

Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et de Informatique
Département de Recherche Opérationnelle



Mémoire

Présenté par :

KHABABA RAYANE & BELKHIRI NOUR HOUDA

Pour l'obtention du diplôme de :

Master

Filière : Mathématiques appliquées

Spécialité : Méthodes et outils pour la recherche opérationnelle

Thème

Codes identifiants dans le produit cartésien de deux chemin.

Soutenu publiquement le 13 juillet 2021 devant le jury composé de :

MR- I.ADDOUNE Président
MR- H.TOUATI Encadreur
MR- B.BRAHMI Examineur

Promotion 2020/2021

Remerciements

Nous remercierons « ALLAH » le tout puissant de nous ouvrir les portes du savoir, de nous aidé dans les moments difficiles, et de nous avoir permis d'être ce que nous devenus.

Nos vifs remerciements à notre encadreur « M.Hillal Touati », pour sa disponibilités, son aide, et ses enseignements si précieux. Qu'il trouve ici toute notre profonde gratitude.

Nos vifs remerciements vont également au membre du jury pour L'intérêt qu'il ont porté à notre recherche on acceptant d'examiner notre travail et de l'enrichir par leur propositions.

Nous souhaitons adresser nos remerciements les plus sincères au corps professoral et administratif du département des mathématiques et informatiques et tous les enseignants de la spécialité Recherche Opérationnelle pour la richesse et la qualité de leur enseignement et qui déploient grand efforts pour assures à leur étudiants une formation actualisée.

Enfin on tient également a exprimer toute notre reconnaissance et notre gratitude a l'ensemble des personnes qui nous ont encouragé , aider, et soutenues dans la réalisation de ce mémoire

Dédicace

A mes chers parents,

♡ Khababa Nour eddine ♡ *et* ♡ Rouabeh Fatiha ♡

pour tous leurs sacrifices,

leur amour, leur tendresse,

leur soutien et leurs prières tout au long de mes études.

A ma chère sœur et mes chères tantes ,

pour leurs encouragements permanents,

et leur soutien moral.

A mon cher frère et mes chers oncles ,

pour leur appui et leur encouragement.

A la mémoire de mon grand-père :

♡ Rouabeh Ammar ♡

A mon grand-père et mes grands-mères, pour leur amour et leur soutien.

A mon âme sœur, mon compagnon, mon bonheur et mon sourire :

♡ Lounici Zineb ♡

A toute mes amies : Rania , Amani, Amina

pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués,

et le fruit de votre soutien infailible,

Merci d'être toujours là pour moi.

*En fin de compte, je ne peux pas terminer cette dédicace sans
remercier ma collègue Nour houda, le travaille avec elle est magnifique donc
merci beaucoup*

De Rayane

Dédicace

A mes chers parents,

♡ Belkhiri.T ♡ et ♡ chara .M ♡

pour tous leurs sacrifices,

leur amour, leur tendresse,

leur soutien et leurs prières tout au long de mes études.

A mes chères soeurs : Meriem Ilhem, Assia, Amel et Rania

pour leurs encouragements permanents,

et leur soutien moral.

A mon cher frère : Mohamed

pour leur appui et leur encouragement.

A mes chères neveux :

♡ Abd El Rahman, Djamel El Dinne, Ayoub, Souhaib Nayar et Adam ♡

A mon âme sœur, mon compagnon, mon bonheur et mon sourire :

♡ Saidi Nesrine ♡

*A toute mes amies : Lounici Zineb , Djeddou Chaima, Houamed
Kenza, Mekhoukh Lilia*

*pour leur soutien tout au long de mon parcours universitaire,
Que ce travail soit l'accomplissement de vos vœux tant allégués,*

et le fruit de votre soutien infailible,

Merci d'être toujours là pour moi.

*En fin de compte, je ne peux pas terminer cette dédicace sans
remercier ma collègue Rayane, le travaille avec elle est magnifique donc merci
beaucoup*

De Nour Houda

Table des matières

Introduction	7
1 Généralités sur les graphes	9
1.1 Définitions et concepts de base	9
1.1.1 Degré dans un graphe	11
1.1.2 Notion de connexité	13
1.1.3 Hypergraphes	13
1.1.4 Différents types de graphes	14
1.1.5 Représentation matricielle	16
1.2 Opérations sur les graphes	17
1.2.1 Suppression et Contraction	17
1.2.2 Union et Intersection	18
1.2.3 Produit de graphes	18
2 Les Codes Identifiants	25
2.1 Codes Identifiants dans les graphes	25
2.1.1 Problème de couverture par tests	25
2.1.2 Définition de codes Identifiants	26
2.1.3 Application Pratique	27
2.2 Quelques généralisations et variantes possibles	28
2.2.1 Identification à distance $r \geq 1$	28
2.2.2 La densité des codes identifiants	29
2.2.3 Identification d'ensembles de sommets	29
2.2.4 Identification dans les graphes orienté	30
2.2.5 Identification Robuste	30
2.2.6 Identification Dynamique	31
2.2.7 Existence d'un code identifiant	31
2.2.8 Résultats généraux	32
2.3 Code identifiant dans les grilles	32
3 Codes Identifiants dans le produit cartésien de deux chemins	35
3.1 Préliminaires	35
3.2 Code identifiant dans les chemins	36
3.3 Code identifiant dans $P_2 \square P_n$	38
3.4 Code identifiant dans $P_4 \square P_n$	39
Conclusion Générale	49

Introduction

La recherche opérationnelle (RO), est la discipline des mathématiques appliquées, qui traite des questions d'utilisation optimale des ressources dans l'industrie et dans le secteur public. Elle née pendant la seconde guerre mondiale, par des efforts conjugués d'éminents mathématiciens (dont Neumann, Dantzig, Blackett) à qui il avait été demandé de fournir des techniques d'optimisation des ressources militaires.

La recherche opérationnelle, aussi appelée aide à la décision pour élaborer de meilleurs décisions. C'est un ensemble de méthodes et techniques visant à résoudre des problèmes d'optimisation, modélisant des problèmes réels dans différents domaines (économie, finance, gestion, transport, logistique, communication, etc.). La théorie des graphes est parmi les techniques mathématiques les plus importantes auxquelles recourt la recherche opérationnelle.

La théorie des graphes constitue un outil très puissant pour schématiser les modèles des liens et relations entre les objets. En effet, le premier problème de la théorie des graphes est apparu en 1736 dans la ville de Königsberg qui contient 7 ponts. Le problème qui se posait à l'époque : peut-on se promener dans la ville en traversant chaque pont, une et une seule fois ?

Le mathématicien Leonhard Euler (1707-1783), a résolu et modélisé ce problème en le présentant sous la forme d'un graphe, où les parties terrestres sont représentées par les sommets, et les ponts, par les arêtes.

La théorie des graphes, est devenue très utile comme modèle mathématique. Parmi les problèmes les plus importants de celle-ci est les problèmes de couverture par tests, Ces problèmes possèdent de nombreuses applications pratiques, dans des domaines variés comme ceux de la reconnaissance de formes, de séquençage d'ADN, d'aide au diagnostic médical ou encore de communication dans des réseaux multi-utilisateurs.

Les codes identifiants dans les graphes sont liés à la famille de couverture par tests. Introduit pour la première fois en 1998 par Karpovsky, Chkrabarty et Levitin pour modéliser un problème de détection et localisation de processeurs défectueux dans les réseaux multi-processeurs. Elles sont présentées de façon détaillée dans le deuxième chapitre de ce mémoire.

Ce manuscrit est structuré comme suit :

Le premier chapitre est consacré à la présentation des généralités sur les graphes, où nous donnons quelques définitions, notions de base et quelques opérations liées à la théorie des graphes qui nous seront nécessaires dans la suite.

Dans le deuxième chapitre, nous avons abordé le problème de code identifiant. Il s'agit de rechercher un sous ensemble de sommet appelé C dans un graphe G , de sorte que chaque sommet de graphe à un voisinage unique dans C .

On donne la définition des codes identifiants, quelques généralisations et variantes possibles, puis on parle sur l'existence d'un code identifiant.

Le troisième chapitre, concerne l'étude des codes identifiants dans le produit cartésien de deux chemins. On présentera les premiers résultats de la cardinalité minimum d'un code identifiants dans $P_4 \square P_n$.

Enfin, nous terminons le mémoire par une conclusion générale et un résumé.

Généralités sur les graphes

Pour une meilleur compréhension des autres chapitres, il est évident de rappeler quelques généralités sur la théorie des graphes et quelques opérations sur les graphes.

