N°D'ORDRE :23/2013-M/MT

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

Université des Sciences et de la Technologie Houari Boumediene
Faculté de Mathématiques

MÉMOIRE
Présenté Pour L'Obtention Du Diplôme De MAGISTÈRE
En : RECHERCHE OPÉRATIONNELLE
Spécialité : MATHÉMATIQUE DE GESTION

Par : **Fatah CHEURFA**

Sujet

Optimisation de plongement de copies de graphes dans l'hypercube

Soutenue le 09/05/2013 devant le jury composé de :

M.	M.E-A. CHERGUI	Maitre de conférence/A à USTHB	**Président**
M.	A.BERRACHEDI	Prof USTHB	**Directeur de mémoire**
M.	A.SEMRI	Maitre de conférence/A à USTHB	**Examinateur**
Mlle	K.MESLEM	Maitre de conférence/B USTHB	**Invitée**
M.	KABYL Kamal	Maitre Assistant/A à U.BEJAIA	**Invité**

Table des matières

Table des figures

Liste des Algorithmes

Remerciements

Premièrement, je remercie Dieu le Miséricordieux, pour m'avoir donné la volonté et la force pour accomplir ce modeste travail, elhamdou li llah.

Je tiens à exprimer ma profonde gratitude au professeur A.BERRACHEDI pour l'honneur qu'il ma fait en acceptant de m'encadrer. Ses conseils précieux ont permis une bonne orientation.

Je tiens ensuite à remercier spécialement Mr K.KABYL pour leurs suivi et conseils, aussi pour la documentation qu'il a mis à ma disposition et l'intérêt qu'il a porté à mon travail.

Je remercie Mr M.E-A CHERGUI qui m'a fait l'honneur d'accepter d'être président du jury.

Mes remerciements chaleureux s'adressent également à Mr A.SEMRI et Melle K .MESLEM pour avoir accepté d'examiner ce travail.

Je n'oublierai pas de remercier ma famille qui ma toujours encouragé et soutenu.

Introduction générale

"Un petit dessin vaut mieux
qu'un grand discours".
Napoléon.

Les graphes sont utilisés dans de nombreux domaines très variés, et ne sont pas seulement un outil mathématique. Les résultats de recherches sur les graphes trouvent des applications pratiques. Ils permettent ainsi de formaliser et modéliser des situations et problèmes.

Plusieurs 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.

C'est naturellement cependant, que l'informatique, l'algorithmique et les mathématiques théoriques sont les domaines où les graphes sont les plus étudiés. Les résultats des recherches effectués dans ces différents domaines sont souvent appliqués dans d'autres secteurs. On voit alors des applications pratiques dans le domaine de l'architecture parallèle par exemple, de nombreuses études ont été élaborées pour résoudre certains problèmes liées à l'architecture des systèmes. Le but de ces études est de proposer des structures (en gardant celle de l'hypercube comme structure de base) ou des méthodes, pour améliorer le temps de communication entre composants et le fonctionnement du système en présence des éléments défectueux.
Dans notre mémoire, l'objectif considéré porté sur la détermination du nombre

maximum de copies disjointes d'un graphe qu'on peut plonger dans un hypercube de dimension n.

Nous présentons ci-après nos contributions et le plan de ce manuscrit.
Le premier chapitre de ce mémoire est consacré à la présentation des différentes notions utilisées en théorie des graphes de manière courante et dont nous nous servons dans ce document.

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

Le troisième chapitre concerne le plongement de graphes dans l'hypercube, et nous avons présenté quelques classes de graphes plongeables dans l'hypercube.

Le quatrième chapitre est consacré à un autre type de problèmes,qui consiste à optimiser le nombre de copies disjointes de graphes comme l'étoile $K_{1,3}$, double quasi-étoile $S(1,1,1;1,1,1)$,k-double quasi-étoile et les grilles, qu'on peut plonger dans l'hypercubes de dimension n.

Nous concluons ce mémoire par quelques algorithmes permettant de déterminer le nombre maximum de copies de graphes comme l'étoile $K_{1,3}$ et double quasi-étoile $S(1,1,1;1,1,1)$,plongeables dans l'hypercube de dimension n.

Chapitre 1

Notions élémentaires de la théorie des graphes

Introduction

La notion de graphe est une structure combinatoire permettant de représenter de nombreuses situations rencontrées dans des applications ; circuits électriques, réseaux de transport (ferrés, routiers, aériens), réseaux d'ordinateurs, ordonnancement d'un ensemble de tâches...

Nous donnons ici les définitions nécessaires à la lecture de ce mémoire et quelques propriétés fondamentales.
On adoptera la terminologie de Berge [3].

1.1 Graphe

Définition *1.1.1*
Un graphe G est constitué d'un ensemble V fini d'éléments $\{v_1, v_2, ..., v_n\}$, appelés sommets, et d'une famille E de paires distinctes de V ; appelées arêtes.

On utilisera la notation simplifiée xy ou yx pour l'arête $\{x, y\}$. Une arête de type xx est appelée boucle de G.
Si $e = xy$ est une arête de G, on dit que x et y sont voisins, ou adjacents dans G et qu'ils forment les extrémités de e.

Une arête e est dite incidente à un sommet v si v est une extrémité de e. Le degré d'un sommet $v \in V$, note $d(v)$, est le nombre d'arêtes incidentes à v.
L'ordre d'un graphe est le cardinal de son ensemble de sommets, noté $|G|$.

Si tous les sommets ont le même degré, on dira que le graphe G est **régulier**.
(voir figure 1.1)

Graphe simple

Un graphe G est *simple* s'il ne contient pas de boucles et s'il ne contient pas plus qu'une arête entre deux sommets, nous considérons dans le document uniquement les graphes simples.

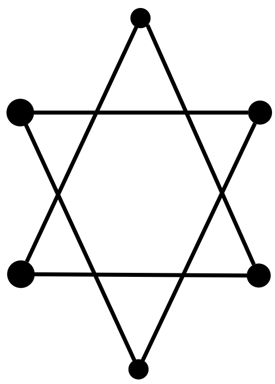


FIGURE 1.1 – Graphe 2-régulier

- Le graphe complémentaire de G noté $\bar{G}$, est le graphe dont l'ensembles de sommets est V et deux sommets distincts de $\bar{G}$ soient adjacents si et seulement s'ils ne sont pas adjacents dans G.

La figure 1.2 représente le graphe complémentaire du graphe G de la figure 1.1.

Sous graphe

– Un *sous-graphe* de G est un graphe $H = (W, E(W))$ tel que W est un sous-ensemble de V, et $E(W)$ sont les arêtes induites par E sur W,

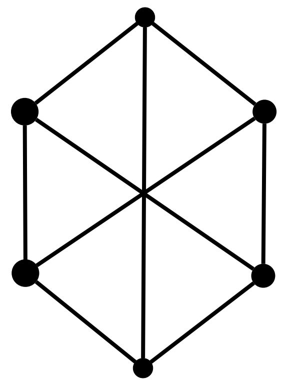


FIGURE 1.2 – Graphe complémentaire

c'est à dire les arêtes de E dont les 2 extrémité sont des sommets de W.

$$E(W) = \{(x, y) \in E | x, y \in W\}$$

– Un *graphe partiel* de G est un graphe $I = (V, F)$ tel que F est un sous-ensemble de E.

Un sous-graphe H de G est entièrement défini (induit) par ses sommets W, et un graphe partiel I par ses arêtes F.

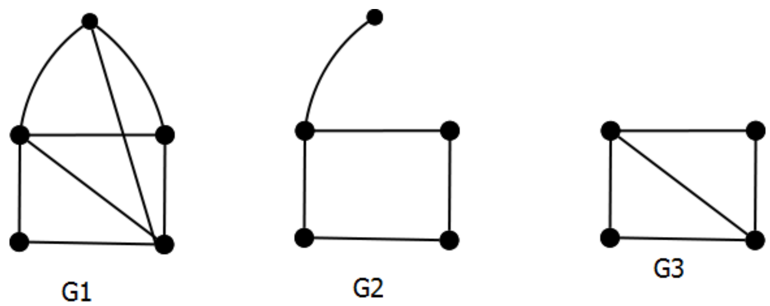


FIGURE 1.3 – Un graphe et deux sous-graphes.

Dans le premier sous-graphe G_2 de la figure 1.3 , on a enlevé uniquement certaines arêtes. Dans le second graphe G_3, on a enlevé un sommet et les arêtes adjacentes.

Chaînes

On appelle chaîne entre deux sommets x et y d'un graphe G, une suite de sommets $x_1, ..., x_k$ dont deux consécutifs sont adjacents, avec $x_1 = x$ et $x_k = y$. Une chaîne qui n'utilise pas deux fois le même sommet est dite élémentaire,notée xy-chaîne induite une chaîne qui n'utilise pas deux fois la même arête est dite simple. Une chaîne élémentaire est donc une chaîne simple.

cycles

On appelle cycle dans un graphe, une chaîne simple dont les extrémités initiale et finale sont confondues.

Chaînes et cycles hamiltoniens

On appelle chaîne (resp. cycle) hamiltonienne (resp. hamiltonien) une chaîne (resp. cycle) passant, une fois et une seule, par tous les sommets d'un graphe G.
Un graphe G qui possède un cycle hamiltonien est dit graphe hamiltonien.

1.1.1 Connexite dans les graphes

Un graphe est dit connexe si pour toute paire de sommets $\{u, v\}$, il existe une chaîne reliant u à v.

1.2 Graphes Particuliers

Dans la suite de ce chapitre nous présentons quelques type de graphes.

1.2.1 Graphe complet

Un graphe $G = (V, E)$ est dit complet si tous les sommets sont deux à deux adjacents. le graphe complet simple à n sommets est noté K_n(figure 1.4(d)).

1.2.2 Une clique

Une clique d'un graphe simple $G = (V, E)$ est un sous-graphe complet de G.
La taille d'une clique est le nombre de sommets qui la composent. La taille maximale d'une clique de G est notée $\omega(G)$.

1.2.3 Un stable

C'est un ensemble des sommets qui sont deux à deux non adjacents.

