République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département recherche opérationnelle

Mémoire De Fin de cycle

En vue d'obtention du diplôme de master en recherche opérationnelle spécialité : MMTD

Plongement des graphes dans l'hypercube.

Réalisé par :

- Melle LASMI Oumana
- Melle ZERROUKI Hanane

Soutenu le 09/11/2020 devant le jury composé de :

Président	**Mr TAOUINET Smail**	**MAA à U.Béjaia.**
Examinatrice	M^{eme}**YOUNSI Leila**	**MAA à U.Béjaia.**
Encadreur	**Mr KABYL Kamal**	**MCB à U.Béjaia.**

2019/2020

Remerciements

Nous tenons à exprimer nos vifs remerciements à :

Allah qui nous a donné la volonté de réaliser ce travail.

Notre promoteur et encadreur **Mr KABYL Kamal**, pour ses conseils, son aide, sa gentillesse, sa disponibilité à tous les instants ainsi que pour sa patience qui ont été d'un grand support moral afin de mener à terminer ce travail.

Mr TAOUINET Smail, pour l'honneur et le plaisir qu'elle nous a fait en acceptant de présider ce jury.

M^{eme} **YOUNSI Leila**, qui a bien voulu examiner ce mémoire et d'accepter à participer au jury.

Aux enseignants du Département de Recherche Opérationnelle nos parents, chacun en son nom, pour nous avoir soutenus.

Dédicaces

Je dédie mon modeste travail :

A la mémoire de mon chèr père ZERROUKI Mohand :
Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne veut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation. Enfin je réalise ton rêve, donc je t'offre ce cadeau, je sais que quelque part, vous êtes heureux pour ta chère fille Hanane. puisse Dieu t'accueillir dans son infinie Miséricorde.

A ma très chers mère :
Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon chèr frère Hafid :
Mon chèr frère qui m'est le père et la mère, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour vous. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.

A ma très chère sœur Wahiba :
En témoignage de l'attachement, de l'amour et de l'affection que je porte pour vous, vous êtes toujours dans mon cœur. Je vous remercie pour votre hospitalité sans égal et votre affection si sincère. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.

A la plus chère personne de ma vie Billal :
Quand je t'ai connu, j'ai trouvé l'homme de vie, mon âme sœur la lumière de mon chemin Que dieu réunisse nos chemins pour un long commun serein et que ce travail soit témoignage de nos reconnaissance et de mon amour sincère et fidèle.

A ma chère binôme **Oumana**.
Hanane.

Dédicaces

Je dédie ce modeste travail.

A celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite à ma maman **BENAMRAOUI Malika**.

A mon père **LASMI Ahmed**, école de mon enfance, qui a été mon ombre durant toutes les années des études, Qui dieu les gardes et les protégé.

A mon très cher frère FARAS.

A mes adorables sœurs ZINA, NOURA, RAZIKA, IKRAME.

A mes anges IMANE, INASE, SERINE, ADEM, LINA.

A toutes mes amies qui m'ont encouragé, et a qui souhaite plus de succès, tout particulièrement FEYROUZ, SARAH, OUIDADE, ROSA, SOUAD, KAMI, CHAYMA, OUAHIBE.

A celle avec qui j'ai partagé ce modeste travail ma binôme HANANE.

A tout ceux qui j'aime.

OUMANA.

Table des matières

Table des Figures vi

Introduction générale **1**

1 Notations et Notions de Base **3**

1.1 Concepts de base 3

1.1.1 Graphes : 3

1.1.2 Graphe non orienté 4

1.1.3 Graphe orienté : 5

1.1.4 Degré d'un sommet : 5

1.2 Chaînes, Chemins, Cycles et Circuits 9

Chaînes et cycles : 9

Chemins et circuits : 10

1.3 La notion de connexité dans les graphes 11

1.3.1 Graphe connexe 11

1.3.2 Composante connexe 12

1.3.3 Graphe fortement connexe 12

1.3.4 Composante fortement connexe 12

1.4 Représentation matricielle des graphes 13

1.4.1 Matrice d'adjacence 13

1.4.2 Matrice d'incidence 14

1.5 Opérations sur les graphes 16

1.5.1 Somme Cartésienne de deux graphes : 16
1.5.2 Produit Cartésien 16
1.5.3 Subdivision d'un graphe 17
1.5.4 Isomorphisme 17
1.5.5 Homomorphisme 17
1.6 Quelques Graphes particuliers 18
1.6.1 Graphe complet 18
1.6.2 Graphe biparti 19
1.6.3 Graphe biparti complet 19
1.6.4 Graphe biparti équilibré 19
1.6.5 Graphe planaire 20
1.6.6 Graphe régulier 21
1.6.7 Graphe distance monotone 22
1.6.8 Graphe sphérique 22
1.6.9 Graphe de Hamming 23
1.6.10 Arbre 23

2 Hypercube et quelques caractérisation **24**
2.1 Hypercube 24
2.1.1 Degré 27
2.1.2 Nombre de sommets 27
2.2 Caractérisation de l'hypercube 27
2.3 Projection et anti-projection 28
2.4 Plongement 29
2.4.1 Paramètres de plongement 29
2.4.2 Plongement optimal 30
2.5 Graphes et dimensions cubiques 30
2.6 Plongement dans Q_n 31
2.6.1 Condition nécessaires de plongement d'un graphe G dans Q_n 32

2.7 La notion de C_n-valuation 33
2.7.1 Théorème 33

3 Plongement de certaines classes d'arbres dans l'hypercube 34
3.1 Quelques types d'arbres 34
3.1.1 Arbres binaires complets : 34
3.1.2 Arbres binaires obtenues par transformation des arbres binaires complets D_n . 35
3.1.3 Classes d'arbres obtenus par subdivision de $\widehat{\widehat{D_n}}$ 37
Type (A) 37
Type (B) 38
3.1.4 Arbre de Fibonacci 38

4 Nouvelle classes d'arbre plongeable dans Q_n 40
4.1 Décomposition de Q_n en des copies disjointes de Q_i 42

5 Application 50
5.1 Arbre couvrant de poids minimum 50
5.2 Algorithme de Kruskal (Bis)(1956) 50
5.3 Algorithme de Prim (Bis)(1957) 51
5.4 Application sur le $C++$: 52
5.4.1 Présentation de langage C++ 52
5.4.2 Exemple d'application de cet algorithme sur un graphe 53
5.4.3 Application numérique 53

Conclusion générale 59

bibliographie 60

Table des figures

1.1 Exemple d'un graphe 4
1.2 graphe non orienté 4
1.3 Graphe orienté 5
1.4 Sous graphe 7
1.5 graphe simple 8
1.6 Un graphe contient une clique 8
1.7 Un stable 9
1.8 Une chaîne de longueur 3 10
1.9 Chemin élémentaire 11
1.10 Un graphe connexe 11
1.11 composantes connexes d'un graphe 12
1.12 Graphe non orienté à 5 sommets 13
1.13 Graphe orienté à 5 sommets 14
1.14 Graphe non orienté à 5 sommets et 6 arêtes 15
1.15 Somme cartésienne de G et H 16
1.16 Le produit cartésien de $G_1 \otimes G_2$ 17
1.17 Homomorphisme 18
1.18 Graphe K_4 18
1.19 Un graphe biparti 19
1.20 Un graphe biparti équilibré 20
1.21 Un graphe planaire 20

1.22 Un graphe régulier 21
1.23 Un arbre 23

2.1 Hypercube Q_1, Q_2 et Q_3 25
2.2 Décomposition canonique de Q_3 26
2.3 Un cycle dans l'hypercube Q_3 26
2.4 Graphe $K_{2,3}$ 31
2.5 Arbre équilibre 2^4 sommets non plongeable dans Q_4 32

3.1 Arbre binaire complet D_2 35
3.2 L'arbre B_2 36
3.3 L'arbre $\widehat{\widehat{D_2}}$ 36
3.4 Arbres binaires $a(A_2^0)$ et $b(A_2^2)$ 37
3.5 Arbres binaires $a(b_1^1)$ et $b(b_2^2)$ 38
3.6 Arbres de Fibonacci 39

4.1 Arbres binaires 40
4.2 Arbre binaire $\alpha(T_1, T_2)$ 41
4.3 Exemple 42
4.4 Représentation des sommets de Q_3 43
4.5 Décomposition de Q_4 en des copies disjointes de Q_3 43
4.6 Q_2 44
4.7 Arbre $\overline{R_1}$ 45
4.8 Arbre $\overline{R_2}$ 45
4.9 Q_3', Q_3'' et Q_4 46
4.10 Construction de 4 copies disjointes de $K_{1,2}$ 47
4.11 Arbre $\overline{R_3}$ 48
4.12 $\overline{R_{n+1}}$ 49

5.1 L'interface de code blocks 53
5.2 Probabilité d'interception des messages dans un réseau de type hypercube 54

5.3 Résultat de la première itération 55
5.4 Résultat de la deuxième itération 56
5.5 Arbre couvrant de poids minimum 57
5.6 Résultat après l'éxécution de l'algorithme de Prim pour Q_3 58

Introduction

L'histoire de la théorie des graphes débute avec les travaux d'Euler au 21^{eme} siècle et trouve son origine dans l'étude de certains problèmes, tel que celui des ponts de Königsberg, la marche du cavalier sur l'échiquier ou le problème de coloriage de cartes. La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales. Depuis le début du XXème siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de Köenig, de Cayley puis de Berge et beaucoup d'autres. De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseaux de communications, réseaux routiers, interaction de diverses espèces animales, circuits éléctriques,..

Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs.

L'hypercube a fait naître de nombreuses études engendrant une littérature trés consistante aussi bien en mathématiques qu'en informatique. Grâce à sa structure, la motivation première de l'intêret sans cesse croissant qui est porté sur l'hypercube et son utilisation dans de nombreux domaines (architecture parallèle, transfert de l'information, réseaux d'intrconnexion).

Il demeure encore le centre d'intérêt de plusieurs travaux récents focalisés sur la manière de caractériser les graphes comme étant des sous graphes de l'hypercube (problème de plongement). En effet, de nombreux efforts ont été consacrés pour déterminer des conditions (nécessaires et/ou suffisantes) selon lesquelles un graphe G est un sous-graphe de l'hypercube Q_n. L'accent est surtout mis sur le plongement d'arbres. Cet intérêt résulte de l'utilisation des arbres dans plusieurs domaînes, à savoir : informatique, sciences sociales, recherche opérationnelle, théorie des réseaux électriques,...

Si un graphe est plongeable dans l'hypercube de dimension n, Q_n, alors il est plongeable dans Q_p, $\forall p \geq n$. Le problème consiste à trouver le plus petit k tel que G est plongeable dans Q_k. On parle alors de dimension cubique du graphe G. Il est connu que tous les arbres sont plongeable dans l'hypercube. Par conséquent, pour un arbre donné, on s'intéresse à sa dimension cubique.

Ce problème a été traîté par plusieurs auteurs tel que I.Havel, F.Harary, M.Laborde, ce qui a permis de caractériser certaines classes d'arbres. Dans ce mémoire, on a établi la dimension cubique de certaines classes d'arbres.

Ce mémoire est développé en cinq chapitres dont le premier est consacré aux rappels nécessaires pour la suite. Le chapitre 2 est une présentation de l'hypercube et de certaines de ses caractéristiques, on y trouve aussi une présentation du problème de plongement de graphes dans l'hypercube.

Létude du plongement d'arbres dans l'hypercube fera l'objet du chapitre 3. Dans ce chapitre, nous présenterons certains résultats existant concernant ce problème. On établit au chapitre 4 la dimension cubique de deux nouvelles classes d'arbres. Le cinquième chapitre représente une application.

Chapitre 1

Notations et Notions de Base

1.1 Concepts de base

1.1.1 Graphes :

un graphe G est un couple (V, E) où V est un ensemble d'objets appelés les sommets du graphe avec |V|=n et $E \subseteq V.V$ est une relation binaire sur V.V. Les éléments de E sont appelés les arêtes du graphe. Le nombre de sommets du graphe G est appelé ordre de G qu'on note O(G). Le nombre d'arcs du graphe G est dit taille de ce graphe qui est égal à m.
La figure $1,1$ montre un exemple de graphe :

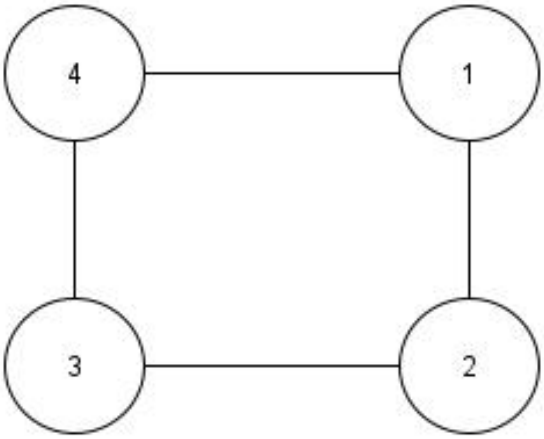

FIGURE 1.1 – Exemple d'un graphe

1.1.2 Graphe non orienté

Un graphe non orienté G = (V, E) est constitué d'un ensemble fini non vide de sommets et d'un ensemble fini E dont les éléments sont appelés arêtes. Une arête $''e''$ de l'ensemble E est défini par une paire non ordonnée de sommets. Si l'arête $''e''$ relie les sommets 1 et 2, on dira que ces deux sommets sont adjacents ou bien que l'arête $''e''$ est incidente à 1 et 2. La figure $1, 2$ montre un exemple d'un graphe non orienté.

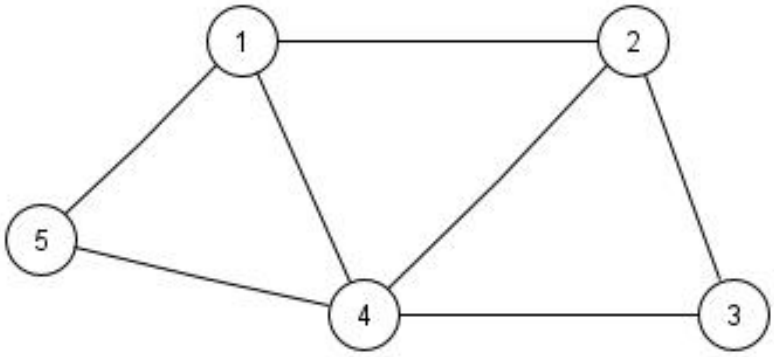

FIGURE 1.2 – graphe non orienté

1.1.3 Graphe orienté :

En donnant un sens aux arêtes d'un graphe, on aura un graphe orienté G = (V, E) qui est constitué d'un ensemble fini non vide de sommets noté V et d'un ensemble fini E dont les éléments sont appelés arcs. Un arc $''e''$ de l'ensemble E est défini par une paire ordonnée de sommets. Lorsque l'arc e = (1,2), on dit que 1 est l'éxtrémité initiale de $''e''$ et 2 est l'éxtrémité terminale de $''e''$. La figure $1,3$ montre un exemple d'un graphe orienté :

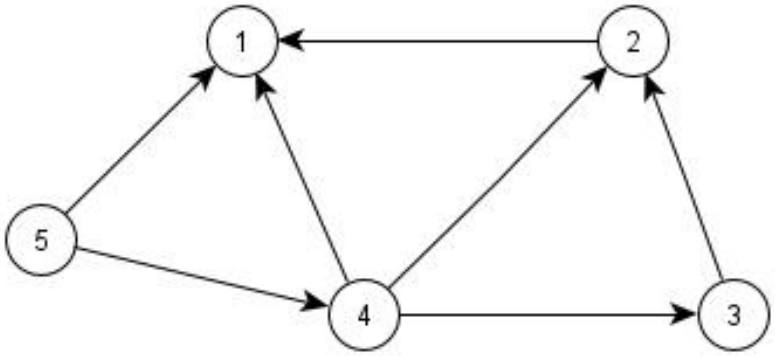

FIGURE 1.3 – Graphe orienté

1.1.4 Degré d'un sommet :

Soit G = (V,E) un graphe orienté :

— Le demi degré éxtérieur d'un sommet x : est égal au nombre d'arcs ayant x comme extrémité initiale (c'est à dire les arcs qui sort de x) on le note $d_G{}^+(x)$ avec :
$d_G{}^+(x) = |\{e \in E/\ \mathbf{I(e) = x}\}|$.

— Le demi degré intérieur d'un sommet x : est égal au nombre d'arcs ayant x comme extrémité terminale (c'est à dire les arcs qui entrant en x) on le note $d_G{}^-(x)$ avec :
$d_G{}^-(x) = |\{e \in E/\ \mathbf{T(e) = x}\}|$.