Nous commencerons par rappeler certaine définitions élémentaires sur les graphes, ainsi en profiter pour expliciter les notations que l'on retrouvera fréquemment par la suite.

Nous définirons ensuite les opérations sur les graphes (suppression, contraction, produit cartésien, produit direct,...).

Toutes les informations qui sera dans cette partie proviennent de [1, 21, 8]

1.1 Définitions et concepts de base

De manière générale, on appelle graphe noté $G = (V, E)$ le couple constitué par :

- Un ensemble de sommets $V = \{v_1, \dots, v_n\}$
- Une famille $E = \{e_1, \dots, e_n\}$ d'éléments du produit cartésien $V \times V = \{(v_1, v_2) \mid v_1, v_2 \in V\}$

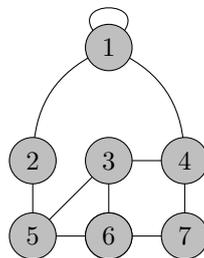


FIGURE 1.1 – Exemple de graphe avec :

$V = \{1, 2, 3, 4, 5, 6, 7\}$ et $E = \{(1, 1), (1, 2), (1, 4), (2, 5), (3, 5), (3, 6), (3, 4), (4, 7), (5, 6), (6, 7)\}$.

Deux classes de graphe existes, à savoir :

- Graphe non orienté :
Est noté $G = (V, E)$, où V est un ensemble fini non vide et E un ensemble de couple non ordonnées d'éléments de V .

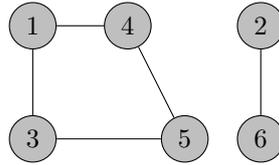


FIGURE 1.2 – Exemple d'un graphe non orienté $G=(V,E)$ avec :
 $V = \{1, 2, 3, 4, 5, 6\}$ $E = \{(1, 4), (1, 3), (3, 5), (5, 4), (2, 6)\}$.

- Un graphe non orienté est dit simple s'il ne comporte pas de boucle¹, et s'il ne comporte jamais plus d'une arête entre deux sommets.
- Ordre d'un graphe, noté $|V|$, est le nombre de ses sommets, et taille d'un graphe, noté $|E|$, est le nombre de ses arêtes.
- Graphe orienté :
 Un graphe orienté, noté $G = (V, A)$, est un graphe où les arêtes sont orientées (on les appelle arcs).

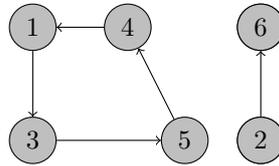


FIGURE 1.3 – Exemple d'un graphe orienté $G=(V,A)$ avec :
 $V = \{1, 2, 3, 4, 5, 6\}$ $A = \{(1, 4), (1, 3), (3, 5), (5, 4), (2, 6)\}$

Définition 1.1.1 (Graphe multiple). :

On appelle graphe multiple, c'est un graphe où au moins 1 couple de sommets ont plus d'une arête entre eux. Lorsqu'on a au plus p arêtes on parle de p -graphe.

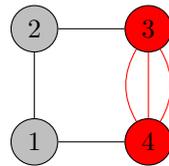


FIGURE 1.4 – Exemple d'un 3-graphe, il y a 3 arêtes entre le sommet 3 et le sommet 4.

Définition 1.1.2 (Graphe partiel). :

Un graphe partiel G_p d'un graphe G , engendré par un ensemble d'arêtes, est simplement le même graphe en gardant seulement les arêtes de l'ensemble.

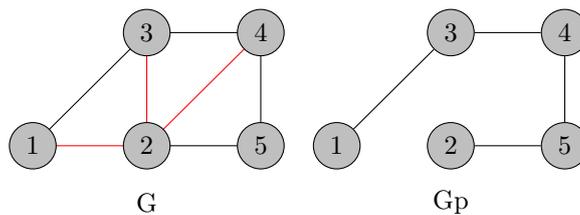


FIGURE 1.5 – Exemple d'un graphe partiel G_p de G .

1. Une boucle est une arête reliant un sommet à lui-même.

Définition 1.1.3 (sous graphe). :

Un sous graphe G_s d'un graphe G , engendré par un ensemble de sommet V_s , est un graphe ayant comme ensemble de sommet V_s et les arêtes joignant ses sommets.

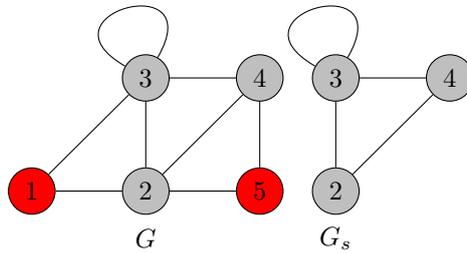


FIGURE 1.6 – Exemple d'un Sous Graphe G_s

1.1.1 Degré dans un graphe

Dans un graphe non orienté, le degré d'un sommet est le nombre d'arêtes incidentes à ce sommet.

Dans le cas d'un graphe simple, le degré d'un sommet représente le nombre de sommets adjacents à celui-ci.

Dans un graphe orienté, le demi degré extérieur ou demi degré sortant (resp. demi degré intérieur ou demi degré entrant) d'un sommet v noté $d^+(v)$ (resp. $d^-(v)$), représente le nombre d'arêtes ayant v comme extrémités initiale (resp. terminale). Le degré d'un sommet v est alors la somme des degré entrant et sortant $d(v) = d^+(v) + d^-(v)$.

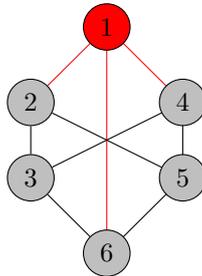


FIGURE 1.7 – Le degré du sommet $\{1\}$ dans ce graphe simple est $d(1) = 3$.

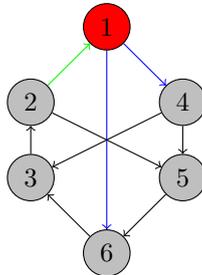


FIGURE 1.8 – Dans ce graphe orienté on a $d(1) = d^-(1) + d^+(1) = 1 + 2 = 3$.

Théorème 1.1.1. :

La somme des degrés des sommets, d'un graphe quelconque est égale à deux fois le nombre d'arêtes (ou d'arcs). C'est-à-dire :

$$\sum_{v \in V} \text{deg}(v) = 2 | E | .$$

Un résultat directe qui découle du théorème précédent, est :

Corollaire 1.1.1. :

Le nombre de sommets ayant un degré impair est pair.

Définition 1.1.4 (chemin). :

Soit $G = (V, A)$ un graphe orienté, un chemin d'un sommet u vers un sommet v est une séquence $\langle s_0, s_1, \dots, s_k \rangle$ de sommet tels que : $u = s_0, v = s_k$ et $(s_{i-1}, s_i) \in E$ pour tout $i \in \{1, \dots, k\}$ on dira que le chemin contient les sommets s_0, s_1, \dots, s_k et les arcs $(s_0, s_1), (s_1, s_2), \dots, (s_{k-1}, s_k)$.

La longueur du chemin, est le nombre d'arcs dans le chemin c'est à dire k . S'il existe un chemin de u à v on dira que v est accessible à partir de u .

Définition 1.1.5 (circuit). :

Un chemin $\langle s_0, \dots, s_k \rangle$ forme un circuit si $s_0 = s_k$, et si le chemin comporte au moins un arc ($k \geq 1$).

Une boucle est un circuit de longueur 1.

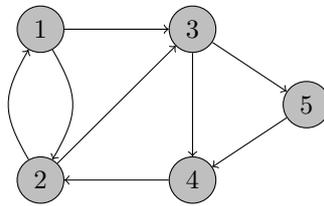


FIGURE 1.9 – Exemple d'un graphe orienté avec un chemin et un circuit :

Un chemin dans ce graphe est : $\langle 1, 2, 3, 4 \rangle$

Un circuit dans ce graphe est : $\langle 1, 3, 5, 4, 2, 1 \rangle$

Remarque 1.1.1. :

Dans le cas non orienté, on parle de chaîne au lieu de chemin et de cycle au lieu circuit.

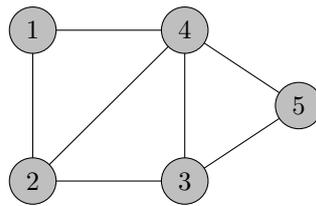


FIGURE 1.10 – Exemple d'un graphe non orienté avec un chaîne et un cycle :

Un chaîne dans ce graphe est : $\langle 1, 2, 3, 4, 5 \rangle$

Un cycle dans ce graphe est : $\langle 2, 3, 4, 1, 2 \rangle$.