1.2.4 Graphe biparti

Un graphe G est dit biparti si l'ensemble de ses sommets V peut être partitionné en deux parties $V = V_1 \bigcup V_2$,telle que tout élément de E a une extrémité dans V_1 et l'autre dans V_2(figure 1.4(a)).
Si de plus $|V_1| = |V_2|$, G est appelé biparti équilibré.

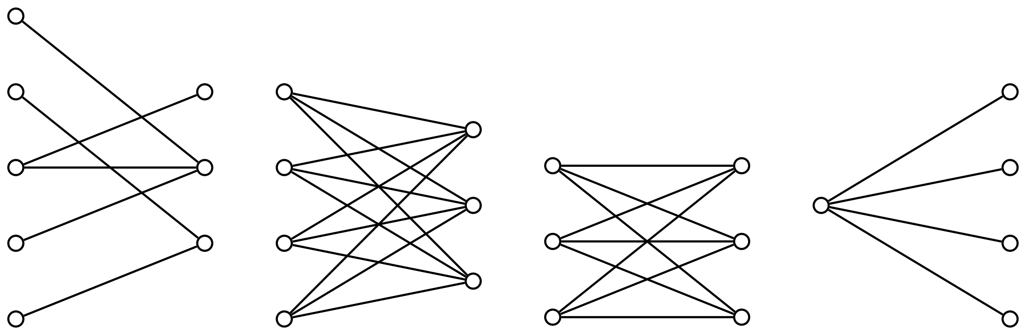
1.2.5 Graphe biparti complet

Un graphe biparti est complet si tout sommet de V_1 est adjacent à tout sommet de V_2(figure1.4(b)).
Un tel graphe est noté $K_{p,q}$ avec : $p =| V_1 |$ et $q =| V_2 |$.

1.2.6 Graphe biparti équilibré

Un graphe $G = (V_1 \cup V_2; E)$ est dit équilibré si et seulement si : chaque sous ensemble de la bipartition contient le même nombre de sommets ;$|V_1| = |V_2|$

Les cycles de longueurs pairs sont des graphes équilibrés, les graphes bipartis complets où $p = q$ sont aussi des graphes équilibré.
La figure 1.4(c) montre le graphe équilibré $K_{3,3}$.



(a) ***Biparti*** (b) ***Biparti complet*** (c) ***Graphe équilibré*** (d) ***Graphe complet***

FIGURE 1.4 – Graphes Particuliers

1.2.7 Arbre

Définition *1.2.1*
Un graphe G=(X,E) est un arbre s'il est connexe sans cycle.

1.2.8 Quelques propriétés des arbres

Théorème *1.2.1 (C.BERGE [3]) Soit G un graphe d'ordre n. Les propriétés suivantes sont équivalentes.*

1. *G est sans cycle et connexe.*
2. *G est sans cycle et possède $n-1$ arêtes.*
3. *G est connexe et possède $n-1$ arêtes.*
4. *G est sans cycles et minimal pour cette propriété (lorsqu'on lui ajoute une arête on crée un cycle et un seul).*
5. *G est connexe et maximal pour cette propriété (lorsqu'on lui supprime une arête quelconque on va le déconnecter.*
6. *Tout couple de sommet (u, v) est relie par une chaîne et une seule.*

Théorème *1.2.2 (Berge [3]) Un graphe G= (V, E) admet un arbre T comme graphe partiel si et seulement si G est connexe.*

Théorème *1.2.3 (Berge [3]) Un arbre T admet au moins deux sommets pendants.*

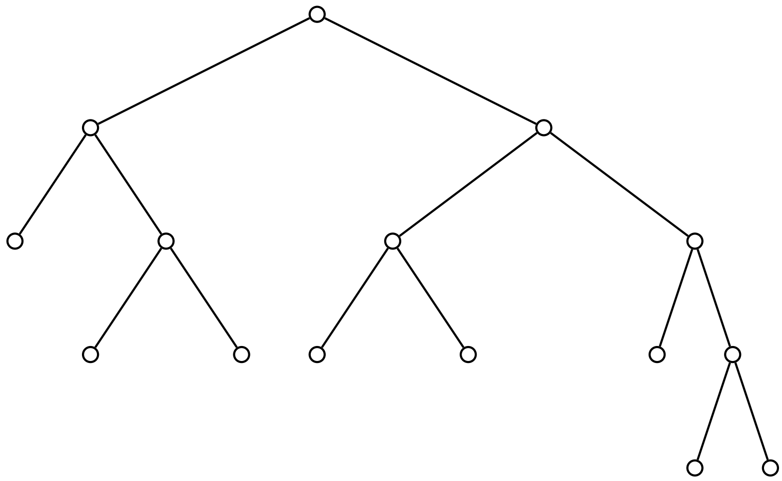


FIGURE 1.5 – Arbre.

1.3 Matrice d'adjacence

Définition *1.3.1*
Soit $G = (V, E)$ un graphe non orienté, avec $V = \{u_1, u_2, ..., u_n\}$:
La matrice d'adjacence du graphe G est la matrice $M(G) \in M_n(\mathbb{R})$ dont les coefficients $m_{i,j}$ sont définis par :

$$m_{i,j} = \begin{cases} 1 & si \quad (u_i, u_j) \in E \\ 0 & si \quad (u_i, u_j) \notin E \end{cases}$$

***Exemple** 1* L'étoile E_n à n branches (et $n+1$ sommets) a pour matrice d'adjacence

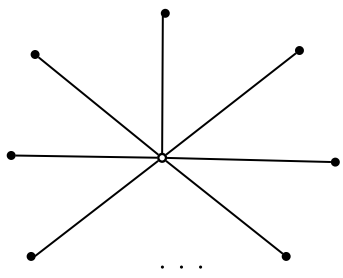


FIGURE 1.6 – Étoile E_n.

$$M_{E_n} = \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix}$$

1.4 Opérations classiques sur les graphes

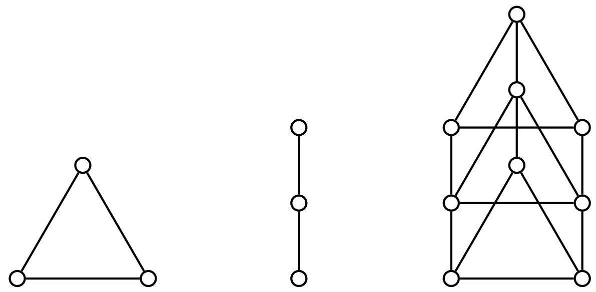
Étant donné deux graphes plusieurs opérations courantes permettent de construire d'autres graphes. Commençons par les plus simples.

1.4.1 Le produit cartésien de deux graphes

On appelle produit cartésien de deux graphes $G = (V, E)$ et $H = (V', E')$, noté $G\square H$, le graphe K dont l'ensemble de sommets est le produit cartésien $V(G) \times V(H)$ et où deux sommets (u,u') et (v, v') sont adjacents si et seulement si :

- $u = v$ et $u'v' \in E'$
  ou
- $u' = v'$ et $uv \in E$.

La figure 1.7 montre le produit cartésien $K_3\square P_3$. Il est à noter que le nombre de sommets dans $G\square H$ est $\mid V \mid . \mid V' \mid$
et que le nombre d'arêtes est $\mid V \mid . \mid E' \mid + \mid V' \mid . \mid E \mid$·



(a) K_3 **(b)** P_3 **(c)** $K_3 \square P_3$

FIGURE 1.7 – Produit cartésien de deux graphes

1.4.2 Morphismes de graphe

Morphisme

Dans un cadre général, un morphisme ou homomorphisme est une application entre deux ensembles qui « préserve » certaines propriétés.
Soit E un ensemble muni d'une relation binaire S et F un ensemble muni d'une relation binaire R, alors une application f de E dans F est un morphisme de (E, S) dans (F, R) si

$$\forall (x, y) \in E^2, xSy \Rightarrow f(x)Rf(y).$$

Morphisme de graphes

Dans le cadre des graphes, la relation considérée est communément la relation d'adjacence.
Soient deux graphes $G = (V, E)$ et $H = (V^{',E^{')}}$. Une application de V dans $V^{'}$ est un morphisme du graphe G dans le graphe H si

$$\forall (u, v) \in V^2, uv \in E \Rightarrow f(u)f(v) \in E^{'}$$

.

1.4.3 Isomorphisme

Deux graphes $G = (V, E)$ et $G^{'} = (V^{'}, E^{'})$ sont dits isomorphes si et seulement si il existe une application bijective $\varphi : V \rightarrow V^{'}$ qui vérifie la condition suivante :

$$xy \in E \Leftrightarrow \varphi(x)\varphi(y) \in E^{'}$$

pour toute paire de sommets x, y dans V.

Exemple *2*
Avec les graphes G et H de la figure 1.8, on voit facilement qu'on a un homomorphisme de G dans H mais pas de H dans G.

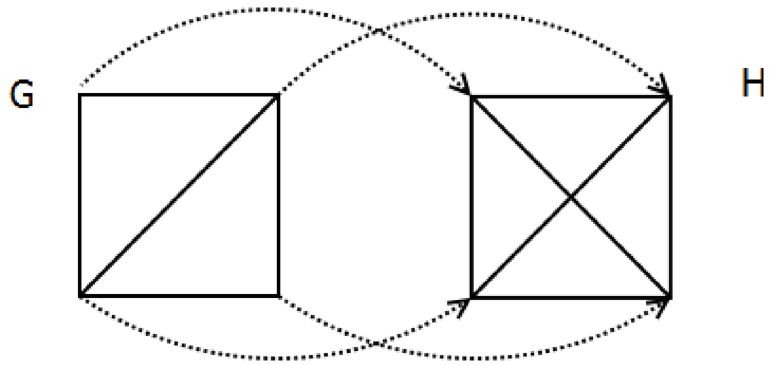


FIGURE 1.8 – Homomorphisme de G dans H.

A la figure 1.9 on donne un autre exemple d'homomorphisme entre deux graphes G et H. Cela montre que $f : V_1 \rightarrow V_2$ n'est pas nécessairement injectif.