— Le degré d'un sommet x : c'est le nombre d'arcs ayant x comme extrémité initiale ou terminale, on le note $d_G(x)$ avec :
$d_G(x) = d_G{}^-(x) + d_G{}^+(x)$.

Propriété 1.1 (Lemme des poignées de mains :)[1]
Pour un graphe non oriénté G = (V, E), on a :

$$\sum_{x \in V} d_G(x) = 2|E|$$

Propriété 1.2[1]
Pour un graphe oriénté G=(V, E), on a :

$$\sum_{x \in V} d_G^+(x) = \sum_{x \in V} d_G^-(x) = |E|$$

•**Sous graphe :** Si G est un graphe dont les sommets sont l'ensemble V et les arêtes sont l'ensemble E, et si V' est une partie de V : on appelle sous graphe de G engendré par V' le graphe G'=(V', E') dont les sommets sont les éléments de V' et les arêtes sont les éléments de E reliant deux sommets de V'. Dans la figure $1,4$: on a tracé un graphe et le sous-graphe formé à partir des sommets A, B, C.

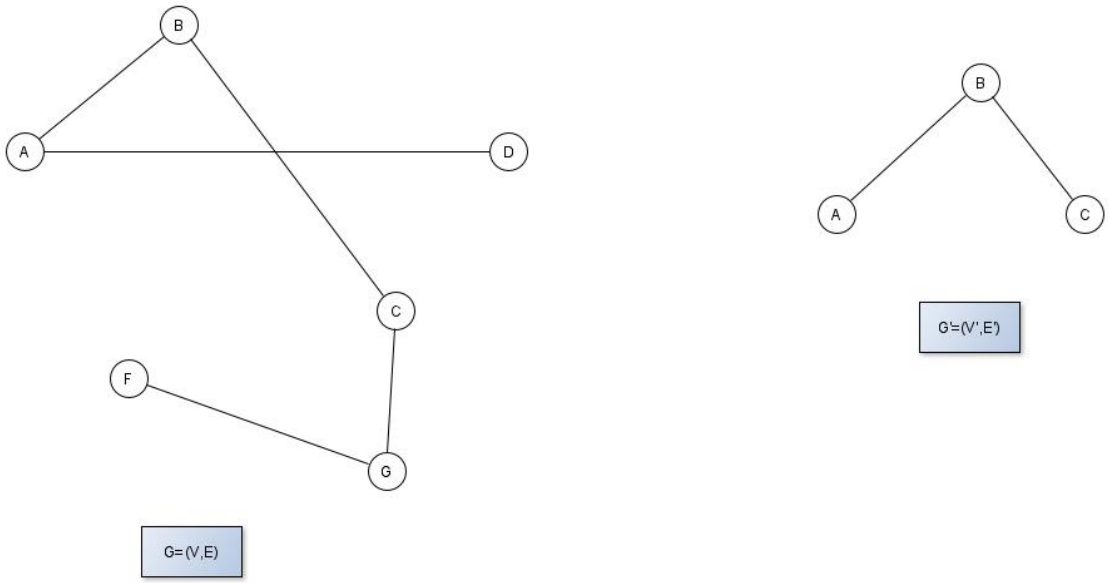

FIGURE 1.4 – Sous graphe

•**Graphe partiel :**

Soit G = (V, E) un graphe. Le graphe G'=(V, E') est un graphe partiel de G si E' est inclus dans E. Autrement dit, on obtient G' en enlevant une ou plusieurs arêtes au graphe G.

•**Sous-graphe partiel :** Un sous-graphe partiel est un graphe partiel d'un sous graphe.

•**Graphe simple :** Un graphe simple est un graphe dans lequel chaque paire de sommets est relié au plus par une arête et aucun sommet ne possède de boucle. L'exemple d'un graphe simple est donné par la figure $1,5$:

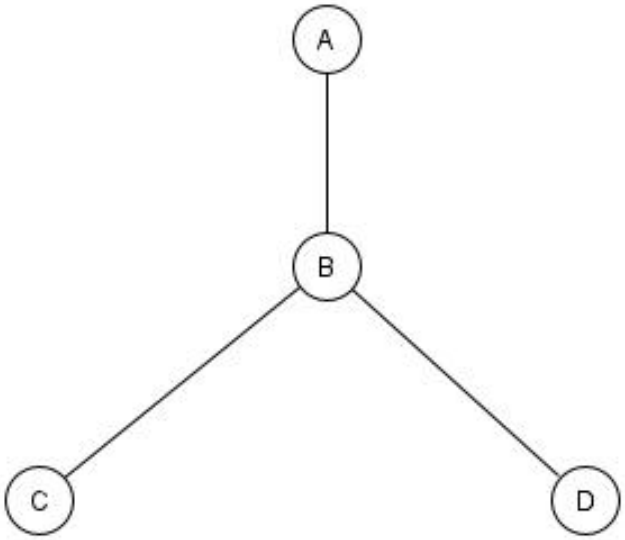

FIGURE 1.5 – graphe simple

•**Graphe multiple :** C'est un graphe qui possède des boucles ou bien des arêtes (arcs) multiples.
•**Clique :** Une clique d'un graphe non orienté est un sous-ensemble des sommets de ce graphe dont le sous graphe induit est complet, autrement dit deux sommets quelconques de la clique sont toujours adjacents. Dans le graphe de la figure $1,6$, le sous ensemble de sommets 1,2,5 forme une clique.

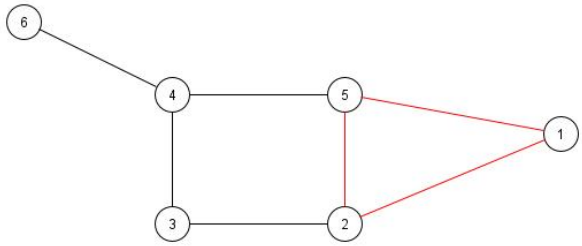

FIGURE 1.6 – Un graphe contient une clique

•**Stable :** Un stable (ensemble indépendnt) est un ensemble de sommets deux à deux non adjacents. La taille d'un stable est égal au nombre de sommets qu'il contient. On remarque dans le graphe de la figure 1, 7 que l'ensemble $\{2, 4, 6\}$ est un stable.

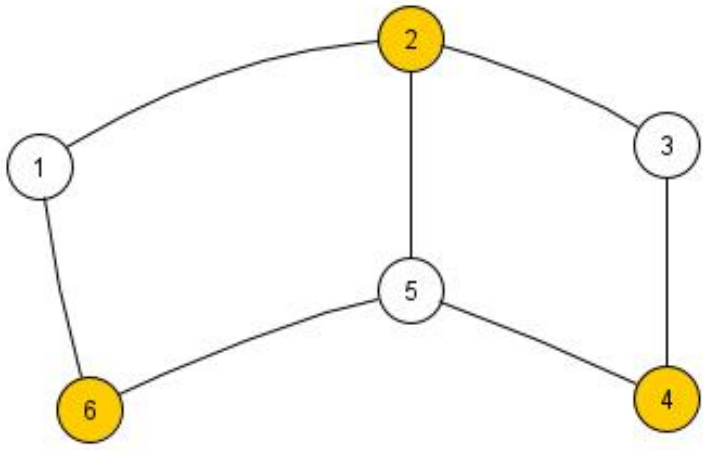

FIGURE 1.7 – Un stable

•**Graphe complémentaire :** Soit G= (V, E) un graphe simple. Le graphe $\bar{G}$=(V, $\bar{E}$) est le graphe complémentaire de G si : $\forall e \in E \Leftrightarrow e \notin \bar{E}$.

1.2 Chaînes, Chemins, Cycles et Circuits

Chaînes et cycles :

• **Chaîne :** On appelle chaîne toute succession d'arêtes dont l'éxtrémité de l'une (sauf la dernière) est l'origine de la suivante. Le nombre d'arêtes qui composent une chaîne est appelé longeur de la chaîne.

Une chaîne est *simple* si la séquence d'arêtes qui la constitue ne comporte pas plusieurs fois le même élément. Une chaîne *élémentaire* est une chaîne ne passant pas deux fois par un même sommet.

Dans le graphe de la figure 1, 8,

— E-A-C-B est une chaîne de longueur 3.

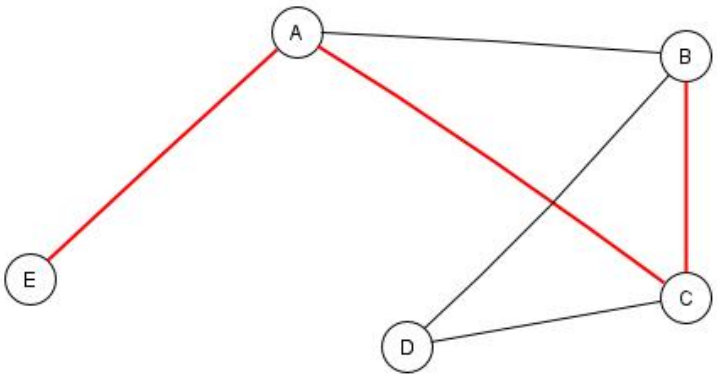

FIGURE 1.8 – Une chaîne de longueur 3

• **Cycle :** Dans un graphe non orienté, un cycle est une suite d'arêtes consécutives (chaîne simple) dont les deux sommets éxtrémités sont identiques. On dira que le cycle a pour longeur K lorsque le nombre d'arêtes du cycle est K. Dans le graphe de la figure $1,8$, ABCA est un cycle de longueur 3.

Chemins et circuits :

• **Chemin :** Un chemin d'origine x et d'éxtrémité y est défini par une suite finie d'arcs consécutifs reliant x à y.

Un chemin *simple* est un chemin ne passant pas deux fois par un même arc. Une chemin *élémentaire* est un chemin qui passe une et une seul fois par ses sommets. Dans le graphe de la figure $1,9$, il existe un chemin élémentaire $\langle 1,4,2,5\rangle$. Par exemple, $\langle 3,6,6,6\rangle$ est un chemin non élémentaire.

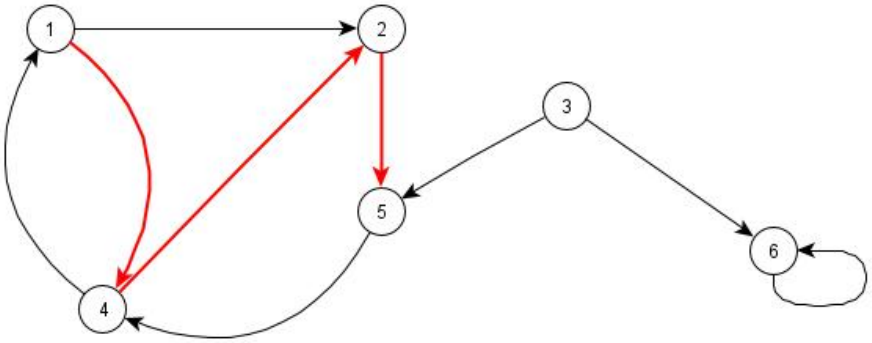

FIGURE 1.9 – Chemin élémentaire

• **Circuit :** Un circuit est un chemin dont les deux éxtrémités sont identiques. Dans l'exemple précédent, $\langle 1, 2, 5, 4, 1\rangle$ est un circuit élémentaire et $\langle 1, 2, 5, 4, 2, 5, 4, 1\rangle$ est un circuit non élémentaire.

1.3 La notion de connexité dans les graphes

1.3.1 Graphe connexe

On dit que le graphe G est connexe si et seulement s'il existe une chaîne entre n'importe quelle paire de sommets distincts du graphe. Par exemple, le graphe de la figure $1, 10$ est connexe.

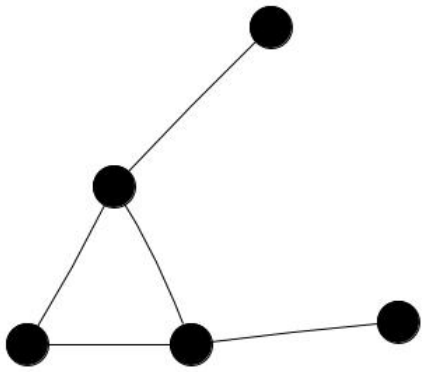

FIGURE 1.10 – Un graphe connexe

1.3.2 Composante connexe

Une composante connexe est un sous graphe induit maximal connexe, c'est à dire un ensemble de sommets qui ont deux à deux la relation de connexité. Le graphe de la figure 1, 11 est composé de deux composantes connexes : la premiére est le sous-graphe défini par les sommets {a, b, c, d} et la seconde est le sous-graphe défini par les sommets {e, f, g}.

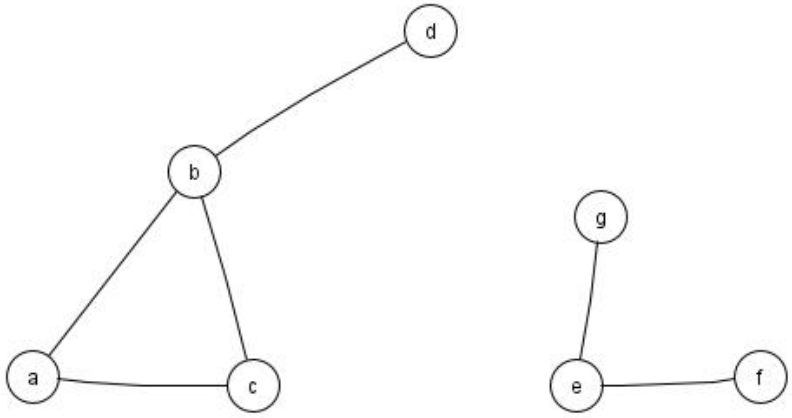

FIGURE 1.11 – composantes connexes d'un graphe

1.3.3 Graphe fortement connexe

Un graphe orienté $G = (V, E)$ est dit fortement connexe si $\forall x, y \in V$, il existe un chemin de x à y et un autre chemin de y à x.

1.3.4 Composante fortement connexe

Une composante fortement connexe est un ensemble de sommets qui ont deux à deux la relation de forte connexité, de plus, tout sommet en dehors de la composante n'a pas de relation de forte connexité avec aucun élément de cet composante.

1.4 Représentation matricielle des graphes

1.4.1 Matrice d'adjacence

Soit un graphe G d'ordre n dont on numérote les sommets de 1 à n. On appelle matrice d'adjacence de G la matrice carrée d'ordre n dont chaque terme $a_{i,j}$ est égal au nombre d'arêtes reliant les sommets i et j ($1 \leq i \leq n, 1 \leq j \leq n$).

Remarque : Dans le cas d'un graphe G non orienté, la matrice d'adjacence de G est symétrique par rapport à une diagonale. Dans le cas d'un graphe simple, elle est booléenne, elle ne comporte que des 1 et des 0. La matrice suivante est une matrice d'adjacence d'un graphe non orienté qui est représenté dans la figure 1, 12 :

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

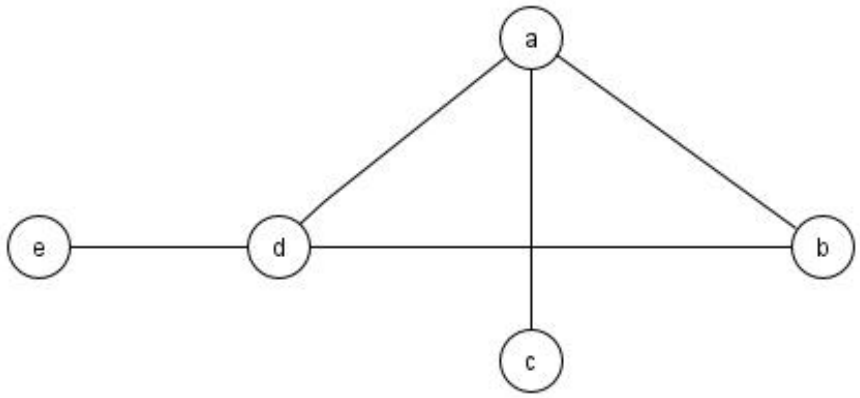

FIGURE 1.12 – Graphe non orienté à 5 sommets

La matrice suivante est une matrice d'adjacence de graphe orienté de la figure 1, 13 :

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

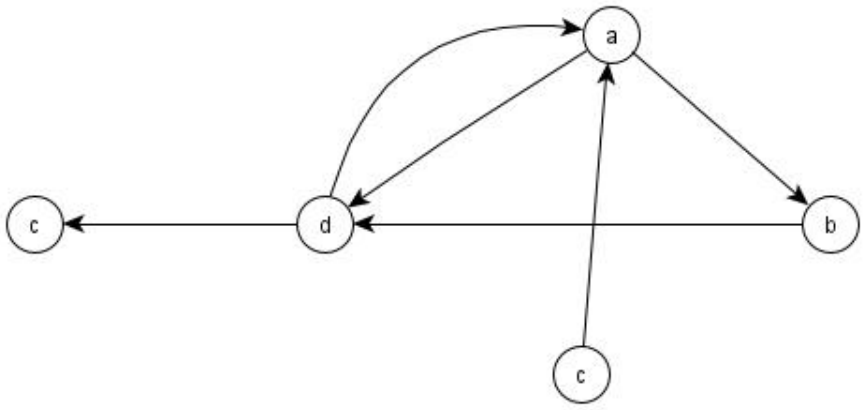

FIGURE 1.13 – Graphe orienté à 5 sommets

Propriété : Soit A la matrice d'adjacence d'un graphe G d'ordre n. On suppose que les sommets de G sont numérotés de 1 à n. Le terme $a_{i,j}$ de la matrice A^k ou k est un entier naturel non nul est le nombre de chaînes de longueur k reliant le sommet i au sommet j.