La notion de longueur de chemin, nous permet ensuite de définir la notion de distance dans un graphe.

Définition 1.1.6 (Distance). :

Soit un graphe $G = (V, E)$, la distance d'un sommet à un autre, est la longueur minimum entre deux sommets, ou ∞ s'il n'y a pas :

$$\forall x, y \in V, d(x, y) = \begin{cases} k & \text{si la longueur minimum entre } x \text{ et } y \text{ est égale à } k \\ \infty & \text{sinon} \end{cases}$$

- Diamètre du graphe, est la plus grande distance entre deux sommets.

1.1.2 Notion de connexité

Définition 1.1.7 (Graphe et sous graphe connexes). :

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

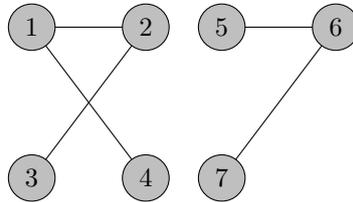


FIGURE 1.11 – Exemple :Le graphe n'est pas connexe car il n'existe pas de chaîne entre 1 et 7 en revanche le sous graphe induit par les sommets $\{1, 2, 3, 4\}$ est connexe.

Définition 1.1.8 (Composante connexe). :

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

Par exemple :Le graphe précédent(figure 1.11) est composé de 2 composantes connexes, la première est le sous graphe induit par les sommets $\{1, 2, 3, 4\}$, et la seconde et le sous graphe induit par les sommets $\{5, 6, 7\}$.

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

On retrouve les différents notions de connexités dans les graphes orientés en remplaçant naturellement la notion de chaîne par celle de chemin. dans le cas orienté on parle de graphe fortement connexe au lieu de connexe, et de composante fortement connexe au lieu de composante connexe.

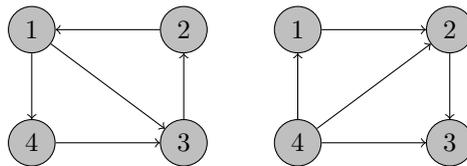


FIGURE 1.12 – Exemple :Le graphe de gauche est fortement connexe, tandis que celui de droite ne l'est pas.

Lorsque le graphe n'est pas fortement, on parle de composantes fortement connexes. Par exemple l'ensemble constitué du sommet 1 est une composante fortement connexe.

1.1.3 Hypergraphes

Un hypergraphe, est la donnée d'un couple (V, ε) où V est un ensemble et ε est un sous-ensemble de l'ensemble des parties de V . Les éléments de V sont les sommets et les éléments de ε sont les hyperarêtes de l'hypergraphe : ce sont des sous-ensembles de sommets. Un hypergraphe est régulier si tout sommet est dans exactement r hyperarêtes et k -uniforme si toutes ses arêtes sont de taille k . La notion d'hypergraphe généralise la notion de graphe : les graphes sont exactement les hypergraphes 2-uniformes.[1]

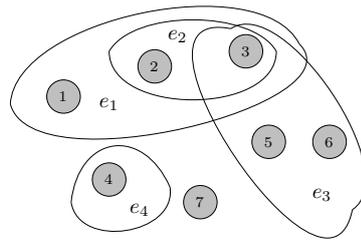


FIGURE 1.13 – Exemple d’un hypergraphe.

1.1.4 Différents types de graphes

Définition 1.1.9 (Graphe complet). :

Un graphe complet, est le graphe ayant chaque sommet adjacent à tous les autres. On le note K_n .

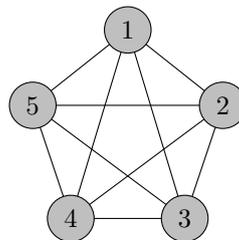


FIGURE 1.14 – Exemple d’un graphe complet K_5 .

Définition 1.1.10 (Graphe planaire). :

Un graphe est dit planaire, s’il admet une représentation où deux arêtes n’intersectent pas.

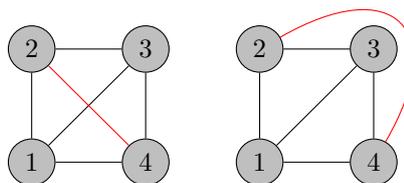


FIGURE 1.15 – Dans le graphe qu’est à gauche, on a l’arête 2-4 qui intersecte l’arête 1-3. Le graphe qu’est à droite est une représentation planaire de celui de gauche.

Parmi les résultats les plus fondamentaux dans ce type de graphe on a, le résultat suivant :

Théorème 1.1.2 (Euler²). :

Soit $G = (V, E)$, un graphe connexe et planaire d’ordre n de taille m et ayant f faces (régions), alors on a : $n - m + f = 2$.

2. Leonhard Euler : né le 15 avril 1707 à Bâle (suisse) et mort à 76 ans le 7 septembre 1783 (18 septembre 1783 dans le calendrier grégorien) à Saint-Petersbourg, est un mathématicien et physicien suisse. Il était notamment membre de l’Académie royale des sciences de prusse à Berlin. Euler fit d’importantes découvertes dans des domaines aussi variés que le calcul infinitésimal et la théorie des graphes. Il introduisit également une grande partie de la terminologie et de la notation des mathématiques modernes, en particulier pour l’analyse mathématique, comme la notion de fonction mathématique.

Définition 1.1.11 (Graphe régulier). :

Un graphe régulier, est un graphe dont les degrés des sommets sont tous égaux. Ainsi on dit qu'un graphe est k -régulier pour tout sommet v on a $d(v) = k$.

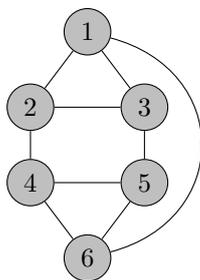


FIGURE 1.16 – Exemple d'un graphe 3-régulier.

Définition 1.1.12 (Graphe biparti). :

On dit qu'un graphe $G = (V, E)$, est biparti si son ensemble de sommets peut être divisé en deux stable³.

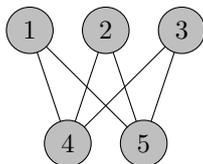


FIGURE 1.17 – Exemple d'un graphe biparti.

- Un graphe sans cycle est dit acyclique.

Définition 1.1.13 (Arbre). :

On appelle arbre, qu'on note $T = (V, E)$, tout graphe simple connexe et sans cycle.

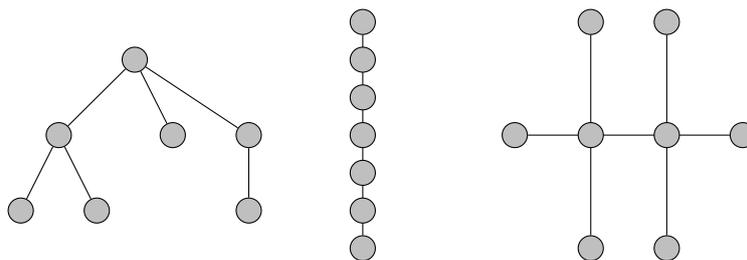


FIGURE 1.18 – Des exemples d'arbres

Définition 1.1.14 (Forêt). :

Une forêt, est un graphe non connexe et sans cycle. c'est à dire que chaque composante connexe est un arbre.

Définition 1.1.15 (Arborescence). :

Dans le cas orienté, on parle d'arborescence, c'est à dire une arborescences est un arbre orienté ayant une racine⁴.

3. Un stable est un sous ensemble de sommets $S \subseteq V$ deux à deux non adjacents.

4. On appelle racine tout sommet r tel que tout autre sommet peut être atteint à partir de r par un chemin.

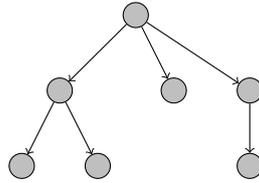


FIGURE 1.19 – Exemple d’arborescence.

1.1.5 Représentation matricielle

On peut considérer cette sous-section comme traitant la représentation "algébrique" d’un graphe. Les matrices classiques permettant de représenter un graphe sont la matrice d’adjacence, à valeurs booléennes, et la matrice d’incidence, à valeurs 0,+1 et -1.

Définition 1.1.16 (Matrice d’adjacence). :

On définit la matrice d’adjacence par : $M = (m_{ij})$ où :

$$m_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E \\ 0 & \text{sinon} \end{cases}$$

Si on se donne le graphe suivant :

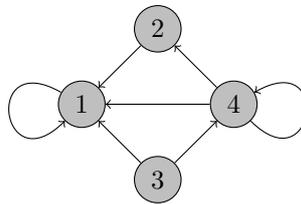


FIGURE 1.20 – Graphe orienté.