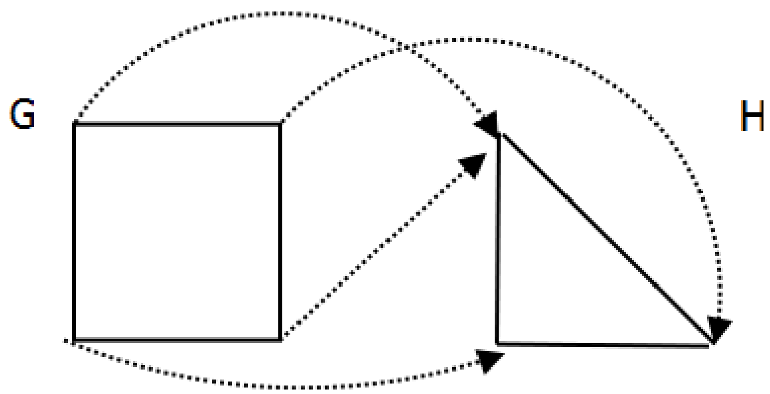


FIGURE 1.9 – Homomorphisme de G dans H.

1.5 Distances et intervalles dans les graphes

1.5.1 Distances dans les graphes

Étant donné deux sommets u et v d'un graphe $G = (V, E)$, on appelle distance entre u et v, la longueur d'une plus courte $(u, v)-$chaîne.
Une telle distance est notée $d_G(u, v)$ (ou $d(u, v)$ s'il n'y a pas de confusion).

- L'éxcentricité d'un sommet u noté $e_G(u)$ (ou $e(u)$ s'il n'y a pas de confusion) est le nombre suivant :

$$e_G(u) = Max \ \ \{d(u, v), v \in V(G)\}.$$

- La diamètre de G noté $D(G)$ (ou D s'il n'y pas de confusion) est la pus grande excentricité :

$$D(G) = Max_{u \in V}[e(u)]$$

- le rayon de G noté $R(G)$ (ou R s'il n'y pas de confusion) est la plus petite excentricité :

$$R(G) = Min_{u \in V}[e(u)]$$

- Le centre de G est l'ensemble des sommets de G dans l'excentricité est égale au rayon.

1.5.2 Intervalle dans les graphes

L'intervalle $I_G(u, v)$ (ou $I(u, v)$ s'il n'y pas de confusion) c'est l'ensemble des sommets de G appartenant aux plus courtes $(u, v)-$chaîne.

$$I_G(u, v) = \{w \in V, \text{ w est sur une plus courte (u,v)-chaine}\}$$

1.6 Les graphes médians

Les graphes médians ont été introduits indépendamment par Avann [1] et par Nebesky [30].

Définition *1.6.1*
Un graphe G est dit médian si :

$$|I(u,v) \cap I(u,w) \cap I(v,w)| = 1 \qquad \forall \;\; u,v,w \;\;\in V(G).$$

Autrement dit pour tout triplet de sommets u, v et w de G, il existe un unique sommet x qui vérifie :
$x \in I(u,v) \cap I(v,w) \cap I(u,w)$.
Ce sommet x est appelé médian.

1.7 Complexité des algorithmes

La complexité (temporelle) d'un algorithme est le nombre d'opérations élémentaires (affectations, comparaisons, opérations arithmétiques) effectuées par un algorithme. Ce nombre s'exprime en fonction de la taille n des données. On s'intéresse au coût exact quand c'est possible, mais également au coût moyen (que se passe-t-il si en moyenne sur toutes les exécutions du programme sur des données de taille n), au cas le plus favorable, ou bien au cas le pire. On dit que la complexité de l'algorithme est $O(f(n))$ où f est d'habitude une combinaison de polynômes, logarithmes ou exponentielles. Ceci reprend la notation mathématique classique, et signifie que le nombre d'opérations effectuées est borné par $cf(n)$, où c est une constante, lorsque n tend vers l'infini.

Considérer le comportement à l'infini de la complexité est justifié par le fait que les données des problèmes sont de grande taille et qu'on se préoccupe surtout de la croissance de cette complexité en fonction de la taille des données. Une question systématique à se poser est : que devient le temps de calcul si on multiplie la taille des données par 2 ?
De cette façon, on peut également comparer des algorithmes entre eux.

Les algorithmes usuels peuvent être classés en un certain nombre de grandes classes de complexité.

Les algorithmes sous-linéaires, dont la complexité est en général en $O(\log n)$. C'est le cas de la recherche d'un élément dans un ensemble ordonné fini de cardinal n.

Les algorithmes linéaires en complexité $O(n)$ ou en $O(n \log n)$ sont considérés comme rapides, comme l'évaluation de la valeur d'une expression composée de n symboles ou les algorithmes optimaux de tri.

Plus lents sont les algorithmes de complexité située entre $O(n^2)$ et $O(n^3)$,c'est le cas de la multiplication des matrices et de parcours dans les graphes.

Au delà, les algorithmes polynômiaux en $O(n^k)$ pour $k > 3$ sont considérés comme lents, sans parler des algorithmes exponentiels (dont la complexité est supérieure à tout polynôme en n) que l'on s'accorde à dire impraticables dès que la taille des données est supérieure à quelques dizaines d'unités.

Chapitre 2

L'hypercube

Introduction

Les propriétés des hypercubes les rendent intéressants pour la construction de machines dédiées au calcul parallèle. Au début des années 1960, des idées furent proposées pour concevoir un ordinateur parallèle avec une architecture en hypercube : il y a n puissance 2 modules, chacun connecté directement à n autres ; en particulier, chaque module est placé sur le sommet d'un cube à n-dimensions, et les arêtes de ce cube sont les câbles. Les justifications dans le choix de l'hypercube peuvent paraître faible au regard des connaissances actuelles sur les familles de graphes, mais il s'avère que l'hypercube a de nombreuses propriétés et plus d'une trentaine de caractérisation(Foldes [8]).
L'objectif de ce chapitre est de citer quelques propriétés des hypercubes qui utilisées dans la suite.

2.1 Le graphe de l'hypercube

L'hypercube de dimension n (noté Q_n) est le graphe de 2^n sommets qui peuvent être considérés comme étant tous les vecteurs booléen 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.

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

$$\underbrace{K_2 \square K_2 ... \square K_2}_{nfois}$$

est donc $Q_{n+d} = Q_n \square Q_d$.

La figure 2.1 montre les hypercubes de petites dimensions.

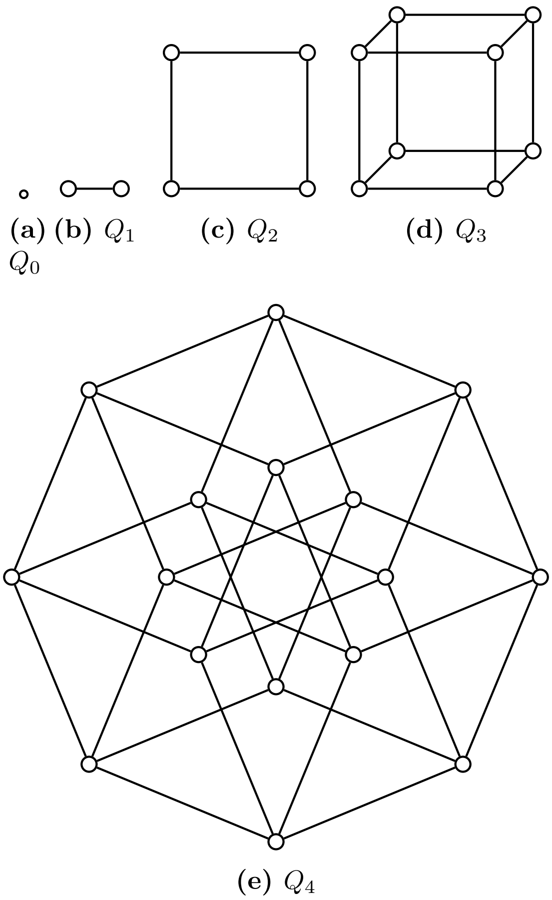


FIGURE 2.1 – Graphes de l'hypercube Q_n pour $n = 0, \ldots, 4$.

2.2 Propriétés élémentaires de l'hypercube

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 à exactement n voisins : tout sommet a pour degré n, autrement dit le graphe est n-régulier.

Nombre de sommets : Par la construction recursive, 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

$$|V_n| = 2 * |V_{n-1}| = 2^2 * |V_{n-2}| = 2^n,$$

c'est-à-dire que le graphe a 2^n sommets.

2.3 Quelques caractérisations de l'hypercube

Une première caractérisation de l'hypercube est due à Foldes[8].

Théorème *2.3.1 un graphe connexe $G = (x, y)$ est un hypercube si et seulement s'il vérifie les conditions suivantes :*

1. *G est biparti.*
2. *pour tout couple de sommets u et v de G, le nombre de plus courtes (u, v)-chaînes est $d(u, v)!$.*

Laborde - Rao Hebbare [25] et Mulder [28] ont indépendamment démontre le théorème 2.3.1 qui repose sur le concept de $(0, 2)$-graphe.

Définition *2.3.1* Un graphe connexe G est un $(0, 2)-$*graphe* si et seulement si toute paire d'arêtes adjacentes de G appartient à exactement un cycle de longueur 4.

Cette même définition peut se formuler de manière équivalente en termes de sommets et de chaînes.

Définition *2.3.2* (Mulder) [29]
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 [29] a donné les propositions suivantes sur les $(0, 2) -$ *graphes*.

Proposition (Mulder) [29]
Si G est un $(0,2)-graphe$; alors G est régulier.

Proposition (Mulder) [29]
Si G est un $(0,2)-graphe$ de degré n, alors $|V(G)| \leq 2^n$.

Théorème *2.3.2 (Mulder, Laborde, Rao) [25, 28]*
Soit $G=(V,E)$ un $(0,2)-graphe$, alors

1. *G est régulier de degré n*
2. *$|V(G)| \leq 2^n$.*
3. *$|V(G)| = 2^n$. si et seulement si G est un hypercube de dimension n.*

Une autre caractérisation de l'hypercube est en terme d'intervalle donnée par Mulder et Bandelt [2].

