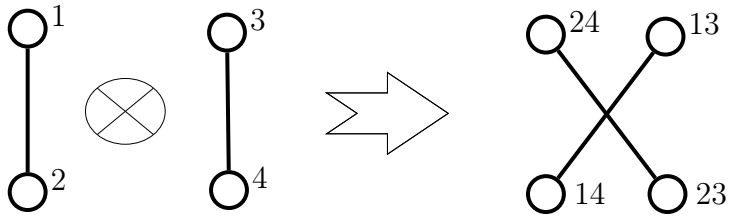

FIGURE 1.6 – Produit Cartésien de deux graphes

1.3.3 Homomorphisme de graphe

Soient $G = (V(G), E(G))$ et $H = (V(H), E(H))$ deux graphes. Un homomorphisme de G dans H est une application : $F : V \to V(H) telque$:

$$xy \in E \Rightarrow f(x)f(y) \in E(H)$$

La Figure.1.7 montre un Homomorphisme de graphe G dans H .

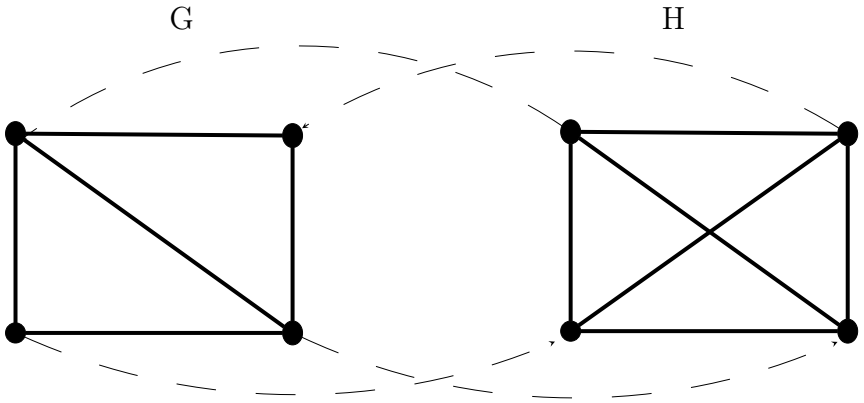

FIGURE 1.7 – Homomorphisme de graphe G dans H

1.3.4 Isomorphisme de graphe

Deux graphes non orientés $G = (V(G), E(G))$ et $H = (V(H), E(H))$ sont isomorphes s'il existe une bijection $f : V \to V(H)$ qui est telle que

$$\{x, y\} \in E \Leftrightarrow \{f(x), f(y)\} \in E(H)$$

1.3.5 Subdivisions de graphes

- ▷ Une subdivision élémentaire d'un graphe G est un graphe obtenu par insertion d'un sommet de degré deux sur une arête de G.
- ▷ Une subdivision de G est un graphe obtenu à partir de G par une succession de subdivisions élémentaires.
- ▷ Une p-subdivision de G est un graphe obtenu par insertion de p sommets sur chaque arête.

La Figure.1.8 donne une subdivisions de graphe

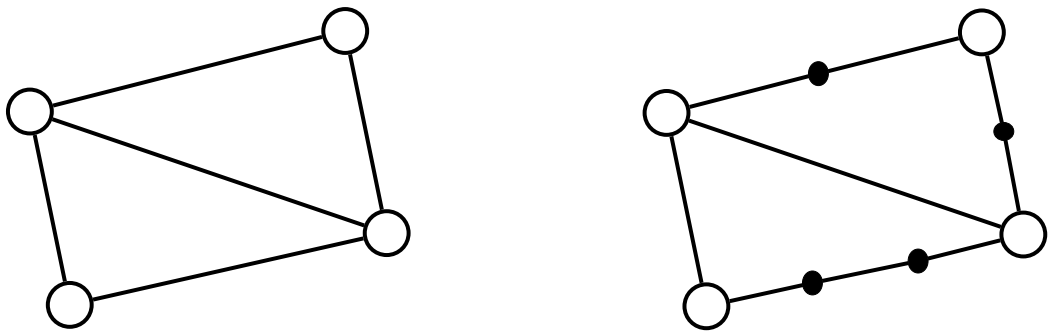

FIGURE 1.8 – Subdivisions de graphes

1.3.6 Distance

La distance entre deux sommets u et v, notée $d(u, v)$, est la longueur du plus court chemin reliant ces deux sommets, s'il existe. S'il n'existe pas de chemin entre u et v, c'est à dire s'ils ne sont pas dans la même composante connexe, on dit que la distance $d(u, v)$ est infinie. La distance entre deux sommets voisins est 1.

Le diamètre d'un graphe est le maximum des distances entre deux de ses sommet.

1.3.7 Intervalle

L'intervalle $I(u, v)$ est l'ensemble des sommets de G appartenant aux chaines géodésiques entre u et v.
Proposition de base : Soient u et v deux sommets de G, alors :

- $u, v \in I(u, v)$
- $I(u, v) = I(v, u)$
- si $w \in I(u, v)$, alors $I(u, w) \subset I(u, v)$
- si $w \in I(u, v)$ et $z \in I(u, w)$, alors $w \in I(z, v)$

1.4 Quelques graphes particuliers

1.4.1 Graphe simple

Un graphe $G = (V, E)$est simple si au plus une arête relie deux sommets et s'il n'y a pas de boucle sur un sommet. On peut imaginer des graphes avec une arête qui relie un sommet à lui-même (une boucle), ou plusieurs arêtes reliant les deux mêmes sommets. On appellera ces graphes des multigraphes.

La Figure.1.9 représente un graphe simple et multigraphes.

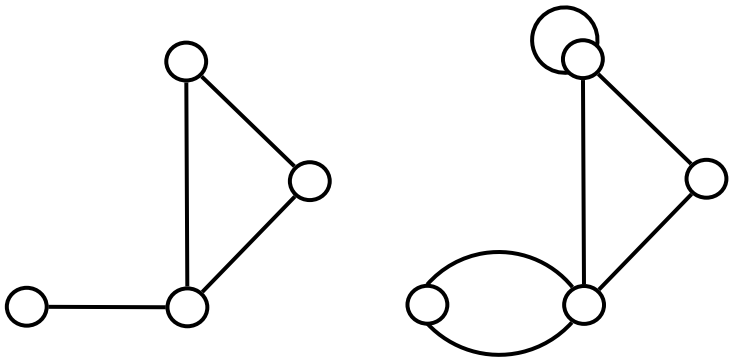

FIGURE 1.9 – Graphe simple et multigraphes

1.4.2 Graphe complet

Un graphe $G = (V, E)$ est dit complet si tous les sommets sont deux à deux adjacent, le graphe complet à n sommet est noté K_n.
La Figure.1.10 définit un exemple de graphe complet.

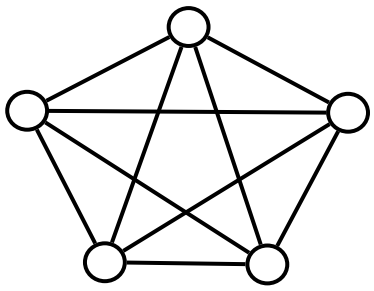

FIGURE 1.10 – Graphe complet

1.4.3 Sous graphe et graphe partiel

***Définition* 1.4.1** *(Graphe partiel) Soit $E_p \subset E$ un sous ensemble des arcs E de G, $G_P = (V, E_P)$ est un graphe partiel de G*

***Définition* 1.4.2** *(Sous graphe) Soit $G = (V, E)$ un graphe. Un sous graphe (V_P, E_P) est tel que :*

$$\begin{cases} V_p \subset V \\ E_p \subset E \end{cases}$$

On note usuellement G_{V_p} le sous graphe engendré par V_p, c'est à dire celui pour lequel on a :

$$E_p = E \cap (V_p \times V_p)$$

1.4.4 Graphe biparti

Un graphe G est biparti si ses sommets peuvent être divisés en deux ensembles X et Y, tel que toutes les arêtes du G relient un sommet dans X à un sommet dans Y .
Un graphe biparti est donné dans la Figure.1.11 .

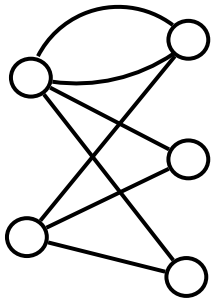

FIGURE 1.11 – Graphe biparti

Graphe biparti complet

Un graphe biparti complet est un graphe biparti tel que chaque sommet de $\mathtt{X}$ est relié avec tous les sommets de $\mathtt{Y}$ qu'on note $\mathtt{K}_{p,q}$ ou $|\mathtt{X}| = p$ et $|\mathtt{Y}| = q$.

Graphe biparti équilibré

Soit **G** un graphe biparti. Si $|\mathtt{X}| = \mathtt{Y}| = p$ alors, on dit que **G** biparti équilibré.

1.4.5 Graphe planaire

Un graphe est dit planaire si on peut le dessiner sur un plan de telle sorte que les arête ne se coupent pas, en dehors de leurs extrémités.

1.4.6 Graphe réguliers

En théorie des graphes, un graphe régulier est un graphe où tous les sommets ont le même nombre de voisins, c'est-à-dire le même degré ou valence. Un graphe régulier dont les sommets sont de degré k est appelé un graphe $k-$régulier ou graphe régulier de degré k.
La Figure.1.12 représente un graphe 2-régulier

Un graphe fortement régulier est un graphe régulier où chaque paire de sommets adjacents a le même nombre l de voisins en commun et où chaque paire de sommets non-adjacents a le même nombre n de voisins en commun.

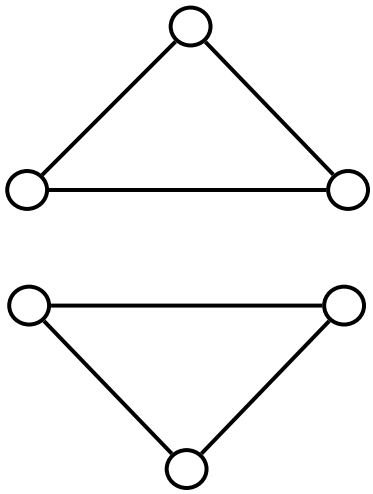

FIGURE 1.12 – Graphe 2-régulier

1.4.7 Graphe médians

Un graphe $G = (V, E)$ est dit médian si pour tout triplet de sommet u, v et w de G, on a $|I(u, v) \cap I(v, w) \cap I(u, w)| = 1$. Autrement dit il existe un unique sommet x simultanément à une plus courte $(u, v)-$chaine, à une plus courte $(v, w)-$chaine et à une plus courte (w, u)-chaine [5].

1.4.8 Graphe pondéré

Un graphe étiqueté est un graphe où chaque arête est affectée d'une chaîne.

Un graphe pondéré $G = (V, E)$ est un graphe étiqueté où chaque arête est affectée d'un nombre réel positif, appelé poids de cette arête. Le poids d'une chaîne est la somme des poids des arêtes qui la composent.

1.4.9 Arbre

***Définition* 1.4.3** *Un arbre est un graphe connexe sans cycle .Un tel arbre est noté T*

Les propriétés suivantes (qui s'appliquent à un graphe comptant n sommets) sont équivalentes à la définition précédente :

- ▷ Un arbre est un graphe connexe qui compte exactement $n - 1$ arcs
- ▷ Un arbre est un graphe sans cycle qui compte exactement $n - 1$ arcs. On parle de graphe acyclique minimal
- ▷ Un arbre est un graphe sans cycle tel que si l'on rajoute un arc quelconque, on crée un cycle
- ▷ Un arbre est un graphe connexe tel que la suppression d'un arc quelconque engendre la séparation en 2 composantes connexes. On parle de graphe connexe minimal

▷ Dans un arbre, tout couple de sommets est relié par une et une seule chaîne.

La Figure.1.13 montre un arbre.

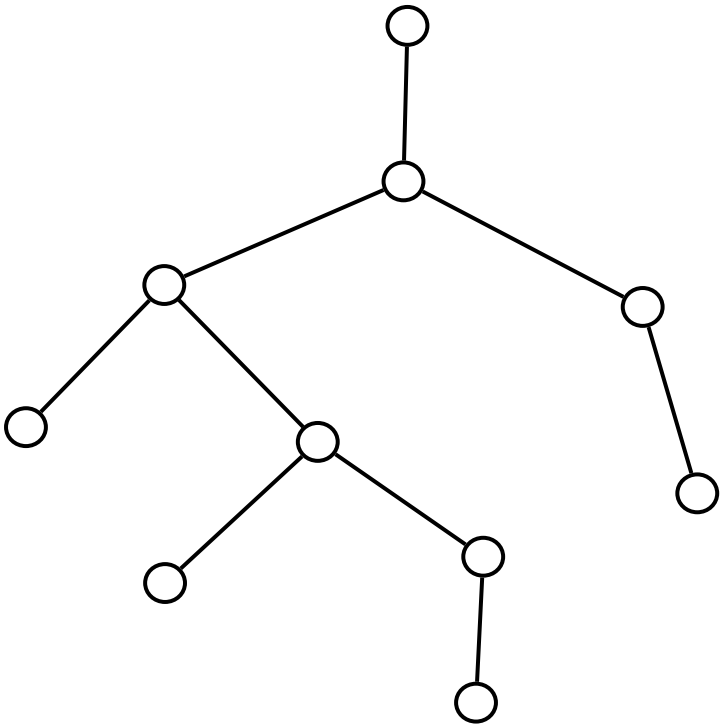

FIGURE 1.13 – Arbre

Conclusion

Dans ce premier chapitre, nous avons donné quelques notations de base de la théorie des graphes et dans ce qui suit on va citer quelques propriétés et caractérisations de l'hypercube.

2
L'hypercube

Introduction

Dans les années 1980, des ordinateurs furent réalisés avec plusieurs processeurs connectés selon un hypercube, chaque processeur traite une partie des données et ainsi les données sont traitées par plusieurs processeurs à la fois, ce qui constitue un calcul parallèle. L'hypercube est couramment introduit pour illustrer des algorithmes parallèles, et de nombreuses variantes ont été proposées, soit pour des cas pratiques liés à la construction de machines parallèles, soit comme objets théoriques.

Dans ce chapitre nous donnons quelques propriétés sur les hypercubes qui seront utilisées dans la suite. Nous décrivons aussi les différentes caractérisations d'un hypercube.

2.1 Définition de l'hypercube

L'hypercube de dimension n, note Q_n est le graphe dont l'ensemble des sommets qui forment des n-uplets binaires, deux sommets sont adjacents si et seulement s'ils diffèrent exactement en une seule composante (coordonnée). Un hypercube n-dimensionnel est aussi appelé un n-cube.
Q_n C'est les graphes de 2^n sommets qui peuvent être considères comme étant tous les vecteurs booléens $(0,1)^n$.
Une direction i dans l'hypercube de dimension n ($i \leq n$) est l'ensemble des arêtes de Q_n, dont les extrémités ont des vecteurs associés qui différent à la i-ème composante.

Pour représenter un hypercube de dimension n, on procède comme suit :

▷ **dimension 1 :** un point est un hypercube de dimension zéro. Si l'on déplace ce point d'une longueur unité, il balaiera un segment de droite, qui est un hypercube unité de dimension 1.
Hypercubes Q_0 et Q_1 sont montrés dans la Figure.2.1.

FIGURE 2.1 – Hypercubes Q_0 et Q_1

▷ **dimension 2 :** si l'on déplace ce segment d'une longueur unité dans une direction perpendiculaire à partir de lui-même ; il balaie un carré bi-dimensionnel.
Hypercube Q_2 est donné dans la Figure.2.2.

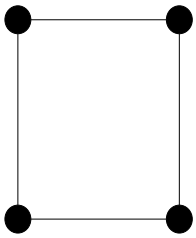

FIGURE 2.2 – Hypercube Q_2

▷ **dimension 3 :** si l'on déplace le carré d'une longueur unité dans la direction perpendiculaire à l'emplacement de celui-ci, il engendrera un cube tri-dimensionnel.
Hypercube Q_3 est donné dans la Figure.2.3.

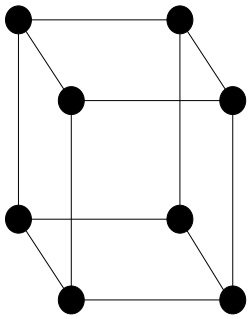

FIGURE 2.3 – Hypercube Q_3

▷ **dimension 4 :** si on déplace le cube d'une longueur unité dans la quatrième dimension, il engendrera un hypercube unité quadri-dimensionnel.
Hypercube Q_3 est donné dans la Figure.2.4.

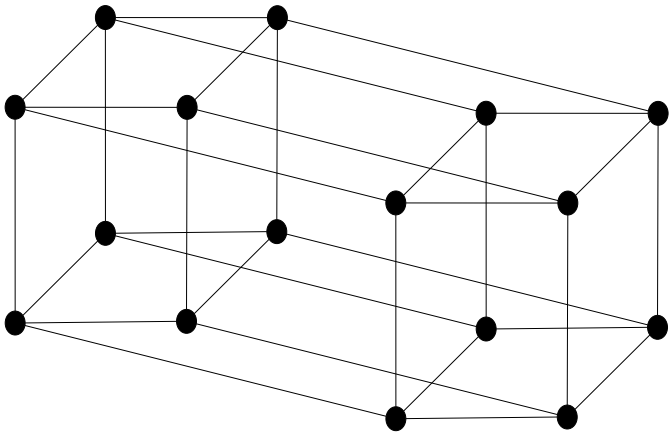

FIGURE 2.4 – Hypercube Q_4

▷ **dimension n :** Notons $Q_0 = K_1; Q_1 = K_2$ et que d'une manière générale ,Q_n peut être défini récursivement en utilisant la somme cartésien par $Q_{n+1} = Q_n \oplus K_2$ d'où pour $Q_n (n \geq 1)$ est isomorphe à

$$K_2 \oplus K_2 \oplus K_2 .....K_2 n \text{ fois}$$

et donc $Q_{n+x} = Q_n \oplus Q_x$

2.2 Propriété élémentaire de l'hypercube

L'hypercube de dimension n est un graphe n-régulier, avec 2^n sommets et donc $n.2^{n-1}$arêtes. Il est sommet-transitif.