Alors, sa matrice d’adjacence est donnée par :

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

Dans le cas non orienté, la matrice d’adjacence est toujours symétrique.

Définition 1.1.17 (Matrice d’incidence). :

On définit la matrice d’incidence par $Q = (q_{ij})$ où :

$$q_{ij} = \begin{cases} 1 & \text{si } v_i \text{ est incident à } e_j \\ 0 & \text{sinon} \end{cases}$$

Soit le graphe suivant :

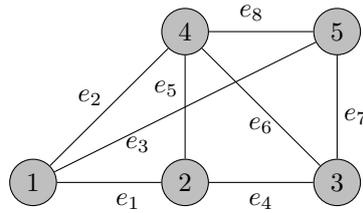


FIGURE 1.21 – Graphe non orienté.

Sa matrice d'incidence est :

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

1.2 Opérations sur les graphes

Dans cette partie, on présente quelques opérations qui peuvent être effectuées sur les graphes. Parmi celles-ci, on cite la suppression d'une arête et d'un sommet, la contraction, l'union de deux graphes, leur intersection et produit de graphes (directe et cartésien).

1.2.1 Suppression et Contraction

Définition 1.2.1 (La suppression d'une arête). :

La suppression d'une arête n'engendre que la suppression de celle-ci.

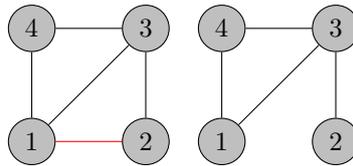


FIGURE 1.22 – Exemple :Suppression de l'arête 1-2.

Définition 1.2.2 (La suppression d'un sommet). :

La suppression d'un sommet engendre la suppression des arêtes qui lui sont incidentes.

Exemple :

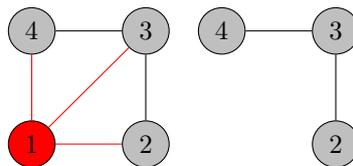


FIGURE 1.23 – Exemple :Suppression de sommet 1.

Définition 1.2.3 (La contraction d'une arête). :

La contraction d'une arête est la suppression de celle-ci en fusionnant les deux extrémités.

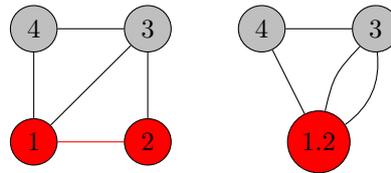


FIGURE 1.24 – Exemple :Contraction de l’arête 1-2.

1.2.2 Union et Intersection

Définition 1.2.4 (Union). :

L’union entre 2 graphes $G_1=(V_1,E_1)$ et $G_2=(V_2,E_2)$, ($G = G_1 \cup G_2$) est un graphe avec :
 $V = V_1 \cup V_2$ et $E = E_1 \cup E_2$.

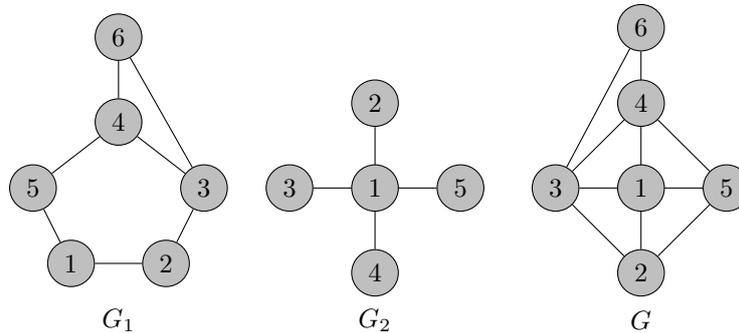


FIGURE 1.25 – G est l’union des deux graphes G_1 et G_2 qu’on note $G_1 \cup G_2$.

Définition 1.2.5 (Intersection). :

L’intersection entre 2 graphes $G_1=(V_1, E_1)$ et $G_2=(V_2, E_2)$, ($G = G_1 \cap G_2$) est un graphe avec :
 $V = V_1 \cap V_2$ et $E = E_1 \cap E_2$.

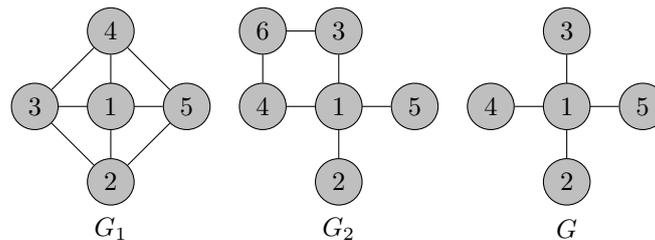


FIGURE 1.26 – G est l’intersection des deux graphes G_1 et G_2 qu’on note $G_1 \cap G_2$

Les définition de 1.2.1 jusqu’au 1.1.5 ont été tiré des cours suivis lors de notre formation en master 1 du module théorie des graphes [8].

1.2.3 Produit de graphes

Dans cette partie, on présente le produit de graphe (directe et cartésien). Pour plus d’informations sur les produit de graphes on peut consulter [6, 23, 14, 17, 18].

Produit directe

Définition 1.2.6. :

Soient deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$. Le produit direct ou croisé de G_1 et G_2 ,

noté $G_1 \times G_2$ est le graphe ayant pour ensemble de sommets $V_1 \times V_2$ et dont deux sommets $u = (u_1, v_1), v = (u_2, v_2)$ sont adjacents si et seulement si :

$$\begin{cases} u_1 u_2 \in E_1 \\ \text{et} \\ v_1 v_2 \in E_2 \end{cases}$$

En d'autres termes, deux sommets de $G_1 \times G_2$ sont, adjacents si et seulement si chaque deux composantes appartenant au même graphe sont adjacentes.

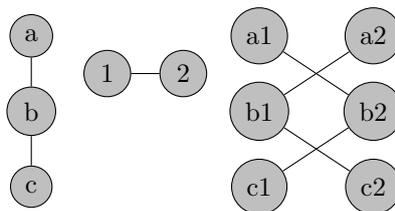


FIGURE 1.27 – Exemple :Produit direct de deux graphes G_1 et G_2 tel que $G = G_1 \times G_2$.

Produit cartésien

Définition 1.2.7. :

Soient $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$. Le produit cartésien ou carré de G_1 et G_2 , noté $G_1 \square G_2$ est le graphe ayant pour ensemble de sommets $V_1 \times V_2$ et dont deux sommets $u = (u_1, v_1), v = (u_2, v_2)$ sont adjacents si :

$$\begin{cases} u_1 = u_2 \quad \text{et} \quad v_1 v_2 \in E_2 \\ \text{ou} \\ v_1 = v_2 \quad \text{et} \quad u_1 u_2 \in E_1 \end{cases}$$

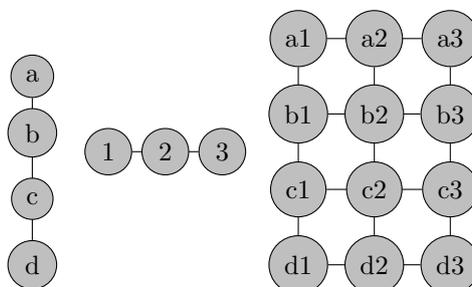


FIGURE 1.28 – Exemple :Produit cartésien de deux graphes G_1 et G_2 tel que $G = G_1 \square G_2$.

Un autre exemple de produit cartésien entre un cycle et une clique⁵ ($C_4 \square K_3$) :

5. Clique : est un sous-graphe complet.

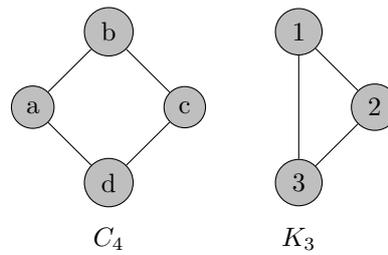


FIGURE 1.29 – Représentation d’un cycle C_4 et une clique K_3 .

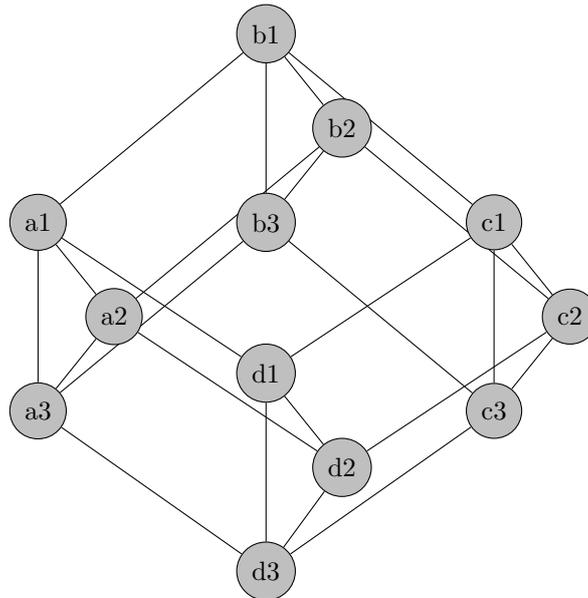


FIGURE 1.30 – Représentation de produit cartésien entre un cycle et une clique $C_4 \square K_3$.