Théorème *2.3.3 (Mulder et Bandeit) [2] Soit G un graphe biparti connexe, les conditions suivantes sont équivalentes :*

1. *G est un hypercube ;*
2. *Tout intervalle dans G engendre un hypercube ;*
3. *Tout intervalle dans G engendre un $(0,2)-graphe$;*
4. *Tout intervalle $I(u,v)$ dans G contient exactement $2^{d(u,v)}$ sommets ; ,*
5. *Tout intervalle $I(u,v)$ dans G engendre un graphe avec exactement $d(u,v).2^{d(u,v)-1}$ arêtes.*

Théorème *2.3.4 (Mulder) [29]*
Un graphe G connexe $G=(V,E)$ est un hypercube si et seulement si G est un graphe médian régulier.

Proposition (Muldcr) [29]

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 caractérisations en terme de graphes distance monotones, [6], en terme de graphes intervalles reguliers et des graphes sphériques [8] peuvent être trouvés.

Définition *2.3.3*
Une projection (resp. anti projection) d'un sommet u d'un graphe G sur un ensemble S de G et 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 projections (resp. anti projections) de u sur S.

Considérons les propriétés suivantes :

1. Pour tout triplet de sommets (u, v, w) on a $|P(u, I(v, w))| = 1$ (P1)
2. Pour tout triplet de sommets (u, v, w) on a $|AP(u, I(v, w))| = 1$ (P2)

Un graphe vérifiant l'une de ces deux propriétés est un graphe biparti. [27] a donné une caractérisation de l'hypercube en termes d'anti projection sur les intervalles.

Proposition Un graphe $G = (V, E)$ est un hypercube si et seulement s'il vérifie la propriété 2.

Berrachedi [4] a donné un résultat analogue en considérant les projections sur les intervalles.

Définition *2.3.4*
$(0, 2) - \textit{graphe}$ qui vérifie la propriété 1 est un hypercube.

Chapitre 3

Plongement de graphes dans l'hypercube

Introduction

La mise en oeuvre d'algorithmes parallèles sur des architectures multiprocesseurs à mémoire distribuée a conduit au développement de la notion de plongement d'un graphe G dans un graphe H.
Bien souvent, un algorithme distribué A est décrit en supposant l'existence d'une topologie logique S, sur laquelle A est défini. Parmi les topologies logiques les plus utilisées, se trouvent les arborescences et les hypercubes. Un plongement permet à ce qu'un réseau soit simulé par un autre : aux sommets du réseau d'origine sont associés des sommets dans le réseau simulant, et deux sommets voisins sont séparés par un chemin. Ainsi, un algorithme conçu spécialement pour un réseau peut être réutilisé dans un autre grâce à un plongement.

La qualité d'un plongement permet de savoir quelles sont les différentes pertes en performance de l'algorithme. Pour cela, on considère plusieurs facteurs (Behrooz Parhami[34]) : Charge,Dilatation et Congestion.

Le problème de plongement de graphes dans l'hypercube à été traité par plusieurs auteurs. Ainsi, I.Havel [15, 19, 22, 20] Harary [10, 12], Nebesky [31, 33] et plusieurs auteurs ont donné des familles de graphes qui sont plongeables dans l'hypercube.

3.1 Définitions générales

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 xy de G, une chaîne reliant les sommets $\varphi(x)$ et $\varphi(y)$ dans H.

3.1.1 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 $dil(\varphi)$ est la longueur maximale des chaînes $P_\varphi(xy)$ de H, associées aux arêtes xy de G. Dans le cas où l'on considère des chaînes de plus courte longueur, la longueur de $P_\varphi(xy)$ est alors égale à la distance $d_H(\varphi(x), \varphi(y))$ et la dilatation s'exprime uniquement en fonction de φ, par :

$$dil(\varphi) = \underset{xy \in E(G)}{Max}\, d_H(\varphi(x), \varphi(y))$$

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 xy de G est l'arête $\varphi(x)\varphi(y)$ de H. Si de plus $|V(G)| = |V(H)|$, alors G est un graphe partiel de H.

Exemple *3*
La grille binaire $M(2, 4)$ est plongeable dans l'hypercube Q_3 avec dilatation 1.

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 $cong(\varphi)$, est le maximum, pris sur toutes les arêtes e de H, du nombre de chaînes $P_{\varphi}(xy)$ de H, images d'arêtes de G, qui contiennent e.
Dans la littérature, le calcul de la congestion n'est généralement fait que pour un plongement qui optimise d'abord des contraintes sur la dilatation et l'expansion. Cependant, il semblerait que ce paramètre soit plus important que la dilatation (dépendant des modèles de communication) et qu'il serait intéressant de le minimiser, sans tenir compte de la dilatation.

3.2 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 $dim(G)$.

Firsov [7] a remarqué que 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 3.1). $K_{2,3}$ 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.

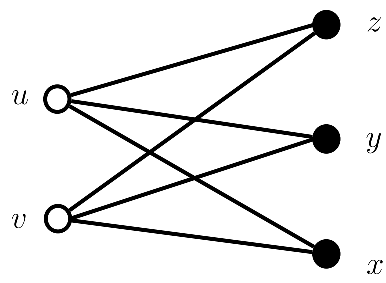


FIGURE 3.1 – Graphe $K_{2,3}$

3.2.1 Décider si un graphe G est cubique

Havel et *Morávek* [18] ont donnés les conditions nécessaires et suffisantes pour dire si un graphe G donné est cubique. Ce concept est connu sous le nom de la Cn-valuation définie comme suit :
Un graphe G est dit Cn-valuaé si et seulement si on peut étiqueter les arêtes de G par des entiers appartenant à l'ensemble $\{1, ..., n\}$, de telle sorte que :

1. Toutes les arêtes de G incidentes à un même sommet x admettent des étiquetages différents,
2. Pour toute chaîne P de G, il existe un entier $i \in \{1, ..., n\}$ qui apparaît un nombre impair de fois dans l'étiquetage des arêtes de P,
3. Pour tout cycle C de G, aucun entier $i \in \{1, ..., n\}$ n'apparaît un nombre impair de fois dans l'étiquetage des arêtes de C.

En utilisant ces conditions, Havel et Liebl [21, 22] ont montré, indépendamment du résultat de Firsov [7], que les arbres sont des graphes cubiques et ont donné des plongements efficaces des arbres binaires dans l'hypercube. Ils ont aussi montré que les cycles sont des graphes cubiques si et seulement si ils sont d'ordres pairs.

3.3 Plongement de graphes dans l'hypercube

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 .

3.3.1 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, $\triangle(G) \leq n$.

Toutes ces conditions sont nécessaires pour que un graphe G qui est plongeable dans Q_n, mais pas suffisantes, comme le montre l'exemple suivant (figure 3.2) :

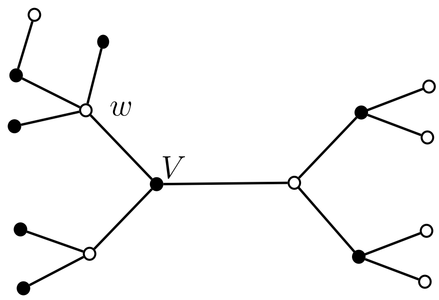


FIGURE 3.2 – Arbre équilibré à 16 sommets qui n'est pas plongeable dans Q_4

Cet arbre T n'est pas plongeable dans Q_4 pour la simple raison que dans T tous les sommets sont à distance au plus 3 du sommet v, alors que dans Q_n, pour tout sommet donné v, il existe un unique sommet u tel que $d(u, v) = n$ (Q_n est antipodal). L'antipodal de v devrait être à distance 4 dans $Q4$.

3.4 Quelques classes de graphes plongeable dans l'hypercube

3.4.1 plongement des arbre dans Q_n

Quelques propriétés des arbres

Théorème *3.4.1 (Havel) [17]*
Un arbre T est plongeable dans Q_n si et seulement s'il existe une $C_n -$ valuation de T.

Soit T un arbre d'ordre n. L'hypercube $Q_{dim(T)}$ est appelé hypercube optimal de T. D'après la remarque précédente tout arbre est plongeable dans l'hypercube, donc on s'intéressera à la recherche de dimension de ces arbres (plongement optimal de ces arbres dans l'hypercube).

3.4.2 Arbre Binaires

Définition *3.4.1*
Un arbre T est dit binaire si son degré maximum $\Delta(T) \leq 3$. Un résultat concernant les arbres binaires a été donné par I.Havel [16].

Proposition (I.Havel) [16]
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 est plongeable dans Q_n.

Définition *3.4.2* 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 u, tel que u est relié par une arête à un sommet de degré 2 de T_1 et par une autre arête à un sommet de degré 2 de T_2.

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 possède $2^{n+1} - 1$ sommets.

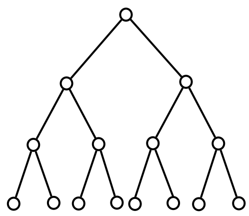


FIGURE 3.3 – L'arbre D_3

Définition *3.4.3* D_n peut être défini aussi de la manière suivante $D_n = (V, E)$ où V c'est l' ensemble $(1, 2, ..., 2^{(n+1)} - 1)$ de sommets, deux sommets p et q sont adjacents si l'une des conditions suivantes est vérifiées :

1.) $1 \leq min(p, q) \leq 2^n - 1$ et $max(p, q) = 2 \;\; min(p, q)$.
2.) $1 \leq min(p, q) \leq 2^n - 1$ et $max(p, q) = 2 \;\; min(p, q) + 1$.

Le résultat suivant est dû à I.Havel [14].

Théorème *3.4.2 (I.Havel [14]) Soit $n \geq 2$. D_n est plongeable dans Q_{n+2} et $dim(D_1) = 2$, $dim(D_n) = n + 2$. A partir de l'arbre binaire complet D_n on définira d'autres arbres plongeables dans l'hypercube.*

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 un arête.

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.

Exemple *4*

I.Havel [16] a montré que B_n est plongeable dans Q_{n+1}, donc D_n est plongeable dans Q_{n+2}.