De plus, entre chaque paire de sommets v et u il existe $d(v, u)!$ géodésiques.

Lors de son étude du problème de plongement de graphes dans l'hypercube, Kobeissi [13] a suggéré une étude détaillée sur l'hypercube. On peut montrer, par induction, qu'entre deux sommets quelconques de l'hypercube, il existe n chaînes deux à deux sommets disjointes, ce qui prouve que sa sommet-connexité (et donc son arête connexité) est égale à n. Rappelons ici que, la sommet-connexité (resp. arête-connexité) d'un graphe connexe G est le nombre de sommets (resp.arêtes) qu'il faut supprimer pour que G ne soit plus connexe.

***Proposition* 2.2.1** *[6] Un graphe $G = (V, E)$ est dit intervalle-régulier si et seulement si pour tout couple de sommets (u, v) de V, le sous graphe induit par l'ensemble des arête entre niveau de $I_G(u, v)$est un hypercube de dimension $d_G(u, v)$.*

***Proposition* 2.2.2** *[13] Il existe $n-1$ chaînes de longueur inférieure ou égale à n et une chaîne de longueur inférieur ou égal à $n+1$, deux à deux sommets disjointes, entre toute paire de sommets distincts de l'hypercube de dimension n*

***Proposition* 2.2.3** *[13] L'hypercube est hamiltonien, de plus par tout arête passe un cycle hamiltonien.*

***Proposition* 2.2.4** *[14] Pour deux sommets u et v qui sont à distance k dans Q_n, il existe $k!$ plus courtes $(u\,;\,v)$ chaines.*

2.3 Caractérisation de l'hypercube

Il existe plus d'une trentaine de caractérisation de l'hypercube à ce jour. Foldes [12] à donné une des première caractérisation de l'hypercube.

***Théorème* 2.3.1** *[10] Un graphe connexe $G = (V; E)$ est un hypercube si et seulement s'il vérifie les conditions suivantes :*

- ▷ G est biparti.
- ▷ Pour tout couple de sommets v et u de G, le nombre de plus courtes chaines est $d(v; u)!$.

***Théorème* 2.3.2** *[11] Soit $G = (V; E)$ un $(0; 2)$- graphe, alors :*

- ▷ G est régulier de degré n.
- ▷ $|E(G)| \leq 2^n$.

***Théorème* 2.3.3** *[10] soit G un $(0; 2)$ graphe tel qu'il existe une décomposition en couches où tout cycle de longueur 4 rencontre 3 couches, alors G est un hypercube.*

***Théorème* 2.3.4** *[9] Un graphe G est un hypercube de dimension n si et seulement si l'ensemble des sous graphes convexes de G est $Q_1, Q_2, ..., Q_n$.*

2.4 Projection et anti-projection

Une projection (resp. anti-projection) d'un sommet u d'un graphe $G = (V, E)$ sur un ensemble de sommets S de G est un sommet v de S à distance minimum (resp. maximum) de u pour tout ensemble de sommets S de G et pour tout sommet u, on désigne par $P(u, S)$ (resp. $AP(u, S)$) l'ensemble des projection (resp. anti-projection) de u sur S.

Considérons les propriétés suivantes :

▷ pour tous triple de sommets (u, v, w), on a $|P(u, I_G(v, w))| = 1$ .....(P_1)

▷ pour tous triple de sommets (u, v, w), on a $|AP(u, I_G(v, w))| = 1$ .....(P_2)

Un graphe vérifiant l'une des propriétés (P_1) et (P_2) est un graphe biparti. molard [4] a donné une caractérisation de l'hypercube en termes d'anti-projections sur les intervalles.

***Proposition* 2.4.1** *[4] un graphe G = (V, E) est un hypercube si seulement s'il vérifie la propriété ((P_2))*

2.5 Graphes de Hamming

Soient a_1, a_2, ..., a_n des entiers positifs. Le graphe de Hamming de dimension n sur un alphabet à n éléments est un graphes défini sur n sommets qui sont tous les $n - uplets$ $(a_1, a_2, ..., a_n)$. Deux sommets sont adjacents si et seulement si leurs $n - uplets$ respectifs diffèrent sur une seule coordonnée.

2.6 Décomposition en couches des graphes de Hamming

Une décomposition en couche de G, à partir d'un sommet v donné, est formé de l'ensemble : $\{v, N_1(v), N_2(v), ...N_k(v)\}$, avec k le nombre de couches. On a les propriété suivantes :

***Propriété* 2.6.1** *i) Pour tout sommet u de $N_2(v)$, il existe exactement deux arêtes reliant u à des sommets de $N_1(v)$*

ii) $|N_i(v)| = C_n^i$

***Propriété* 2.6.2** *Pour tout sommet u de $N_i(v)$ et tout sommet z de $N_{i-1}(v)(i \geq 2)$, le nombre d'arêtes entre u et $N_{i-1}(v)$ est égale au nombre d'arêtes entre z et $N_{i-2}(v)$ plus 1. plus précisément, le nombre d'arêtes entre u et $N_{i-1}(v)$ est égale à $i - 1$.*
la Figure.2.5 représente un exemple de décomposition en couches.

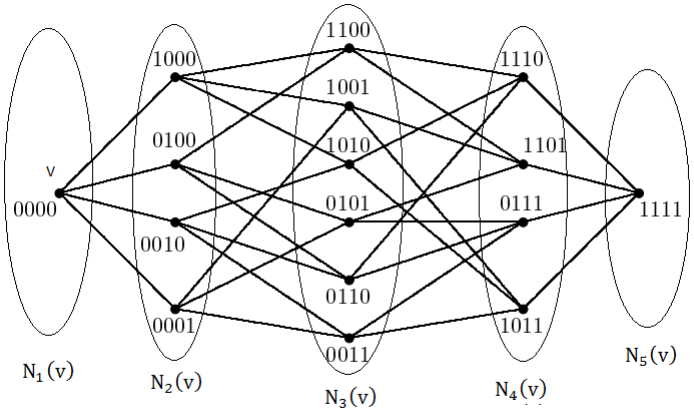

FIGURE 2.5 – Exemple de décomposition en couches

2.7 Décomposition de Q_n en des copies disjointes de Q_i

Les sommets de Q_n sont répartis en deux parties disjointes $V_1(Q_n)$ et $V_2(Q_n)$, tel que $|V_1(Q_n)| = |V_2(Q_n)| = 2^{n-1}$ et que pour tout $n \geq 3$. Q_n peut être décomposé en 2^{n-i} copies disjointes de Q_i, $0 \leq i \leq n$. Comme Q_2 est décomposé en une copie de $K_{1,2}$ et une copie de Q_0, donc Q_n peut être décompose en 2^{n-2} copies disjointes de $K_{1,2}$ et 2^{n-2} copies disjointes de Q_0.

Supposons que les sommets de $V_1(Q_n)$, de$V_1(T)$, de $V_1(Q_0^i)$, de $V_1(Q_0^i)$ et de $V_1(K_{1,2}^i)$ sont représentés par des cercles et ceux de $V_2(Q_n)$, de $V_2(T)$, de $V_2(Q_1^i)$, de $V_2(Q_0^i)$ et de $V_2(K_{1,2}^i)$ sont représentés par des carrés.
Nous considérons les deux compositions suivantes :

- ▷ Q_n est décomposée en 2^{n-1} copies disjointes de Q_1, notées $Q_n^i, i \in 1, 2, ..., 2^{n-2}$.
- ▷ Q_n est décomposé en 2^{n-2} copies disjointes de $K_{1,2}$, notées $K_{1,2}^i$ et 2^{n-2} copies disjointes de Q_0,notées Q_0^i, tels pour tout $i \in 1, 2, ..., 2^{n-3}$.

La Figure.2.6 montre la représentation des sommets de $|V_1(Q_3)|$ par des cercles et ceux de $|V_2(Q_3)|$ par des carrés.

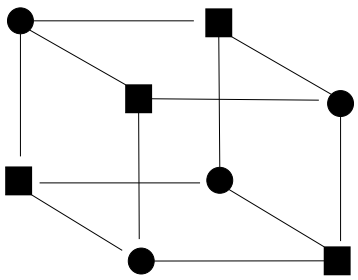

FIGURE 2.6 – Représentation des sommets de Q_3

La Figure.2.7 montre Décomposition de Q_3 en des copies disjointes de Q_i.
La Figure (a) montre décompositions de Q_3 en 8 sommets isolés, La Figure (b) montre la

décomposition de Q_3 en 4 copies disjointes de Q_1 et la Figure (c) montre la décompositions de Q_3 en 2 copies disjointes de $K_{1,2}$ et 2 copies disjointes de Q_3.

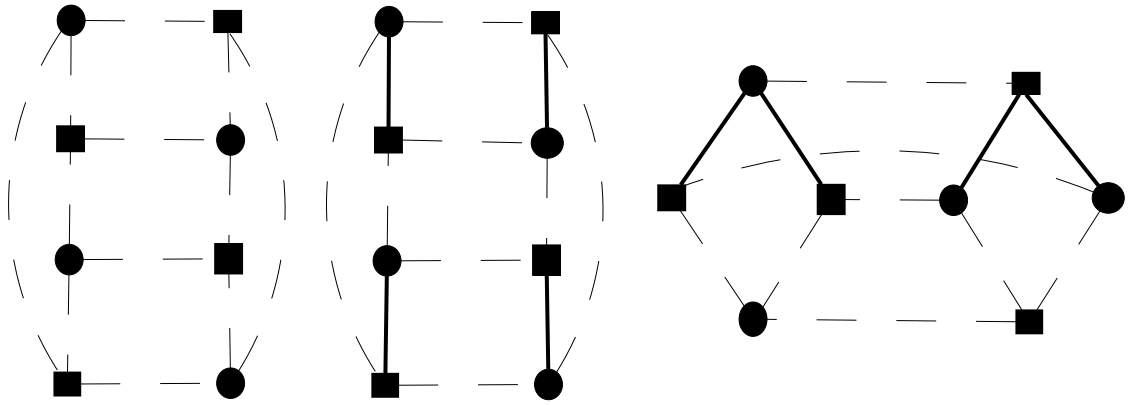

FIGURE 2.7 – Décomposition de Q_3 en des copies disjointes de Q_i

Conclusion

Dans ce chapitre nous avons donné la définition ainsi quelque propriétés et caractérisations de l'hypercube. Nous sommes à présent armés pour aborder les différents problèmes spécifiques de ce mémoire. La notion de graphe est bien entendu présente dans chacun des chapitres suivants.

3

Plongement de graphe dans l'hypercube

Introduction

Le problème de plongement de graphes dans l'hypercube a été traité par plusieurs auteurs. On peut citer : A. Berrachedi [21], S. Bezrukov [22], I. Havel [23], F. Harary [24], M. Kobeissi [25], M.Laborde [26] et plusieurs auteurs ont donné des familles de graphes qui sont plongeables dans Q_n.

La recherche d'un plongement optimal d'un graphe G dans un hypercube Q_n consiste a trouver la plus petite dimension n de l'hypercube dans lequel G y est plongeable, l'entier n sera appelé dimension de G est notée $Cd(G)$.Un graphe $G = (V, E)$ est dit cubique s'il est plongeable dans Q_n pour un certain n.

Dans ce chapitre, on donne quelques résultats connus concernant les plongements dans l'hypercube.

3.1 Définition

***Définition* 3.1.1** *Un plongement d'un graphe G dans un graphe H est défini par la donnée d'une application injective φ de l'ensemble des sommets de G dans l'ensemble des sommets de H, et d'une application P_φ de l'ensemble des arêtes de G dans l'ensemble des chaînes de H, qui associe à chaque arête vu de G, une chaîne reliant les sommets $\varphi(v)$ et $\varphi(u)$ dans H.*

3.2 Plongement dans Q_n

L'hypercube de dimension n, noté Q_n est le graphe où les sommets représentent les n-uplets de $\{0,1\}^n$ et où deux sommets sont adjacents si et seulement si les vecteurs associés à ces sommets différent exactement en une seule composante. Le plongement d'un graphe G dans l'hypercube revient à voir si G est isomorphe à un sous-graphe de Q_n. Chercher un plongement optimal d'un graphe G dans un graphe d'une famille donnée, revient à plonger G dans le graphe H de cette famille ayant le plus petit nombre de sommets possible, supérieur ou égale à celui de G. On dit alors que H est optimal pour G. Dans le cas où cette famille de graphes est réduite à un seul graphe qui est le graphe de l'hypercube, alors la recherche d'un plongement optimal d'un graphe G dans un hypercube Q_n consiste à trouver la plus petite dimension n de l'hypercube pour le quel G y est plongeable.

3.2.1 Paramètres de plongement

Beaucoup de paramètres ont été définis pour mesurer l'efficacité des plongements. Nous donnerons la définition de ceux d'entre eux qui sont le plus souvent étudiés, à savoir la dilatation, l'expansion et la congestion.

3.2.1.1 Dilatation

La dilatation d'un plongement ϕ d'un graphe G dans un graphe H,notée $dil(\phi)$, est la longueur maximale des chaîne $P_\phi(v,u)$ de H, associées aux arêtes (v,u) de G. Dans le cas où l'on considère des chaînes de plus courte longueur, la longueur de $P_\phi(v,u)$ est alors égale à la distance $d_H(\phi(v),\phi(u))$, et la dilatation s'exprime uniquement en fonction de ϕ, par :

$$dil(\phi) = Max_{v,u\in E(G)} d_H(\phi(v);\phi(u))$$

Dire que G est plongeable avec dilatation 1 est équivalent à dire que G est un sous-graphe de H. Dans ce cas, l'image de l'arête (v,u) de G est l'arête $(\phi(v);\phi(u))$ de H. Si de plus $|V(G)| = |V(H)|$, alors G est un graphe partiel de H.

3.2.1.2 Expansion

L'expansion d'un plongement d'un graphe G dans un graphe H est le rapport du nombre de sommets de H, sur le nombre de sommets de G.Ce paramètre est une mesure du degré d'utilisation des processeurs dans le cas d'un algorithme modélisé par G, et implémenté sur le réseau de processeurs modélisé par H.
Une expansion égale à 1 peut correspondre à une utilisation optimale, ou du moins très efficace des processeurs.

3.2.1.3 Congestion

La congestion d'un plongement ϕ d'un graphe G dans un graphe H, notée $cong(\phi)$, est le maximum pris sur toutes les arêtes e de H du nombre de chaines $P_\phi(v,u)$ de H images d'arêtes de G qui contiennent e.

3.2.2 Plongement optimal

Chercher un plongement d'expédition minimum d'un graphe G dans un graphe d'une famille donnée revient donc à plonger G dans le graphe H de cette famille ayant le plus petit nombre de sommets possible supérieur ou égal à celui de G. On dit que H est optimal pour G. La seule borne inférieure connue pour la dilatation valable pour tous les graphe, si l'expansion du plongement vaut 1 (l'application ϕ est alors bijective),est donnée par la proposition suivante :

***Proposition* 3.2.1** *[6] Si l'expansion d'un plongement ϕ d'un graphe G dans un graphe H vaut 1, alors la dilatation de ϕ est au moins égale au rapport du diamètre de H sur le diamètre de G.*

3.3 Démentions cubiques

Un graphe G est dit cubique s'il admet un plongement de dilatation 1 dans Q_n pour un certain n. Le plus petit entier n pour lequel G est plongeable dans Q_n est appelée dimension cubique, notée $Cd(G)$.

Firsov [19] a remarquée que les arbres sont des graphes cubiques.
Il a aussi montrée que tout graphe cubique est nécessairement biparti, mais la réciproque n'est pas vrai en général.
Un exemple de graphe biparti $K_{2,3}$ est donné dans la Figure.3.1 .

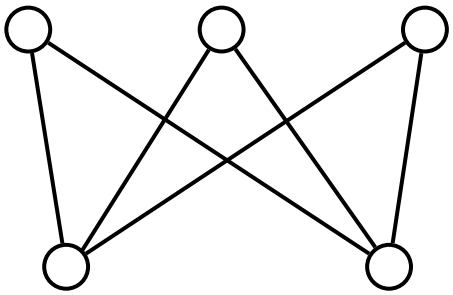

FIGURE 3.1 – Graphe $K_{2,3}$

$K_{2,3}$ n'est pas un graphe cubique, il n'admet pas de plongement dans Q_n quelque soit la valeur de n. En effet, supposons qu'il existe un tel plongement, comme u et v sont à distance 2 dans $K_{2,3}$, alors leurs images respectives $p = \varphi(u)$ et $q = \varphi(v)$ seront aussi à distance 2 dans Q_n Or, deux sommets à distance 2 dans Q_n appartiennent à exactement 2 chaînes sommet-disjointes de longueurs 2 dans Q_n, ce qui n'est pas possible car les 3 chaînes sommet-disjointes de longueurs 2 dans $K_{2,3}$ doivent se plonger dans 3 chaînes sommet-disjointes dans Q_n.