Propriétés :

- **La commutativité :**

Le produit cartésien est commutative. En d’autres termes, on note

$$G_1 \square G_2 \cong G_2 \square G_1.$$

- **L’associativité :**

Le produit cartésien est associative, En d’autres termes, on note

$$(G_1 \square G_2) \square G_3 \cong G_1 \square (G_2 \square G_3).$$

- **La connexité :**

Le produit cartésien $G_1 \square G_2$, est connexe si et seulement si G_1 et G_2 sont connexe.

- **Le degré d’un sommet :**

Le degré d’un sommet du produit cartésien, est égale à la somme des degrés de ses coordonnées.

- **Le diamètre :**

- Soit $H = G_1 \square G_2$, le diamètre D_H du $G_1 \square G_2$ est la somme des diamètres de G_1 et G_2 .

- **L’indépendance :**

L’ensemble indépendant : est toute simplement est un stable.

La cardinalité d’un stable dans G , notée $\alpha(G)$, est appelée nombre d’indépendance .

L'idée centrale qui est utilisée pour produire des limites inférieures pour le nombre d'indépendance de produit cartésiens est le fait que si I est un stable de G_1 , et J est un stable de G_2 , alors $I \times J$ est indépendant dans $G_1 \square G_2$, avec

$$\alpha(G_1 \square G_2) \geq \alpha(G_1)\alpha(G_2).$$

- Pour tous graphes G_1 et G_2 , on a :[18]

$$\alpha(G_1 \square G_2) \geq \alpha(G_1)\alpha(G_2)\min\{|G_1| - \alpha(G_1), |G_2| - \alpha(G_2)\}.$$

$$\alpha(G_1 \square G_2) \leq \min\{\alpha(G_1)|G_2|, \alpha(G_2)|G_1|\}.$$

Graphes résultant du produit cartésien :

- **Hypercube :**

Définition 1.2.8. *L'hypercube de dimension n ou le n -cube 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.*

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 isomorphe⁶ à

$$K_2 \square K_2 \square \dots \square K_2.$$

et donc

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

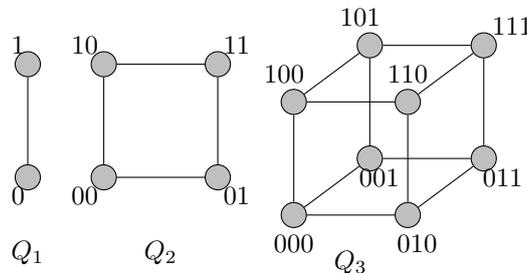


FIGURE 1.31 – Les premiers hypercubes

- **Graphe de Hamming :**

6. Isomorphe entre deux graphes est une fonction conserve l'adjacence et la non adjacence.

Définition 1.2.9. *Un graphe de Hamming, est un produit cartésien de cliques de tailles quelconques. Un graphe de Hamming produit de d cliques de taille q est noté $H(d, q)$, Nombre des sommets q^d . Nombre d'arêtes $\frac{d(q-1)q^d}{2}$. Distribution des degrés $d(q-1)$ - régulier. L'hypercube de dimension d est donc le cas particulier du graphe de Hamming $H(d, q)$.*

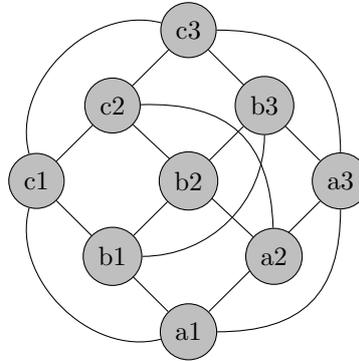


FIGURE 1.32 – Graphe de Hamming $H(3, 3) = K_3 \square K_3$.

Représentation matricielle

La présentation d'une matrice d'adjacence de produit cartésien $G_1 \square G_2$, est donnée comme suit :

$$A_{G_1 \times G_2} = A_{G_1} \otimes I_{n_{G_2}} + I_{n_{G_1}} \otimes A_{G_2}$$

avec G_1 et G_2 sont d'ordre n_{G_1} et n_{G_2} respectivement, et \otimes est le produit de Kronecker, et I_m est la matrice identité $m \times m$. Rappelons que le produit de Kronecker, de deux matrice $A_{m \times n} \otimes B_{k \times l}$ est la matrice $mk \times nl$ donnée par :

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}$$

En général, le produit Kronecker n'est pas commutatif.

Nous allons décomposer la formule pour voir pourquoi le produit Kronecker est nécessaire. la matrice d'adjacent dans chaînes P_3 et P_2 respectivement, sont :

$$A_{P_3} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad A_{P_2} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

On prend $P_3 \times P_2$ alors $A_{P_3 \times P_2} = A_{P_3} \otimes I_2 + I_3 \otimes A_{P_2}$

les dimensions des matrices d'identité "croisent" pour garantir que nous obtenons la bonne dimension pour la matrice d'adjacence de $P_3 \times P_2$, qui est 6×6

$$A_{P_3} \otimes I_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Et

$$I_3 \otimes A_{P_2} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Les deux matrices sont des matrices d'adjacence des deux graphes à 6 sommets, Notez que nous avons maintenant deux copies de P_3 dans un graphe et trois copies de P_2 dans un graphes, comme représenté dans la figure suivante :

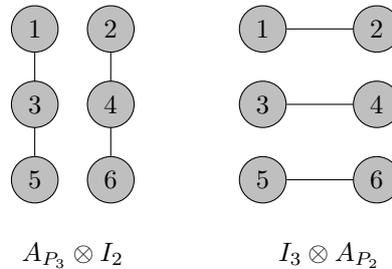


FIGURE 1.33 – Les graphes de $A_{P_3} \otimes I_2$ et $I_3 \otimes A_{P_2}$.

Maintenant, on va ajouter les deux ensembles matricielle pour obtenir la matrice d'adjacence de $A_{P_3 \times P_2}$.

$$A_{P_3 \times P_2} = A_{P_3} \otimes I_2 + I_3 \otimes A_{P_2} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Cette matrice représente le graphe suivant :

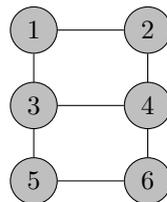


FIGURE 1.34 – Le graphe de $A_{P_3 \times P_2}$.

Les Codes Identifiants

Dans ce chapitre, on présente les codes identifiants comme un cas particulier du problème général de couverture par tests. On présente quelques définitions, application pratique, plusieurs variantes et généralisations possibles de la notion de code identifiant, des liens avec d'autres types de codes, l'existence et la construction de codes identifiants sont également présentés. Les notions et les définitions dans cette partie émanent des références suivante : [12, 22, 7, 3, 20, 13, 19, 4, 9].

2.1 Codes Identifiants dans les graphes

Dans cette partie, on présente le problème de couverture par tests, quelques définitions et application pratique.

2.1.1 Problème de couverture par tests

Les codes identifiants font partie de la famille des problèmes de couverture par tests, définis de façon générique comme suit :

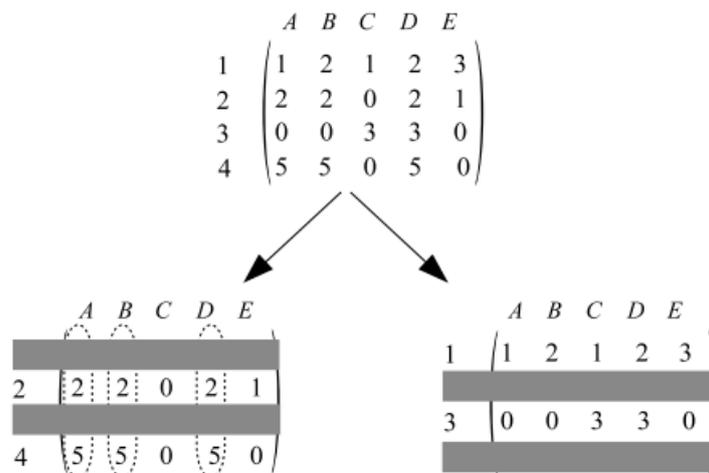


FIGURE 2.1 – Exemple de ce problème dans le cas d'une matrice M à coefficients entiers.