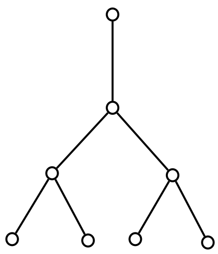


FIGURE 3.4 – L'arbre B_3

Théorème *3.4.3 (I.Havel) [16]*
Pour tout $n \geq 2$, B_n est plongeable dans Q_{n+1}, $dim(B_n) = n+1$. L'arbre B_n peut être généralise comme suit :

soit $n \geq 2$ et $k \geq 1$, on peut définir l'arbre noté $B_n^{(k)}$ de la manière suivante : $B_n^1 = B_n$ et $B_n^{(k)}$ est l'arbre obtenu par subdivision de chaque arête de l'arbre binaire, complet D_{n-1} par $k-1$ sommets et l'arête pendante de B_n adjacente à la racine de D_{n-1} par k sommets.

Il est clair que $|V_n^{(k)}| = K.2^{n+1}$ I.Havel [16] a démontré aussi la proposition suivante,qui concerne le plongement de B_n^2 dans l'hypercube.

Proposition (Havcl) [16] Pour $n \geq 2$, $|V(B_n^2)| = 2^{n+2}$,et B_n^2 est plongeable dans Q_{n+2}, $dim(B_n^2) = n+2$.

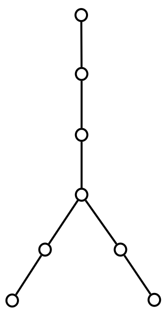


FIGURE 3.5 – L'arbre D_1^2

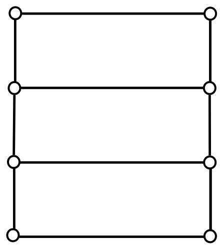
3.5 Autres graphes plongeables dans Q_n

3.5.1 Plongements des grilles et des échelles

Définition *3.5.1* Une n-grille $M = M(d_1 \times d_2 \times ... \times d_n)$ est le produit cartésien de n chaînes $P_1, P_2, ..., P_n$ d'ordres respectifs $d_1, d_2, ..., d_n$.

Grilles binaires

Une grille est dite binaire si d_i est une puissance de 2 pour tout i,en particulier. Si $d_i = 2 \ \ \forall i$, alors M est l'hypercube de dimension n.



$M(2 \times 4) = P_2 \square P_4$

FIGURE 3.6 – Grille binaire

Exemple *5* Harary et Lewinter [13] ont montré que les grilles binaires sont des graphes cubiques. La démonstration repose sur le lemme suivant :

Lemme 3.1 (Hanlry et Lewinter [13]) Si G et H deux graphes plongeables dans G' et H' respectivement, alors $G \square H$ est plongeable dans $G' \square H'$.

Preuve : on a $V(G \square H) \subseteq V(G' \square H')$, et par définition de la somme cartésienne, on a $V(G \square H) \subset V(G' \square H')$.

Théorème *3.5.1 ((Kobeissi) [24])*
La n-grille est plongeable dans Q_m si et seulement si $d_1 \times ... \times d_n \leq 2^m$

Preuve : (M. Kobeissi) [24]

La condition nécessaire vient du fait que $|V(M)| = d_1.d_2...d_n$, alors que $|V(Q_m)| \leq 2^m$.
Maintenant supposons que $d_1.d_2...d_n = 2^m$. Dans ce cas M est le produit cartésien de chaînes d'ordres d_i (ou d_i est une puissance de 2 pour tout i). Chaque chaîne d'ordre d_i se plonge dans son hypercube optimal de dimension $\log d_i$ car l'hypercube est hamiltonien,

Échelles

Définition *3.5.2* Soient $P_1 = a_1, a_2, ..., a_k$ et $P_2 = b_l, b_2, ..., b_k$ deux chaînes d'ordres K, tel que les sommets a_i, et b_i $(i = 1...K)$ sont reliés par des chaînes d'ordres $r_1, r_2, ..., r_k$ de sorte que les extrémités de r_i soient reliée l'une par une arête à a_i et l'autre par une autre arête à b_i. Pour $i = 1, ..., K$ (a_i et b_i seront reliés par une arête si $r_i = 0$) .Un tel graphe est appelé grille . Les chaînes entre a_i el b_i sont appelées les rangs.

Exemple *6*

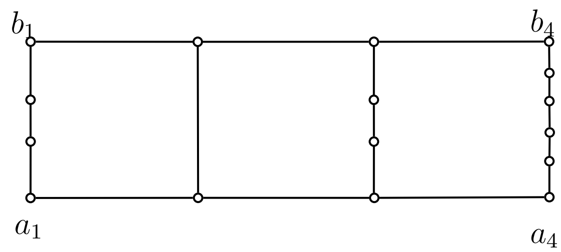


FIGURE 3.7 – Échelle de rangs $2, 0, 2$ *et* 4

Le théorème suivant sur le plongement des échelles dans l'hypercube est dû à Bezrukov [5]

Théorème *3.5.2 Toute échelle équilibrée E est plongeable dans son hypercube optimal.*

3.5.2 Plongement des quasi-étoiles et des double quasi-étoiles

Après avoir rappeler des résultats sur le plongement de certains graphes dans l'hypercube, nous nous intéressons à présent dans cette section à deux familles de graphes, à savoir les quasi-étoiles et les doubles quasi-étoiles. Par opposé aux plongements précédents qui concernaient les graphes à faible degré, on s'intéresse ici aux plongements de graphes ayant un degré maximum par rapport à l'hypercube. Une étoile est, comme son nom l'indique, un arbre avec exactement un sommet u qui n'est pas pendant. Ce sommet est appelé jonction, et son degré est le nombre d'arêtes qui sont incidentes à u. Un tel arbre est noté $K_{1,n}$

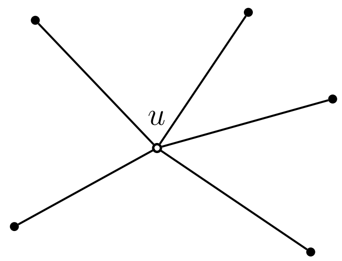


FIGURE 3.8 – Étoile $K_{1,n}$

Il est évident qu'une étoile est cubique et que $dim(K_{1,n}) = n$. Une quasi-étoile est une étoile dont les arêtes sont subdivisées. Le lemme 3.2 donne une condition nécessaire et suffisante pour qu'une quasi-étoile soit équilibrée.

Plongement doubles quasi-étoiles

Lemme 3.2 (I.Havel) [29] Une quasi-étoile est équilibrée si est seulement si elle possède exactement une seule chaîne de longueur impaire.

Le degré d'une quasi-étoile est le nombre de chaînes qui sont incidentes u . On notera $S(a_1, a_2, ..., a_K)$ une quasi-étoile de degré K, dont les chaînes sont d'ordres respectifs $a_1 a_2, ..., a_K$. Une quasi- étoile de degré K, à 2^n sommets, est appelée K-quasi-étoile.

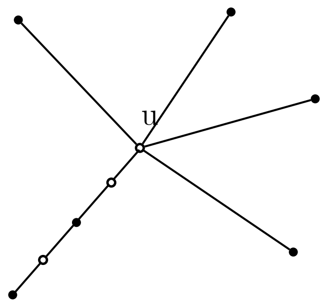


FIGURE 3.9 – Quasi-étoile $S(1,1,1,1,4)$

Définition *3.5.3* Une double étoile est formée de deux étoiles dont les jonctions u et v sont reliées par une arête.
A noter que u et v ne sont pas de même degré.

Définition *3.5.4* Une double quasi-étoile est une subdivision d'une double étoile, dans laquelle l'arête reliant u et v n'est pas subdivisée .
Une double quasi-étoile dont les sommets u et v sont de degrés respectifs k et s $(k \geq s)$, est notée $S(a_1, ..., a_k; b_1, b_2, ..., b_s)$.La figure 3.10 montre un example de tel graphe.

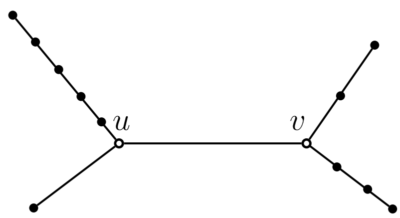


FIGURE 3.10 – Double quasi-étoile $S(1,4;2,3)$

Une double quasi-étoile est équilibrée si est seulement si le nombre de chaînes d'ordres impairs qui sont incidentes à u est égale au nombre de chaînes d'ordres impairs incidentes à v.
Une double quasi-étoile équilibrée $S(a_1, a_2, ..., a_K, b_1, b_2, ..., b_s)$ est appelé K-double quasi-étoile équilibrée.

I.Havel [9] a montré que les 3-quasi-étoiles équilibrée à 2^n sommets sont plongeables dans Q_n, $(Nebesk\grave{y})$ [32] a étendu ce résultat aux 4-quasi-étoiles équilibrées et 5-quasi-étoiles équilibrées. Limaye [26] a prouvé ce même résultat pour $k = 6$.

Théorème *3.5.3 Toute k-quasi-étoile équilibrée à* 2^n *sommets avec* $(k \leq n)$ *est plongeable dans* Q_n*.*

Chapitre 4

Plongement de copies de graphes dans l'hypercube

Introduction

L'objectif considéré portait sur la détermination de nombre maximum de copies disjointes de graphe qu'on peut plonger dans un hypercube de dimension n.