1.4.2 Matrice d'incidence

La matrice d'incidence est une matrice $n * p$ où n est le nombre de sommets du graphe et p est le nombre de liens (arêtes ou arcs). Soit un graphe orienté sans boucle G=(V,E) avec $|V| = n$ et $|E| = m$: On appelle matrice d'incidence (aux arcs) de G la matrice $M = m_{ij}$ de dimension $(n \times m)$ telle que :

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

La matrice suivante est une matrice d'incidence de graphe non orienté de la figure 1, 14 :

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

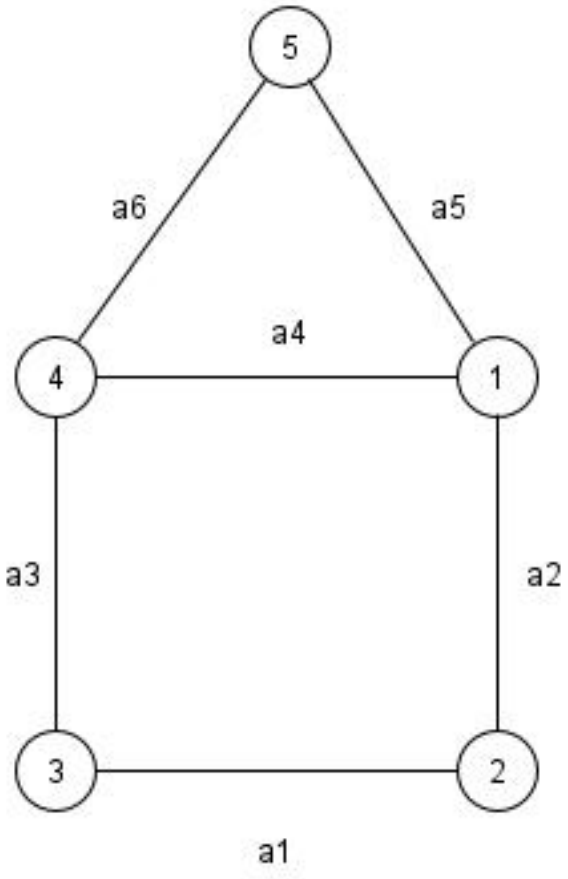

FIGURE 1.14 – Graphe non orienté à 5 sommets et 6 arêtes

1.5 Opérations sur les graphes

1.5.1 Somme Cartésienne de deux graphes :

Soient $G = (V, E)$ et $H = (Y, F)$ deux graphes, la somme cartésienne de G et H, notée $G \diamond H$ est le graphe défini sur l'ensemble de sommets $V(G) * Y(H)$ tel que deux sommets (u,u') et (v,v') sont adjacents si et seulement si $(u = v)$ et $(u'v' \in F(H))$. La figure $1,15$ montre la somme cartésienne de G et H :

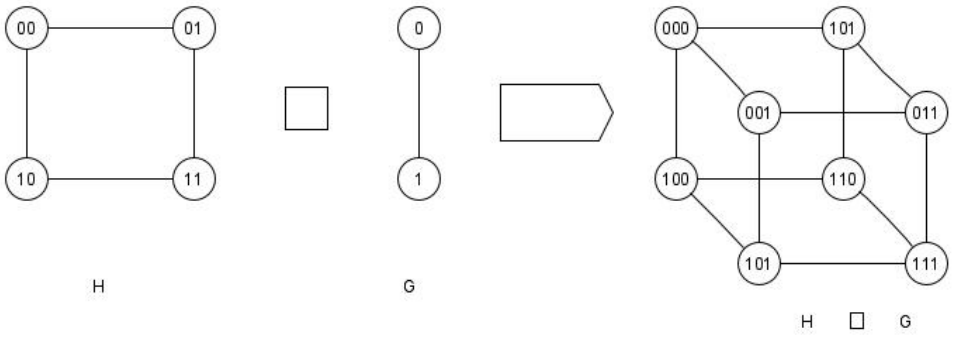

FIGURE 1.15 – Somme cartésienne de G et H

1.5.2 Produit Cartésien

Soient $G = (V(G), E(G))$ et $H = (V(H), E(H))$ deux graphes, on définit le graphe $G \otimes H = (S, T)$ appelé produit cartésien de G et H, tel que $S = V(G) * V(H)$ et où deux sommets sont adjacents si et seulement si $uv \in E$ et $u'v' \in E'$. La figure $1,16$ montre le produit cartésien de G_1 et G_2. Il est à noter que le nombre de sommets dans $G * H$ est $|V| * |V'|$ et que le nombre d'arêtes est $|V| * |E'| + |V'| * |E|$.

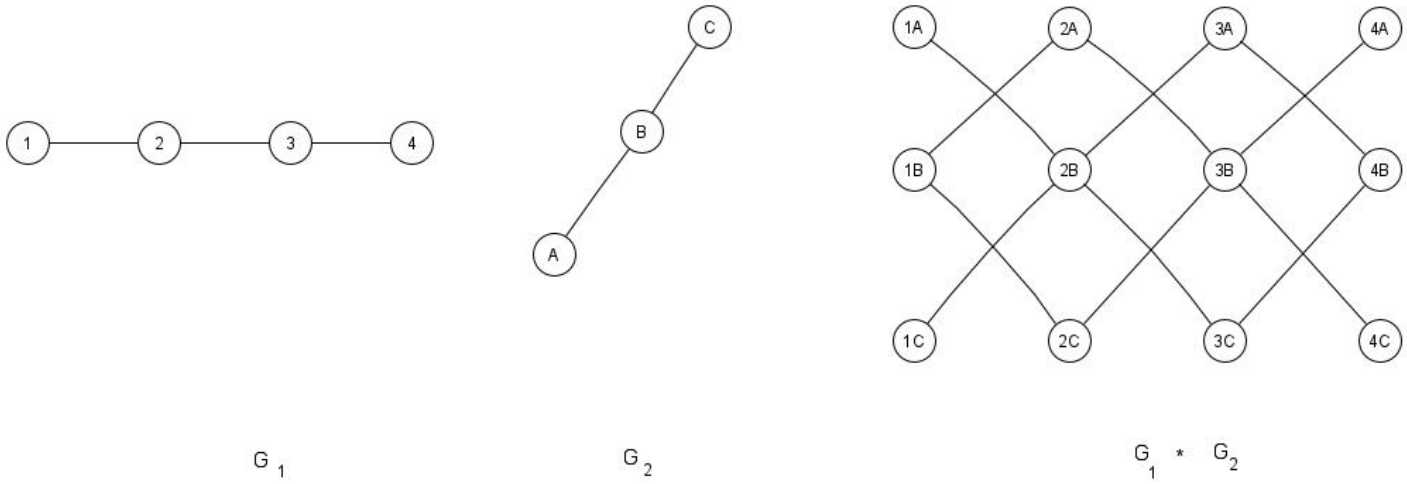

FIGURE 1.16 – Le produit cartésien de $G_1 \otimes G_2$

1.5.3 Subdivision d'un graphe

Une subdivision d'un graphe G (parfois appelée expansion de graphe) est le graphe résultant de la subdivision d'arête uv qui est conduit à un graphe contenant un nouveau sommet w et ou l'on a remplacé l'arête uv par deux nouvelles arêtes uw, wv.

1.5.4 Isomorphisme

Deux graphes $G = (V, E)$ et $G' = (V', E')$ sont dit isomorphes si et seulement s'il existe une application bijective $\wp : V \to V'$ qui vérifie la condition suivante : $xy \in E \leftrightarrow \wp(x)\wp(y) \in E'$ pour toute paire de sommets x, y dans V.

1.5.5 Homomorphisme

Soient $G = (V(G), E(G))$ et $H = (V(H), E(H))$ deux graphes. Un homomorphisme de G dans H est une application : $F : V(G) \to V(H)$ tel que : $xy \in E \to F(x)F(y) \in E(H)$. La figure $1,17$ montre un homomorphisme de graphe G et H :

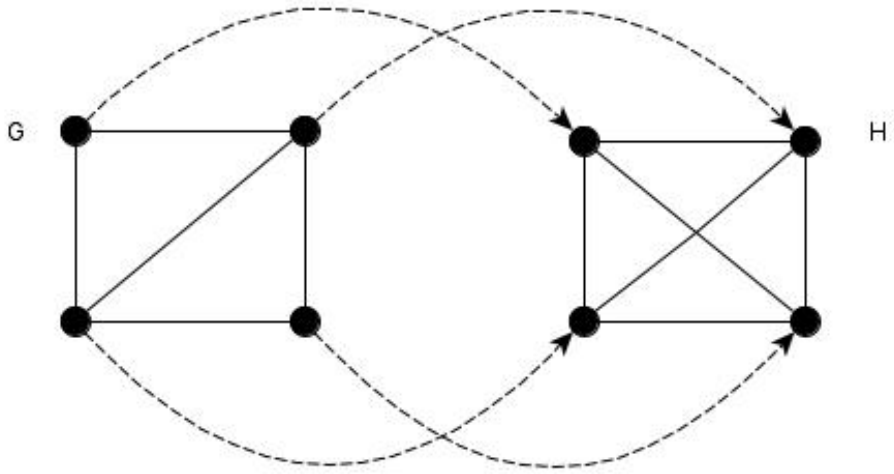

FIGURE 1.17 – Homomorphisme

1.6 Quelques Graphes particuliers

1.6.1 Graphe complet

Un graphe complet est un graphe simple dont tous les sommets sont deux à deux adjacents . Si le graphe est orienté, on dit qu'il est complet si chaque paire de sommet est relié par exactement deux arcs. Un graphe complet à n sommets est noté K_n et contient $\frac{n*(n-1)}{2}$ arêtes. La figure 1, 18 nous illustre le graphe K_4 :

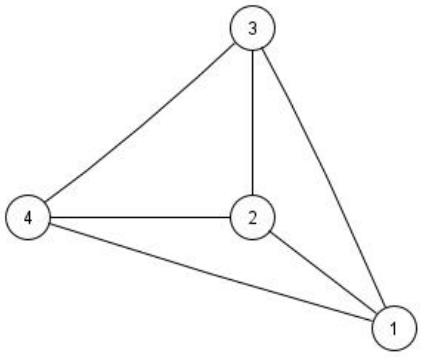

FIGURE 1.18 – Graphe K_4

1.6.2 Graphe biparti

Un graphe G est dit biparti si on peut partitionner son ensemble de sommets en deux parties A et B tel qu'il n'y ait aucune arête entre les éléments de A et aucune arête entre les éléments de B. La figure 1, 19 montre un graphe biparti :

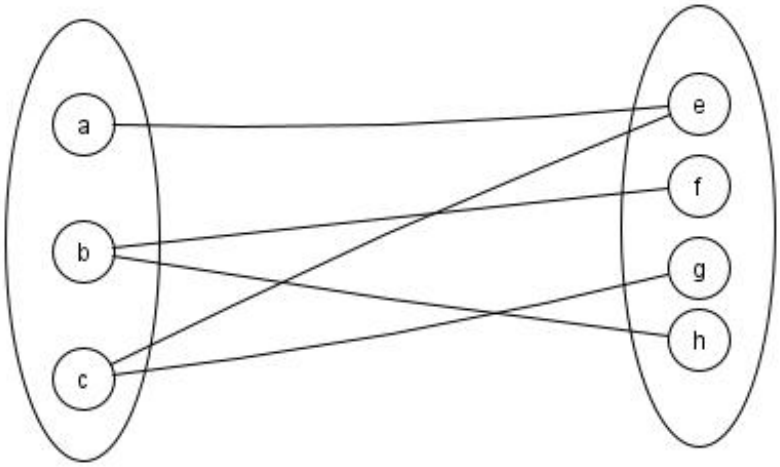

FIGURE 1.19 – Un graphe biparti

1.6.3 Graphe biparti complet

Un graphe G est dit biparti complet si chaque sommet de A est adjacent à un sommet de B. On note ce graphe par $K_{p,q}$ ou $p = |A|$ et $q = |B|$. Le graphe $K_{p,q}$ a $p + q$ sommets et $p * q$ arêtes.

1.6.4 Graphe biparti équilibré

Un graphe G est dit équilibré si et seulement si :

— Chaque sous-ensemble de la bipartition contient le même nombre de sommets. La figure 1, 20 montre un graphe biparti équilibré :

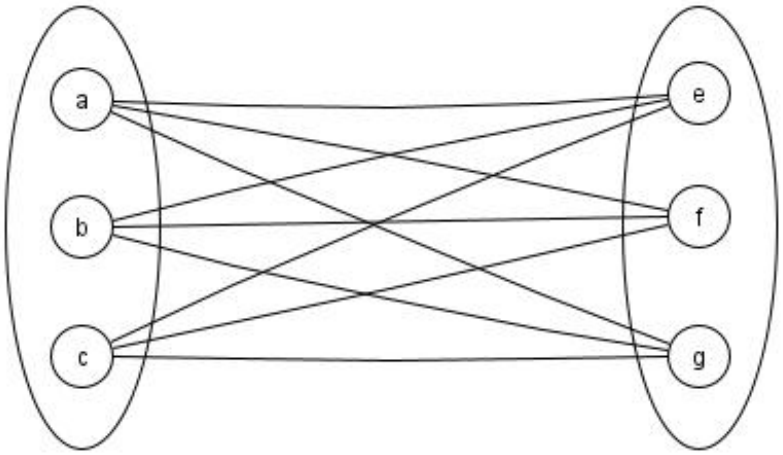

FIGURE 1.20 – Un graphe biparti équilibré

1.6.5 Graphe planaire

Un graphe planaire est un graphe qui a la particularité de pouvoir se représenter sur un plan tel que ses arêtes ne se rencontrent pas en dehors de leurs éxtrémités. La figure $1,21$ représente un graphe planaire :

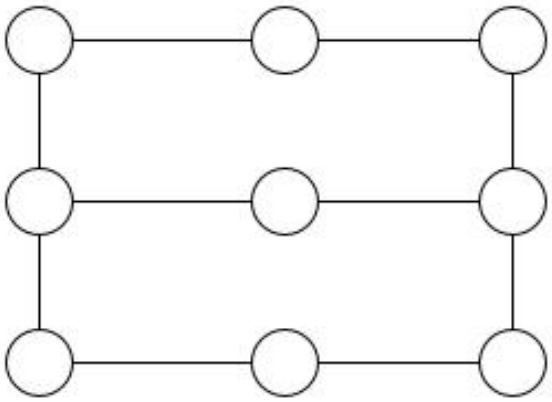

FIGURE 1.21 – Un graphe planaire

1.6.6 Graphe régulier

Un graphe G est dit régulier si tous ses sommets ont le même degré. La figure 1, 22 représente un graphe régulier.