Condition nécessaire pour qu'un graphe G soit plongeable dans Q_n Si un graphe $G = (V; E)$ est plongeable dans le graphe Q_n, alors nécessairement on a :

- ▷ $|V(G)| \leq 2^n$.

- ▷ G est biparti.
- ▷ Le degré maximum de G, est $\Delta(G) \leq n$.

Si de plus $|V(G)| = 2$ alors G doit être équilibré.

A noter qu'un graphe G est cubique si et seulement si toutes ses composantes connexes. Havel et Moràvek ont donnée les conditions nécessaires et suffisantes suivantes pour dire si un graphe G donnée est cubique, Ce dernier est connu sous le nom de la C_n**-valuation** [18] [20].

3.4 La notion de C_n-valuation

Un graphe G peut être plongé dans Q_n si et seulement si on peut marquer les arêtes de G par des entiers appartenant à l'ensemble $\{1, ..., n\}$, de telle sorte que :

- ▷ Toutes les arêtes de G incidentes à un même sommet x admettent des marques différents,
- ▷ Pour toute chaîne P de G, il existe un entier $k \in \{1, ..., n\}$ qui apparaît un nombre impair de fois dans le marquage des arêtes de P,
- ▷ Pour tout cycle C de G, aucun entier $k \in \{1, ..., n\}$ n'apparaît un nombre impair de fois dans le marquage des arêtes de C.

***Théorème* 3.4.1 (7)** *Un Graphe G est plongeable dans Q_n si et seulement s'il existe un C_n-valuation de G.*

3.4.1 Complexité

Chercher une C_n-valuation d'un graphe G n'est pas un problème facile, c'est un probleme NP-complet. Garey et Graham ont montré que la complixité de trouvé une C_n-valuation pour un graphe estNP-Complet [15]. Arfati, Papadimiriou et Papageogiou [16] ont étendu ce résultat dans le cas où H est le graphe l'hypercube. Enfin, Wagner et Coreil [17] ont montré que ce problème reste NP-Complet même dans le cas particulier où G est un arbre.

3.5 C_n-Valuation de Quelques Classes de Graphes

Un Graphe G est plongeable dans Q_n si et seulement s'il existe une $C_n$valuation de G.

3.5.1 Arbre binaires

***Définition* 3.5.1** *Un arbre T est dit binaire si son degré maximum $\triangle(T) \leq 3$.*

La Figure.3.2 montre un arbre binaire.

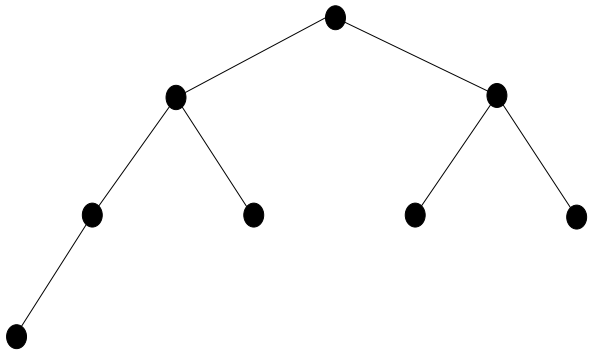

FIGURE 3.2 – Arbre binaire

***Proposition* 3.5.1** *[23] Soit T un arbre binaire d'ordre 2^n avec $n \geq 3$. Si T est équilibré et possède deux sommets de degré 3 alors T admet une C_n-Valuation (T est plongeable dans Q_n.*

***Définition* 3.5.2** *[18] L'arbre binaire complet D_n est le graphe défini inductivement comme suit :*

- ▷ Pour $n = 1$. $D_1 = K_{1,2}$ est un graphe biparti complet.
- ▷ Pour $n \geq 2$, D_n est obtenu à partir de deux copies disjointes T_1, T_2 de D_{n-1} et d'un nouveau sommet v, tel que v est relié par une arête à un sommet de degré 2 de T_1 et T_2.
  La Figure.3.3 montre un arbre binaire complet D_2 .

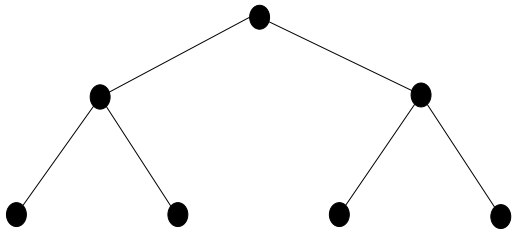

FIGURE 3.3 – Arbre binaire complet D_2

- ▷ Il est clair que D_n possède 2^n sommets pendants, $2^n - 2$ sommets de degré 3 et un seul sommet de degré 2 appelé la racine de D_n, donc D_n possède $2^{n+1} - 1$ sommets.
  La Figure.3.4 montre un arbre binaire complet D_3

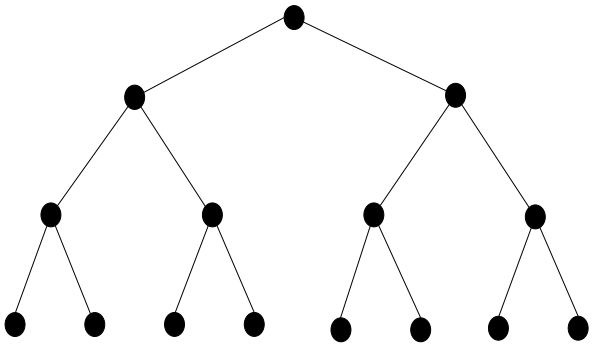

FIGURE 3.4 – Arbre binaire complet D_3

***Théorème* 3.5.1** *Soit $n \geq 2$. D_n admet une $C_{n+2}-$Valuation et $cd(D_1) = 2, cd(D_n) = n + 2$. [29]*

A partir de l'arbre binaire complet D_n on définira d'autres arbres plongeables dans l'hypercube.

1. Pour $n \geq 2$, B_n et un arbre binaire obtenu à partir de l'arbre binaire complet D_{n-1}, et d'un sommet u, tel que u soit relié à la racine de D_{n-1} par une arête.
   La Figure.3.5 montre un arbre binaire B_3

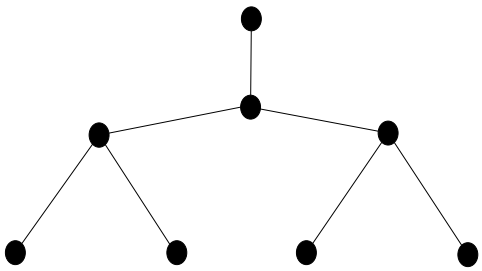

FIGURE 3.5 – Arbre binaire B_3

Il est clair que B_n possède $2^{n-1} + 1$ sommets pendants et $2^{n-1} - 1$ sommets de degré 3, donc B_n possède 2^n sommets .

***Théorème* 3.5.2** *Pour tout $n \geq 2$, B_n admet une $C_{n+1}-$Valuation, $Cd(B_n) = n + 1$.*

L'arbre $B_n^{(k)}$ peut être généralisé comme suit :

Soit $n \geq 2$ et $k \geq 1$, on peut définir l'arbre noté $B_n^{(k)}$ de la manière suivante :

▷ pour $k = 1$, $B_n^{(1)} = B_n$

▷ pour $k \geq 2$, $B_n^{(k)}$ est l'arbre obtenu par subdivision de chaque arête de l'arbre binaire complet D_{n-1} par $k-1$ sommets et l'arête supplémentaire pour obtenir B_n par k .
La Figure.3.6 montre un arbre binaire B_1^2.

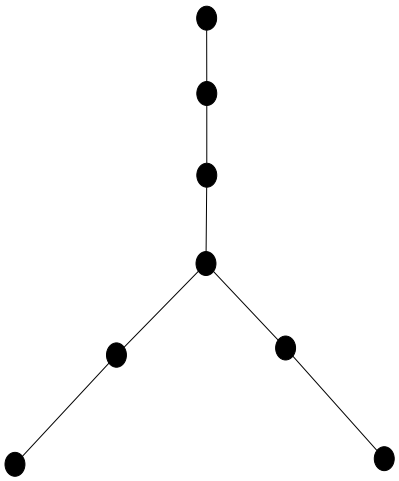

FIGURE 3.6 – Arbre binaire $B_2^{(2)}$

***Proposition* 3.5.2** *[30] Pour $n \geq 2$ et $s \geq 1$, $|V(B_n^{(2s)})| = 2^{n+1+s}, et B_n^{(2s)}$ admet une $C_{n+s+1}-$Valuation avec $cd(B_n^{(2s)}) = n+s+1$.*

2. $\hat{\hat{D}}_n$ l'arbre formé à partir de deux copies disjointes de D_n, tel que leurs racines sont reliées par une arête appelée arête axiale de $\hat{\hat{D}}_n$. Cet arbre possède 2^{n+2} sommets pendants et $2^{n+2}-2$ sommets de degré 3. Donc $\hat{\hat{D}}_n$ a $2^{n+2}-2$ sommets.
$\hat{\hat{D}}_n$ est montré dans la Figure.3.7 :

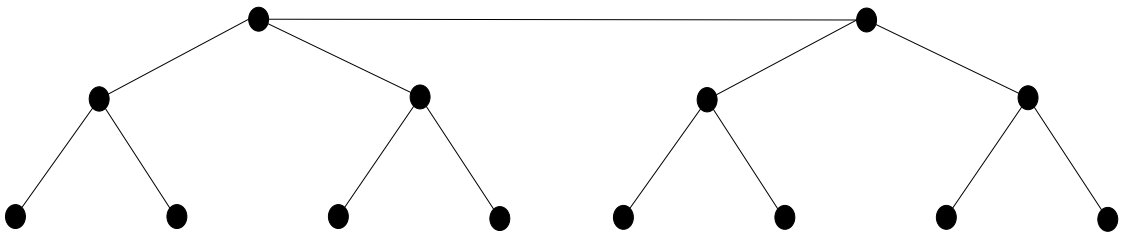

FIGURE 3.7 – Arbre binaire $\hat{\hat{D}}_2$

***Théorème* 3.5.3** *[29] Pour tout $n \geq 1$, l'arbre $\hat{\hat{D}}_n$ admet une $C_{n+2}-$Valuation, avec $Cd(\hat{\hat{D}}_n) = n+2$.*

Dans le cas des arbres binaires équilibré T, Havel à donné la conjecture suivante :

***Conjecture* 3.5.1** *[18] Tout arbre binaire équilibré T, ayant 2^n sommets est C_n-Valué.*

3. Soit $n \geq 1$, on désigne par :

   ▷ $\hat{D}_n$ l'arbre formé à partir de $\hat{\hat{D}}_n$ en insérant deux nouveaux sommets au niveau de l'arête axiale, et la chaine obtenue à partir de l'arête axiale sera appelée chaine axiale de $\hat{D}_n$.

   $\hat{D}_2$ est montré dans la Figure.3.8.

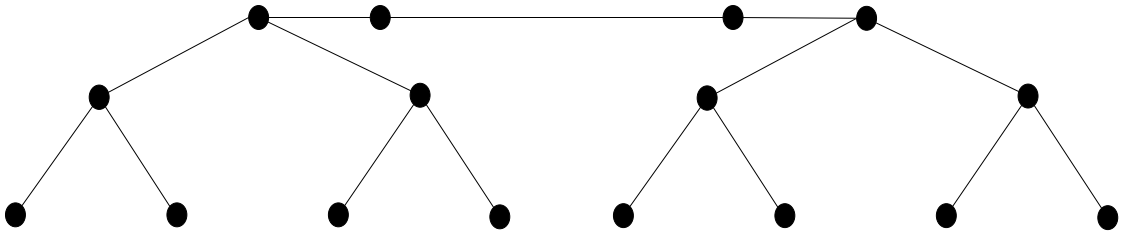

FIGURE 3.8 – Arbre binaire $\hat{D}_2$

   ▷ L'arbre $\check{D}_n$ peut être défini à partir de $\hat{\hat{D}}_n$ en insérant deux nouveaux sommets de degré 2 au niveau d'une arête pendante de $\hat{\hat{D}}_n$.

   La Figure.3.9 donne un Arbre binaire $\check{D}_n$.

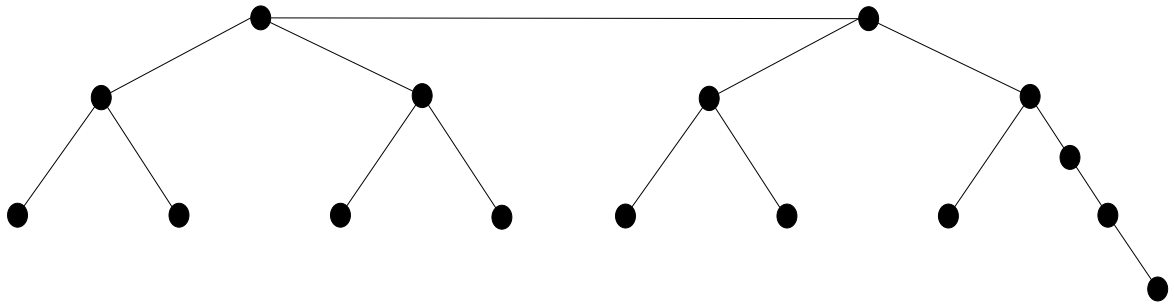

FIGURE 3.9 – Arbre binaire $\check{D}_n$

   Évidement que $\check{D}_n$ et $\hat{D}_n$ sont équilibrés, ils possèdent le même nombre de sommets c-à-d, possède deux sommets de degré 2, 2^{n+1} sommets pendants et $2^{n+1}-2$ sommets de degré 3, qui est équivalant à dire qu'ils ayant chacun 2^{n+2} sommets.

***Théorème* 3.5.4** *[29] Pour tout $n \geq 1$, $Cd(\check{D}_n) = Cd(\hat{D}_n) = n+2$.*

4. Toute les division possible de l'arbre binaire a double racine $\hat{\hat{D}}_n$ ont été donnés dans [31] afin d'obtenir un arbre a 2^{n+2} sommets et ils ont prouvé le résultat suivant.

Type (A) : l'arbre A_n^k est obtenu en subdivisant deux fois une arête de niveau k, $0 \leq k \leq n$, dans $\hat{\hat{D}}_n$, $n \geq 1$.

Type (B) : l'arbre B_n^k est obtenu en subdivisant deux arêtes distinctes de la même niveau k, $1 \leq k \leq n$, dans $\hat{\hat{D}}_n$, n'appartenant pas à la même copie de D_n.

Type (C) : l'arbre C_n^k est obtenu en subdivisant deux arêtes distinctes de la même niveau k, $1 \leq k \leq n$, dans $\hat{\hat{D}}_n$, les deux appartenant à la même copie de D_n.

Type (D) : l'arbre $D_n^{k,l}$ est obtenue en subdivisant deux arêtes distinctes de niveaux k et l, $0 \leq k \leq l \leq n$, dans $\hat{\hat{D}}_n$, n'appartenant pas à la même copie de D_n.

Type (E) : l'arbre $E_n^{k,l}$ est obtenue en subdivisant deux arêtes distinctes de les niveaux k et l, $1 \leq k \leq l \leq n$, dans $\hat{\hat{D}}_n$, les deux appartenant à la même copie de D_n, tels que qu'aucun de ces bords n'est l'ancêtre de l'autre.

Type (F) : l'arbre $F_n^{k,l}$ est obtenue en subdivisant deux arêtes distinctes de les niveaux k et l, $1 \leq k \leq l \leq n$, dans $\hat{\hat{D}}_n$, les deux appartenant à la même copie de D_n, tels que que le bord du niveau k est l'ancêtre du bord du niveau l.