4.1 Nombre de copies de $K_{1,3}$ dans Q_n

Dans cette section on cherche à déterminer le nombre de copies de $K_{1,3}$ dans un hypercube de dimension n.
Soit $F_n(k_{1,3})$ le nombre de copies de $k_{1,3}$ dans Q_n avec $n \geq 3$.
On choisit un sommet x dans Q_n parmi 2^n sommets pour construire $K_{1,3}$, ensuite multiplions par nombre de combinaison de 3 parmi n (C_n^3) sommets adjacent au sommet x, et comme y 2^n sommets dans Q_n, alors il faut multiplier par(2^n).

$$F_n(k_{1,3}) = C_n^3 \;\; 2^n = \frac{n!}{(n-3)!3!} \cdot 2^n \qquad\qquad n \geq 3$$

Exemple *7* Le nombre de copies de $K_{1,3}$ dans Q_3 .

$$F_3(K_{1,3}) = C_3^3 \ \ 2^3 = 8$$

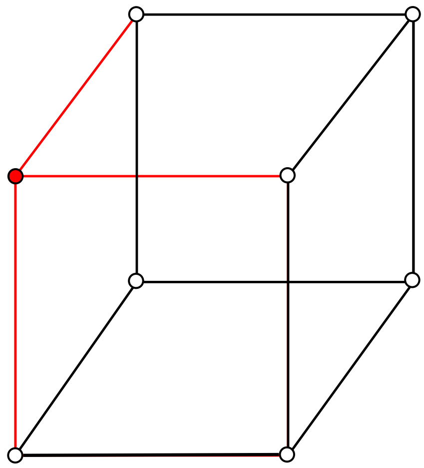


FIGURE 4.1 – Nombre de copies de $K_{1,3}$ dans Q_3

4.2 Nombre de copies de $K_{1,v}$ dans Q_n

Notons par $F_n(K_{1,v})$ le nombre de copies de $K_{1,v}$ (étoile E_v) dans Q_n, $(v+1)$ le nombre de sommets de $K_{1,v}$ et v nombre d'arêtes.
Il y a exactement $2^n \cdot C_n^v$ copies de $K_{1,v}$ dans Q_n avec $n \geq v$.

En effet, il faut tout d'abord choisir un sommet parmi 2^n dans Q_n pour construire $K_{1,v}$, ensuite il faut multiplier par le nombre de combinaison possible (C_n^v) et de faire avec 2^n sommets de (Q_n).
La formule qui déterminer le nombre de copies de $K_{1,v}$ dans Q_n.

$$F_n(k_{1,v}) = C_n^v \cdot 2^n = \frac{n!}{(n-v)!v!} \cdot 2^n \qquad\qquad n \geq v$$

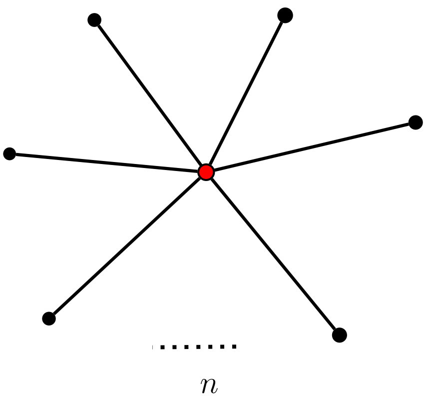


FIGURE 4.2 – $K_{1,n}$

4.3 Nombre de copies de double quasi-étoile $S(1,1,1;1,1,1)$ dans Q_n

Notons par $F_n(s)$ le nombre de copies de $S(1,1,1;1,1,1)$ dans Q_n figure 4.3,

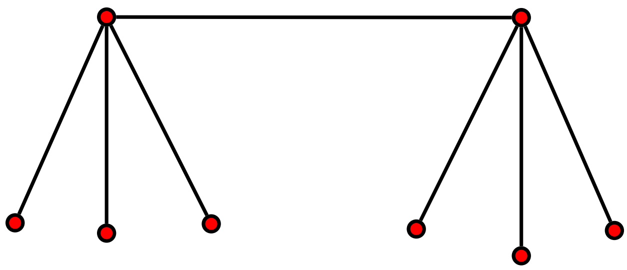


FIGURE 4.3 – 3-Double quasi-étoile $S(1,1,1;1,1,1)$

$$F_n(s) = C_n^3 \cdot C_n^3 \quad \cdot 2^{n-1} = (\frac{n!}{(n-3)!3!})^2 \cdot 2^{n-1} \qquad\qquad n \geq 4$$

4.4 Plongement de copies disjointes de Q_i dans Q_n

On sait que dans un hypercube de dimension n (Q_n) on peut plonger deux copies disjointes de Q_{n-1}, et dans Q_{n-1} on peut plonger deux copies disjointes de Q_{n-2}, donc par induction sur n, on obtient :
le nombre maximum de copies disjointes de Q_i que on peut plonger dans Q_n est 2^{n-i} avec $0 \leq i \leq n$.

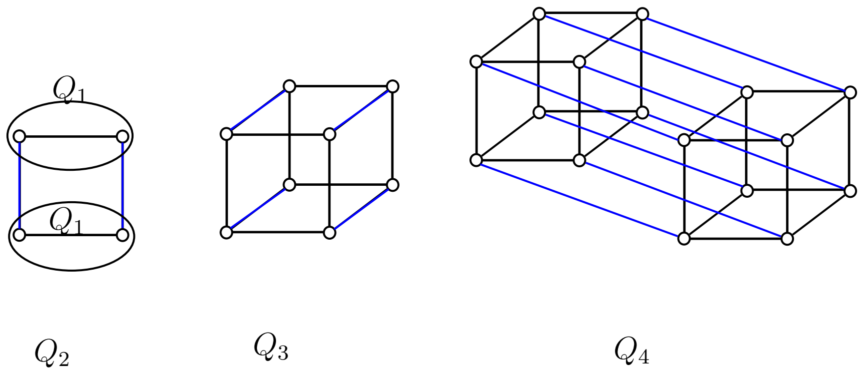


FIGURE 4.4 – Nombre de copies de Q_{n-1} dans Q_n

4.5 Plongement de copies disjointes de $K_{1,3}$ dans Q_n

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.

Soit DF_n le nombre de copies disjointes de $K_{1,3}$ qu'on peut plonger dans Q_n.
Pour $n = 3$ on a $DF_3 = 2$ et

Pour $n \geq 4 \quad DF_n = 2 \cdot DF_{n-1}$ Par induction sur n on obtient :

$$\forall n \geq 3 \quad DF_n = 2^{n-2}$$

Le nombre de copies maximum qu'on peut plonger dans Q_n est 2^{n-2}.

Exemple *8* Plongement de copies disjointes de $k_{1,3}$ dans Q_3

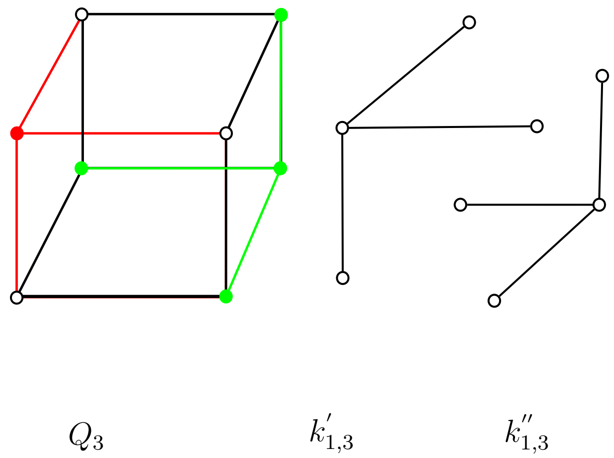


FIGURE 4.5 – Plongement de copies disjointes de $K_{1,3}$ dans Q_n

4.6 Plongement de copies disjointes de double quasi-étoile $S(1,1,1;1,1,1)$ dans Q_n

Soit DS_n le nombre de copies disjointes de $S(1,1,1;1,1,1)$ qu'on peut plonger dans Q_n. Pour $n = 4$ $DS_4 = 2$ et pour $n \geq 5 \quad DS_n = 2 \cdot DS_{n-1}$. Par induction sur n on obtient :

$$\forall n \geq 4 \quad DF_n = 2^{n-3}$$

Dans un hypercube de dimension n on peut plonger 2^{n-3} copies disjointes de $S(1,1,1;1,1,1)$.

Exemple *9*

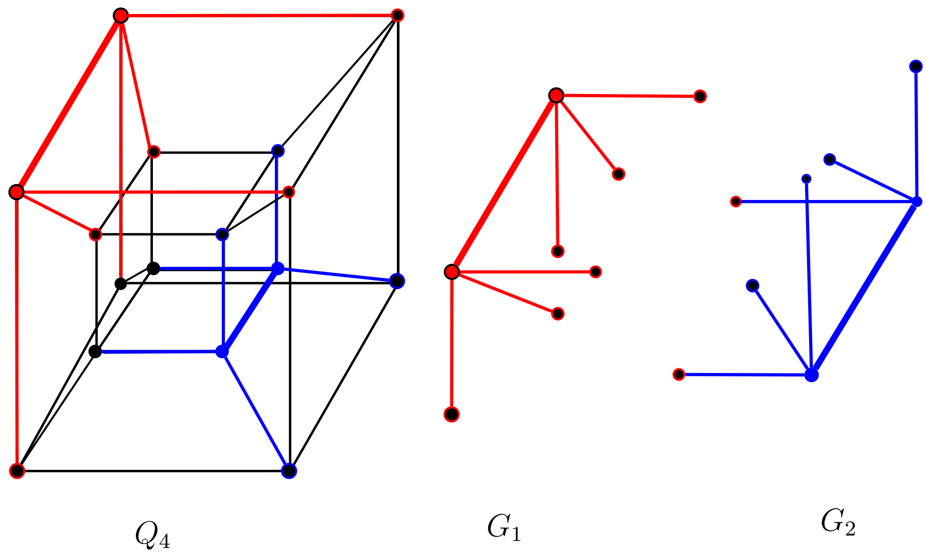


FIGURE 4.6 – Plongement de copies disjointes de double quasi-étoile $S(1,1,1;1,1,1)$ dans Q_n

4.7 Plongement de copies disjointes du graphe $S_k(1,1,1;1,1,1)$ dans Q_n

Le graphe $S_k(1,1,1;1,1,1)$ est l'union de k copies de double quasi-étoile $S(1,1,1;1,1,1)$ par une arête, comme il le montre la figure4.7.