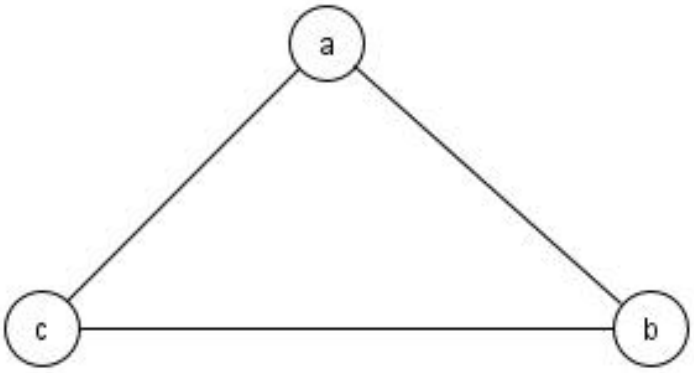

FIGURE 1.22 – Un graphe régulier

Ainsi, on doit définir les notions suivantes :

• **Distances dans les graphes :**

Soient deux sommets u et v d'un graphe $G = (V(G), E(G))$, on appelle distance entre u et v, la longueur d'une plus courte (u,v)- chaîne (en terme de nombre d'arêtes) qui les relies et on la note $d_G(u, v)$ (ou $d(u, v)$ s'il n'y a pas de confusion). Une telle chaîne s'appelle géodésique (La ligne géodésique est la ligne la plus courte entre deux points d'une surface).

- L'excentricité d'un sommet u noté $e_G(u)$ (ou e(u) s'il n'y a pas de confusion) est le nombre suivant : $e_G(u) = \max_{u \in V(G)} d_G(u, v)$.

-Le diamètre de G noté $D(G)$ (ou D s'il n'y pas de confusion) est la plus grande excentricité : $D(G) = \max_{u \in V(G)}[e(u)]$. Un sommet $u \in V(G)$ est dit diamétral si $d_G(u, v) = D(G)$. Si chaque sommet de G admet un unique sommet diamétral, on dira que G est diamétral.

-Le rayon de G noté $R(G)$ (ou R s'il n'y a pas de confusion) est la plus petite excentricité : $R(G) = \min_{u \in V(G)}[e(u)]$.

-Le centre de G est l'ensemble des sommets de G dont l'excentricité est égal au rayon.

• **Intervalles :**[5] L'intervalle $I(G)(u,v)$ est l'ensemble des sommets de G appartenant aux chaînes géodésiques entre u et v.

$I(u,v) = \{w \in V$, w appartient à une chaîne géodésique entre u et v$\}$. Trivialement, un sommet $w \in I(u,v)$ si et seulement si : $d(u,w) + d(w,v) = d(u,v)$.

Proposition 1 :[5] Soient u et v deux sommets d'un graphe 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 $w \in I(u,v)$, alors $I(u,w) \cap I(w,v) = \{w\}$.
- Si $w \in I(u,v)$ et $z \in I(u,w)$, alors $w \in I(z,v)$.

Proposition 2 :[5] Pour tout triplet u, v, w d'un graphe G, il existe un sommet z dans $I(u,w) \cap I(u,v)$ tel que : $I(z,w) \cap I(z,v) = \{z\}$.

Proposition 3 :[5] Soient u, v, w et z quatres sommets d'un graphe G, z est l'unique sommet de $I(u,w) \cap I(u,v)$ tel que $I(z,w) \cap I(z,v) = \{z\}$ si et seulement si : $I(u,w) \cap I(u,v) = I(u,z)$.

1.6.7 Graphe distance monotone

Un graphe connexe est dit distance monotone si, pour tout intervalle $I(x,y)$ et tout sommet z hors de l'intervalle, il existe un sommet z' dans $I(x,y)$ tel que $d(z,z') > d(x,y)$.[3]

Il est clair qu'un intervalle $I(x,y)$ est tel que pour tout z, z' dans l'intervalle , $d(z,z') \leq d(x,y)$. Dans un graphe distance monotone, la propriété de la définition d'un graphe distance monotone est donc caractéristique des sommets de l'intervalle.

1.6.8 Graphe sphérique

Un graphe $G = (V(G), E(G))$ est dit sphérique (resp. hypersphérique) si dans tout intervalle $I_G(u,v)$ et pour tout sommet $w \in I_G(u,v)$, il existe un unique (resp. au moins un sommet $\overline{w} \in I(G)(u,v)$ tel que $d_G(w,\overline{w}) = d_G(u,v)$, $\overline{w}$ sera appelé antipodal de w dans l'intervalle $I_G(u,v)$.

1.6.9 Graphe de Hamming

Soient $x_1, x_2, ..., x_n$ des entiers positifs, le graphe de Hamming $H_{x_1,x_2,...,x_n}$ est le graphe dont les sommets sont les éléments de la somme cartésienne d'ensemble de x_i éléments et où deux sommets sont adjacents si et seulement si les vecteurs correspondants diffèrent en exactement une composante.

1.6.10 Arbre

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.[3]

- 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.
- Un graphe vérifiant l'une des propriétés ci-dessus caractérise un arbre

La figure 1, 23 montre un arbre :

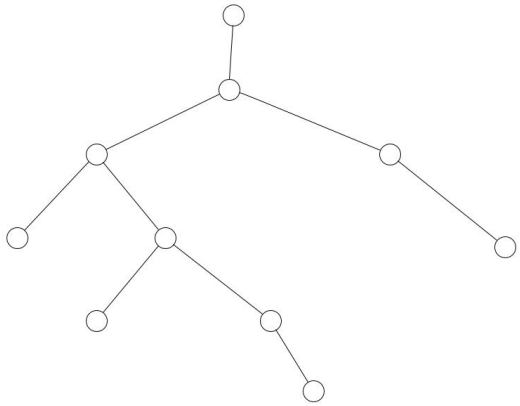

FIGURE 1.23 – Un arbre

Chapitre 2

Hypercube et quelques caractérisation

2.1 Hypercube

Définition

L'hypercube de dimension n, noté Q_n est le graphe de 2^n sommets qui peuvent être considérés comme étant tous les vecteurs booléens sur $\{0,1\}^n$, et où deux sommets sont adjacents si et seulement si les vecteurs associés à ces sommets diffèrent exactement en une seule composante. La figure $2,1$ montre les hypercubes de petites dimensions :

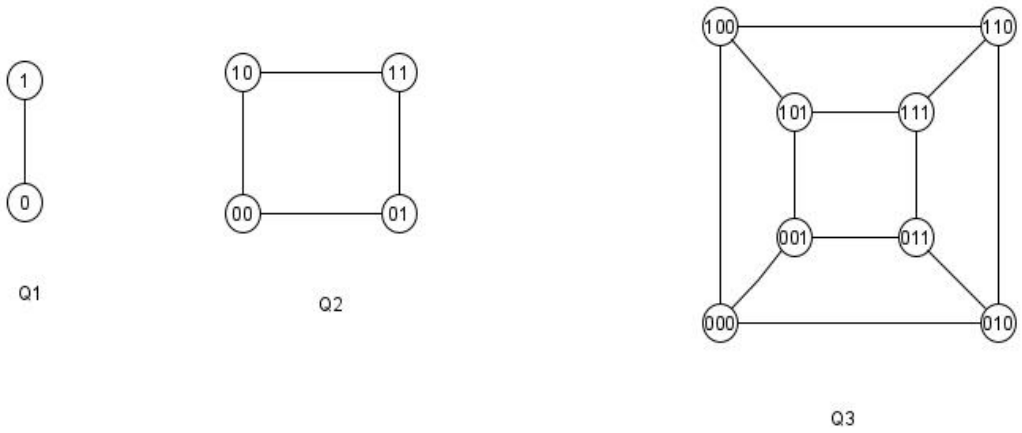

FIGURE 2.1 – Hypercube Q_1, Q_2 et Q_3

Notons que $Q_0 = K_1, Q_1 = K_2$ et que d'une manière générale, Q_n peut être défini récursivement en utilisant la somme cartésienne par $Q_{n+1} = Q_n \diamond K_2$. Il est donc clair que $Q_n, (n \geq 1)$ est isomorphe à : $K_2 \diamond K_2 ..... \diamond K_2$ (n fois).

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.

La figure $2, 2$ montre une décomposition de Q_3 en deux copies de Q_2 suivant la direction 1 (les arêtes en gras) :

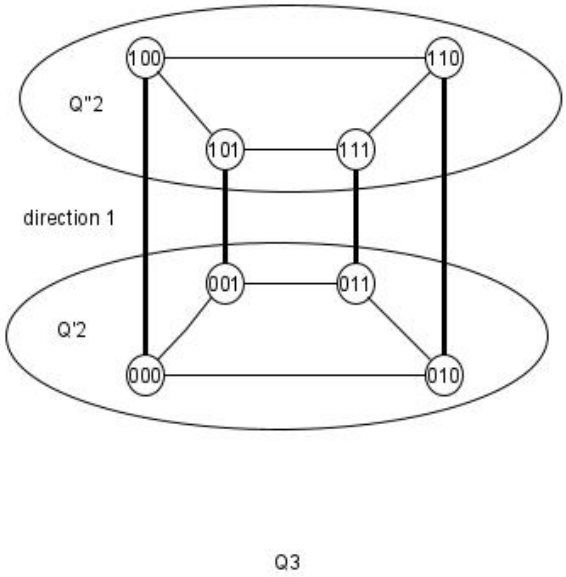

FIGURE 2.2 – Décomposition canonique de Q_3

Un cycle dans un hypercube Q_3 constitué de deux hypercubes Q_2. En rouge, un chemin dans le premier hypercube Q_2. En bleu les arêtes le reliant à sa copie et en vert le chemin dans la copie. Le cycle est de longueur 2+2+2=6, qui est pair. Comme le montre le graphe de la figure $2,3$:

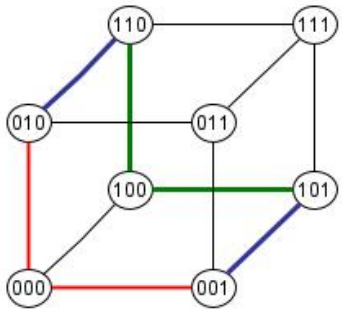

FIGURE 2.3 – Un cycle dans l'hypercube Q_3

2.1.1 Degré

Deux sommets sont connectés s'ils diffèrent exactement sur un symbole de leurs étiquettes. Comme L'étiquette a n symboles, chaque sommet est connecté exactement à n voisins, donc, le degré de chaque sommet est n, autrement dit le graphe est n-régulier.

2.1.2 Nombre de sommets

Par la construction récursive, on voit que pour passer de Q_{n-1} à Q_n, il faut faire une copie du graphe, autrement dit, le nombre de sommets est doublé. Si V_n est le nombre de sommets du graphe Q_n, on obtient ainsi $V_n = 2 * V_{n-1}$, et le premier cas est $V_1 = 2$, en déroulant la récurrence, on obtient $V_n = 2 * V_{n-1} = 2^2 * V_{n-2} = 2^n * V_0$, c'est à dire que le graphe a 2^n sommets.

2.2 Caractérisation de l'hypercube

On a plusieurs caractérisations :

•**Théorème (Foldes)** : Un graphe connexe $G = (V, E)$ est un hypercube si et seulement s'il vérifie les conditions suivantes :[4]

— G est biparti

— Pour tout couple de sommets u et v de G, le nombre de plus courtes (u,v)-chaîne est $d(u, v)$

•**Théorème (Mulder)** : Un graphe connexe G est un $(0, 2)$-graphe si et seulement si pour tout couple de sommets (u,v) de G, il existe soit exactement deux chaînes de longueur 2 reliant u à v, soit aucune chaîne de longueur 2 reliant u à v : Mulder[10] a donné les propositions suivantes sur les (0,2) graphes :[10]

•**Proposition(Mulder)** : Si G est un (0,2)- graphe, alors G est régulier.[10]

•**Proposition(Mulder)** : Si G est un $(0, 2)$-graphe de degré n alors $|V(G)| \leq 2^n$.[10]

•**Théorème (Mulder, Laborde, Rao)** : Soit $G = (V, E)$ un $(0, 2)$- graphe, alors :[6]

— G est régulier de degré n.

— $|V(G)| \leq 2^n$.

— $|V(G)| \leq 2^n$ si et seulement si G est un hypercube de dimension n.

Une autre caractérisation de l'hypercube est en terme d'intervalle donné par (Mulder et Bendelt).

•**Théorème(Mulder,Laborde, Rao)** : Soit $G = (V, E)$ un $(0, 2)$- graphe, alors :[6]

— G est régulier de degré n.

— $|V(G)| \leq 2^n$.

— $|V(G)| = 2^n$, si seulement si G est un hypercube de dimension n.

• **Théorème (Bandelt et Mulder)** : Soit G un graphe biparti connexe, les conditons suivantes sont équivalentes :[7]

— G est un hypercube.

— tout intervalle dans G engendre un hypercube.

— tout intervalle dans G engendre $(0, 2)$ -graphe .

— tout intervalle $I(u, v)$ dans G contient exactement $2^{d(u,v)}$ sommets.

— tout intervalle $I(u, v)$ dans G engendre un graphe avec exactement $d(u, v).2^{d(u,v)-1}$ arêtes.

•**Théorème (Mulder)** : Un graphe G connexe est un hypercube si et seulement si G est un graphe médian régulier [5].

Proposition (Mulder) :[10] L'hypercube est hamiltonien, de plus par toute arête passe un cycle hamiltonien. D'autres caractérisations de l'hypercube en terme d'intervalles ont été données, en particulier des caractérisation en terme de graphes distance monotones, en terme de graphes intervalles réguliers et des graphes sphériques peuvent être trouvés.

2.3 Projection et anti-projection

Une projection (resp. anti projection) d'un sommet u d'un graphe G sur un ensemble de sommets S et 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'esemble des projections(resp. antiprojection) de u sur S.

on considère les propriétés suivantes :

-P1 :Pour tout u, v et w , $|P(u, I(v, w))| = 1$.

-P2 :Pour tout u,v et w, $|AP(u, I(u, w))| = 1$.

Un graphe vérifiant l'une de ces deux propriétés est un graphe biparti.

2.4 Plongement

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 uv de G, une chaîne reliant les sommets $\phi(u)$ et $\phi(v)$ dans H.

2.4.1 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.

— **Dilatation :** La dilatation d'un plongement ϕ d'un graphe G dans un graphe H, notée $dil(\phi)$, est la longueur maximale des chaîne $P_\phi(x, y)$ de H ,associées aux arêtes (x, y) de G, Dans le cas où l'on considère des chaîne de plus courte longueur, la longueur de $P_\phi(x, y)$ est alors égale à la distance $d_H(\phi(x), \phi(y))$, et la dilatation s'exprime uniquement en fonction de ϕ ,par : $dil(\phi) = \max_{x,y \in E(G)} d_H(\phi(x), \phi(y))$. 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 (x,y) de G est l'arête $(\phi(x), \phi(y))$ de H. Si de plus $|V(G)| = |V(H)|$,alors G est un graphe partiel de H.

— **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.

— **Congestion :** La congestion d'un plongement ϕ d'un graphe G dans un graphe H, notée cong(ϕ), est le maximum, pris sur toutes les arêtes $''e''$ de H, du nombre de chaînes $P_\phi(u, v)$

de H images d'arêtes de G qui contiennent $''e''$.

2.4.2 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 optimale pour G. La seule borne inférieure comme pour la dilatation valable pour tous les graphes, si l'expansion du plongement vaut 1 (l'application ϕ est alors bijective), est donnée par la proposition suivante :

proposition

Si l'expansion d'un plongement ϕ d'un graphe G dans un graphe H vaut 1, alors la dilation de ϕ est au moins égale au rapport du diamètre de H sur le diamètre de G.[8]

2.5 Graphes et dimensions 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 appelé *dimension cubique*, noté $cd(G)$.

Firsov [10] a remarqué que tous les arbres sont des graphes cubiques. Il a aussi montré que tout graphe cubique est nécessairement biparti, mais la réciproque n'est pas vrai en général. Un exemple de graphe biparti est le graphe $K_{2,3}$ ci-dessous :

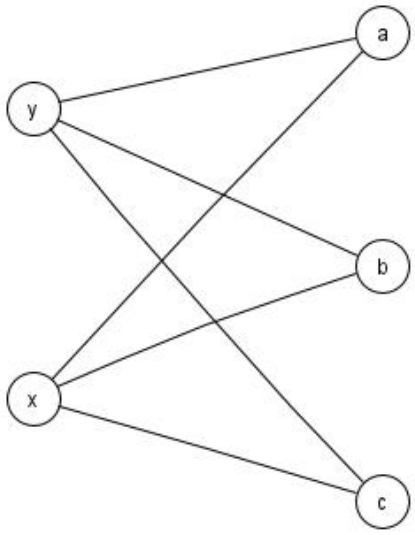

FIGURE 2.4 – Graphe $K_{2,3}$

$K_{2,3}$ 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 x et y sont à distance 2 dans $K_{2,3}$, alors leurs images respectives $p = \varphi(x)$ et $q = \varphi(y)$ seront aussi à distance 2 dans Q_n. Or, deux sommets à distance 2 dans Q_n appartiennent à exactement 2 chaînes sommets - disjointes de longeur 2 dans Q_n, qui n'est pas possible car les 3 chaînes sommets - disjointes de longeur 2 dans $K_{2,3}$ doivent se plonger dans 3 chaînes sommets - disjointes dans Q_n.