Les types d'arbres définis ci-dessus sont illustrés à la Figure.3.10, les subdivisions sont dessinées sous forme de carrés

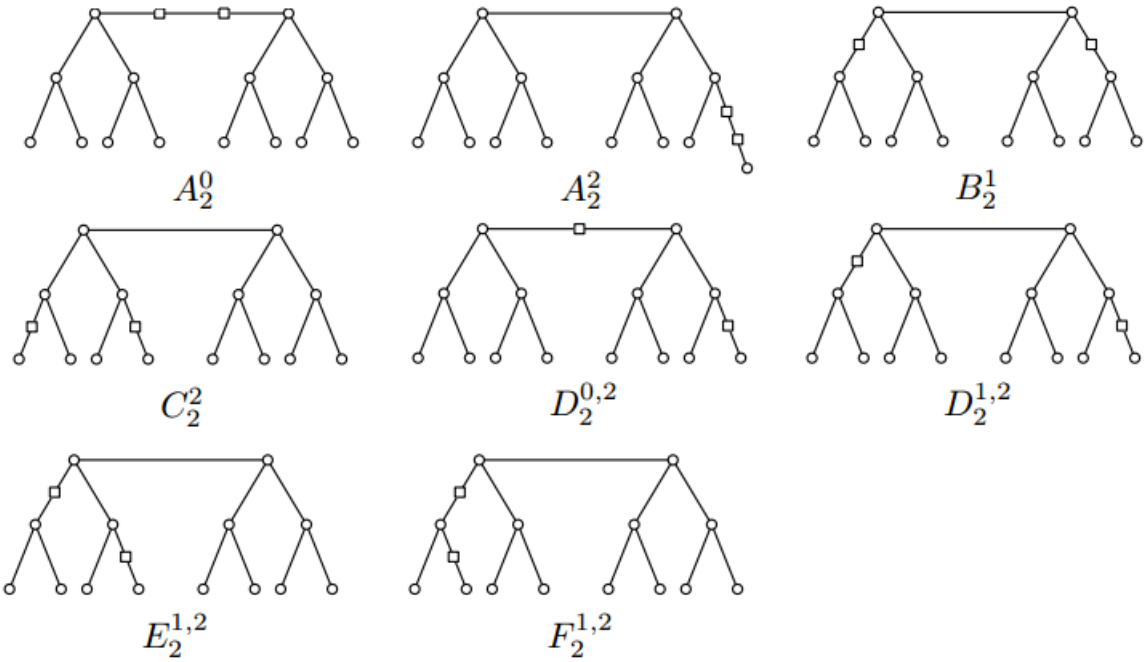

FIGURE 3.10 – Échantillon des arbres binaires complets obtenus par la subdivision de $\hat{\hat{D}}_n$

***Théorème* 3.5.5** *[31] Soit T un arbre obtenu en subdivisant l'arbre binaire $\hat{\hat{D}}_n$. Si T est équilibré alors il est C_{n+2}-valué, si non il est de C_{n+3}-valué :*

- ▷ *$Cd(T) = n + 2$ si T est de type (A) ou (B)*
- ▷ *$Cd(T) = n + 3$ si T est de type (C) , (D) , (E) ou (F)*

3.5.1.1 Arbres AVL(Adelson-Velskii et Landis)

***Définition* 3.5.3** *(Arbres binaires de recherche) : Soit T un arbre binaire tel que chaque sommet est représenté par son poids (valeur entière positive). On dit que T est un arbre de recherche si pour tout sommet u de T et pour tout sommet v du sous-arbre gauche (resp. droit) de u, on a le poids de v est strictement inférieur (resp. strictement supérieur) au poids de u.*

***Définition* 3.5.4** *Un arbre de recherche T est un arbre AVL, si pour tout sommet u de T, la différence $b_T(u)$ entre les hauteurs des sous-arbres gauche et droit de u est au plus 1, par convention la hauteur d'un sous-arbre vide est égale à -1.La différence $b_T(u)$ est appelée facteur d'équilibrage du sommet u. L'arbre AVL est aussi appelé Arbre équilibré en hauteur. La Figure.3.11 montre un exemple d'arbre AVL à 13 sommets.*

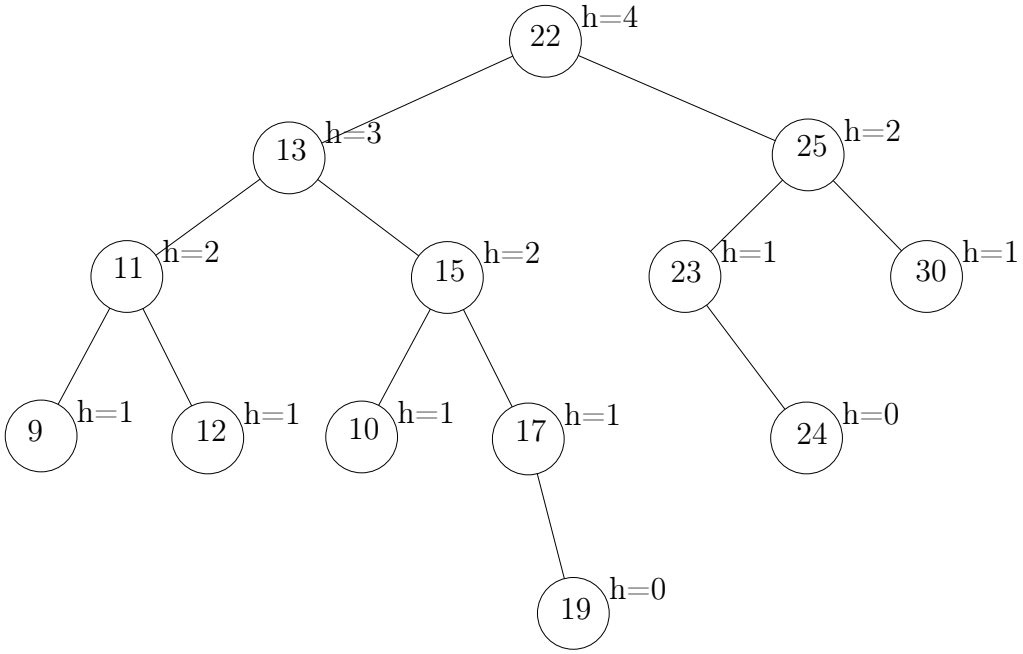

FIGURE 3.11 – Arbre AVL

***Proposition* 3.5.3** *[32] Pour tout $h \geq 0$, l'arbre à hauteur équilibré T_{P_h} admet une $C_{h+1}-$Valuation.*

3.5.1.2 Arbre de Fibonacci

Les arbre de Fibonacci sont des arabes binaires obtenus de la manier suivante :

- ▷ $\mathcal{F}_0$ est l'arbre réduit à un seul sommet : ●
- ▷ $\mathcal{F}_1$ est une chaine de langueur 1 : ●——●
- ▷ $\mathcal{F}_2$ est une chaine de langueur 2 :

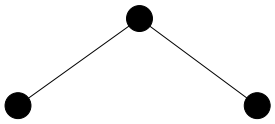

FIGURE 3.12 – Arbre de Fibonacci $\mathcal{F}_2$

▷ pour $n \geq 3$, $\mathcal{F}_n$ est arbre contenant une racine avec $\mathcal{F}_{n-1}$ pour un sous-arbre gauche et $\mathcal{F}_{n-2}$ pour un sous-arbre droit.
$F4$ est donné dans la Figure.3.13.

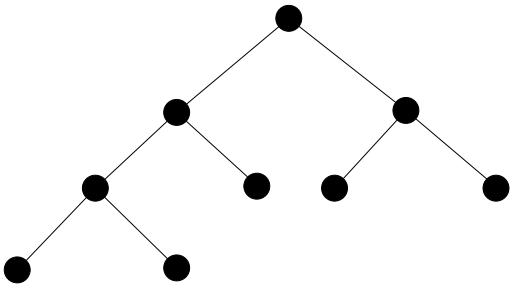

FIGURE 3.13 – Arbre de Fibonacci $\mathcal{F}_4$

Un arbre de Fibonacci $\mathcal{F}_h$ est arbre à hauteur équilibré T_{P_h}. Donc pour tout $h \geq 0$, l'arbre de Fibonacci $\mathcal{F}_h$ admet une C_{h+1}valuation.

***Proposition* 3.5.4** *[19] pour tout $h \geq 0$, $\mathcal{F}_h$ admet une C_{dh}valuation.*

$$d_h = \begin{cases} \dfrac{3 \times h + 4}{4} \text{ si } h = 0(mod4) \\ \dfrac{3 \times h + 5}{4} \text{ si } h = 1(mod4) \\ \dfrac{3 \times h + 6}{4} \text{ si } h = 2(mod4) \\ \dfrac{3 \times h + 3}{4} \text{ si } h = 3(mod4) \end{cases}$$

3.5.2 Grilles

***Définition* 3.5.5** *Une grille est la somme Cartésienne de deux chaines.*
Une n-grille $M = (d_1 \times d_2 \times, ..., \times d_n)$ est la somme Cartésien de n chaines $P_1, P_2, ..., P_n$ d'ordres respectifs $d_1, d_2, ..., d_n$.
Une grille est dite binaire si d_i est une puissance de 2 pour tout $i \in \{1, ..., n\}$.

Grille binaire $M(2 \times 4) = P_2 \oplus P_4$ est montré dans la Figure.3.14.

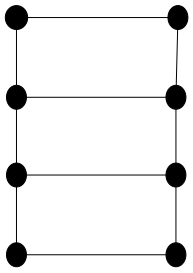

FIGURE 3.14 – Grille binaire $M(2 \times 4) = P_2 \oplus P_4$

Propriétés :

- ▷ Les grilles sont des graphes médians, en effet les chemins sont médians, et la propriété est conservée par la somme cartésien.
- ▷ Une grille carrée de taille n a une largeure d'arbre égale à n^2.
- ▷ Si $d_i = 2$ pour tout i, alors M est l'hypercube de dimension n.

***Théorème* 3.5.6** *[13] La $n-$grille est un graphe C_m-Valué si et selement si $d_1 \times d_2 \times, ..., \times d_n \leq 2^m$ Kobeissi .*

3.5.3 Échelles

***Définition* 3.5.6** *soient deux chaines d'ordres k $P_1 = u_1, u_2, ..., u_k$ et $P_2 = v_1, v_2, ..., v_k$, tels que les sommets u_i et v_i avec $i = 1, ..., k$ sont reliés par des chaines d'ordres $r_1, ..., r_k$ de telle sorte que les extrémités de r_i soient reliées l'une par une arête à u_i et l'autre par une arête à v_i pour $i = 1, ..., k$.En résulte un graphe qui appelée une échelle E, et les chaines entre u_i et v_i sont appelées les rangs de cette échelle.*
Une échelle de rang 0, 2, 1 et 0 est donné dans la Figure.3.15.

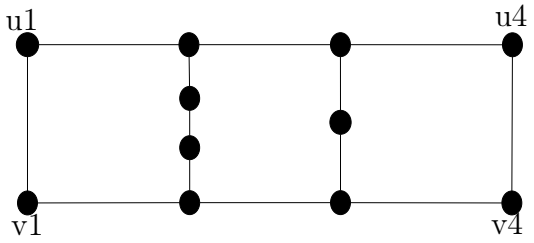

FIGURE 3.15 – Une échelle de rang 0, 2, 1 et 0

***Théorème* 3.5.7** *(Bezrukov) [27] Toute échelle équilibrée est plongeable dans son hypercube optimal.*

3.5.4 Quasi-étoiles et des double quasi-étoiles

Après avoir rappelé des résultats sur le plongement de certains graphes dans l'hypercube, nous nous intéressons à présent deux familles de graphes, à savoir les quasi-étoiles et les doubles quasi-étoiles.

On s'intéresse a prisent aux plongements de graphe ayant un degré maximum par rapport à l'hypercube.

***Définition* 3.5.7** *Une étoile est un graphe d'arbre avec un sommet u qui n'est pas pendant. ce sommet est appelé jonction, et son degré est le nombre d'arêtes qui sont incidents à u. Il est clair qu'une étoile est cubique. Un tel arbre est noté* $K_{1,n}$.

La Figure.3.16 donne une étoile $K_{1,5}$.

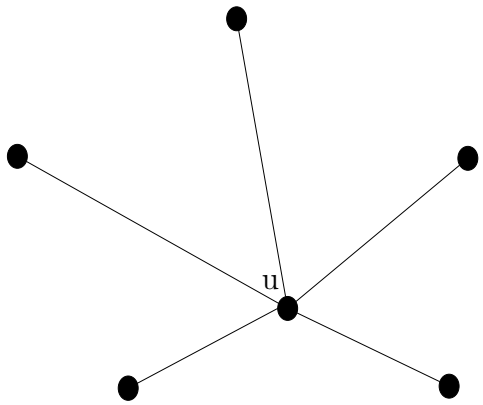

FIGURE 3.16 – Une étoile $K_{1,5}$

Il est évident qu'une étoile est cubique et que $cd(K_{1,n}) = n$.

***Définition* 3.5.8** *Une double étoile est formée de deux étoiles dont les jonctions u et v sont reliées par une arête, u et v ne sont pas forcément de même degré.*

***Définition* 3.5.9** *Une quasi-étoile est une étoile dont les arêtes sont subdivisées. Le degré d'une quasi-étoiles est le nombre de chaines qui sont incidentes a u. On notera* $S(a_1, a_2, ..., a_k)$ *une quasi-étoiles ayant k chaines incidentes à u, d'ordre respectifs* $a_1, a_2, ..., a_k$. *Une quasi-étoiles de degré k, à* 2^n *sommets, est appelée* $k-$*quasi-étoiles.*

Exemple d'une quasi-étoile est montré dans la Figure.3.17.

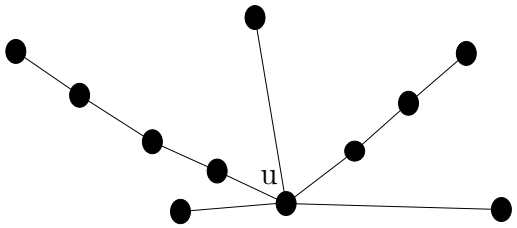

FIGURE 3.17 – Exemple d'une quasi-étoile

la proposition donne une condition nécessaire et suffisante pour qu'une quasi-étoiles soit équilibrée.

***Proposition* 3.5.5** *Une quasi-étoiles est équilibrée si et seulement si elle possède exactement une seule chaîne de longueur impaire.*

***Définition* 3.5.10** *Une double quasi-étoiles est une subdivision d'une double étoile, dans laquelle l'arête qui relier u et v n'est pas subdivisée.*

Une double quasi-étoiles est donné dans la Figure.3.18.

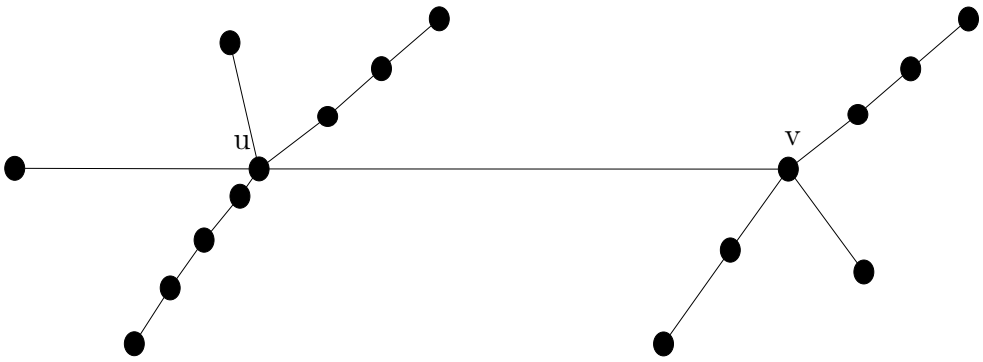

FIGURE 3.18 – Une double quasi-étoiles

Une double quasi-étoiles dont les sommets u et v sont de degrés respectifs k et s avec $(k \geq s)$ est notée $S(a_1, a_2, ..., a_k; b_1, b_2, ..., b_s)$.

Propriété :

- ▷ Une double quasi-étoiles est équilibré si et seulement si le nombre de chaines d'ordres impaires qui sont incidentes àu est égale au nombre de chaines d'ordres impaires qui sont incidentes à v.
- ▷ Une double quasi-étoiles équilibrées $S(a_1, a_2, ..., a_k; b_1, b_2, ..., b_s)$ est appelée $k-$double quasi-étoiles équilibrée.

***Théorème* 3.5.8** *[28] Toute $k-$quasi-étoiles équilibrées à 2^n sommets, avec $k \leq n$ admet une C_n-Valuation .*

Conclusion

Dans ce chapitre, on a présenté la définition de plongement et ses propriétés, on a ainsi parlé sur la notion de C_n-valuation, et on a présenté les différentes classes de graphe plongeable dans l'hypercube de quelques classes de graphes qui seront utiliser par la suite.

4

C_n- valuation de nouvelles classes d'arbres

Introduction

La recherche d'un plongement optimal d'un graphe G dans un hypercube Q_n consiste à trouver si le graphe G admet en C_n- valuation

4.1 Arbre T_n

On définit l'arbre binaire équilibré T_n inductivement comme suit :

▷ Pour n=1, T_1 est une arête donnée par la Figure.4.1 :

FIGURE 4.1 – Arbre T_1

▷ Pour n=2, T_2 est une chaine de longueur 3 donnée par la Figure.4.2 :

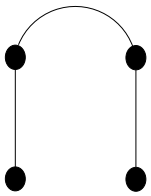

FIGURE 4.2 – Arbre T_2

▷ Pour n=3, T_3 est obtenu à partir de deux copies disjointes T_2' et T_2'' de T_2, tel que un seul sommet de degré 2 de T_2' est relié à son analogue dans T_2'', T_3 est donné par la Figure.4.3 :

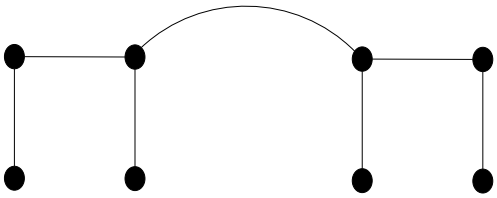

FIGURE 4.3 – Arbre T_3

▷ Pour $n \geq 4$, T_n est obtenu à partir de deux copies disjointes T'_{n-1} et T''_{n-1} de T_{n-1}, tel que un seul sommet de degré 2 de T'_{n-1} est relié à son analogue dans T''_{n-1}.
La Figure.4.4 montre un arbre T_4.

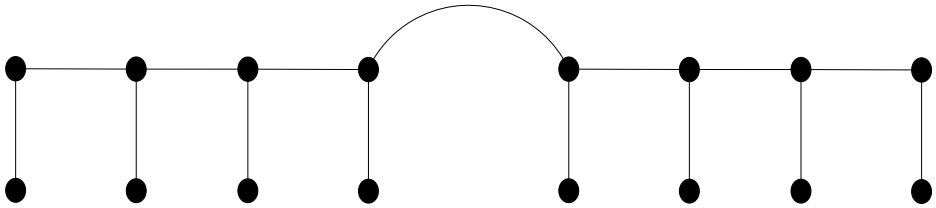

FIGURE 4.4 – Arbre T_4

En reliant à chaque fois les deux sommets de degré 2 on aura T_n un arbre binaire équilibré ayant 2^n sommets.

$\forall n \geq 2 : T_n$ possède 2^n sommets avec 2^{n-1} sommets pendants, 2 sommets de degré 2 et $(2^{n-1} - 2)$ sommets de degré 3.