Soit $DS_n^{(k)}$ le nombre de copies disjointes de $S_k(1,1,1;1,1,1)$ qu'on peut plonger dans Q_n.
pour $n = k$ et $k \geq 4$ on a $DS_n^{(k)} = 2$ et pour $n > k \geq 4 \quad DS_n^{(k)} = 2 \cdot DS_{n-1}^{(k)}$
Par induction sur n on obtient :

$$\forall n \geq k \geq 4 \quad DS_n^{(k)} = 2^{n-k+1}$$

Donc dans l'hypercube Q_n on peut plonger 2^{n-k+1} copies disjointes de $S_k(1,1,1;1,1,1)$.

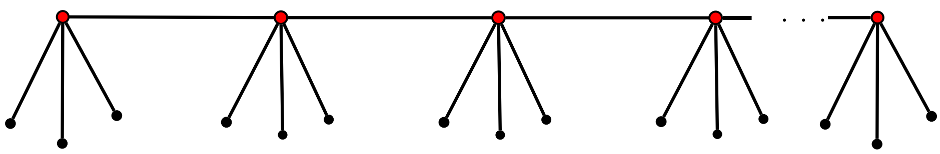


FIGURE 4.7 – Le graphe $S_k(1,1,1;1,1,1)$

4.8 Plongement de copies disjointes de la grille binaire $M(2,4)$ dans Q_n

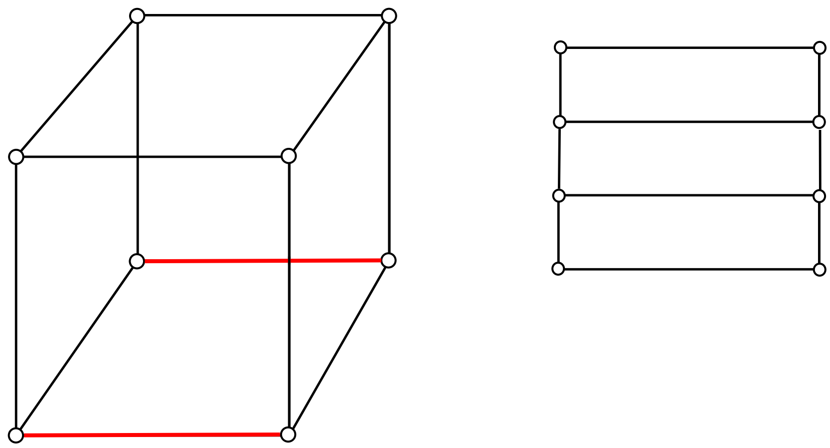
Soit $M_{2,4}(Q_n)$ le nombre de copies disjointes de graphe $M(2,4)$ qu'on peut plonger dans Q_n.

pour $n = 3$ $M_{2,4}(Q_3) = 1$
pour $n \geq 4$
$M_{2,4}(Q_n) = 2 \cdot M_{2,4}(Q_{n-1})$
par induction sur n on obtient :

$$\text{donc} \quad \forall n \geq 3 \quad M_{2,4}(Q_n) = 2^{n-3}$$

Ce résultat montre que dans l'hypercube Q_n on peut plonger 2^{n-3} copies disjointes de $M(2,4)$.

Exemple *10* plongement de la grille binaire(2,4) dans Q_3



Q_3 $\qquad$ $M(2 \times 4) = P_2 \square P_4$

FIGURE 4.8 – Plongement de la grille binaire(2,4) dans Q_3

4.9 Plongement de copies disjointes de la grille binaire $M(2,8)$ dans Q_n

Soit $M_{2,8}(Q_n)$ le nombre de copies disjointes de $M(2,8)$ qu'on peut plonger dans Q_n.
Pour $n = 4$ $M_{2,8}(Q_4) = 1$

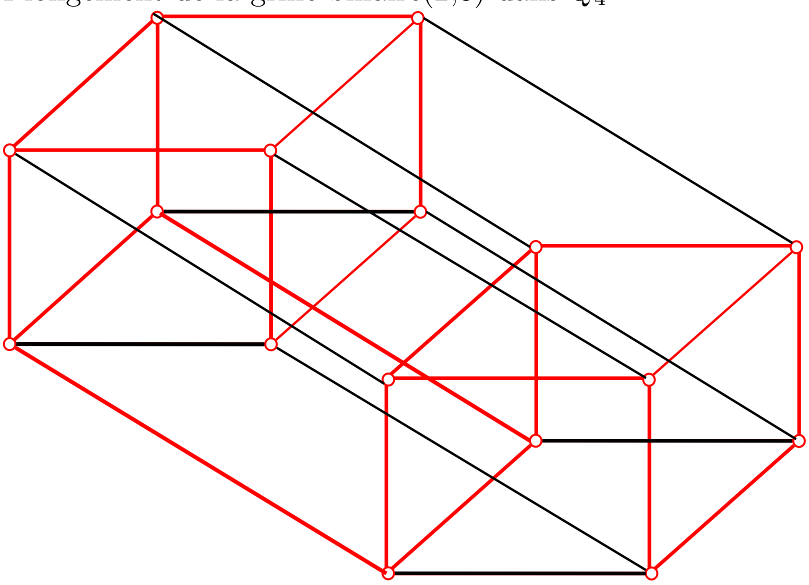
$$M_{2,8}(Q_n) = 2 \cdot M_{2,8}(Q_{n-1}) \qquad \forall n \geq 5$$

Par induction sur n on obtient :

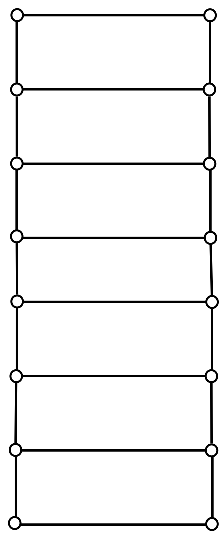
$$M_{2,8}(Q_n) = 2 \cdot M_{2,8}(Q_{n-1}) \qquad n \geq 4$$

Ce résultat montre que dans l'hypercube Q_n on peut plonger 2^{n-3} copies disjointes de $M(2,8)$.

Plongement de la grille binaire(2,8) dans Q_4



Q_4



$M(2 \times 8) = P_2 \square P_8$

FIGURE 4.9 – Plongement de la grille binaire(2,8) dans Q_4

4.10 Plongement de copies disjointes de grille binaire $M(2, 4 \cdot 2^k)$ dans Q_n

On pose $M_{2,4\cdot 2^k}(Q_n)$ avec $k \in N$ est le nombre de copies disjointes de $M(2, 4 \cdot 2^k)$ qu'on peut plonger dans Q_n, on obtient ainsi

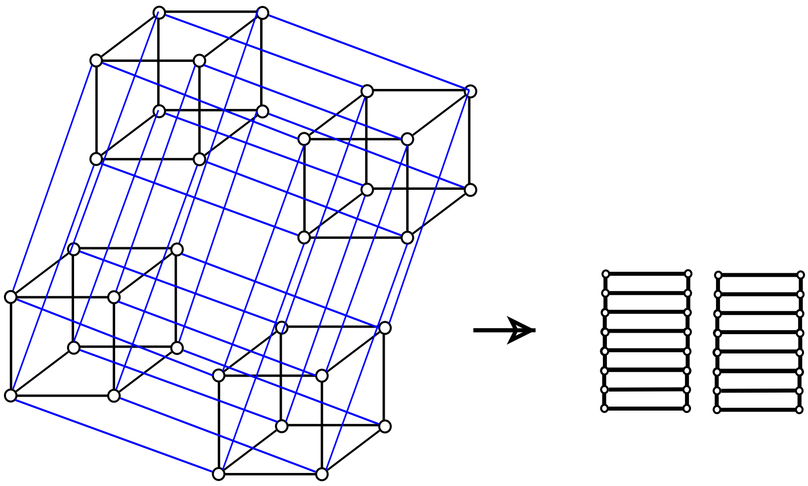
$$M_{2,4\cdot 2^k}(Q_n) = 2 \cdot M_{2,4\cdot 2^k}(Q_{n-1}) \quad \text{avec} \quad n \geq k+3 \geq 4.$$

$$\text{Pour} \qquad n = k+3 \quad M_{2,4\cdot 2^k}(Q_n) = 1$$

Par induction sur n on obtient :

$$\forall k \geq 0, \forall n \geq 2^k + 2 \quad M_{2,4\cdot 2^k}(Q_n) = 2^{n-(k+3)}$$

Ce résultat montre que dans l'hypercube Q_n on peut plonger $2^{n-(k+3)}$ copies disjointes de $M(2, 4 \cdot 2^k)$.



Q_5 $\qquad$ $M(2 \times 8) = P_2 \Box P_8$

FIGURE 4.10 – Plongement de 2 grille binaire(2,8) dans Q_5

4.11 La mise en oeuvre de l'algorithme

Dans cette section, nous décrivons deux algorithmes qui permettent de déterminer le nombre maximum de copies disjointes de graphes $k_{1,3}$ et double quasi-étoile $s(1,1,1;1,1,1)$, dans un hypercube de dimension n.

Le but de cet algorithme est d'exposer une méthode de décomposition de l'hypercube en copies disjointes. La nécessite d'une telle décomposition apparaît par exemple lorsqu'on veut répartir un ensemble de composants électroniques sur des cartes ou des plaquettes en minimisant les connexions inter-cartes ;

4.11.1 Algorithme de calcul du nombre de copies de $K_{1,3}$ dans Q_n

Entrée :n dimension de l'hypercube.
Sorte N nombre de copies de $k_{1,3}$.
s : **Tableau** ;
V : **Tableau** ;
x_0 : **entier** ;
x_1 : **entier** ;

$v \leftarrow [1, 2, ..., 2^n]$
Si $(n < 3\ )$ **Alors**
  $N \leftarrow 0$ *[Les hypercubes de dimension < 3]*
Sinon
  $x_0 \leftarrow$ sommet initial d'étiquette $\{0\}^n$
  Tant que $(v \neq \varnothing)$ **faire**
    s =procedure_sommets_adjacent(x_0, k)
    [La procédure retourne un ensemble S de k sommets adjacents]
    $V \leftarrow V \cap (V \setminus s)$
    Ncopie $\leftarrow$ Ncopie + 1 ;
    Si $(itra \mod 2 = 0)$ **Alors**
      $x_1 \leftarrow$ le sommet a distance 2 de x_0
      [x_1 Le sommet suivant]
    Sinon
      $x_1 \leftarrow$ le sommet adjacent de x_0
    Fin Si
    $x_0 \leftarrow x_1$
  Fait
Fin Si
Retourne (N)

Algorithme 1: Algorithme de calcul du nombre de copies de $K_{1,3}$ dans l'hypercube Q_n

***Exemple** 11* Appliquons de l'algorithme précédent pour trouver N nombre de copies de $k_{1,3}$ dans (Q_i avec $i = 4..12$).