2.6 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 seulment 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 petite 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.

2.6.1 Condition nécessaires de plongement d'un graphe G dans Q_n

Si un graphe G=(V,E) est plongeable dans le graphe Q_n , alors :

- $|V(G)| \leq 2^n$.
- G est biparti.
- Le degré maximum de G : $\Delta(G) \leq n$.
- Ces conditions sont nécessaires mais pas suffisantes. Comme le montre le graphe de la figure $2,5$:

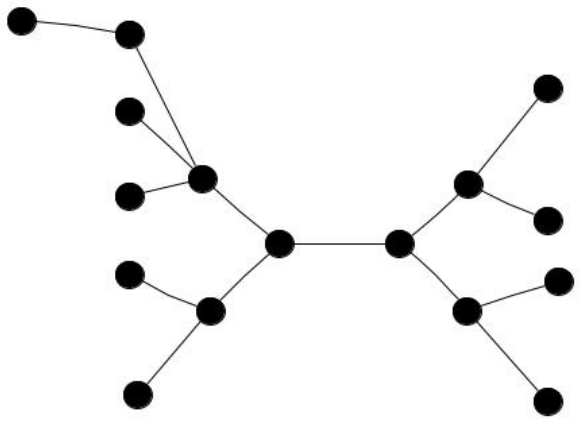

FIGURE 2.5 – Arbre équilibre 2^4 sommets non plongeable dans Q_4

2.7 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.

2.7.1 Théorème

Un Graphe G est plongeable dans Q_n si et seulement s'il existe un C_n-valuation de G.[10]

Chapitre 3

Plongement de certaines classes d'arbres dans l'hypercube

3.1 Quelques types d'arbres

Soit T un arbre d'ordre n. L'hypercube $Q_{dim(T)}$ est appelé hypercube optimal de T. Comme tous les arbres sont cubique. Donc on s'intéressera á la recherche de la dimension cubique de ces arbres(plongement optimal de ces arbres dans l'hypercube).

Définition : Un arbre T est dit binaire si son degré maximum inférieur ou égal à 3. [11]

3.1.1 Arbres binaires complets :

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 obtenue à partir de deux copies disjointes T_1, T_2 de D_{n-1} (D'_{n-1} et D''_{n-1}) est d'un nouveau sommet w, tel que w est relié par une arète à un sommet de degré 2 de T_1 et par une autre arête à un sommet de degré 2 de T_2

L'arbre binaire complet D_n possède 2^n sommets pendants, $2^n - 2$ sommets de degré 3 et un seul sommet de degré 2. Le sommet de degré 2 sera appelé la racine de D_n. Donc D_n est un arbre de hauteur n qui possède $2^{n+1} - 1$ sommets. D_2 est montré dans la figure $3,1$:

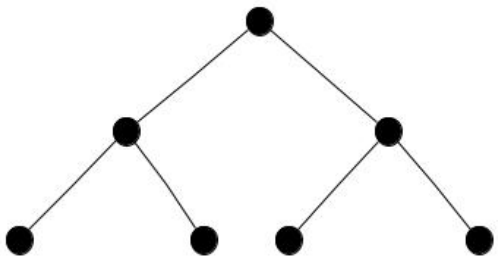

FIGURE 3.1 – Arbre binaire complet D_2

Proposition

L'arbre binaire complet de hauteur n, D_n ne peut se plonger avec dilatation 1 dans son hypercube optimal Q_{n+1}, pour $n \geq 2$.[11]

Preuve : Les deux graphes sont bipartis. Le graphe D_n étant connexe, un plongement de dilatation 1 doit envoyer chaque partie de la bipartition de D_n dans une classe de la bipartition de Q_n. Or, l'hypercube a le même nombre de sommets dans les deux classes de la bipartition, alors que le nombre de sommets de l'arbre D_n dans une classe de la bipartition est strictement supérieur à 2^n, si $n > 1$.

3.1.2 Arbres binaires obtenues par transformation des arbres binaires complets D_n

- **L'arbre** B_n

Pour $n \geq 2$, B_n est un arbre binaire obtenue à partir de l'arbre binaire complet D_{n-1} et d'un sommet r', tel que r' soit relié à r la racine de D_{n-1} par une arête. B_n possède $2^{n-1} + 1$ sommets de degré 1 et $2^{n-1} - 1$ sommets de degré 3. Donc B_n possède 2^n sommets.

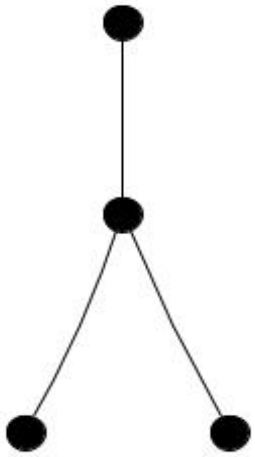

FIGURE 3.2 – L'arbre B_2

B_n n'est pas équilibré. Havel.I[10] a montré que B_n est plongeable dans Q_{n+1}, donc D_n est plongeable dans Q_{n+2}.

Théorème (Havel.I [10]) : Pour tout $n \geq 2$, on a B_n est plongeable dans Q_{n+1}, $dim(B_n) = n + 1$.

• **L'arbre** $\widehat{\widehat{D_n}}$

Pour $n \geq 1$, on définit $\widehat{\widehat{D_n}}$ l'arbre obtenue à partir de deux copies disjointes de D_n tel que leurs racines sont reliées par une arête axiale. L'arbre $\widehat{\widehat{D_n}}$ a $2^{n+2} - 2$ sommets. La figure $3,3$ représente l'arbre $\widehat{\widehat{D_2}}$

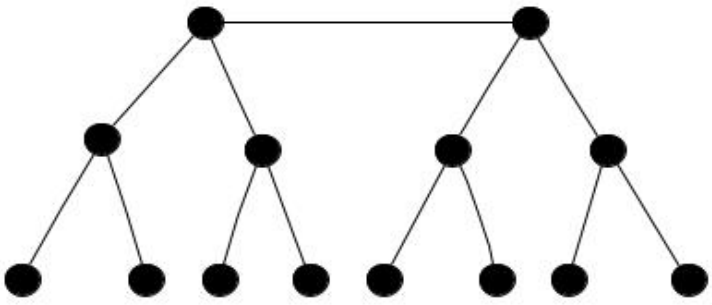

FIGURE 3.3 – L'arbre $\widehat{\widehat{D_2}}$

Proposition (Havel. I [10]) : L'arbre $\widehat{\widehat{D_n}}$ est plongeable dans Q_{n+2}, $dim(\widehat{\widehat{D_n}}) = n + 2$, pour tout $n \geq 1$.

3.1.3 Classes d'arbres obtenus par subdivision de $\widehat{\widehat{D_n}}$

Type (A)

Pour tout $n \geq 1$, $0 \leq k \leq n$, l'arbre A_n^k est obtenue par une double subdivision de l'arète de niveau k dans D'_n. A_2^0 et A_2^2 sont montrés dans la figure suivante :[11]

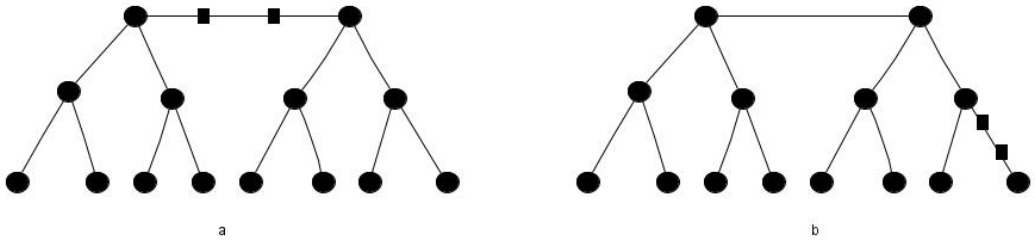

FIGURE 3.4 – Arbres binaires $a(A_2^0)$ et $b(A_2^2)$

Type (B)

Pour tout $n \geq k \geq 1$, l'arbre B_n^k est obtenue à partir de l'arbre $\widehat{\widehat{D_n}}$, en subdivisant une seule arête de niveau k dans chaque copie de D_n. B_2^1 et B_2^2 sont montrés dans la figure suivante :[11]

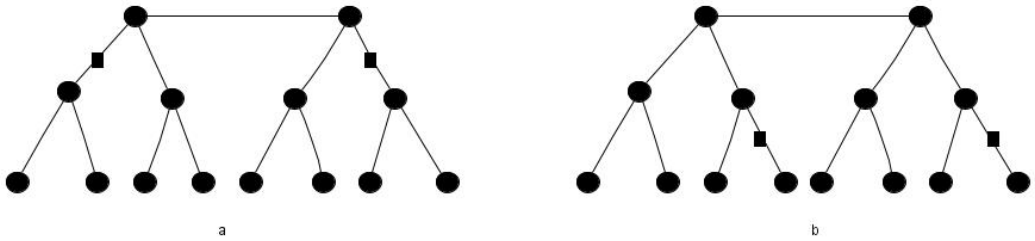

FIGURE 3.5 – Arbres binaires $a(b_1^1)$ et $b(b_2^2)$

3.1.4 Arbre de Fibonacci

Les arbres de Fibonacci sont des arbres binaires obtenus de la manière suivante :
F_0 est l'arbre réduit à un seul sommet, F_1 est la chaîne de longueur 1 (une arête) et pour $n \geq 2$, F_n est un arbre contenant une racine avec F_{n-1} pour un sous arbre gauche et F_{n-2} pour un sous arbre droit. F_0, F_1, F_2 et F_3 sont donnés dans la figure suivante :

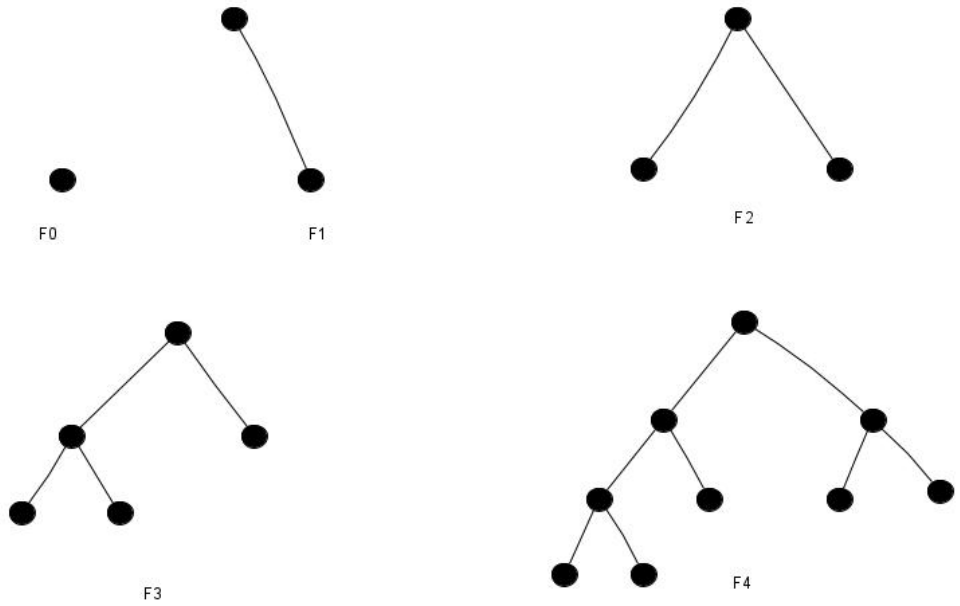

FIGURE 3.6 – Arbres de Fibonacci

Il est clair q'un arbre de Fibonacci F_h est un arbre à hauteur équilibré T_{Ph}, donc pour tout $h \geq 0$, l'arbre de Fibonacci F_h est plongeable dans Q_{h+1}.

Chapitre 4

Nouvelle classes d'arbre plongeable dans Q_n

Définition : Soient T_1 et T_2 deux arbres binaires tels que T_1 possède une chaîne de longueur p donnée par : $\mu_p = \{U_0, U_1, ..., U_i, ..., U_p\}$ et T_2 possède une arête xy. Soient v, w deux sommets de degré 2 de μ_p, on définit l'arbre $\alpha(T_1, T_2)$ par l'arbre obtenue à partir de T_1 et T_2 en reliant v à x par une arête, w à y par une autre arête et en supprimant l'arête xy. Soient T_1 et T_2 les arbres de la figure 4.1 avec T est un arbre binaire quelconque.

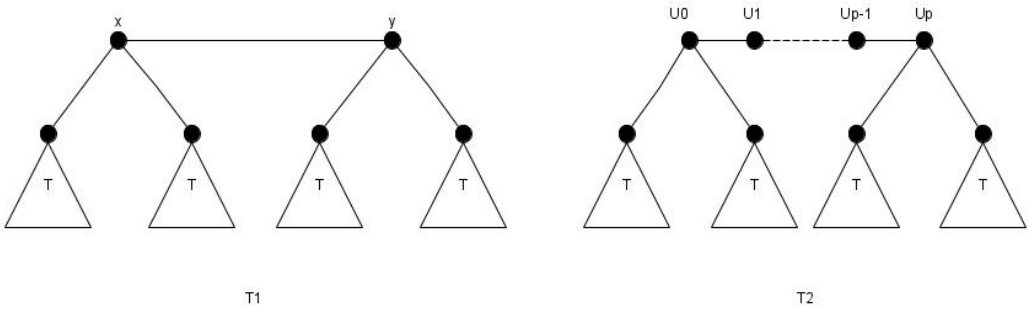

FIGURE 4.1 – Arbres binaires

alors $\alpha(T_1, T_2)$ est donné par la figure 4.2.

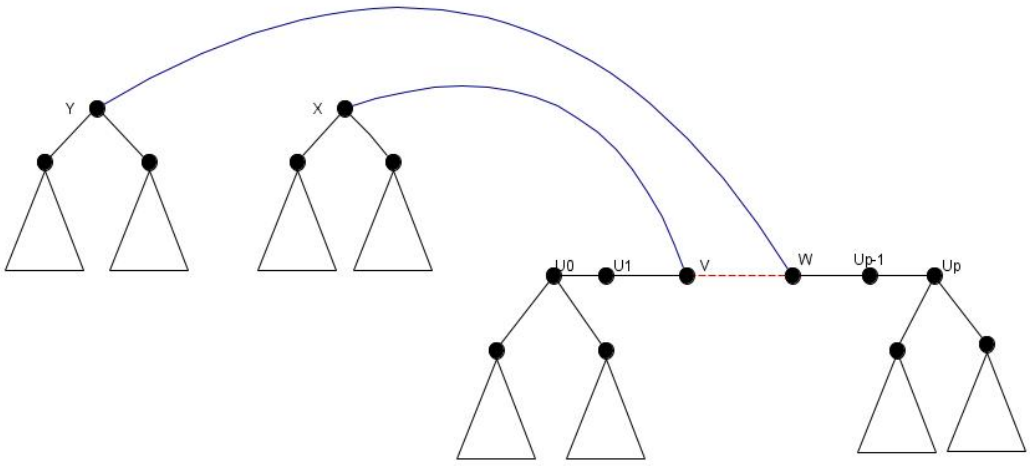

FIGURE 4.2 – Arbre binaire $\alpha(T_1, T_2)$

Théorème :

Considérons deux arbres binaires T_1, T_2 tel que T_1 possède une chaîne μ_p et T_2 possède une arête xy, si T_1 est plongeable dans Q_n et T_2 est plongeable dans Q_m, alors l'arbre binaire $\alpha(T_1, T_2)$ est plongeable dans $Q_{[\max\{n,m\}+1]}$ avec $m \neq n$. Si de plus $|V(\alpha(T_1, T_2))| > 2^{\max\{n,m\}}$ ou bien s'il existe $i \in \{1, 2\}$ tel que $|V_i(\alpha(T_1, T_2))| > 2^{\max\{n,m\}-1}$, alors la dimension cubique de $\alpha(T_1, T_2)$ est $\max\{n, m\} + 1$.

Preuve :

-Comme T_1 est plongeable dans Q_n, alors T_1 est C_n valué c'est- à- dire $\exists k \in \{1, 2, ..., n\}$, tels que pour toute chaîne de T_1, un nombre impair de ses arêtes est marqué par k.