***Théorème* 4.1.1** *$\forall n \geq 1$ T_n est C_n-valué, donc est plongeable dans Q_n.*

Démonstration :
Raisonnons par récurrence

▷ Il est clair que pour $1 \leq n \leq 4$, T_n est C_n-valué, comme le montre la Figure.4.5

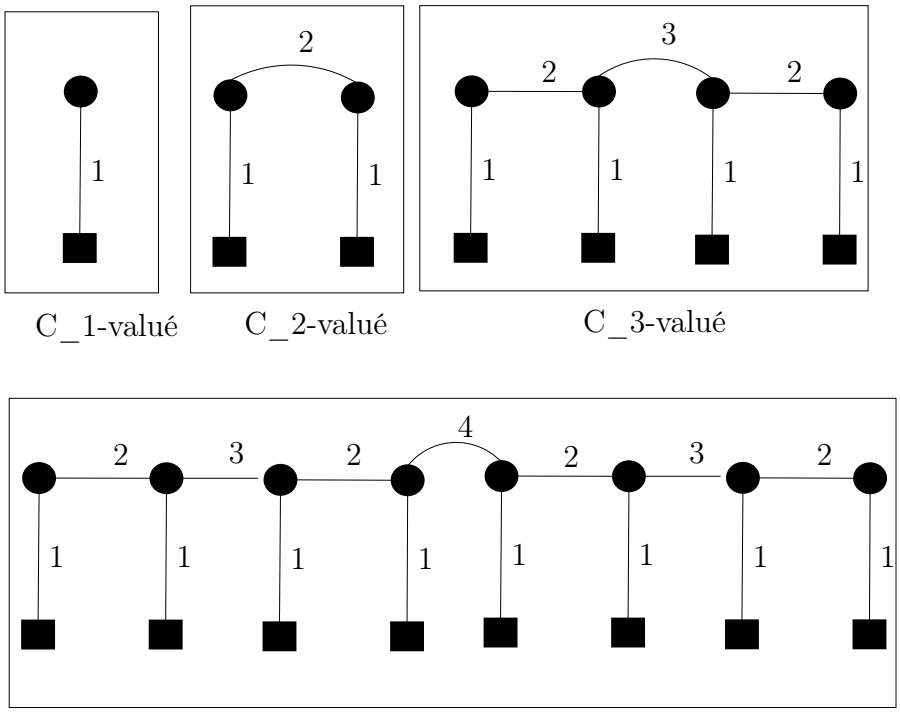

FIGURE 4.5 – C_4-valuation de T_4

▷ Pour $n > 4$: supposons que T_n est C_n-valué $\forall n$, montrons que T_{n+1} est C_{n+1}-valué.

Comme T_{n+1} est obtenu à partir de deux copies disjointes T'_n et $T"_n$ de T_n tel que un seul sommet de degré de 2 de T'_n est relié à son analogue de $T"_n$. comme (d'après l'hypothèse de la récurrence) T'_n et $T"_n$ sont C_n-valué alors pour avoir la $C_{n+1}-$valuation de T_{n+1} on marque l'arête reliant les deux copies disjointes T'_n et $T"_n$ par $n+1$. comme le montre la Figure.4.6 :

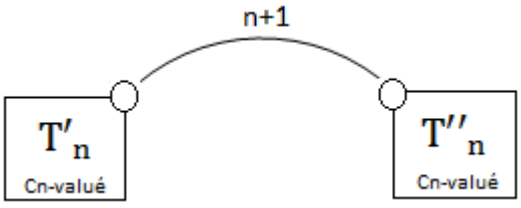

FIGURE 4.6 – $C_{n+1}-$valuation de T_{n+1}

D'où T_{n+1} est plongeable dans Q_{n+1}.

4.2 Arbre $A^k_{n,2^p}$

L'arbre $A^k_{n,2^p}$ est l'arbre défini inductivement comme suit :

▷ Pour $n \geq 1$,et $n \geq k \geq 0$, et $p = 0$ Alors $A^k_{n,2^0} = A^k_n$ est l'arbre obtenu en subdivisant deux fois une arête de niveau k, dans $\hat{\hat{D}}_n$.

▷ Pour $n \geq 1$,et $n \geq k \geq 0$, et $p = 1$ Alors $A^k_{n,2^1}$ est obtenu à partir de deux copies disjointes de (A^k_n)' et (A^k_n)" de A^k_n, en reliant l'un des nouveaux sommet de degré 2 de$A^{k'}_n$ a son analogue de $A^{k''}_n$.

- L'arbre $A^0_{1,2^1}$ donné par la Figure.4.7 :

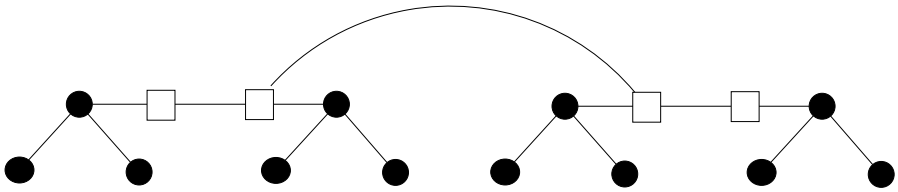

FIGURE 4.7 – L'arbre $A^0_{1,2}$

- L'arbre $A^1_{1,2^1}$ donné par la Figure.4.8 :

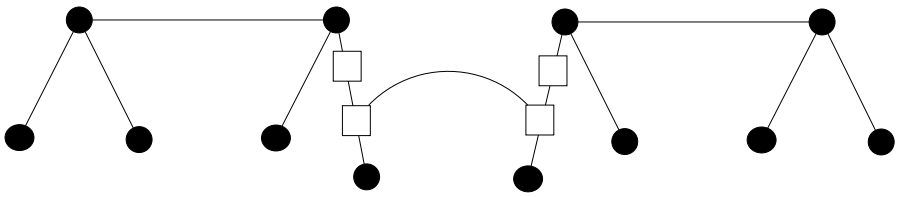

FIGURE 4.8 – L'arbre $A^1_{1,2}$

- L'arbre $A^2_{2,2^1}$ donné par la Figure.4.9 :

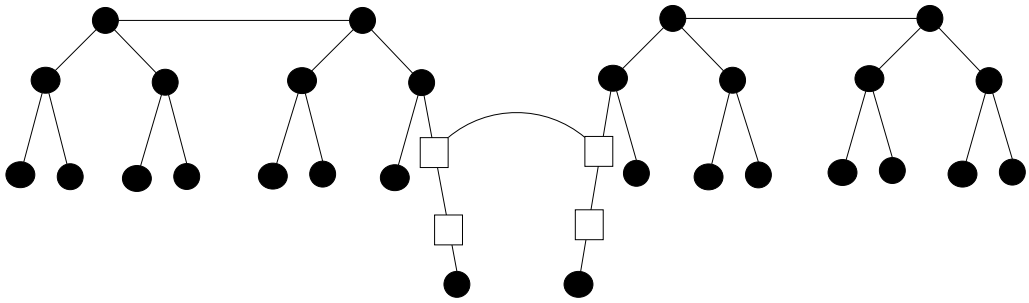

FIGURE 4.9 – L'arbre $A^2_{2,2}$

▷ Pour $n \geq 1$,et $n \geq k \geq 0$, et $p \geq 2$ Alors $A^k_{n,2^p}$ est obtenu à partir de deux copies disjointes de $(A^k_{n,2^{p-1}})$' et $(A^k_{n,2^{p-1}})$" de $A^k_{n,2^{p-1}}$, en reliant l'un des nouveau sommet de degré 2 de($A^k_{n,2^{p-1}}$)' a son analogue de $(A^k_{n,2^{p-1}})$".

- La C_5-valuation de l'arbre $A^0_{2,2}$ est donné dans la Figure.4.10 :

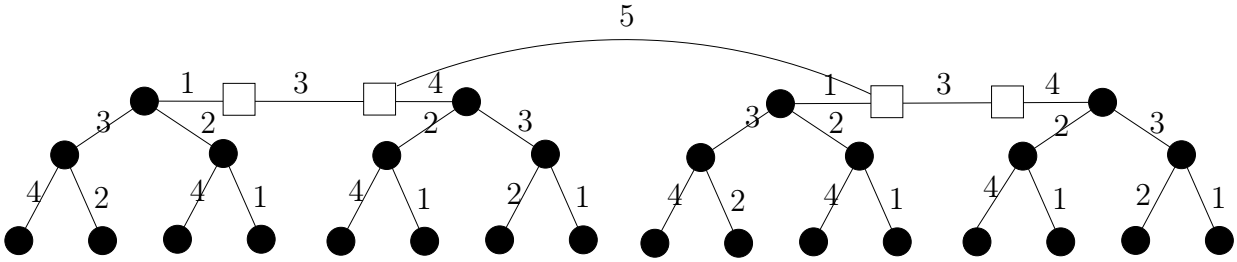

FIGURE 4.10 – C_5-valuation de l'arbre $A^0_{2,2}$

- La C_5-valuation de $A^1_{2,2}$ est donné respectivement dans la Figure.4.11 :

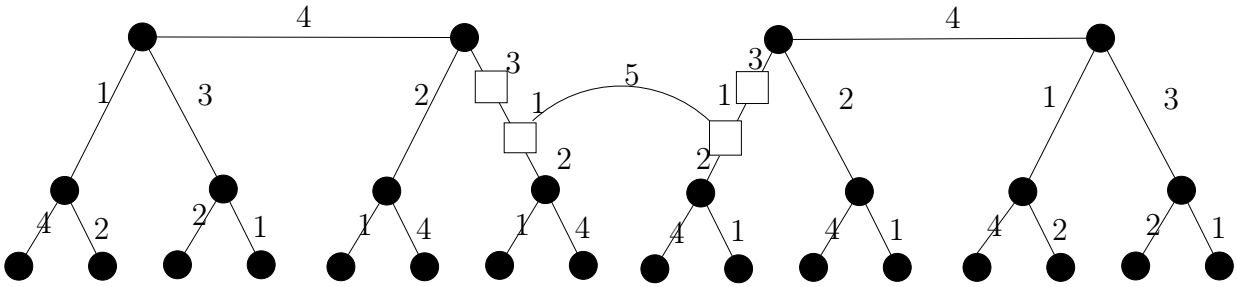

FIGURE 4.11 – C_5-valuation de l'arbre $A^1_{2,2}$

- La C_5-valuation de $A^2_{2,2}$ est donné respectivement dans la Figure.4.12 :

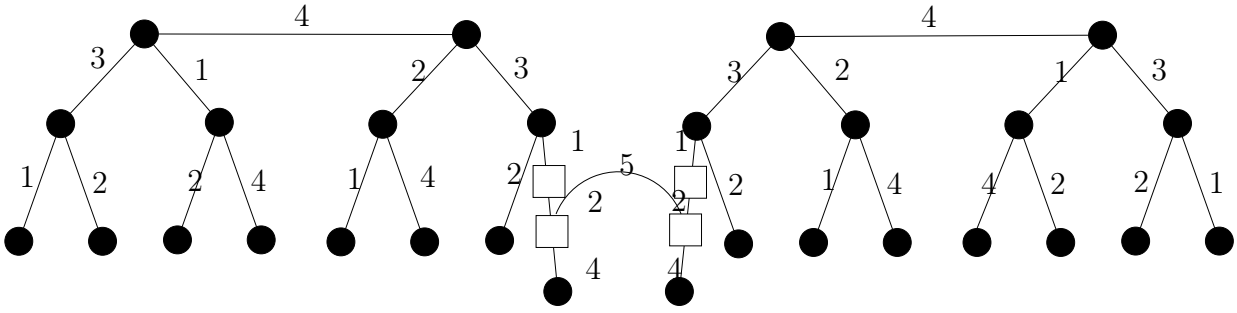

FIGURE 4.12 – C_5-valuation de l'arbre $A^2_{2,2}$

D'après les figure 4.10 ,4.11 et 4.12, et pour $0 \leq k \leq 2$, on remarque que le niveau k n'influence pas sur la C_5-valuation de $A^k_{2,2}$.

Kabyle et al [31] ont prouvé que l'arbre A_n^k admet une $C_{n+2}-$valuation, donc pour trouver la C_n-valuation de $A_{n,2^1}^k$ il suffit de marquer par $n+3$ l'arête reliant (A_n^k)' à (A_n^k)".

La Figure 4.13 montre la $C_{n+2+1}-$valuation de $A_{n,2^1}^k$, la $C_{n+2+2}-$valuation de $A_{n,2^2}^k$, et la $C_{n+2+3}-$valuation de $A_{n,2^3}^k$

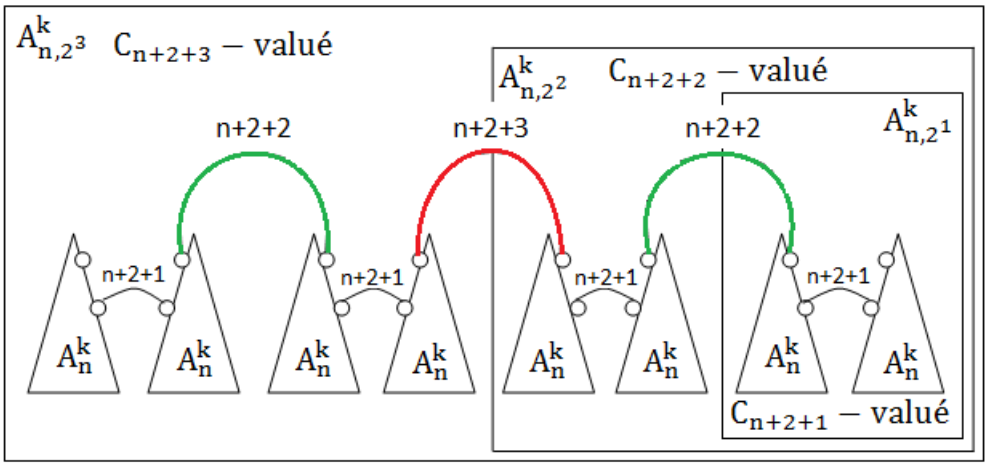

FIGURE 4.13 – C_5-valuation de l'arbre $A_{n,2^3}^k$

Donc l'arbre $A_{n,2^p}^k$ admet une $C_{n+p+2}-$valuation, $Cd(A_{n,2^p}^k) = n+p+2$, comme il est montre dans la Figure.4.14

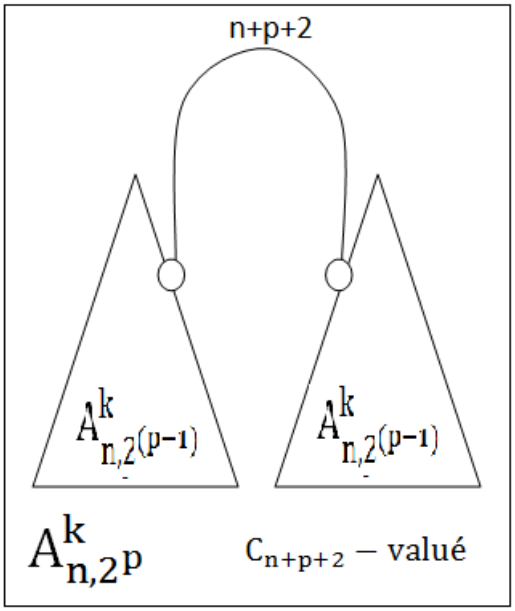

FIGURE 4.14 – $C_{n+p+2}-$valuation de l'arbre $A_{n,2^p}^k$

Démonstration par récurrence :

Pour $A^k_{n,2^0}$ et $A^k_{n,2^1}$ sont respectivement C_{n+2}-valué et C_{n+2+1}-valué.

supposons que $A^k_{n,2^{p-1}}$ est $C_{n+(p-1)+2}$-valué $\forall n$, montrons que $A^k_{n,2^p}$ est aussi C_{n+p+2}-valué.

Comme $A^k_{n,2^p}$ est obtenue à partir de deux copies disjointes $(A^k_{n,2^{p-1}})$' est $(A^k_{n,2^{p-1}})$" de $A^k_{n,2^{p-1}}$ tel que un seul sommet de degré de 2 de $(A^k_{n,2^{p-1}})$' est relié à son analogue de $(A^k_{n,2^{p-1}})$". Comme d'après l'hypothèse de la récurrence, $(A^k_{n,2^{p-1}})$' et $(A^k_{n,2^{p-1}})$" sont $C_{n+(p-1)+2}-$valué alors pour avoir la $C_{n+p+2}-$valuation de $A^k_{n,2^p}$ on marque l'arête reliant les deux copies disjointes $(A^k_{n,2^{p-1}})$' et $(A^k_{n,2^{p-1}})$" par $n+p+2$.

Donc $A^k_{n,2^p}$ est plongeable dans Q_{n+p+2}.

4.3 Arbre AR_n

On définit l'arbre parfaitement équilibré AR_n inductivement comme suit :

▷ Soit AR_1 est donné par la Figure.4.15 :

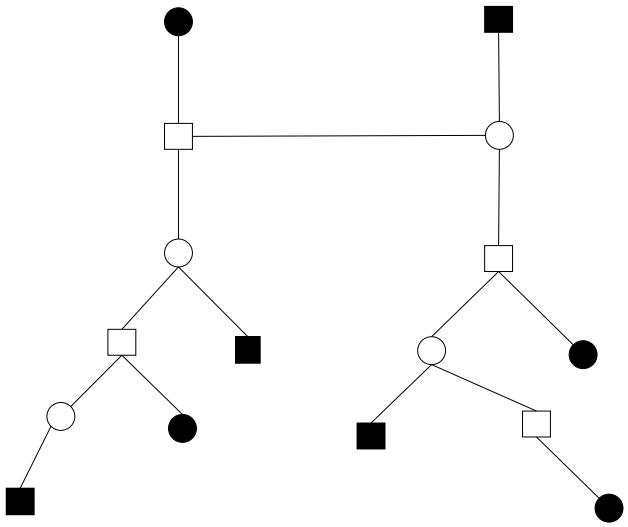

FIGURE 4.15 – L'arbre AR_1

Un arbre T est dit parfaitement équilibré si les sommets de T sont repartis en deux parties $V_1(T)$ et $V_2(T)$ tel que : $|V_1(T)| = |V_2(T)|$ et l'ensemble des sommets pendants de T noté $V^p(T)$ sont repartis en deux parties $V_1^p(T)$ et $V_2^p(T)$ tel que $|V_1^p(T)| = |V_2^p(T)|$.