Étant donnée une matrice M , quels sont les sous-ensembles de lignes de M telles que les colonnes résultantes soient toutes différentes ?

Dans le problème de couverture par tests, il faut déterminer un ensemble de lignes de la matrice permettant de différencier les colonnes entre elles. Dans la Figure 2.1, les restrictions aux lignes 2 et 4 des colonnes A,B et D sont les mêmes : les lignes 2 et 4 ne permettent donc pas de différencier les colonnes de la matrice. Les lignes 1 et 3 quant à elles, permettent d'identifier les colonnes A,B,C,D,E : les traces des colonnes A,B,C,D,E sur les lignes 1 et 3 sont toutes différentes.

Ce problème a été, à l'origine, formulé pour modéliser un problème d'aide au diagnostic médical, dans lequel les lignes de la matrice M correspondent à des symptômes et les colonnes à des maladies. Les coefficients de M correspondent à l'intensité des symptômes dans les maladies. Le problème consistait à déterminer un sous-ensemble de symptômes qui identifiait chaque maladie de façon unique.[12].

2.1.2 Définition de codes Identifiants

Dans un graphes $G = (V, E)$ donné, un code C est un sous-ensemble de sommets, ($C \subseteq V$). Un élément de ce code est appelé mot de code. Notons $N(v)$ les voisin du sommet v .

Définition 2.1.1 (Code couvrant). :

On dit que C est un code couvrant de G , si C est tel que tout sommet v de $V \setminus C$ est voisin d'au moins un sommet de C .

Un code couvrant est aussi appelé dominant du graphe.

En d'autres termes :

Pour un sommet v de G , on définit le voisinage étendu de v comme l'ensemble :

$$N[v] = N(v) \cup \{v\}.$$

Un sous-ensemble de sommets C , est un code couvrant de G si et seulement si pour tout $v \in V$ on a :

$$N[v] \cap C \neq \emptyset.$$

On dira qu'un sommet $c \in C$ couvre le sommet v s'il appartient au voisinage étendu de v .

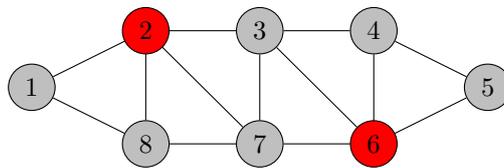


FIGURE 2.2 – Le sous-ensemble des sommets rouges est un code couvrant du graphe : tout sommet non rouge est voisin d'au moins un sommet rouge.

Définition 2.1.2 (Code séparateur). :

Un sous-ensemble $C \subseteq G$, est un code séparateur de G si et seulement si pour toute paire de sommets distincts u, v de G on a :

$$N[u] \cap C \neq N[v] \cap C.$$

D'une façon équivalente, C est un code séparateur si pour toute paire de sommets u et v dans G on a :

$$(N[u] \cap C) \Delta (N[v] \cap C) \neq \emptyset.$$

où $A\Delta B$ désigne la différence symétrique de A et B : $A\Delta B = (A\setminus B)\cup(B\setminus A)$. On dira qu'un sommet $c \in C$, sépare les sommets u et v s'il appartient à la différence symétrique de $N[u] \cap C$ et $N[v] \cap C$. Le sous-ensemble C est un code séparateur de G s'il sépare toutes les paires de sommets distincts de G .

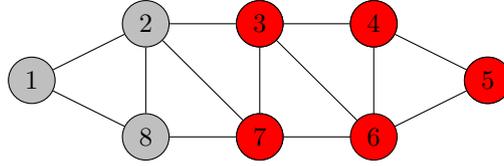


FIGURE 2.3 – Le sous-ensemble de sommets C en rouges représente un code séparateur :toute les paires de sommets du graphe sont séparées par au moins un sommet rouge.

Remarque 2.1.1. *Le code séparateur peut ne pas être un code couvrant. Par exemple dans la figure 2.3,le sommet 1 n'est pas couvert.*

Définition 2.1.3 (Code identifiant). :

On dit que un sous-ensemble C de G , est un code identifiant si et seulement si :

1. Pour tout sommet $v \in V$, on a : $N[v] \cap C \neq \emptyset$ (condition de couverture).
2. pour toute paire de sommets distincts u et v , on a : $N[u] \cap C \neq N[v] \cap C$ (condition de séparation).

D'une autre manière, un code identifiant, est à la fois un code couvrant et un code séparateur .

L'ensemble $N[v] \cap C$ est appelé ensemble identifiant de v , on le note $I(v, C)$ ou simplement $I(v)$ s'il n'y a pas d'ambiguïté.

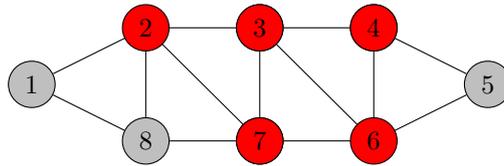


FIGURE 2.4 – L'ensemble des sommets rouges forme un code identifiant du graphe. On a par exemple : $I(2) = \{2, 3, 7\}$ et $I(5) = \{4, 6\}$.

2.1.3 Application Pratique

Les codes identifiants, ont été introduits pour modéliser un problème pratique d'identification de processeurs défectueux dans des réseaux multiprocesseurs.

Supposons que chaque processeur p_i d'un réseau soit capable d'exécuter une procédure $\text{test}(p_i)$, qui s'applique à p_i ainsi qu'aux processeurs voisins de p_i . Cette procédure teste le bon fonctionnement de p_i et de ses voisins, et ne retourne qu'une information de type binaire : par exemple, 0 si une défaillance a été détectée sur p_i ou sur l'un de ses voisins, et 1 sinon. En supposant qu'à tout moment, au plus un processeur du réseau soit défectueux, le problème est de déterminer un sous-ensemble de processeurs C tel que :

- Si au moins un des processeurs de C renvoie 0 après l'exécution de test , alors il y a un unique processeur défectueux dans le réseau, que nous sommes en mesure de localiser d'après les résultats des exécutions de test sur C .

- Si tous les processeurs de C renvoient 1 après l'exécution de test, alors tous les processeurs du réseau sont en bon état de marche.

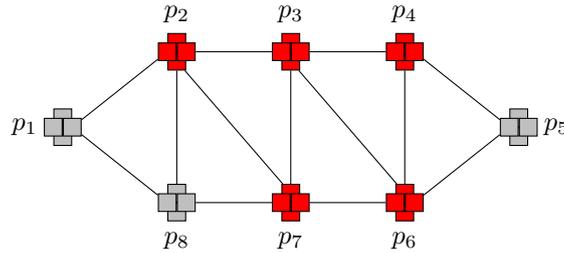


FIGURE 2.5 – Dans le réseaux ci-dessous à 8 processeurs, ceux qui sont en rouges permettent d'identifier de façon unique chaque processeur du réseau.

Il est facile de voir qu'un sous-ensemble de processeurs C vérifie ces deux conditions si et seulement si l'ensemble de sommets correspondant C est un code identifiant du graphe associé au réseau.

En effet, la deuxième condition garantit le fait que si aucun processeur de C ne détecte de défaillance alors aucun processeur du réseau n'est défectueux : cela équivaut à dire que C est un code couvrant du graphe associé au réseau. La première condition, équivaut elle à dire que C est un code séparateur du graphe. En effet, le sous-ensemble I de processeurs de C ayant renvoyé 0 après l'exécution de test est l'ensemble des processeurs de C voisins du processeur défectueux p : c'est l'ensemble identifiant de p . Dire que $I = I(p, C)$ détermine de façon unique p équivaut à dire que C est un code séparateur du graphe du réseau.

Une autre application des codes identifiants, a été réalisée pour le problème de localisation par un réseaux de capteurs à une échelle local (immeuble, station de travail...).

Les réseaux de capteurs est un nouveau moyen permettant d'atteindre et de surveiller notre environnement. D'importantes applications de ces réseaux comprennent la surveillance des bâtiments, stations nucléaires,...ect.

2.2 Quelques généralisations et variantes possibles

Dans cette partie, on présente l'identification à distance $r \geq 1$, identification d'ensembles de sommets, identification dans les graphes orientés, identification robuste et dynamique.