-Comme T_2 est plongeable dans Q_m, alors T_2 est C_m valué c'est- à- dire $\exists k \in \{1, 2, ..., m\}$, tels que pour toute chaîne de T_2, un nombre impair de ses arêtes est marqué par k. Pour la valuation de l'arbre $\alpha(T_1, T_2)$, on distingue deux cas :

• Si $m > n$, alors : on va marquer les nouvelles arêtes de $\alpha(T_1, T_2)$ par $m+1$, comme $m \notin \{1, 2, ..., n\}$, alors, on a m comme valeur unique entre les arêtes de la chaîne $\{x....y\}$, donc, $\alpha(T_1, T_2)$ est C_{m+1} valué.

• Si $m < n$, alors : on remplace n par $n+1$ dans les copies disjointes de T_2 et on marque une arête de

la chaîne $\{x....y\}$ par n et les nouvelles arêtes de $\alpha(T_1, T_2)$ par $n+1$. Donc $\alpha(T_1, T_2)$ est C_{n+1} valué. Prenons l'exemple suivant :

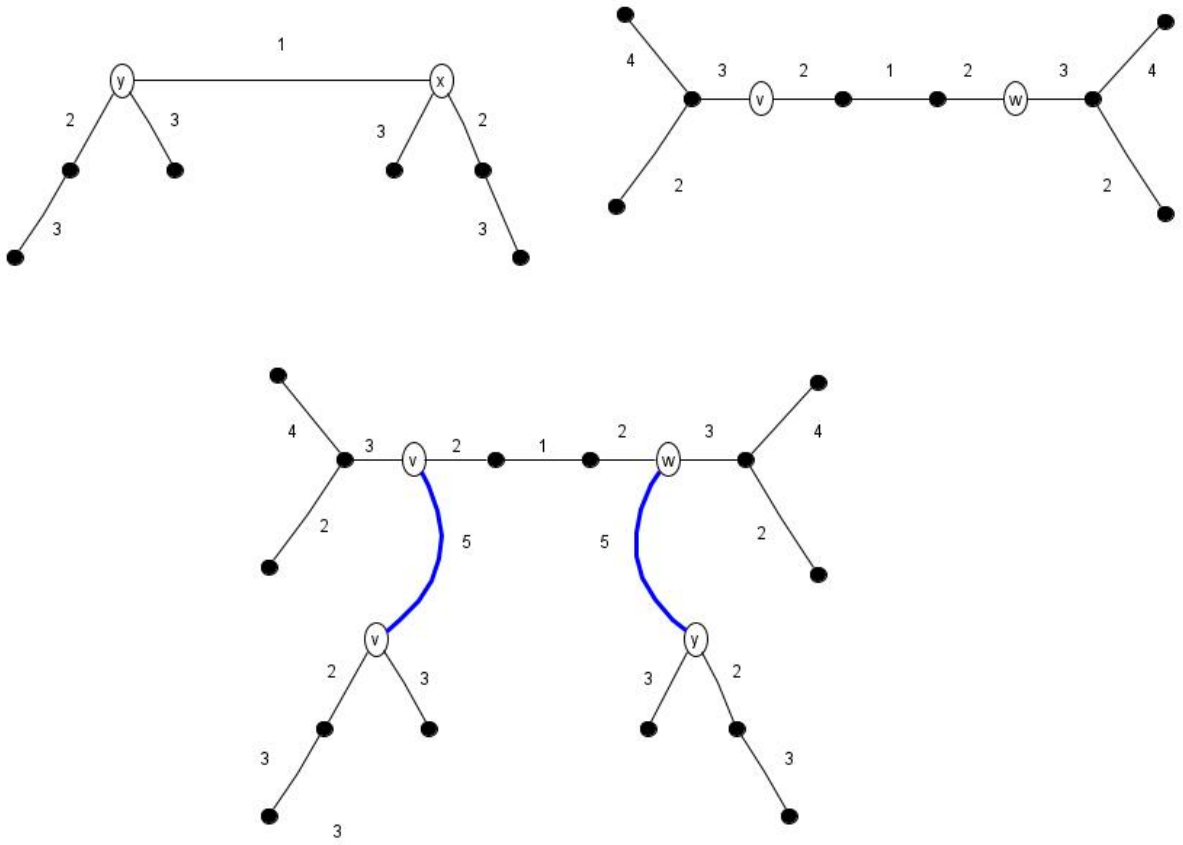

FIGURE 4.3 – Exemple

4.1 Décomposition de Q_n en des copies disjointes de Q_i

Etant donné un hypercube de dimension n noté Q_n. Comme Q_n est biparti, alors nous allons représenter les sommets de la première partie par des carrés et ceux de la deuxième partie par cercles, comme le montre le graphe de la figure 4.4, pour l'hypercube Q_3 :

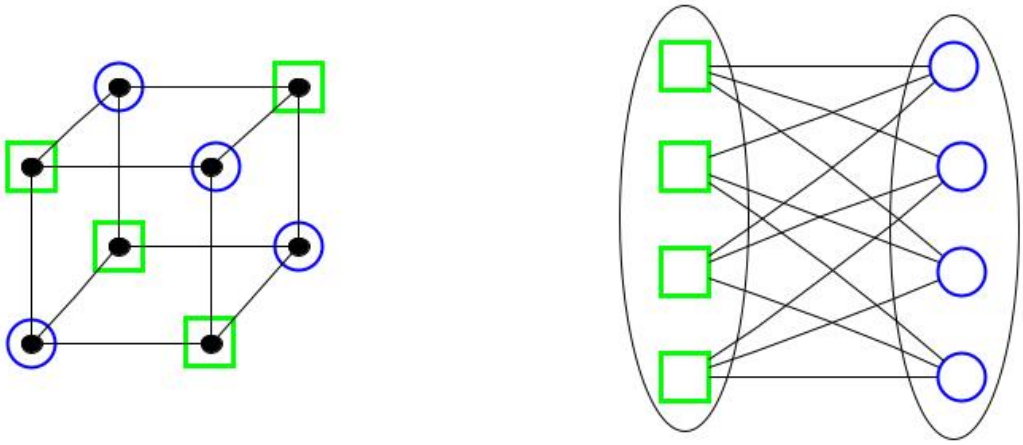

FIGURE 4.4 – Représentation des sommets de Q_3

L'hypercube Q_n peut être obtenue à partir de deux copies disjointes Q'_{n-1} et Q''_{n-1} de Q_{n-1} telles que chaque sommet de type carré (resp. cercle) de Q'_{n-1} est relié à un sommet de type cercle (resp. carré) de Q''_{n-1}, comme le montre le graphe de la figure 4.5, pour l'hypercube Q_4.

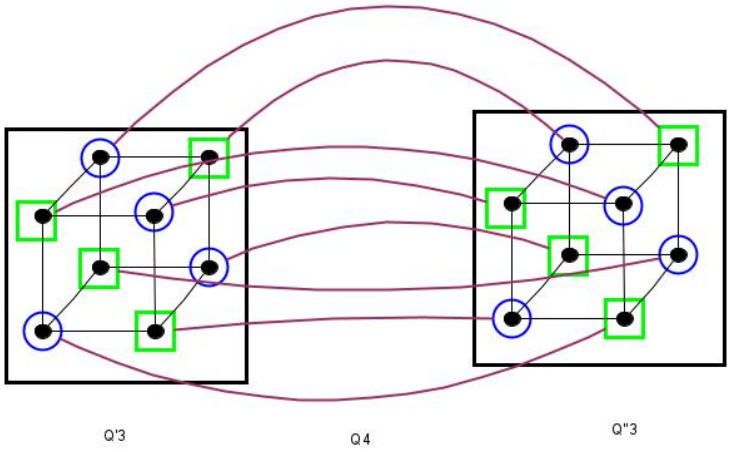

FIGURE 4.5 – Décomposition de Q_4 en des copies disjointes de Q_3

A partir de la construction de l'hypercube Q_n par les copies disjointes Q'_{n-1} et Q''_{n-1} de Q_{n-1} , on peut parler de la décomposition de l'hypercube Q_n en des copies disjointes de Q_i comme suit :

• Q_n peut être décomposé en deux copies disjointes de Q_{n-1}, comme Q_{n-1} peut être décomposé en deux copies disjointes de $Q_{(n-1)-1}$, alors si on va procéder de la même manière, l'hypercube Q_n peut être décomposé en 2^{n-1} copies de Q_1.

Remarque : Q_n peut être décomposé en 2^{n-i} copies disjointes de Q_i avec $i \in \{0, 1, 2, ..., n\}$, il est clair que Q_2 peut être décomposé en un sommet isolé (Q_0) et une copie de $K_{1,2}$ (graphe complet $K_{1,2}$) comme le montre le graphe de la figure 4.6 :

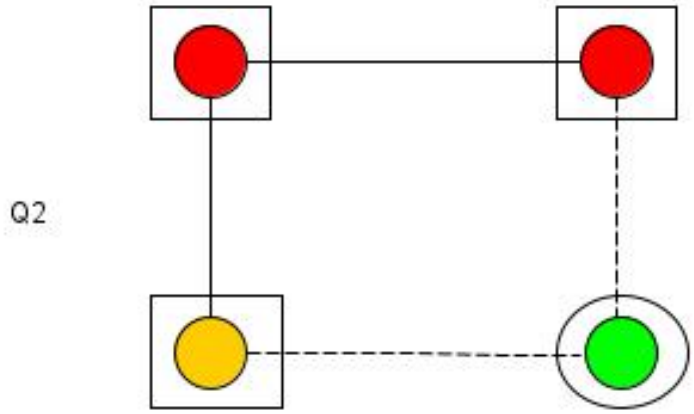

FIGURE 4.6 – Q_2

En procédant de la même manière, on peut décomposer Q_n en 2^{n-2} copies disjointes $\{b_i / i \in \{1, 2, ..., 2^{n-2}\}\}$ de $K_{1,2}$ et 2^{n-2} copies disjointes $\{Q_0^1, Q_0^2, ..., Q_0^{2^{n-2}}\}$ de Q_0 (sommets isolés).

• **Classe d'arbre binaire** $\overline{R_n}$:

Pour $n = 1$, l'arbre $\overline{R_1}$ est donné par la figure 4.7 :

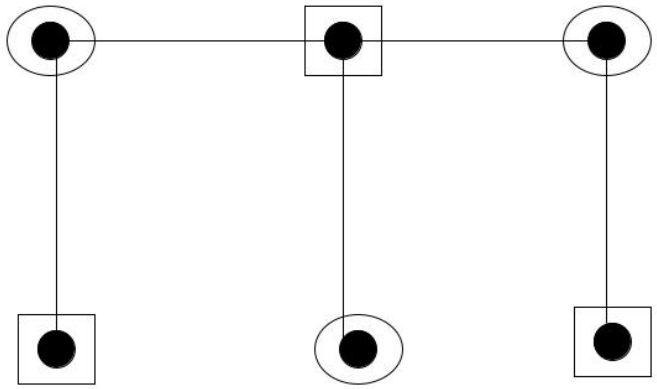

FIGURE 4.7 – Arbre $\overline{R_1}$

Il est clair que $\overline{R_1}$ possède $2^3 - 2 = 6$ sommets, dont 2 sommets pendants de type carré et un sommet pendant de type cercle.

Pour $n \geq 2$, l'arbre $\overline{R_n}$ est obtenu à partir de $\overline{R_{n-1}}$ tel que chaque sommet pendant de $\overline{R_{n-1}}$ est relié à deux nouveaux sommets. Il est clair que $\overline{R_n}$ contient $2^{n+2} - 2$. l'arbre $\overline{R_2}$ est donné par la figure $4,8$:

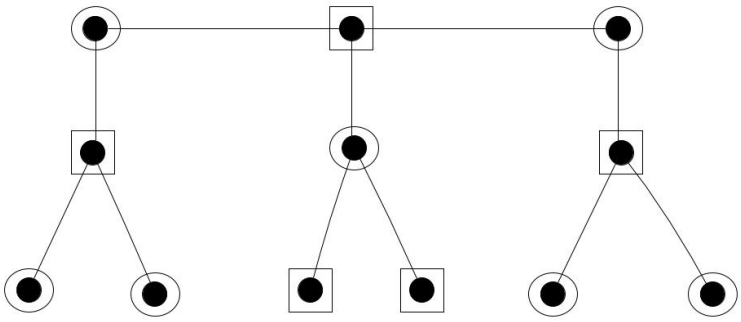

FIGURE 4.8 – Arbre $\overline{R_2}$

Théorème : Pour tout $n \geq 1$, l'arbre $\overline{R_n}$ est plongeable dans Q_{n+3} et $dim(\overline{R_n}) = n + 3$.

Preuve :

Il est clair que $\overline{R_1}$ et $\overline{R_2}$ sont plongeables respectivement dans Q_3 et Q_4, comme $V(\overline{R_1} = 6)$ et $V(\overline{R_2} = 12)$, alors $dim(\overline{R_1} = 3)$ et $dim(\overline{R_2} = 4)$. Pour montrer que $\overline{R_3}$ est plongeable dans Q_5 nous

allons procéder comme suit :

On sait que Q_n peut être décomposé en 2^{n-i} copies disjointes de Q_i, $i \in \{0, ..., n\}$ et que Q_2 est décomposé en une copie de $K_{1,2}$ et un sommet isolé, alors Q_n peut être décomposé en 2^{n-2} copies disjointes ($\{b_1, b_2, ..., b_{2^{n-2}}\}$) de $K_{1,2}$ et 2^{n-2} copies ($\{Q_0^1, Q_0^2, ..., Q_0^{2^{n-2}}\}$) de Q_0 (sommets isolés).

De plus, l'hypercube Q_n est obtenu à partir de deux copies disjointes Q'_{n-1} et Q''_{n-1} de Q_{n-1}, en reliant un sommet de type carré (respectivement cercle) de Q'_{n-1} à un sommet de type cercle (respectivement carré) de Q''_{n-1}.

Considérons deux copies disjointes Q'_3 et Q''_3 de Q_3 et une copie de Q_4, tel que dans Q'_3 on prend seulement l'arbre $\overline{R_1}$, dans Q''_3 on prend 4 copies disjointes ($\{Q_1^1, Q_1^2, Q_1^3, Q_1^4\}$) de Q_1 et dans Q_4 on prend 4 copies disjointes ($\{b_1, b_2, b_3, b_4\}$) de $K_{1,2}$ et 4 sommets isolés dont deux sont de type carré et 2 sont de type cercle, comme le montre la figure 4.9 :

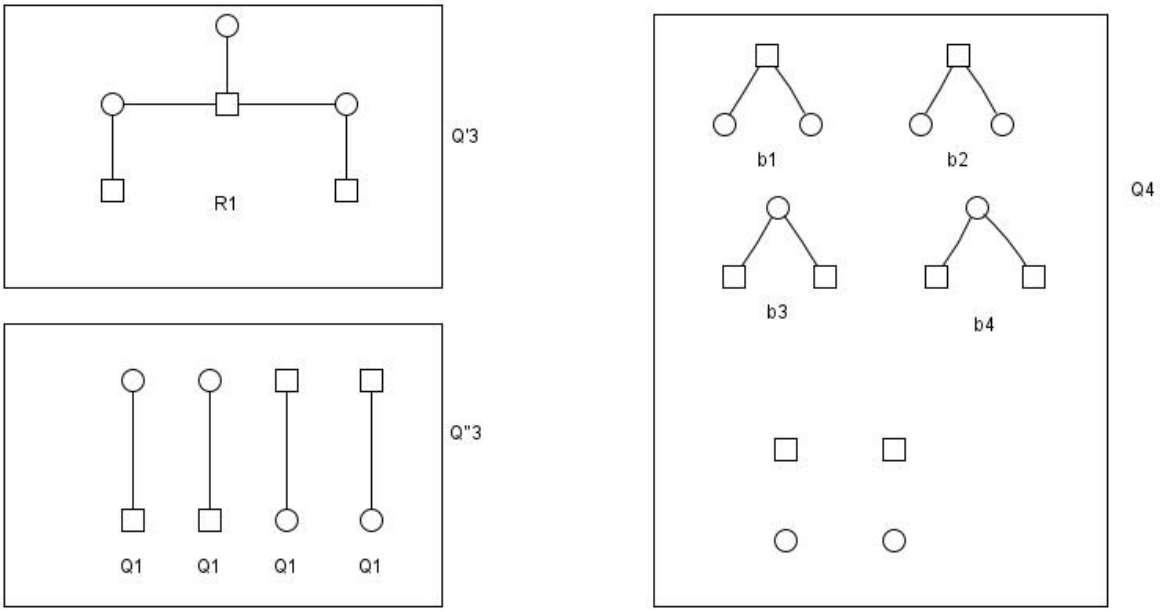

FIGURE 4.9 – Q'_3, Q''_3 et Q_4

Dans cette figure, si chaque sommet, $(1 \leq i \leq 4)$ de type carré (resp. cercle) de Q''_3 est relié par une arête à un sommet isolé de type cercle (resp. carré) de Q_4, alors on obtient 4 copies disjointes $(\{a_1, a_2, a_3, a_4\})$ de $K_{1,2}$ comme le montre le graphe de cette figure 4.10 :