La Figure.4.16 montre la C_4-valuation l'arbre AR_1 :

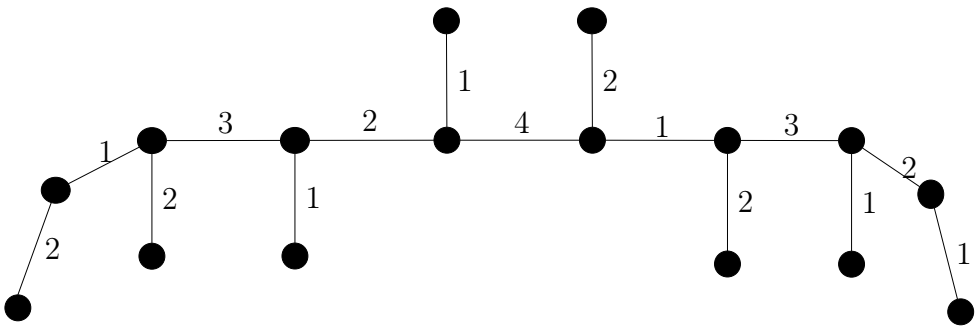

FIGURE 4.16 – La C_4-valuation l'arbre AR_1

L'arbre AR_1 à 16 sommets avec 8 sommets pendant et deux sommets de degré 2 et 6 sommets de degré 3, admet une C_4 valuation, donc il est plongeable dans l'hypercube Q_4.

▷ Soit l'arbre AR_2 est donné par la Figure.4.17

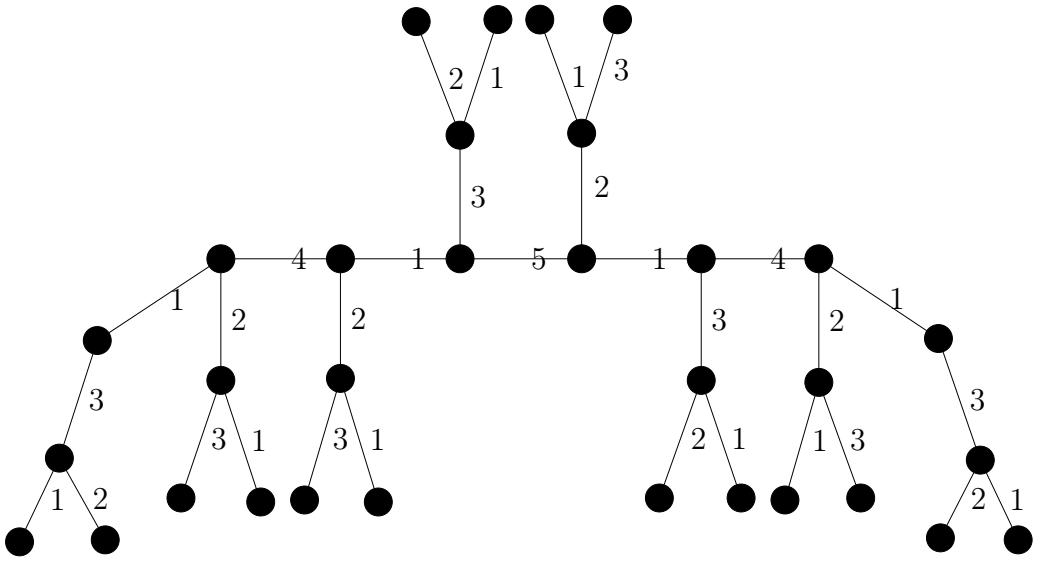

FIGURE 4.17 – l'arbre AR_2

Pour $n \geq 2$, l'arbre parfaitement équilibré AR_n est obtenu à partir de AR_{n-1} tel que chaque sommet pendant de AR_{n-1} est relié à deux nouveaux sommets.

Il est clair que AR_n possède 2^{n+3} sommets, avec 2^{n+2} sommets pendants, deux sommets de degré 2, et $2^{n+2} - 2$ sommets de degré 3.

▷ Prenons deux copies disjoints Q'_4 et $Q"_4$ de Q_4 et une copies Q'_5 de Q_5, tel que dans Q'_4 nous considérons que l'arbre AR_1, dans $Q"_4$ nous prenons 8 copies disjoints de Q_1, et dans Q'_5 nous prenons 4 sommets de types carré, 4 sommets de types cercle, 4 copies disjoints de $K'_{1,2}$ ayant chacune un sommets de degré 2 de type carré, et 4 copies disjointes de $K"_{1,2}$ ayant chacune un sommets de degré 2 de type cercle.(voire la Figure.4.18 :)

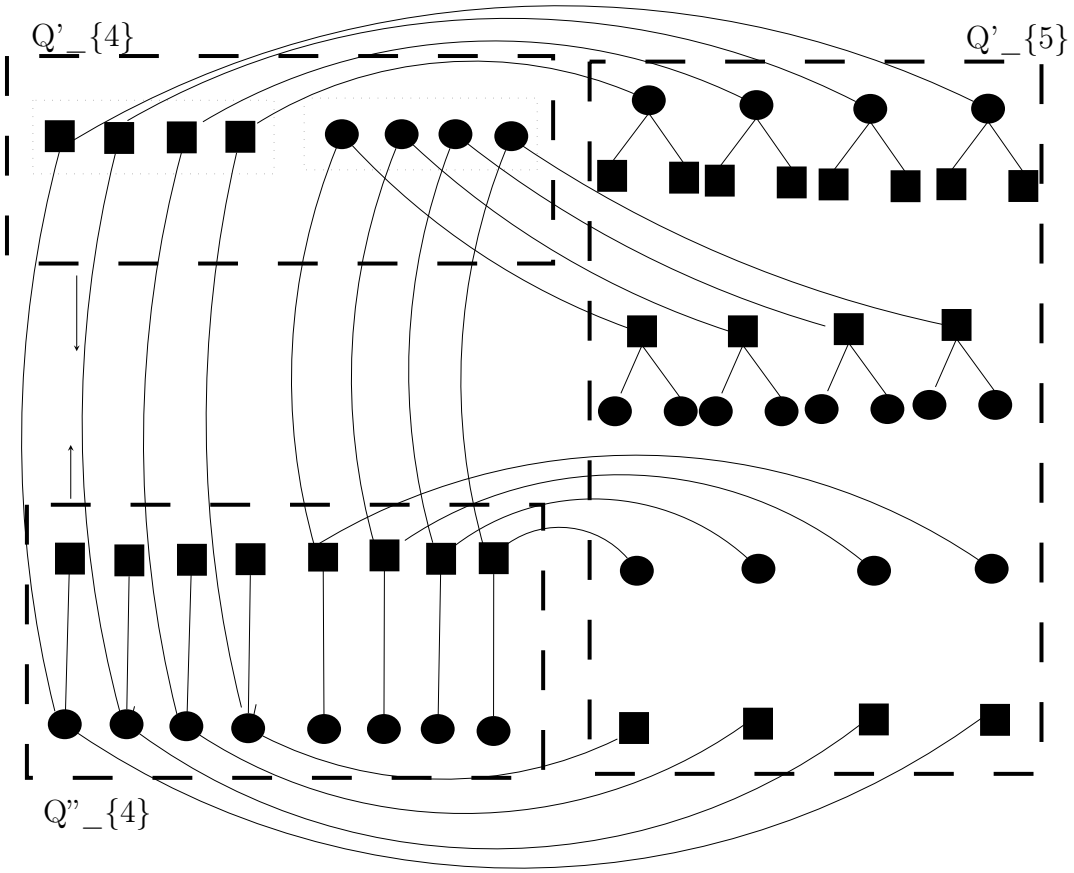

FIGURE 4.18 – Plongement de l'arbre AR_3 dans Q_6

On relie par une arête chaque sommet de type cercle de $Q"_4$ à un sommet isolé de type carré de Q'_5 on crée 4 copie de $K"'_{1,2}$ ayant le sommet de degré 2 cercle, et chaque sommet de type carré $Q"_4$ à un sommet isolé de type cercle Q'_5 on crée 4 copie de $K""_{1,2}$ ayant le sommet de degré 2 carré.
Maintenant on relie chaque sommet de type cercle de Q'_4 à un sommet de degré 2 de $K'_{1,2}$ et $K""_{1,2}$, et relie chaque sommet de type carré de Q'_4 à un sommet de degré 2 de $K"_{1,2}$ et $K'"_{1,2}$, on obtient l'arbre AR_3.

Comme $Q"_5$ est obtenu à partir de deux copies disjoint de Q'_4 et $Q"_4$, et Q_6 est obtenu à partir de deux copies disjoint de Q'_5 et $Q"_5$, Donc l'arbre AR_3 est C_6-valué donc il est plongeable dans Q_6.

***Théorème* 4.3.1** *Pour tout $n \geq 1$, on a AR_n est C_{n+3}-valué*

$$Cd(AR_n) = n + 3$$

Preuve :

La démonstration se fait par récurrence de la même manière que l'arbre AR_3.(Voir la Figure.4.19)

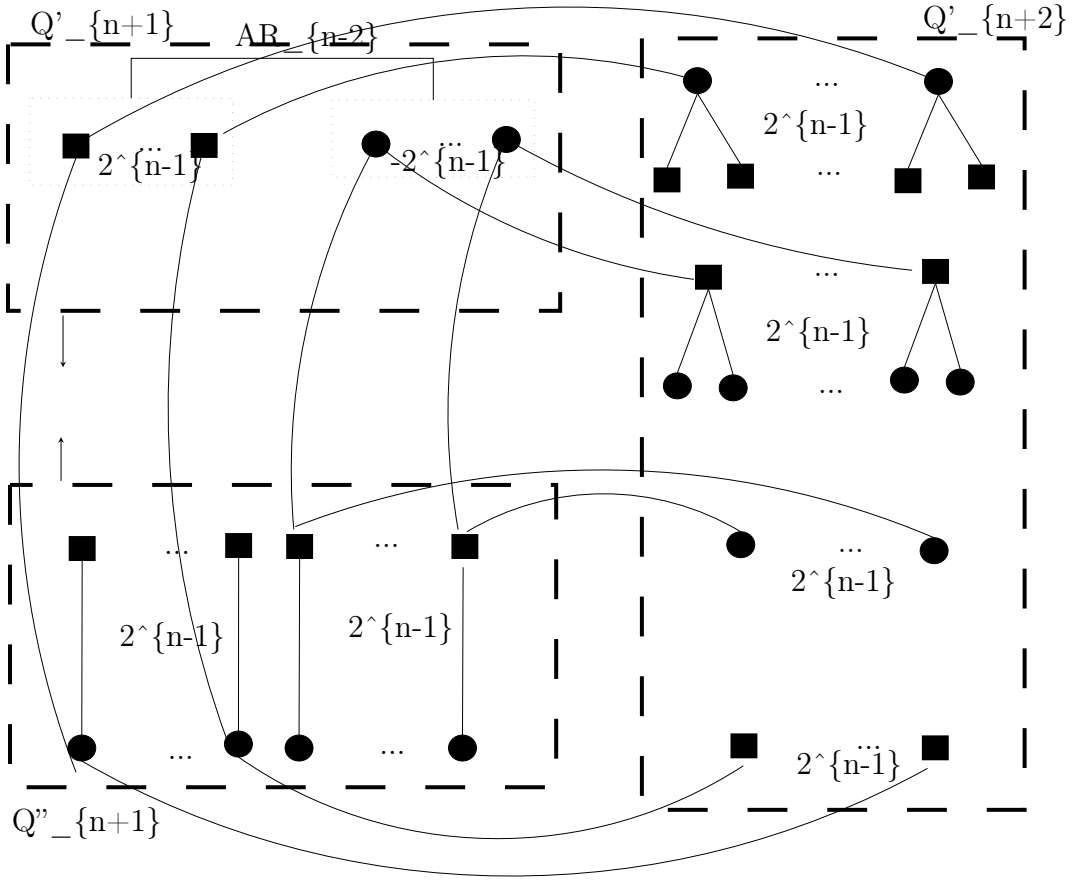

FIGURE 4.19 – l'arbre AR_n

Au faite on a pris deux copies disjointes Q'_{n+1} et Q''_{n+1} de Q_{n+1} et une copie Q'_{n+2} de Q_{n+2}, tel que dans Q'_{n+1} nous considérons que l'arbre AR_{n-2}, dans Q''_{n+1} nous prenons 2^n copies disjoints de Q_1, et dans Q''_{n+1} nous prenons 2^{n-1} sommets de types carré, 2^{n-1} sommets de types cercle, 2^{n-1} copies disjoints de $K'_{1,2}$ ayant chacune un sommets de degré 2 de type carré, et 2^{n-1} copies disjointes de $K"_{1,2}$ ayant chacune un sommets de degré 2 de type cercle.

Conclusion

Dans ce chapitre nous avons présenté des critères de C_n-valuation de quelques graphes obtenus d'une manière récursive dans l'hypercube. Ainsi nous allons essayer de trouver d'autre critère de C_n-valuation des arbres obtenus par la subdivisions des arêtes pour qu'on puisse avoir de nouvelle classes d'arbres binaires C_n-valué.

5
Applications

Introduction

Dans ce chapitre, on vas considérer la construction d'un arbre couvrant de poids minimal. Étant donné un hypercube Q_4 ,un arbre couvrant est un sous-ensemble d'arêtes qui connecte tous les sommets sans former de cycle. Son nombre d'arêtes est exactement le nombre de sommets moins un. Si on attribue un poids $P > 0$ à chaque arête, un arbre couvrant de poids minimal est un arbre couvrant qui minimise la somme des poids de ses arêtes.

Nous allons programmer l'algorithme de Kruskal pour la détermination d'un arbre couvrant minimal. On va considérer que le graphe de référence est l'hypercube Q_4. En fait, la structure de ce graphe sera implicite, on aura seulement à gérer de ensembles d'arêtes.

Comme application, il s'agit d'inter-connecter un ensemble des sommets en minimisant la probabilité totale du graphe.

5.1 Matlab

5.1.1 Historique

MATLAB est une abréviation de Matrix LABoratory (laboratoire de matrice). Il a été conçu par Cleve Moler en 1977 à partir des bibliothèques Fortran, LINPACK et EISPACK2. MATLAB est un langage de développement informatique particulièrement dédié aux applications scientifiques. Matlab est utilisé pour développer des solutions necessitant une très grande puissance de calcul.

MATLAB est un environnement puissant, complet et facile 'a utiliser destiné au calcul

scientifique. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant calcul numérique et visualisation. C'est un environnement performant, ouvert et programmable qui permet de remarquables gains de productivité et de créativité.

MATLAB est un environnement complet, ouvert et extensible pour le calcul et la visualisation. Il dispose de plusieurs centaines (voire milliers, selon les versions et les modules optionnels autour du noyeau Matlab) de fonctions mathématiques, scientifiques et techniques. L'approche matricielle de MATLAB permet de traiter les données sans aucune limitation de taille et de réaliser des calculs numériques et symboliques de façon fiable et rapide. Grâce aux fonctions graphiques de MATLAB, il devient très facile de modifier interactivement les différents paramètres des graphiques pour les adapter selon nos souhaits.

L'approche ouverte de MATLAB permet de construire un outil sur mesure. On peut inspecter le code source et les algorithmes des bibliothèques de fonctions (Toolboxes), modifier des fonctions existantes et ajouter d'autres. MATLAB possède son propre langage, intuitif et naturel qui permet des gains de temps de CPU spectaculaires par rapport à des langages comme le C, le TurboPascal et le Fortran. Avec MATLAB, on peut faire des liaisons de façon dynamique, à des programmes C ou Fortran, échanger des données avec d'autres applications (via la DDE : MATLAB serveur ou client) ou utiliser MATLAB comme moteur d'analyse et de visualisation.

MATLAB comprend aussi un ensemble d'outils spécifiques à des domaines, appelés Toolboxes (ou Boites à Outils). Indispensables à la plupart des utilisateurs, les Boites à Outils sont des collections de fonctions qui étendent l'environnement MATLAB pour résoudre des catégories spécifiques de problèmes. Les domaines couverts sont très variés et comprennent notamment le traitement du signal, l'automatique, l'identification de systèmes, les réseaux de neurones, la logique floue, le calcul de structure, les statistiques, etc...

MATLAB fait également partie d'un ensemble d'outils intégrés dédiés au Traitement du Signal. En complément du noyau de calcul MATLAB, l'environnement comprend des modules optionnels qui sont parfaitement intégrés à l'ensemble :

- ▷ une vaste gamme de bibliothèques de fonctions spécialisées (Toolboxes).
- ▷ Simulink, un environnement puissant de modélisation basée sur les schémas-blocs et de simulation de systèmes dynamiques linéaires et non linéaires.
- ▷ Des bibliothèques de blocs Simulink spécialisés (Blocksets).
- ▷ D'autres modules dont un Compilateur, un générateur de code C, un accélérateur,...
- ▷ Un ensemble d'outils intégrés dédiés au Traitement du Signal : leDSP Workshop.