Q_i	nombre de copies	Durée d'exécution (Secondes)
Q_4	4	0.0283
Q_5	8	0.1077
Q_6	16	0.4362
Q_7	32	1.8276
Q_8	64	7.0125
Q_9	128	28.7698
Q_{10}	256	115.0460
Q_{11}	512	1900.9
Q_{12}	1024	9330.2

4.11.2 Algorithme de calcul du nombre de copies de $S(1,1,1;1,1,1)$ dans Q_n

Appliquons de l'algorithme précédent pour trouver N nombre de copies de $k_{1,3}$ dans (Q_i avec $i = 4..12$).

Entrée :n dimension de l'hypercube.
Retourne (N) nombre de copies de $S(1,1,1;1,1,1)$.
s : **Tableau** ;
V : **Tableau** ;
x_0 : **entier** ;
x_1 : **entier** ;

$v \leftarrow [1,2,...,2^n]$; N $\leftarrow$ 0 ;
Si ($n < 3$) **Alors**
    Rtourne (N=0) ;
Sinon
    $x_0 \leftarrow$ sommet initial d'étiquette $\{0\}^n$
    $x_1 = procedure\_sommets\_adjacent(x_0, 1)$
    [$x_1 \leftarrow$ sommet initial adjacent au sommet x_0]

    Tant que ($v \neq \varnothing$) **faire**
        $s_1 = procedure\_sommets\_adjacent(x_0, 3)$
        $s_2 = procedure\_sommets\_adjacent(x_1, 3)$
        [La procédure retourne un ensemble S de k sommets adjacents au sommet x_0]
        $s \leftarrow s_1 \cup s_2$;
        $V \leftarrow V \cap (V \setminus s)$;
        $N \leftarrow N + 1$;
        $x_3 = procedure\_sommets\_adjacent(x_0, 1)$
        $x_0 \leftarrow x_3$
        $x_1 = procedure\_sommets\_adjacent(x_0, 1)$
    Fait
Fin Si
Retourne (N)

Algorithme 2: Algorithme de calcul du nombre de copies disjointes de $S(1,1,1;1,1,1)$ dans l'hypercube Q_n

Application 2

Q_i	nombre de copies	Durée d'exécution (Secondes)
Q_4	2	0.0293
Q_5	4	0.1198
Q_6	8	0.4790
Q_7	16	1.9886
Q_8	32	8.3063
Q_9	64	31.3699
Q_{10}	128	128.5559
Q_{11}	256	1632.7
Q_{12}	512	7861.3

4.11.3 Complexité

L'algorithme exécute au plus (2^n) itération, et pour chacun d'eux :
y n accès à la matrice d'adjacence de l'hypercube,pour chaque accès on utilise une pile v (un vecteur)de dimension 2^n , pour laquelle on définies les procedures $s = procedure\_sommets\_adjacent(x_0, k)$, qui retourne les successeurs de x_0
la procédure $empile(v, s)$ qui supprime l'ensemble s des sommets de la pile v.

Ainsi que on obtient un algorithme de complexité $O(2^n \cdot n^2)$, la complexité de l'algorithme augmente de façon exponentielle.

Discussion

Dans les deux algorithmes le temps de calcule croit avec la dimension de l'hypercube.
pour Q_4 :dans le premier algorithme la durée t=0.0283 s et pour Q_5 $t_4 = 0.1077 \;\; s = 4 \cdot t_4 \;\; s.$
On remarquer que $t_n = 4 \cdot t_{n-1}$

Conclusion générale

Beaucoup de chercheurs se sont intéressés au plongement de graphe dans l'hypercube Havel [15, 19, 22, 20] Harary [11, 10, 12], Nebesky [33, 33].

Dans ce mémoire,Nous nous sommes intéressés au plongement de copies de graphes dans l 'hypercube. Nous avons introduit des nouvelles classes d'arbre pour les quelles nous avons donné le nombre de copies disjointes maximale qu'on peut placer dans l'hypercube de dimension n , cette étude de placement est très importante dans la caractérisation des nouvelles classes d'arbre plongeable dans l'hypercube.
Nous avons ensuite présenté deux algorithmes qui permettent de donner le nombre de copies qu'on peut placer d'un hypercube Q_n.

La notion de plongement de copies disjointes d'arbres dans l'hypercube Q_n, peut nous aider à construire plusieurs nouvelles classes d'arbres qu'on peut plonger dans leur hypercube optimal.

Bibliographie

[1] S.P. Avann. Metric ternary distributive semi-lattices. *proc.amer math.soc*, pages 407–414, 1961.

[2] H.J. Bandelt and H.M. Mulder. infinite midian graph, $(0, 2)-$graphs and hypercubes. *journale of graph theory*, (7) :487–492, 1983.

[3] C. Berge. *Graphes et hypercubes.* Paris, 1970.

[4] A. Berrachedi. Sur quelques propriétés métriques du graphe de type hypercube. *Thèse de doctorat d'état.Institut de Mathématiques,USTHB*, 1997.

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

[6] G. Burosch.I.Havel and J.M. Laborde. Distance monotone graphs and a new characterization of hypercubes. *Discrete Mathematics*, (110) :9–16, 1992.

[7] V. Firsov. On isometric embeddings of a graph into a boolean cube. *Cybernetics*, 1 :112–113, 1965.

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

[9] M. Gardner. Mathematischer karneval. *Frankfurt am Main*, 14 :69–76, 2008.

[10] F. Harary and M.Lewillter. Spaning subgraphs of a hypercube ii. *Meshes, inter, jour,computer math*, 25 :20–24, 1988.

[11] F. Harary and M.Lewillter. Spanning subgraphs of a hypercube ii. *starlik,computer math.Modelling*, (11) :216–217, 1988.

[12] F. Harary and M.Lewillter. The starlike tree which span a hypercube. *computer math*, 15 :299–302, 1988.

[13] F. Harary and M.lewinter. Spanninng subgraphs of hypercube. *computer Math*, 1988.

[14] I. Havel. Embedding certain trees into hypercube. *in recent advances in graph theory*, pages 201–205, 1974.

[15] I. Havel. On hamiltonian circuits and spanning trees of hypercube. *Casopis*, (109) :103–108, 1984.

[16] I. Havel. On certain trees in hypercube. *Topics in combinatorrics and graph thoery*, (353-358), 1990.

[17] I. Havel and J.Maoravek. B-valuations of graphs. *Czech-math*, (22) :338–351, 1972.

[18] I. Havel and J.Monivek. B-valuations of graphs. *Czech Math Journ.*, (98) :338–351, 1972.

[19] I. Havel and P. Liebl. Embedding the polytomic tree into the n-cube. *Cas Pest mat*, (98) :307–314, 1973.

[20] I. Havel and P. Liebl. One-legged caterpillars span hypercube. *Journal of Graph Theory*, (10) :69–77, 1986.

[21] I. Havel and P.Liebl. Embedding the dichotomie tree into the n-cube. *Cas.Pest. mat*, (97) :201–205, 1972.

[22] I. Havel and P.Liebl. Embedding the polytomic tree into the n-cube. *Cas Pest math*, (98) :307–314, 1973.

[23] C.T Ho and L. Johnsson. Spanning graphs for optimal broadcasting and personalized communication in hypercubes. *Tech. REPT. YALEU/CSD/RR-500 Yale U*, 1986.

[24] M. Kobiessi. Plongement de graphes dans l'hypercube. *Thèse de doctorat ,discipline informatique*, 2001.

[25] J.M. Laborde and S.P.Rao Hebbare. Another characterization of hypercubes. *Discrete Mathematics*, (39), 1982.

[26] N.B. Limaye. 6-quasistars and n-cubes. *Technical Report, Institut IMAG, Saint Martin d'hères, France*, (563), 1986.

[27] M. Mollard. Quelque problemes combinatoires sur l'hypercube et les graphe de hamming. *Discrete Mathématics*, 1989.

[28] H.M. Mulder. (0,λ)-graphs and n-cubes. *Discrete Mathematics*, (28), 1979.

[29] H.M. Mulder. The interval fection of graph. *Mathematicl centerum,Amesterdam*, 1980.

[30] L. Nebesky. Median graphs. *comment Math Univ Corolinea*, (533-544), 1970.

[31] L. Nebesky. On quasistars in n-cubes. *Cas.Pest. mat*, (109) :153–156, 1984.

[32] L. Nebesky. On quasistars in n-cubes. *Cas. Pest. mat.*, (109) :153–156, 1984.

[33] L. Nebesky. Embedding m-quasistars in n-cubes. *C zechoslovak mathematical, journal,praha*, (38) :113, 1988.

[34] B. Parhami. *Introduction to Parallel Processing*, chapter chapitre 13. ISBN 0306459701. 1999.

[35] QF. Stout and B.Wagar. Passing messages in link-bound hypercubes. *Multiprocesseurs Hypercube*, (251-257), 1987.

Index

Résumé

L'étude d'un plongement d'un graphe G dans un graphe H revient à voir si G est isomorphe à un sous graphe de H.Le problème a été traité par Havel [15, 19, 22, 20] Harary [11, 10, 12], Nebesky [31, 33].
Ce problème possède de nombreuses applications (architecture parallèle, transfère de l'information, codage,).Johnsson [23] , Wagar[35].
On s'est intéressé dans ce travail au plongement de copies d'un graphe G dans l'hypercube.
Les graphes considérés sont les étoiles, les quasi-étoiles et les Grilles. Pour les deux premier types, on a développé deux algorithmes donnant le maximum de copies plongeables.

Mots clés : Hypercube, Plongement, Graphe, Isomorphisme, Arbre,Grille.