2.2.1 Identification à distance $r \geq 1$

Une première façon d'étendre la notion de code identifiant est de permettre à chaque processeur p d'un réseau d'exécuter une procédure $test(p)$, qui s'applique à p ainsi qu'aux processeurs voisins de p . On peut imaginer que p puisse tester tous les processeurs à distance au plus r du processeur exécutant la procédure, où r est une constante supérieure ou égale à 1, et où la distance entre deux processeurs est définie comme le nombre minimum d'arêtes d'un chemin entre ces deux processeurs.

On parle dans ce cas de codes r -identifiants.

Définition 2.2.1 (Code r -identifiant). :

On dit qu'un sous-ensemble de sommets $C \subseteq V$, est un code r -identifiant si et seulement si :

- pour tout sommet $v \in V$, on a :

$$B_r(v) \cap C \neq \emptyset$$

- pour toute paire de sommets distincts u, v , on a :

$$B_r(u) \cap C \neq B_r(v) \cap C$$

tel que $B_r(v)$ représente la boule de rayon r centrée en v : $B_r(v)$ est l'ensemble des sommets à distance au plus r de v .

L'ensemble identifiant de v , $B_r(v) \cap C$, est en général désigné par $I_r(v)$.

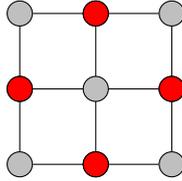


FIGURE 2.6 – Les processeurs en rouges forment un code 1- et 2-identifiant du graphe.

2.2.2 La densité des codes identifiants

Soit $G = (V, E)$ un graphe quelconque, la densité d'un code r -identifiant de G est définie comme suit :

Définition 2.2.2 (La densité). :

Pour tout sommet $v \in V$, et pour tout $r \in \mathbb{N}$. La densité noté $d(C, G)$ d'un code r -identifiant de G , est définie comme étant la limite :

$$d(C, G) = \lim_{r \rightarrow \infty} \sup \frac{|C \cap B_r(v)|}{|B_r(v)|}.$$

2.2.3 Identification d'ensembles de sommets

On peut, de plus supposer qu'à tout moment il y ait au plus ℓ processeurs défectueux dans le réseau, où ℓ est une constante fixée. Pour faire face à cette éventualité, on définit les codes identifiant les ensembles d'au plus ℓ sommets de G .

De manière formelle, on dit que C est un code identifiant les ensembles d'au plus ℓ sommets de $G = (V, E)$ si et seulement si, les ensembles identifiants $I(X, C)$ sont distincts pour tous les sous-ensembles X de cardinalité au plus ℓ de V , où $I(X, C)$ est défini comme l'union des ensembles identifiants des sommets de X :

$$I(X, C) = \bigcup_{v \in X} I(v, C).$$

Dans ce cas, on parle de code $(1, \leq \ell)$ -identifiant, On peut bien entendu combiner ceci avec l'identification à distance $r \leq 1$, on parle alors de code $(r, \leq \ell)$ -identifiant, qu'on définit comme suit :

Définition 2.2.3 (Code $(r, \leq \ell)$ -identifiant). :

On dit qu'un code C , est $(r, \leq \ell)$ -identifiant si et seulement si : les ensembles $I_r(X, C)$ sont distincts pour tous les sous-ensembles d'au plus ℓ sommets X de G , où $I_r(X, C)$ est défini comme :

$$I_r(X, C) = \bigcup_{v \in X} I_r(v, C)$$

2.2.4 Identification dans les graphes orienté

En considérant les liens entre les processeurs du réseau soient directionnels, c'est-à-dire que le graphe du réseau considéré soit orienté. En ce cas, un processeur p_1 peut tester un processeur p_2 si et seulement si il y a un arc de p_1 vers p_2 . Le fait que p_1 puisse tester p_2 n'implique pas que p_2 puisse tester p_1 .

Au niveau graphique, il suffit dans ce cas de remplacer $N[v]$ par $\Gamma^-[v]$ dans la définition d'un code identifiant, où $\Gamma^-[v]$ désigne l'ensemble fermé des voisins entrants de v :

$$\Gamma^-[v] = \Gamma^-(v) \cup \{v\}.$$

Définition 2.2.4 (Identification dans les graphes orienté). :

Un sous-ensemble de sommets C d'un graphe orienté G , est un code identifiant de G si et seulement si :

- pour tout sommet $v \in V$, on a

$$\Gamma^-[v] \cap C \neq \emptyset.$$

- pour toute paire de sommets distincts u, v , On a

$$\Gamma^-[u] \cap C \neq \Gamma^-[v] \cap C.$$

L'ensemble identifiant de v , $\Gamma^-[v] \cap C$, est noté $I^-(v)$.

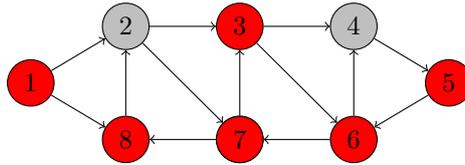


FIGURE 2.7 – Exemple d'un graphe orienté muni d'un code identifiant, $C = \{1, 3, 5, 6, 7, 8\}$.

2.2.5 Identification Robuste

On peut supposer que les processeurs détecteurs peuvent être sujet à des défaillances.

Définition 2.2.5 (Code identifiant r -robuste). [16]

Soit C un code identifiant de $G = (V, E)$, on dit que C est un r -robuste si :

- pour tout $u, v \in V$ et $A, B \subseteq V$ avec $|A|, |B| \leq r$ on a :

$$I(u, C) \Delta A \neq I(v, C) \Delta B$$

En d'autres termes, on peut permettre l'ajout, ou la suppression d'au plus r mot de code de n'importe quelle ensemble identifiant sans que cet ensemble perd sa propriété d'identification. Alternativement, il est possible de redéfinir les codes identifiants r -robustes moyennant la différence symétrique. Soit Δ_{min} la cardinalité minimum de toutes différences symétriques entre deux sommets quelconques, c'est à dire :

$$\Delta_{min}(C) = \min_{u, v \in V, u \neq v} |I(u, C) \Delta I(v, C)|$$

entre les ensembles identifiants de deux sommets quelconques. On a donc le théorème suivant qui est une application directe de la définition.

Théorème 2.2.1. [16]

Le code identifiant C est r -robuste si et seulement si :

$$\Delta_{min}(C) \geq 2r + 1$$

2.2.6 Identification Dynamique

Une chaîne $\langle c_1, c_2, \dots, c_m \rangle$ dans un graphe non orienté $G = (V, E)$, est appelée un code d'identification dynamique, si tous les ensembles $I(v) = \{u \in C : d(u, v) \leq 1\}$ car $v \in V$ ne sont pas vides et aucun d'entre eux n'est le même ensemble. Ici $d(u, v)$ désigne la distance entre u et v . Les codes d'identification dynamiques en grilles carré, grilles triangulaires, maillages hexagonaux et hypercubes binaires [11, 10].

Définition 2.2.6 (Identification Dynamique). Soit G un graphe. Une suite c_1, c_2, \dots, c_m de sommets tel que :

- $d(c_i, c_{i+1}) = 1$ pour tout $i = 1, 2, \dots, m - 1$, où $\langle c_1, c_2, \dots, c_m \rangle$ forme une chaîne dans G ,
- $\{c_1, c_2, \dots, c_m\}$ est un code r -identifiant.

C est appelé un code r -identifiant dynamique dans G . (si $r = 1$, on parle juste d'un code d'identification dynamique).

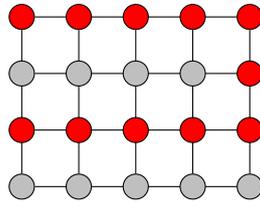


FIGURE 2.8 – Un code d'identification dynamique dans $P_5 \square P_4$.

2.2.7 Existence d'un code identifiant

Il n'existe pas toujours de code identifiant dans un graphe. En effet, si deux sommets ont exactement le même voisinage fermé, alors aucun sommet ne pourra les séparer. Deux sommets u et v tels que $N[u] = N[v]$ sont appelés des sommets jumeaux (voir la figure 2.9). C'est la seule restriction puisque si un graphe n'a pas de jumeaux alors l'ensemble de tous ses sommets est clairement un code identifiant. Les graphes complets K_n , font partie de graphes, qui n'admettant pas de code identifiant.

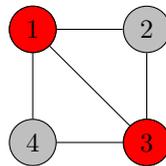


FIGURE 2.9 – Graphe avec des sommets jumeaux , indiqué en rouge.

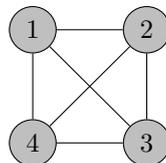


FIGURE 2.10 – Graphe complet K_4 , n'admettant pas de code identifiant.

Proposition 2.2.1. :

Un graphe admet un code identifiant si et seulement s'il est sans jumeaux .