5.1.2 les particularités de MATLAB

MATLAB permet le travail interactif soit en mode commande, soit en mode programmation ; tout en ayant toujours la possibilité de faire des visualisations graphiques. Considéré comme un des meilleurs langages de programmations (C ou Fortran), MATLAB possède les particularités suivantes par rapport à ces langages :

- ▷ la programmation facile,
- ▷ la continuité parmi les valeurs entières, réelles et complexes,

- ▷ la gamme étendue des nombres et leurs précisions,
- ▷ la bibliothèque mathématique très compréhensive.
- ▷ l'outil graphique qui inclus les fonctions d'interface graphique et les utilitaires,
- ▷ la possibilité de liaison avec les autres langages classiques de programmations (C ou Fortran).

Dans MATLAB, aucune déclaration n'est à effectuer sur les nombres. En effet, il n'existe pas de distinction entre les nombres entiers, les nombres réels, les nombres complexes et la simple ou double précision. Cette caractéristique rend le mode de programmation très facile et très rapide. En Fortran par exemple, une subroutine est presque nécessaire pour chaque variable simple ou double précision, entière, réelle ou complexe. Dans MATLAB, aucune nécessité n'est demandée pour la séparation de ces variables.

La bibliothèque des fonctions mathématiques dans MATLAB donne des analyses mathématiques très simples. En effet, l'utilisateur peut exécuter dans le mode commande n'importe quelle fonction mathématique se trouvant dans la bibliothèque sans avoir à recourir à la programmation.

Pour l'interface graphique, des représentations scientifiques et même artistiques des objets peuvent être créées sur l'écran en utilisant les expressions mathématiques. Les graphiques sur MATLAB sont simples et attirent l'attention des utilisateurs, vu les possibilités importantes offertes par ce logiciel.

5.1.3 MATLAB peut-il s'en passer de la nécessité de Fortran ou du C ?

La réponse est non. En effet, le Fortran ou le C sont des langages importants pour les calculs de haute performance qui nécessitent une grande mémoire et un temps de calcul très long. Sans compilateur, les calculs sur MATLAB sont relativement lents par rapport au Fortran ou au C si les programmes comportent des boucles. Il est donc conseillé d'éviter les boucles, surtout si celles-ci sont grandes.

5.2 Problème de l'arbre couvrant de poids minimum

En théorie des graphes, étant donné un graphe non orienté connexe dont les arêtes sont pondérées. L'arbre couvrant de poids minimum est aussi connu sous certains autres noms, tel qu'arbre couvrant minimum ou encore arbre sous-tendant minimum.

L'arbre couvrant minimum peut s'interpréter de différentes manières selon le type de graphe. De manière générale si on considère un réseau où un ensemble d'objets doivent être reliés entre eux, l'arbre couvrant minimum est la façon de construire un tel réseau en minimisant un coût représenté par le poids des arêtes. pour la recherche d'arbre couvrant de poids minimum, nous utiliserons l'un des algorithmes suivants

5.2.1 Algorithme de Prim

L'algorithme de Prim en 1957 est un algorithme qui trouve l'arbre couvrant minimum dans un graphe connexe valué d'une manière plus centralisée. Et si le graphe n'est pas

connexe, alors l'algorithme ne déterminera que l'arbre couvrant minimum d'une composantes connexe du graphe. En effet, à chaque itération de l'algorithme, l'élément le plus proche de l'arbre couvrant est ajouté. Si un graphe possède un nombre S de sommets, il faudra($|S| - 1$) itérations pour couvrir tous les sommets du graphe et obtenir l'arbre couvrant minimum.

L'algorithme de Prim a donc un fonctionnement très simple. L'ensemble de départ contient un sommet $s_i \in S$ et à chaque itération la meilleure arête est choisie pour étendre notre ensemble de façon optimale.

5.2.1.1 Algorithme [$Prim$]

Entrées : soit un graphe pondérés $G = (V, E, P)$. Soit I l'ensemble de sommets déjà inclus dans l'arbre et NI l'ensemble de sommets non encore inclus ;

(0) Initialisation : $I = \oslash, NI =$ l'ensemble de tout les sommets ;

(1) Mettre le sommet de départ dans I ;

(2) Si l'arbre est connexe aller à (5) ;

(3) Trouver un sommet de NI dont la distance à un sommet a quelconque de I est minimal ; On relie ce sommet a au sommet b et faire $I = I \cup \{b\}$, $NI = NI\{b\}$;

(4) Aller à (3) ;

(5) FIN.

Le critère d'arrêt : L'algorithme s'arrête lorsque le nombre d'arcs retenus est égale à $n - 1$. Sorties : Trouver un arbre de poids minimum de G.

5.2.2 Algorithme de Kruskal

Kruskal a conçu cet algorithme suite à la lecture du papier de Borûvka en 1956. Contrairement à l'algorithme de Borûvka et celui de Prim, l'algorithme de Kruskal nécessite auparavant le tri des arêtes. Cette condition permet de simplifier grandement le fonctionnement de l'algorithme. Il ne s'agit en fait que de placer les meilleures arêtes les unes après les autres, tant qu'elles ne créent pas de cycle. L'algorithme suivant représente les étapes nécessaires que fait l'algorithme de Kruskal

5.2.2.1 Algorithme [$Kruskal$]

Entrées : Soit un graphe pondéré $G = (V, E, P)$;

(0) Initialisation : numéroter les arcs de G dans l'ordre des poids croissants : $P(e_1 \leq P(e_2) \leq P(e_m))$, soit $W = \oslash$,$i = 1$;

(1) Si $(V, W \cup e_i)$ contient un cycle aller en (3) sinon aller en (2) ;

(2) On pose $W = W \cup e_i$,aller en (3)

(3) Si $i = m$ terminer.
$T = (V, W)$ est l'arbre couvrant de poids minimum$P(W) = \sum P(e_i)$, pour $e_i \in W$;
Sinon,$i = i + 1$;

Le critère d'arrêt : L'algorithme s'arrête lorsque le nombre d'arcs retenus est égal à $n-1$. Sorties :Un arbre couvrant de poids minimum.

5.2.3 Résolution de problème avec Algorithme de Kruskal

Soit l'hypercube Q_3 (resp Q_4) un réseau de communication de communication de 8 personnes (resp 16 personnes), te que P_{ij} représente la probabilité d'interception des message confidentiel entre i et j par une autre personne étrangère. On va chercher un arbre couvrant qui minimise la probabilité de l'interception, en utilisant l'algorithme de Kruskal sous MATLAB.

La situation est représentée comme suite :

la Figure.5.1 montre Probabilité d'interception des messages dans un réseau de type hypercube Q_3 :

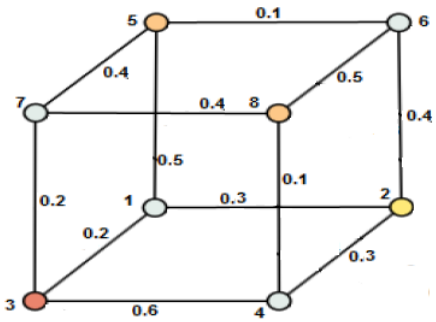

FIGURE 5.1 – Probabilité d'interception des messages dans un réseau de type hypercube Q_3

La Figure.5.2 montre Probabilité d'interception des messages dans un réseau de type hypercube Q_4 :

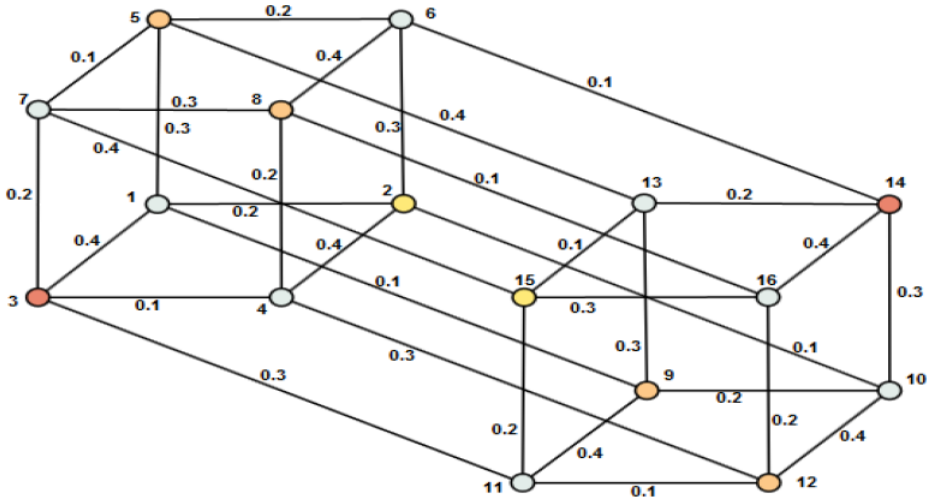

FIGURE 5.2 – Probabilité d'interception des messages dans un réseau de type hypercube Q_4

5.2.3.1 Résolution pour Q_3

Résolution manuelle

Le tableau suivant représente l'ordonnancement croissant des arêtes du graphe selon leurs poids :

i	1	2	3	4	5	6	7	8	9	10	11	12
e_i	(4,8)	(5,6)	(1,3)	(3,7)	(1,2)	(2,4)	(2,6)	(5,7)	(7,8)	(1,5)	(6,8)	(3,4)
$C(e_i)$	0,1	0,1	0,2	0,2	0,3	0,3	0,4	0,4	0,4	0,5	0,5	0,6

TABLE 5.1 – Le tableau d'ordonnancement croissant des arêtes du graphe de Q_3

Initialisation : Soit $W = \oslash; i = 1;$

▷ 1^{er} itération :

$W = W \cup \{e_1\} = W \cup \{(4,8)\} = \{(4,8)\}$ avec $|W| = 1, i = 2\,;$
le graphe ne contient pas de cycle.

▷ Ainsi, 2^{ime} itération de l'algorithme de kruskal est :

$W = W \cup \{e_2\} = \{(4,8);(5,6)\}$ avec $|W| = 2, i = 3\,;$
le graphe ne contient pas de cycle.

▷ Dans, 3^{ime} itération, on obtient :

$W = W \cup \{e_3\} = \{(4,8);(5,6);(1.3)\}$ avec $|W| = 3, i = 4\,;$
le graphe ne contient pas de cycle.

▷ La 4^{ime} itération de l'algorithme de kruskal est :

$W = W \cup \{e_4\} = \{(4,8);(5,6);(1,3);(3,7)\}$ avec $|W| = 4, i = 5\,;$
le graphe ne contient pas de cycle.

▷ En 5^{ime} itération , on trouve :

$W = W \cup \{e_5\} = \{(4,8);(5,6);(1,3);(3,7);(1,2)\}$ avec $|W| = 5, i = 6\,;$
le graphe ne contient pas de cycle.

▷ La 6^{ime} itération , donne :

$W = W \cup \{e_6\} = \{(4,8);(5,6);(1,3);(3,7);(1,2);(2,4)\}$ avec $|W| = 6, i = 7\,;$

le graphe ne contient pas de cycle.

▷ La 7^{ime} itération , donne :

$W = W \cup \{e_7\} = \{(4,8);(5,6);(1,3);(3,7);(1,2);(2,4);(2,6)\}$ avec $|W| = 7, i = 8$; le graphe ne contient pas de cycle.

L'algorithme s'arrête car $|W| = n - 1 =$. 8-1 $= 7$ Alors, $|W| = 7$, l'arbre couvrant de poids minimum il est montré dans le Figure.5.3 :

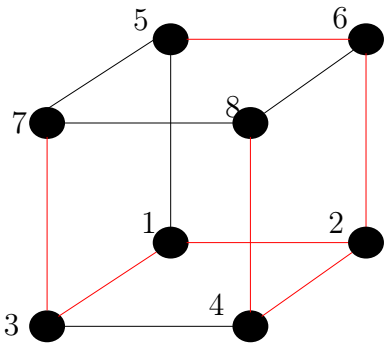

FIGURE 5.3 – Arbre couvrant de poids minimum de Q_3

Résolution sous MATLAB

Premièrement on introduit la matrice de probabilité A de Q_3 dans l'espace commande comme il est représenté dans la Figure.5.4 :

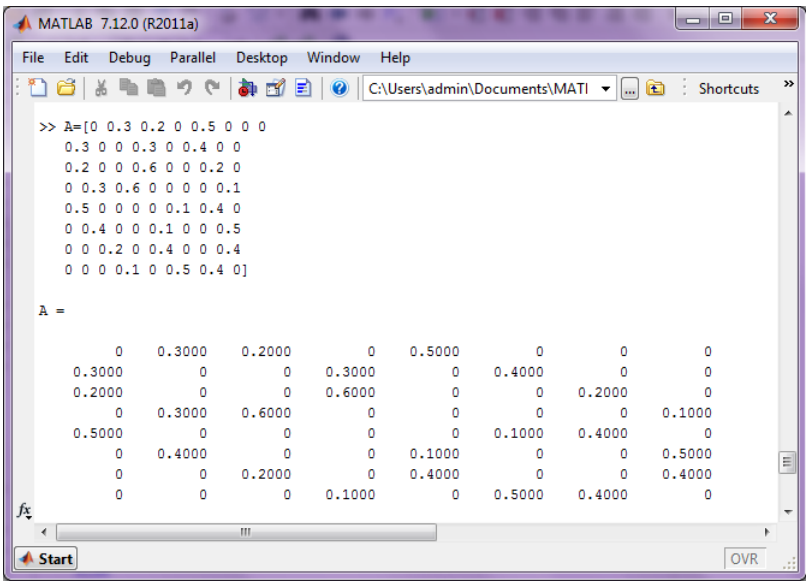

FIGURE 5.4 – La matrice A pour Q_3

En suite, on fait appel a la fonction kruskal(A) dans l'espace commande de MATLAB. On aura la matrice de probabilité de l'arbre couvrant avec son poids minimum. Voir la Figure.5.5 :

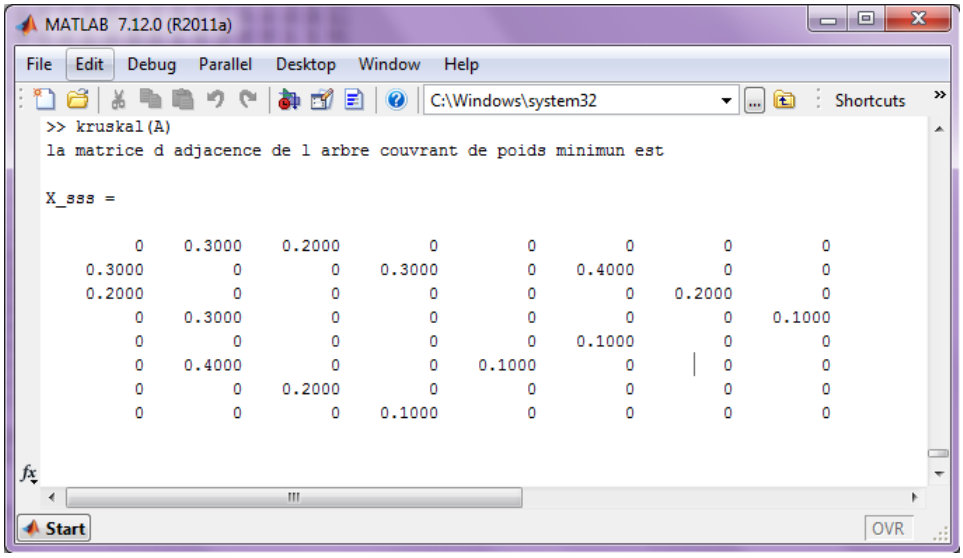

FIGURE 5.5 – Résultat après l'exécution de fonction kruskal pour Q_3

Et la représentation graphique de l'arbre couvrant de Q_3 donné par la Figure.5.6

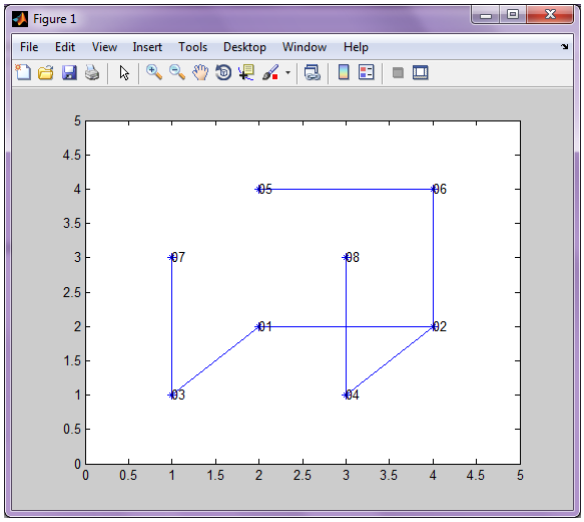

FIGURE 5.6 – Arbre couvrant de poids minimum Q_3

5.2.3.2 Résolution pour Q_4

Résolution manuelle

Le tableau suivant représente l'ordonnancement croissant des arêtes du graphe selon leurs poids :

i	1	2	3	4	5	6	7	8	9	10	11
e_i	(1,9)	(3,4)	(7,5)	(6,14)	(8,16)	(2,10)	(11,12)	(13,15)	(1,2)	(3,7)	(4,8)
$c(e_i)$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,2	0,2	0,2

i	12	13	14	15	16	17	18	19	20	21	22
e_i	(5,6)	(9,10)	(11,15)	(12, 16)	(13, 14)	(1,5)	(2, 6)	(7,8)	(3,11)	(4,12)	(9,13)
$c(e_i)$	0,2	0,2	0,2	0,2	0,2	0,3	0,3	0,3	0,3	0,3	0,3

i	23	24	25	26	27	28	29	30	31	32
e_i	(10,14)	(15,,16)	(10, 12)	(1,3)	(2,, 4)	(6,8)	(7,15)	(5,,13)	(9,11)	(14,16)
$c(e_i)$	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

TABLE 5.2 – Le tableau de l'ordonnancement croissant des arêtes du graphe de Q_4

Initialisation : Soit $W = \oslash; i = 1;$

▷ 1^{er} itération :

$W = W \cup \{e_1\} = W \cup \{(1,9)\} = \{(1,9)\}$ avec $|W| = 1, i = 2\,;$
le graphe ne contient pas de cycle.