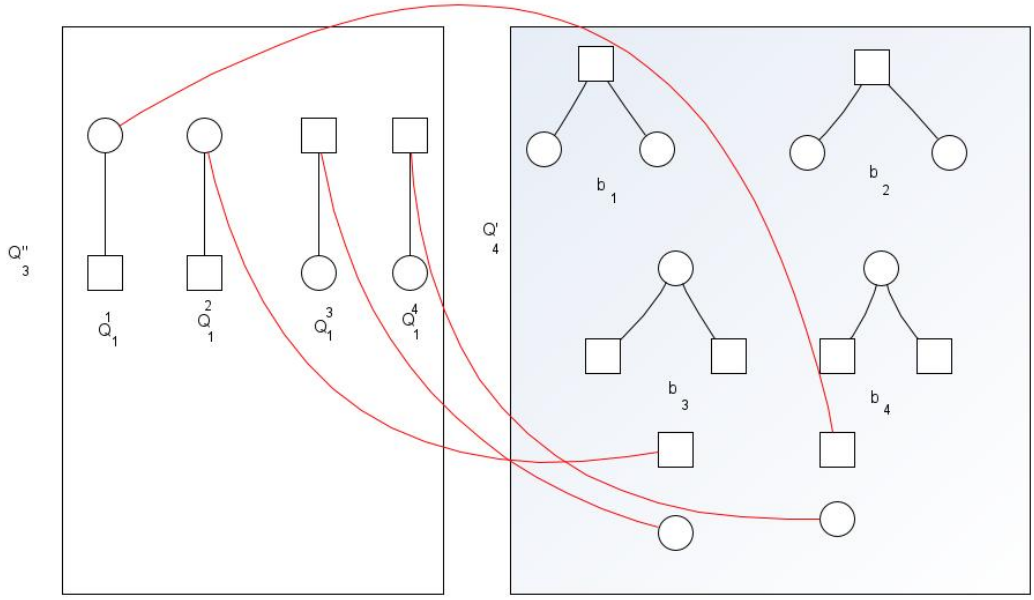

FIGURE 4.10 – Construction de 4 copies disjointes de $K_{1,2}$

Lorsque on relie chaque sommet pendant de type carré (resp. cercle) de $\overline{R_1}$ à un sommet de type cercle (resp. carré) de degré 2 de $b_i, 1 \leq i \leq 4$ par une arête et chaque sommet pendant de type cercle (resp. carré) de $\overline{R_1}$ à un sommet de type carré (resp. cercle) de degré 2 de $(a_i, 1 \leq i \leq 4)$ par une autre arête, on obtient l'arbre $\overline{R_3}$ représenté par la figure $4, 11$:

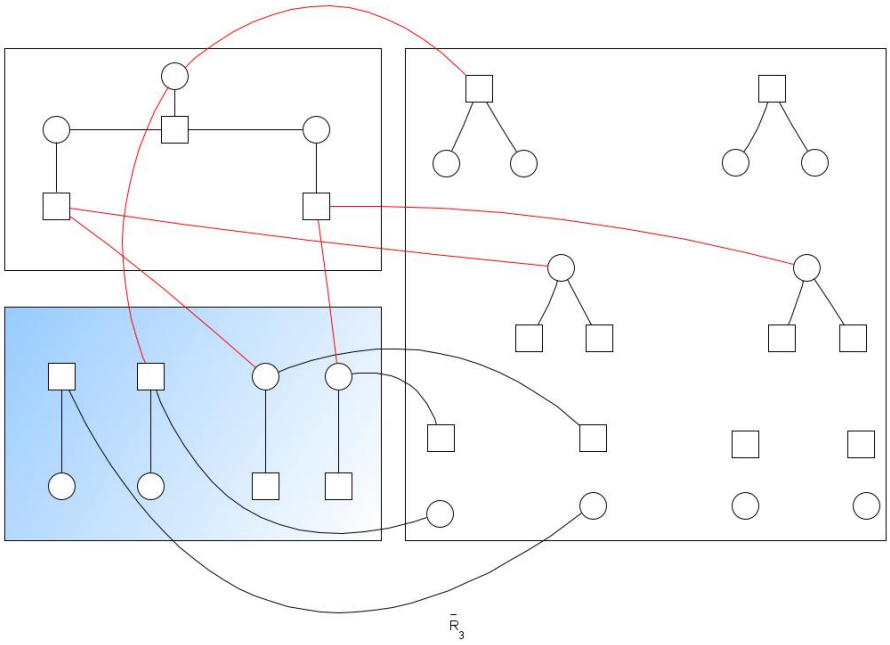

FIGURE 4.11 – Arbre $\overline{R_3}$

Comme Q_4 est obtenu à partir de Q'_3 et Q''_3 tel que chaque sommet de type cercle (resp. carré) de Q'_3 est relié par une arête à un sommet de type carré (resp. cercle) de Q''_3, et que Q_5 est obtenu à partir de deux copies disjointes Q'_4 et Q_4 tel que chaque sommet de type carré (resp. cercle) de Q'_4 est relié à un sommet de type cercle (resp. carré) de Q_4. En conclusion, $\overline{R_3}$ est plongeable dans Q_5. Comme $\overline{R_1}, \overline{R_2}$ et $\overline{R_3}$ sont plongeables respectivement dans Q_3, Q_4 et Q_5, alors, montrons par récurrence sur n que $\overline{R_n}$ est plongeable dans Q_{n+2} :
Supposons que la propriété est vraie pour n c'est à dire $\forall n \geq 1$, $\overline{R_n}$ est plongeable dans Q_{n+2} et montrons que $\overline{R_{n+1}}$ est plongeable dans Q_{n+3}.
Comme $\overline{R_{n-1}}$ et $\overline{R_n}$ sont plongeable respectivement dans Q_{n+1} et Q_{n+2} (hypothèse de récurrence), alors, si on considère deux copies disjointes Q'_{n+1} et Q''_{n+1} de Q_{n+1} et une copie de Q_{n+2} tel que dans Q'_{n+1} nous prenons que l'arbre $\overline{R_{n-1}}$, dans Q''_{n+1} nous prenons 2^n copies disjointes $\{Q_1^1, Q_1^2, ..., Q_1^{2^n}\}$ de Q_1, et dans Q_{n+2} nous allons prendre 2^n copies disjointes $(\{b_1, b_2, ..., b_{2^n}\})$ de $K_{1,2}$ et 2^n copies de Q_0 (sommets isolés). Si on va relier chaque sommet de type carré (resp. cercle) de $b_i, 1 \leq i \leq 2^n$ dans Q''_{n+1} à un sommet isolé de type cercle (resp. carré) de Q_{n+2}, on obtient 2^n copies $(\{a_1, a_2, ..., a_{2^n}\})$

de $K_{1,2}$. Par la suite, si on va relier chaque sommet pendant de type carré (resp. cercle) de $\overline{R_{n-1}}$ à un sommet de degré 2 de type cercle (resp. carré) de $b_i, (1 \leq i \leq n)$ par une arête et à un sommet de degré 2 de type cercle (resp. carré) de $a_i, (1 \leq i \leq 2^n)$, on obtient $\overline{R_{n+1}}$ comme le montre le graphe de la figure suivante :

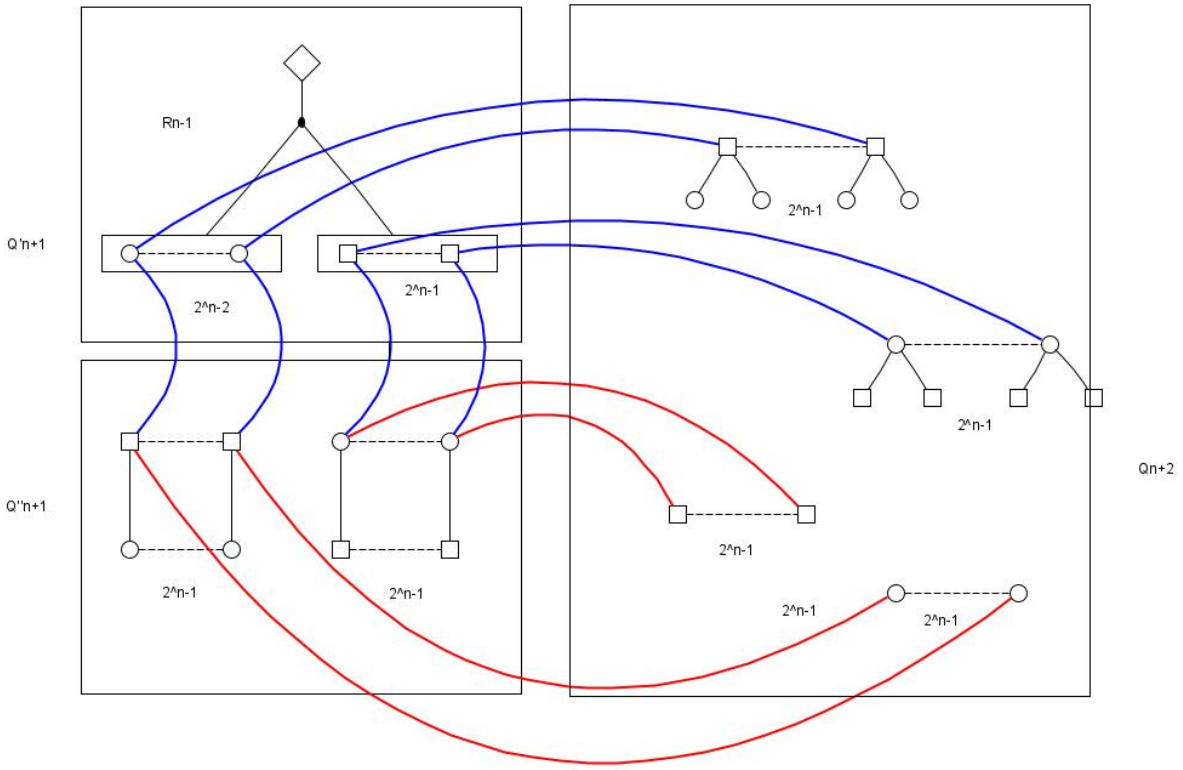

FIGURE 4.12 – $\overline{R_{n+1}}$

Si chaque sommet de type carré (resp.cercle) de Q'_{n+1} est relié par une arête à un sommet de type cercle (resp. carré) de type Q''_{n+1}, on obtient une copie Q'_{n+2} de Q_{n+2}. Il est clair que Q_{n+3} est obtenue à partir de Q'_{n+2} et Q_{n+2} tel que chaque sommet de type carré (resp. cercle) de Q'_{n+2} est relié par une arête à un sommet de type cercle (resp. carré) de Q_{n+2}, alors $\overline{R_{n+1}}$ est plongeable dans Q_{n+3}. Comme $\overline{R_{n+1}}$ contient $(2^{(n+1)+2}) - 2 = 2^{n+3} - 2$ sommets, alors, $\overline{R_{n+1}}$ ne peut pas être plongeable dans Q_{n+2} car : $|V(\overline{R_{n+1}})| > |V(Q_{n+2})|$. Donc $dim(\overline{R_{n+1}}) = n + 3$. En conclusion, $\overline{R_n}$ est plongeable dans Q_{n+2} et $dim(\overline{R_n}) = n + 2$.

Chapitre 5

Application

5.1 Arbre couvrant de poids minimum

On parle ainsi d'un arbre qui couvre le graphe G (il connecte tous ses sommets) et dont le cout de connexion est minimal.
Comment trouver l'arbre couvrant de poids minimum sur un hypercube ?
Pour la recherche d'un arbre couvrant de poids minimum, nous utiliserons l'un des algorithmes suivants :

5.2 Algorithme de Kruskal (Bis)(1956)

En reprenant l'algorithme d'énumération des arbres, on descend l'arborescence à gauche (contraction de l'arête) en choisissant l'arête de longueur minimale à chaque étape. On s'arrête lorsque tous les sommets du graphe sont connectés, ce qui revient au même, lorsque le nombre d'arêtes retenues égale à $n - 1$.
C'est un algorithme glouton, c'est à dire il fait un choix optimal localement dans l'espoir que ce choix mènera à la solution optimale globalement. Ici, il rajoute à chaque étape l'arête de poids minimal. L'arbre obtenu est unique si toutes les arêtes sont initialement de valeurs différentes.
Algorithme de Kruskal (Bis) :

Recherche de l'arbre de poids min.

Données : graphe connexe $G = (V, E, P)$.

Résultats : ensemble d'arcs E.

Début

Renuméroter les arcs dans l'ordre de poids croissant $P(a_1) \leq P(a_2) \leq ...P(a_m)$.

Initialiser $A = \emptyset$.

Pour tout $j = 1, 2...m$ faire

Si $G_j = (V, E \cup a_j)$ ne contient pas de cycle, alors , poser $A = A \cup a_j$.

Fin si ;

Fin pour ;

Fin.

5.3 Algorithme de Prim (Bis)(1957)

•Idée générale

-On part d'un arbre initial réduit à un seul sommet du graphe.

-A chaque itération, on agrandit l'arbre en lui ajoutant le sommet libre accessible de plus petit poids possible.

-On s'arrète quand l'arbre est recouvrant.

• **But :** Recherche de l'arbre de poids min.

Données : graphe connexe $G = (V, E, P)$.

Résultats : ensemble d'arcs E.

• **Méthode**

-Initialiser $A = \emptyset$.

-Répéter :

• Trouver toutes les arètes de G qui relient un sommet de A et un sommet éxtérieur à A.

• Parmi celles-ci, choisir une arête de poids le plus petit possible.

• Ajouter à A cette arête et le sommet correspondant.

-S'arrêter dès que tous les sommets de G sont dans A.

-Retourner A.

5.4 Application sur le $C++$:

Nous proposons d'appliquer l'algorithme de Prim pour trouver l'arbre de poids minimum. Pour le faire, nous utiliserons le langage $C++$.

5.4.1 Présentation de langage C++

C++ est un langage de programmation compilé permettant la programmation sous de multiples paradigmes (comme la programmation orienté objet..). Ses bonnes performances, et sa compatibilité avec le C en font un des langages les plus utilisés dans les applications et pour le développement des logiciels où la performance est critique. Crée initialement par **Bjarne Stroustrup** dans les années 1980, le langage C++ est aujourd'hui normalisé par l'ISO.
Programmer un ordinateur, c'est lui fournir une série d'instructions qu'il doit éxécuter. Ces instructions sont généralement écrites dans un langage dit évolué, puis, avant d'être éxécutées, sont traduites en langage machine (qui est le langage de microprocesseur). Cette traduction s'appelle compilation et elle est effectuée automatiquement par un programme appelé compilateur.
Un programme écrit en C++ se compose généralement de plusieurs fichiers-sources. Il y a deux sortes de fichiers-sources :

- Ceux qui contiennent effectivement des instructions, leur nom possède l'extension.cpp,
- Ceux qui ne contiennent que des déclarations, leurs nom possède l'extension .h (signifiant "header" ou en- tête).

Un fichier.h sert à regrouper des déclarations qui sont communes à plusieurs fichiers.cpp et permet une compilation correcte de ceux-ci. Pour ce faire, dans un fichier.cpp on prévoit l'inclusion automatique des fichiers .h qui lui sont nécessaires, grâce aux directives de compilation include. En supposant que le fichier à inclure s'apelle untel.h, on écrira include < untel.h > s'il s'agit d'un fichier de la bibliothèque standard du C++, ou include "untel.h" s'il s'agit d'un fichier écrit par nous même.[12]
•Utiliser Code : :Blocks sous Windows.

Lorsque on lance Code : :Blocks, nous voyons apparaître l'écran ci-contre :

Pour créer un nouveau projet, il faut choisir dans le menu File puis New puis Project.