▷ Ainsi, 2^{ime} itération de l'algorithme de kruskal est :

$W = W \cup \{e_2\} = \{(1,9);(3,4)\}$ avec $|W| = 2, i = 3\,;$
le graphe ne contient pas de cycle.

▷ Dans, 3^{ime} itération, on obtient :

$W = W \cup \{e_3\} = \{(1,9);(3,4);(7,5)\}$ avec $|W| = 3, i = 4\,;$
le graphe ne contient pas de cycle.

▷ La 4^{ime} itération de l'algorithme de kruskal est :

$W = W \cup \{e_4\} = \{(1,9);(3,4);(7,5);(6,14)\}$ avec $|W| = 4, i = 5\,;$
le graphe ne contient pas de cycle.

▷ En 5^{ime} itération , on trouve :

$W = W \cup \{e_5\} = \{(1,9);(3,4);(7,5);(6,14);(8,16)\}$ avec $|W| = 5, i = 6\,;$

le graphe ne contient pas de cycle.

▷ La 6^{ime} itération , donne :

$W = W \cup \{e_6\} = \{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10)\}$ avec $|W| = 6, i = 7$;
le graphe ne contient pas de cycle.

▷ La 7^{ime} itération , donne :

$W = W \cup \{e_7\} = \{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10);(11,12)\}$ avec $|W| = 7, i = 8$;
le graphe ne contient pas de cycle.

▷ En 8^{ime} itération , on trouve :

$W = W \cup \{e_8\} = \{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10);(11,12);(13,15)\}$ avec $|W| = 8, i = 9$;
le graphe ne contient pas de cycle.

▷ En 9^{ime} itération , on trouve :

$W = W \cup \{e_9\} = \{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10);(11,12);(13,15);(1,2)\}$ avec $|W| = 9, i = 10$;
le graphe ne contient pas de cycle.

$W = W \cup \{e_{10}\} = \{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10);(11,12);(13,15);(1,2);(3,7)\}$ avec $|W| = 10, i = 11$;
le graphe ne contient pas de cycle.

On continue de la même façon jusqu'à la 17^{ime} itération.

▷ La 17^{ime} itération , donne :

$W = W \cup \{e_7\} = \{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10);(11,12);(13,15);(1,2);(3,7);(4,8);(5,6);(11,15);(12,16);(1,5)\}$ avec $|W| = 15$;
L'algorithme s'arrête car $|W| = n - 1 = 16 - 1 = 15$. Alors, $|W| = 15$, l'arbre couvrant de poids minimum il est : $\{(1,9);(3,4);(7,5);(6,14);(8,16);(2,10);(11,12);(13,15);(1,2);(3,7);(4,8);(5,6);(11,15);(12,16);(1,5)\}$.

Résolution sous MATLAB

On introduit la matrice de probabilité A de Q_4 dans l'espace commande comme il est représenté dans la Figure.5.7 :

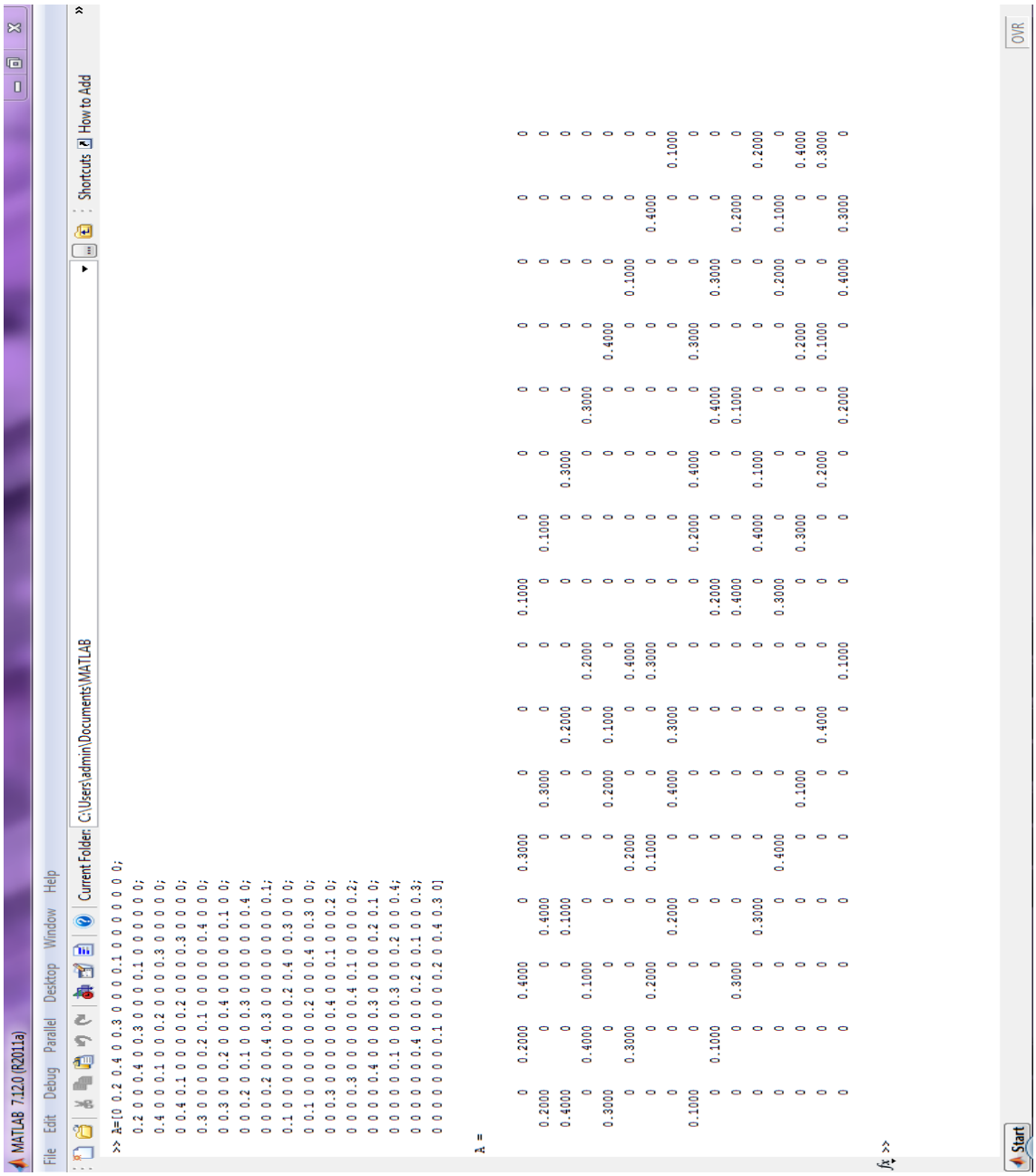

FIGURE 5.7 – La matrice A

En suite, on fait appel a la fonction kruskal(A) dans l'espace commande de MATLAB. On aura la matrice de probabilité de l'arbre couvrant avec son poids minimal. Voir la Figure.5.8 :

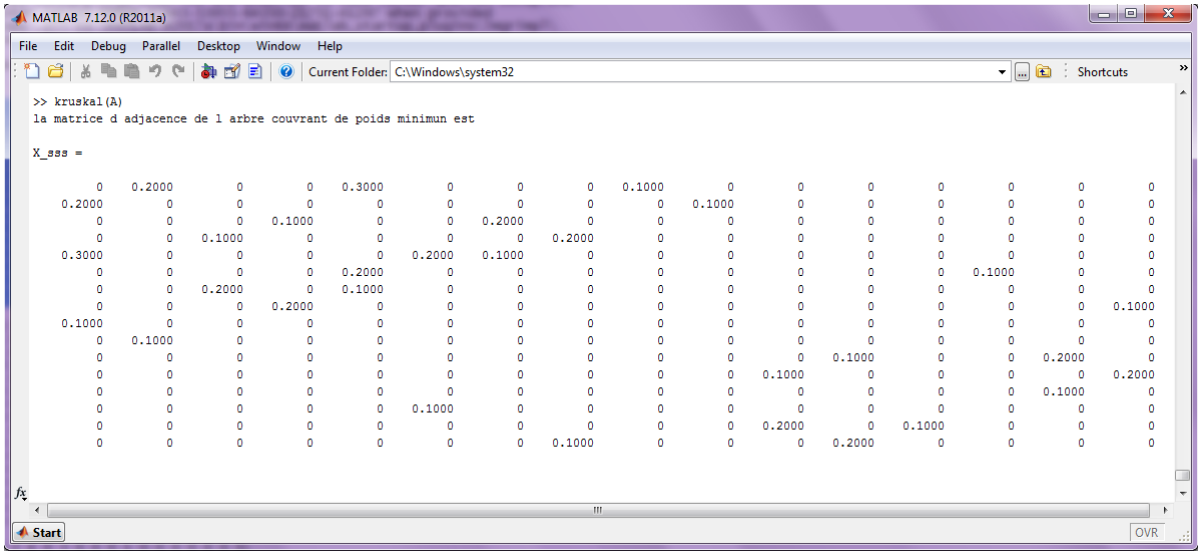

FIGURE 5.8 – Résultat après l'exécution de fonction kruskal

Et la représentation graphique de l'arbre couvrant donné par la Figure.5.9 :

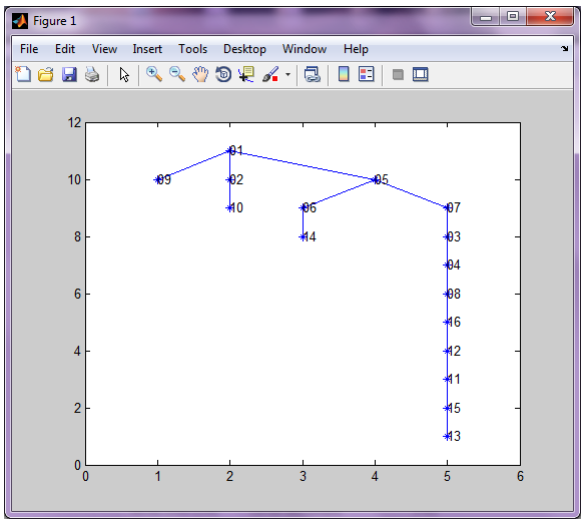

FIGURE 5.9 – Arbre couvrant de poids minimum de Q_4

Conclusion

Dans ce chapitre on a présenté le problème de l'arbre couvrant minimum comme arbre plongeable dans l'hypercube, ainsi deux deux algorithmes pour la résolution (Prim,Kruskal), et on a implémenté l'algorithme de kruskal sous MATLAB.

Conclusion Générale

Les graphes constituent une méthode de représentation qui permet de modéliser une grande variété de problèmes concrets en se ramenant à l'étude de sommets et d'arêtes.

Plusieurs chercheurs se sont intéressés sur ce type de problèmes, ce qui a permet de caractériser plusieurs classes de graphes pour lesquels la dimension est connue.

Dans le même contexte, nous avons réussi à introduire trois nouvelles classes d'arbres binaires équilibrés pour lesquels la dimension cubique.

Par ailleurs nous avons implémenté l'algorithme de Kruskal pour la recherche d'arbres de poids minimum dans un réseau de télécommunication de topologie (structure) hypercube.

Comme perspective, nous allons essayer de caractériser des nouvelles classes d'arbres binaires équilibrés, pour lesquels nous donnerons une technique de plongement optimale. Il est important de caractériser les classes des graphes de Hamming pour lesquels nous déterminerons la C_n-valuation.

Bibliographie

[1] N.Bellharat, *"La théorie des graphes"*, 2010.

[2] Didier Müller, *"Introduction à la théorie des graphes"*, 2012.

[3] Didier Maquin, *"Éléments de Théorie des Graphes"*, 2003.

[4] M. Mulder, *"Quelques problèmes combinatoires sur l'hypercube et les graphes de Hamming"*, PhD thesis, thèses de doctorat, L'Université de Joseph Fourier, 1989.

[5] H.M Mulder, *"n-cubes and median graphs"*, J.Graph Theory, 1982.

[6] H.M Mulder, *"The interval function of a graph"*, Mathematical center tracts 123,Mathimatical centrum, Amesterdam 1980.

[7] M. Meringer, J, *"Graph Theory"*, 1999, 30, 137.

[8] S.Foldes, *"A characterization of hypercube discrete"*, No. 155-159 in 18. 1977.

[9] P.Van de cruyce, *"A chracterization of the n-cube by convex subgraphes. Discrete Mathematics"*, (1982).

[10] A.Berrachedi, *"Sur quelques propriétés métriques du graphe de type hypercube. PhD thesis,Institut de Mathématique, Université des Science et de la Technologie Houari Boumediène"*, (1997).

[11] G.buroch, J. M. L, *"Distance monotone graphe and a new caractérization of hypercube. Discrete Mathématics"*, 110 (1982).

[12] S. Flodes, *"A characterization of hypercubes"*, Discrete Mathematics, (17) :155- 159, 1977.

[13] M. Kobeissi, *"Plongement de graphes dans l'hypercube"*, PhD thesis, Université Grenoble I, 2001.

[14] C.BERGE, *"Graphe et hypercube"*, 1, 1973.

[15] M.R. Garay and R.L. Graham , *" Compurters and Intractability "*,A guide to theiry og NP- completeness. W.M. Freeman and Company, 1979.

[16] J. Arfati C.H. Papadimitriou, and P. Papageorgiou, *"The comlexity of cubical graphe"*, Proceedings of 11th International Colloquium on Automata, Languades and Programming, pages 51-57, 1984.

[17] A. Wagner and D.G. Corneil , *" Embedding trees in a hypercube is np-complet"*, Technical report 197/87, Departement of Computer Science, University of Toronto, Toronto, Onatario, February 1987.

[18] Havel, P. L , *"Embedding the dichotomie tree into the n-cube"* Cas.Pest. mat, 97 1972.

[19] V.Firsov , *"On isometric embeddings of a graph into a boolean cube"*, Cyberne- tics (in Russian) 1. No. 6, pp. 112-113, (1965).

[20] , *"J. M. B valuations of graphs"*, Czech Math Journ, 98 (1972).

[21] Berrachedi, A , *"Sur la dimension cubique de quelques classes d'arbres. Actes du Colloque Cosi'04"*, Colloque sur L'optimisation et les Systèmes d'Information. Université de Tizi-Ouzou.

[22] Bezrukov, S. and Monien, B. Unger, W. and Wechsung, G, *"Embedding ladders and caterpillars into hypercube"*, discrete applied mathematics , 83 (1992) 21-29.

[23] Havel, I., *" On hamiltonian circuits and spanning trees of hypercubes"*, Cas prest. Mat 109 (1984).

[24] Harary, F. Lewinter, M. and Widolski, W., *"On two legged caterpillars which span a hypercube"*, Congr. Numer. 66 (1988).

[25] kobeissi, M. and Mollard, M., *" Spanning graphs of hypercubes starlike and double starlike trees"* Accept discrete Math.

[26] Labord, J.M. and Rao hebbar, S.P, *"Another characterisation of hypercube"*, discrete Math., 39 , (1982).

[27] S.Bezrukov, W. Unger,and G. Wechsung. *"Embedding ladders and caterpillars into hypercube"*. Discrete Applied Mathematics, (83) :21-29,1992.

[28] F.Harary et M. Lewinter, *"The starlike trees which span a hypercube"*,Comput. Math. Appl, 1988.

[29] L. Nebesky, *"On cubes and dichotomic trees"*, Caspis Pest. Mat. 99, No. 2, pp. 164-167, (1974).

[30] Havel, J. M. B , *"valuations of graphs"*, Czech Math Journ, 98 (1972).

[31] Kamal Kabyl, Abdelhafid Berrachedi, Éric Sopena, *"A NOTE ON THE CUBICAL DIMENSIONOF NEW CLASSES OF BINARY TREES"*, Czechoslovak Mathematical Journal, (2015).

[32] G.M. Adelson-Velskki and E.M. Landis, *"An algorithm for the organisation of information."*.

Résumé

L'étude de plongement d'un graphe G dans un hypercube H revient à voir si G admet une C_n-valuation. Dans ce mémoire, nous nous somme intéressé à la C_n-valuation des graphes, dont nous avans introduit trois nouvelles classes d'arbres pour lesquelles la dimension cubique est déterminée. Puis on a implémenté l'algorithme de Kruskal sous MATLAB pour la recherche d'arbres de poids minimum dans un réseau de télécommunication de topologie (structure) hypercube.

Abstract

The study of the embedding of an graph G in an hypercube H is to say if G allowan C_n-valuation. In this dissertation, we were interested in the C_n-valuation of graphs, which we hare given three new classes in which the cubic dimension is determined. Then we implemented the Kruskal algorithm with in MATLAB for search a minimum spanning tree in the hypercube.

Mots-clés : La C_n-valuation, plongement, graphe, Hypercube, arbre.