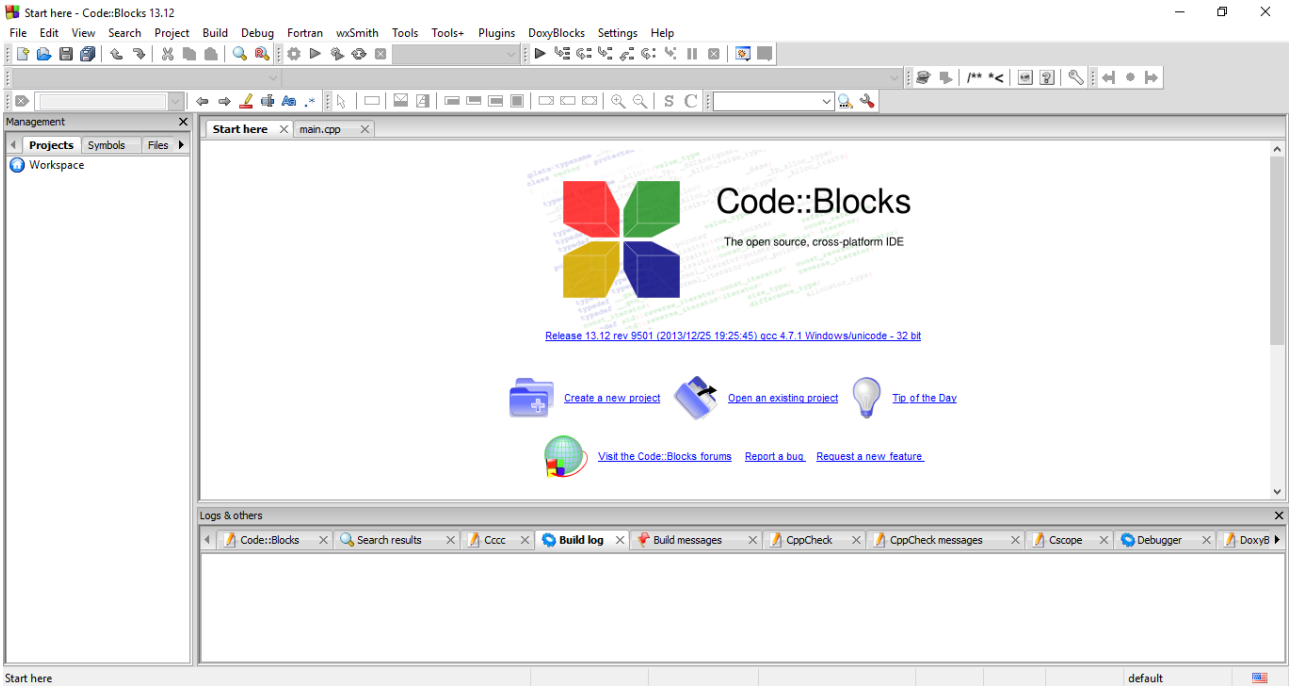

FIGURE 5.1 – L'interface de code blocks

5.4.2 Exemple d'application de cet algorithme sur un graphe

La recherche d'un arbre couvrant sur un hypercube se fait à l'aide de l'algorithme de Prim qu'on va appliquer sur l'hypercube Q_3

5.4.3 Application numérique

Prenons l'hypercube Q_3 comme un réseau de communication de 8 individus, avec $P_{i,j}$ représente la probabilité d'interception des messages confidentiels entre i et j par une personne étrangère. On doit chercher un arbre couvrant qui minimise la probabilité de l'interception, en utilisant l'algorithme de Prim sous langage C++. La situation est représentée par la matrice suivante :

$$M = \begin{pmatrix} 0 & 0.6 & 0 & 0.25 & 0.15 & 0 & 0 & 0 \\ 0.6 & 0 & 0.25 & 0 & 0 & 0.15 & 0 & 0 \\ 0 & 0.25 & 0 & 0.6 & 0 & 0 & 0.15 & 0 \\ 0.25 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0.15 \\ 0.15 & 0 & 0 & 0 & 0 & 0.6 & 0 & 0.25 \\ 0 & 0.15 & 0 & 0 & 0.6 & 0 & 0.25 & 0 \\ 0 & 0 & 0.15 & 0 & 0 & 0.25 & 0 & 0.6 \\ 0 & 0 & 0 & 0.15 & 0.25 & 0 & 0.6 & 0 \end{pmatrix}$$

Le réseau de télécommuication est illustré par l'hypercube Q_3 suivant :

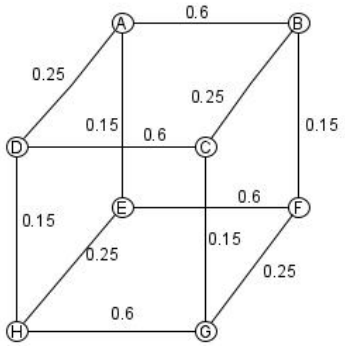

FIGURE 5.2 – Probabilité d'interception des messages dans un réseau de type hypercube

Appliquons maintenant l'algorithme de Prim :

On doit d'abord multiplier la matrice par 10, pour pouvoir appliquer cet algorithme :

-**Résolution manuelle**

•**Initialisation**

$A = \emptyset$

Soit le sommet de départ $C = \{A\}$, $\prod(A) = 0$, $\prod(X) = +\infty \ \forall x \notin C$.

• 1^{er} **itération**

On considère les sommets voisins de sommet A :

$A = A \cup \{(A, E)\} = \{(A, E)\}$ avec $|A| = 1$.

Cette itération est illustré dans ce graphe :

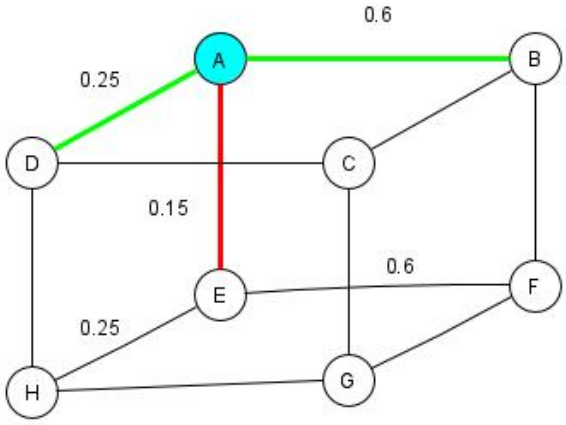

FIGURE 5.3 – Résultat de la première itération

Comme le graphe ne contient pas de cycle , on continue :

• 2^{me} **itération**

On considère les sommets voisins des sommets A et E :

$A = A \cup \{(A, D)\} = \{(A, E), (A, D)\}$ avec $|A| = 2$.

Cette itération est illustré dans le graphe suivant :

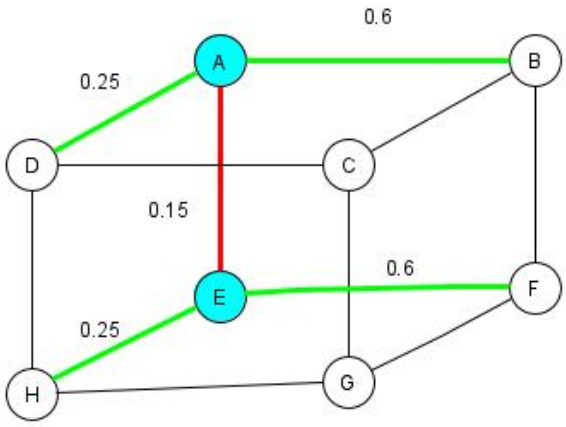

FIGURE 5.4 – Résultat de la deuxième itération

Le graphe ne contient pas de cycle, on continue :

• 3^{me} **itération**

On considère les sommets voisins des sommets A, E et D : et on procédant de la même manière, on aura le résultat suivant :

$A = A \cup \{(D, H)\} = \{(A, E); (A, D); (D, H)\}$ avec $|A| = 3$.

Le graphe ne contient pas de cycle, on continue :

• 4^{me} **itération**

On considère les sommets voisins des sommets A, E, D et H : et on procédant de la même manière, on aura le résultat suivant :

$A = A \cup \{(A, B)\} = \{(A, E); (A, D); (D, H); (A, B)\}$ avec $|A| = 4$.

Le graphe ne contient pas de cycle, on continue :

• 5^{me} **itération**

On considère les sommets voisins des sommets A, E, D, H et B : et on procédant de la même manière, on aura le résultat suivant

$A = A \cup \{(B, F)\} = \{(A, E); (A, D); (D, H); (A, B); (B, F)\}$ avec $|A| = 5$.

Le graphe ne contient pas de cycle, on continue :

• 6^{me} **itération**

On considère les sommets voisins des sommets A, E, D, H, B et F : et on procédant de la même manière, on aura le résultat suivant :

$A = A \cup \{(B, C)\} = \{(A, E); (A, D); (D, H); (A, B); (B, F); (B, C)\}$ avec $|A| = 6$.

Le graphe ne contient pas de cycle, on continue :

• 7^{me} **itération**

On considère les sommets voisins des sommets A, E, D, H, B, F et C : et on procédant de la même manière, on aura le résultat suivant

$A = A \cup \{(C, G)\} = \{(A, E); (A, D); (D, H); (A, B); (B, F); (B, C); (C, G)\}$ avec $|A| = 7$.

Le graphe ne contient pas de cycle. L'algorithme s'arrète car $|A| = n - 1 = 8 - 1 = 7$ alors $|A| = 7$, l'arbre couvrant de poids minimum est montré dans la figure suivante :

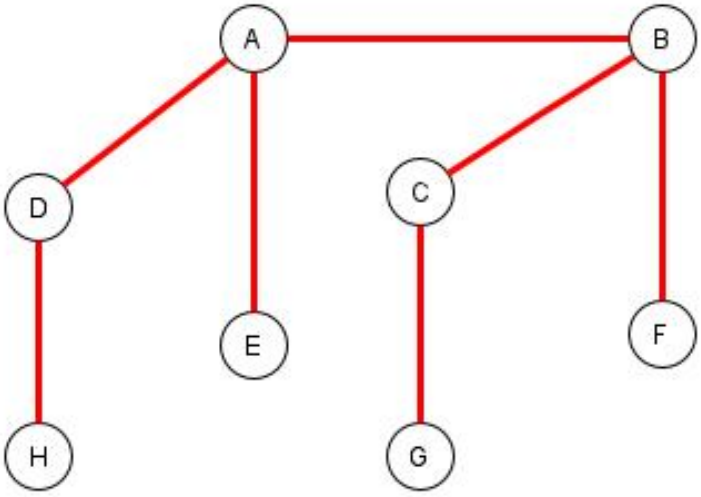

FIGURE 5.5 – Arbre couvrant de poids minimum

En éxécutant l'algorithme de Prim en C++, Le résultat obtenu par la machine est le suivant :

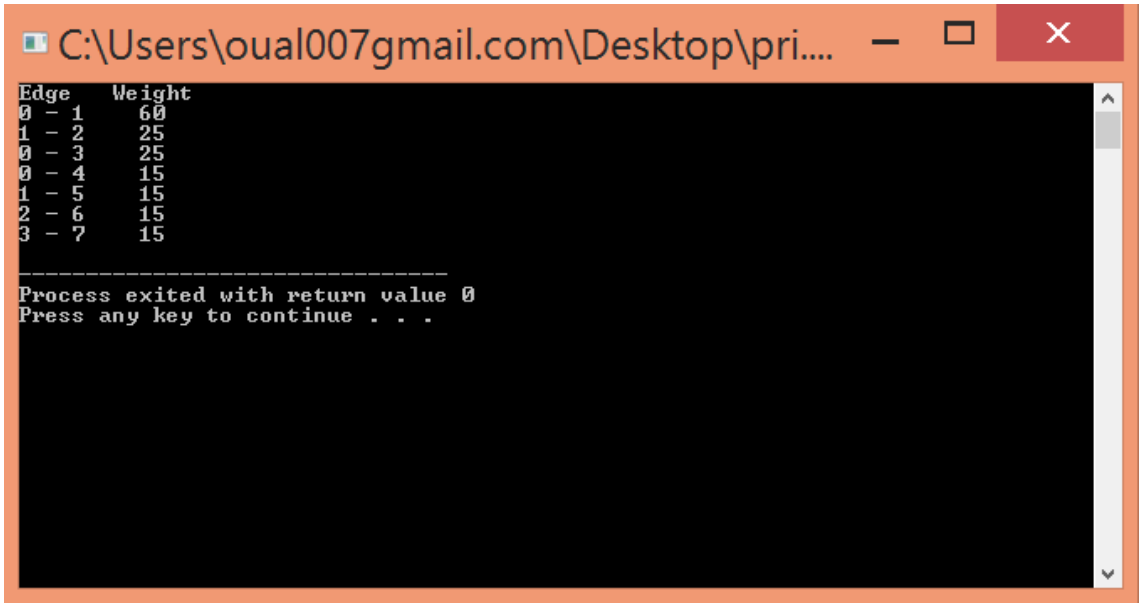

FIGURE 5.6 – Résultat après l'éxécution de l'algorithme de Prim pour Q_3

Conclusion générale

Le plongement des graphes dans l'hypercube Q_n est un problème très étudié en théorie des graphes. En effet, de nombreux efforts ont été consacré pour déterminer les conditions selon lesquelles un graphe G est plongeable dans l'hypercube. L'hypercube est un graphe intéréssant dont la topologie et utilisé en informatique (parallélisme, réseau), il est fondamentale de déterminer quels sont les graphes et surtout les arbres qui sont plongeable dans l'hypercube. Cette importance résulte de l'utilisation des arbres dans plusieurs domaines, donc le problème consiste à donner la plus petite dimension d'un hypercube. Dans le même contexte nous avons introduit une nouvelle classe d'arbres pour laquelle nous avons donné la dimension cubique. Nous avons aussi introduit une autre classe d'arbre binaire pour laquelle nous avons donné le nombre maximum de copies disjointes de sa topologie.

Comme perspective, nous allons essayer de caractériser des nouvelles classes d'arbres binaires, pour lesquelles nous donnerons une technique de plongement optimal. Nous déterminerons ainsi la C_n valuation de tous les arbres obtenus d'une manière récursive.

Bibliographie

[1] S.Taouinet cours du module "Théorie des graphes" destiné aux étudiants de Master 1. Recherche Opérationnelle, Béjaia

[2] Mémoire de Fin d'étude Master Filière Informatique thème :
$\ll Reconnaissancedemotifsdansdesgraphes EMF \gg$ Juillet 2019, Université de 8 Mai 1945 -Guelma-

[3] I-Havel . J.M-Laborde : on distance Monotone Graphs, Colloquia, Mathematica Societatis Janos Balyai52 (1987). 557 - 561.

[4] S.A. Folodes characterization of hypercubes - Discrete Mathematics 17,pp-, 155 - 159, 1977

[5] H.M.Mulder. The interval fonction of a graph - Mathematical Center tracte 132, Mathematics center, Amesterdam 1980.

[6] J.M.Laborde and Rao Hebbare-S.P- another characterisation of hypercube, Discrete Math- (39) : 161-166, 1982.

[7] H.J.Bandelt and Mulder. H.M.Infinity median graph - graph and hypercub - journal of graph theory. pp 487 - 492 - 1983.

[8] P.L.Havel.Embedding the polytomic tree into the n - cube. carpest. mat (1972)

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

[10] I.Havel. an hamiltonian circuit and spamming tree of hypercubes.Cospis Pest. Mat mat.97, No.2, pp - 201 - 205 (1972)

[11] K.Kabyle and A.Berrachedi et Eric Sopera . A note on the cubical dimension of new classes of binary tree - Czechoslovak Mathematical Journal - vol - 95, N01, pp 151 - 160, 2015.

[12] O - Marguin. cours d'informatiques : $\ll C++ : lesBases \gg$ - 2003/2004

Résumé

L'étude de plongement d'un graphe G dans un hypercube H revient à voir si G admet une C_n-valution. Dans ce mémoire, nous nous somme intéressé à la nouvelle classe d'arbre plongeable dans Q_n, puis on a implémenté l'agorithme de Prim sous C^{++} pour la recherche d'arbre de poids minimum dans un réseau de télécommunication de topolgie hypercube.

Abstract

The study of the embedding of a graphe G in hypercub H amounts to seeing if G admits a C_n-valution, in this thesis we are interested in the new tree class that can be plunged into Q_n. Then we implemented the Prim agorithm under C^{++} for the search of a minimum weight tree in hypercub topology telecommunication network .