Proposition 2.2.2. :

Soient G un graphe sans jumeaux et C un code identifiant de G . Alors tout ensemble C' contenant C est un code identifiant de G .

v

2.2.8 Résultats généraux

Dans les parties précédentes, nous avons défini comment sélectionner un code identifiant, et nous avons étudié sa existence, mais le problème le plus difficile, c'est de donner un code identifiant de cardinalité minimum de manière générale. Cependant, nous pouvons quand même borner le cardinal d'un code identifiant d'un graphe.

En ce qui concerne la borne supérieure, on doit le résultat à Sylvain Gravier et Julien Moncel [5].

Théorème 2.2.2. :

Soit G un graphe sans jumeaux possédant au moins une arête. Alors

$$\gamma^{Id}(G) \leq |n - 1|.$$

De plus, cette borne est serrée :elle est atteinte notamment pour les étoiles¹ $K_{1,n}$ avec $n \geq 2$.

D'autre part, concernant la borne inférieure, nous avons :

Théorème 2.2.3. :

Soit G un graphe sans jumeaux possédant au moins une arête. Alors

$$\gamma^{Id}(G) \geq \lceil \log_2(n + 1) \rceil.$$

Une preuve constructive de ce théorème est disponibles dans [12].

2.3 Code identifiant dans les grilles

Les grilles peuvent apparaitre comme un modèle pour les réseaux,un réseaux de multiprocesseur en est un exemple que nous avons abordé précédemment.

On a quatre types de grilles ,"grille carrée,grille triangulée, grille royale et grille hexagonal" mais on a abordé dans notre mémoire deux types des grilles.

Définition 2.3.1 (Grille infinie). [12] :

La grille n -dimensionnelle (infinie) est le produit cartésien de chemins infinis P_∞ ,dont l'ensemble de sommets est \mathbb{Z}^n est l'ensemble d'arêtes $\{uv, d(u, v) = \sum_{i=1}^n |v_i - u_i| = 1\}$ où $d(u, v)$ (distance de Manhattan) représente la longueur d'un plus courts chemin entre u et v . Le mot grille désigne généralement la grille bidimensionnelle \mathbb{Z}^2 .

La densité optimale du code identifiant C dans une grille infinie est donné pa le théorème suivant :

1. sont les graphes biparti complet $K_{1,n}$

Théorème 2.3.1. :

$$d(C, \mathbb{Z}^2) = \frac{7}{20}.$$

Ce résultat a été prouvé en deux temps. En 1999, Cohen et al. ont prouvé que $d(C, \mathbb{Z}^2) \leq \frac{7}{20}$, et en 2005, Ben-Haim et Litsyn ont montré que $d(C, \mathbb{Z}^2) \geq \frac{7}{20}$.

Définition 2.3.2 (Grille finie). [12] :

Une grille $k \times n$ (finie) est le sous graphe $\mathcal{G}_{k \times n}$ de la grille induit par le sous ensemble de sommets $\{1, \dots, k\} \times \{1, \dots, n\}$.

Pour les grilles finies, on a les bornes suivantes :

Théorème 2.3.2. :

Pour tous $k, n \geq 2$, on a :

$$\frac{7}{20} - \frac{1}{2} \frac{k+n}{kn} \leq d(\mathcal{G}_{k \times n}) \leq \frac{7}{20} + 2 \frac{k+n-2}{kn}$$

Codes Identifiants dans le produit cartésien de deux chemins

Dans ce chapitre, nous allons étudier le problème de codes identifiants dans le produit cartésien de deux chemins. Nous présentons les premiers résultats obtenus dans $P_4 \square P_n$, tel que P_4 et P_n sont deux chemins.

Nous énonçons les résultats de cas de chemin et $P_2 \square P_n$ ainsi que les démonstrations.

3.1 Préliminaires

Pour rappels, soit $G = (V, A)$ un graphe orienté.

On dit que le sous-ensemble C est un code identifiant dans G si et seulement si :

- pour tout sommet $v \in V$, on a : $I^-(v) \neq \emptyset$.
- pour toute paire de sommets distincts u, v , on a : $I^-(u) \neq I^-(v)$.

Dans le premier chapitre de ce mémoire, on a évoqué les propriétés du produit cartésien, et parmi ces propriétés on trouve la commutativité c'est à dire :

$$P_m \square P_n = P_n \square P_m.$$

Ce qui veut dire que la recherche d'un code identifiant dans $P_m \square P_n$ est similaire à sa recherche dans $P_n \square P_m$.

Dans ce qui suit, on va utiliser cette notation :

- $\forall n, m \in \mathbb{N}^*$:

$$V(P_m \square P_n) = \{v_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m\}.$$

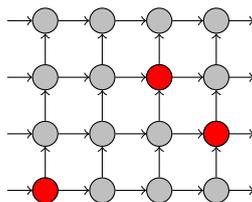


FIGURE 3.1 – Les sommets $v_{1,1}, v_{3,3}$ et $v_{4,2}$ dans $V(P_4 \square P_n)$.

- On notera un code identifiant dans $P_m \square P_n$ par C .

- Soit G un graphe quelconque, on notra $\gamma^{Id}(G)$ la cardinalité minimum d'un code identifiant dans G .

Lemme 3.1.1. :

Soient $v_{1,1}$, $v_{1,2}$ et $v_{2,1}$ trois sommets de $P_m \square P_n$ alors ils doivent être des mots de code .

Démonstration. :

- Si $v_{1,1} \notin C$ alors $I^-(v_{1,1}) = \emptyset$, c'est à dire que $v_{1,1}$ ne sera pas couvert par C . Donc $v_{1,1} \in C$.
- Si $v_{2,1} \notin C$ alors $I^-(v_{2,1}) = I^-(v_{1,1}) = \{v_{1,1}\}$, c'est à dire que $v_{1,1}$ et $v_{2,1}$ ne seront pas séparés. D'où $v_{2,1} \in C$ doit être dans C .
- De même manière, si $v_{1,2} \notin C$ alors les deux sommets $v_{1,2}$ et $v_{1,1}$ ne seront pas séparés . □

Lemme 3.1.2. :

Soient $v_{i,j}, v_{i,j-1}$ et $v_{i-1,j}$ trois sommets de $P_m \square P_n$ avec $i \geq 2, j \geq 2$, alors au moins l'un d'entre eux doit être un mot de code.

Démonstration. :

- On sait que $\Gamma^-[v-i, j] = \{v-i, j, v-i-1, j, v-i, j-1\}$, alors si aucun éléments de $\Gamma^-[v-i, j]$ n'est du mot de code, alors $I^-(v_{i,j}) = \emptyset$ c'est à dire $v_{i,j}$ ne sera pas couvert.

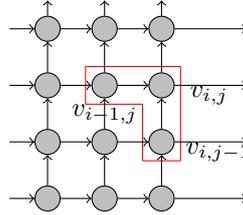


FIGURE 3.2 – Disposition des sommets dans la grille.

□

Ces deux résultats nous seront nécessaire dans la suite.

3.2 Code identifiant dans les chemins

Soit $P_n = \{v_i : 1 \leq i \leq n\}$ un chemin de longueur n .

Lemme 3.2.1. [2] :

Le sous-ensemble C est un code identifiant de P_n si et seulement si :

1. Les deux sommets v_1 et v_2 sont des mots de code,
2. Pour toute paire de sommets consécutifs, v_i et v_{i+1} , $i \in \{3, 4, \dots, n-1\}$, v_i ou v_{i+1} est un mot de code,
3. Pour tout triplet de sommets consécutifs, v_i, v_{i+1} et v_{i+2} , $i \in \{1, 4, \dots, n-2\}$, v_i est un mot de code ou v_{i+2} est un mot de code.

Démonstration. :

- Pour (1) :
On suppose que $v_1 \notin C$ alors $I^-(v_1) = \emptyset$, c'est à dire v_1 ne sera pas couvert par C . Donc $v_1 \in C$.
On suppose que $v_2 \notin C$ alors $I^-(v_2) = I^-(v_1) = \{v_1\}$, c'est à dire v_1 et v_2 ne seront pas séparés. Donc $v_2 \in C$.
- Pour (2) :
On suppose que $v_i \notin C$ et $v_{i+1} \notin C$ alors $I^-(v_{i+1}) = \emptyset$. Donc $v_i \in C$ ou $v_{i+1} \in C$.
- pour (3) :
On suppose que $v_i \notin C$ et $v_{i+2} \notin C$ alors on a deux cas :



FIGURE 3.3 – Les deux cas possibles. Dans la figure a gauche on a $v_{i+1} \in C$, et dans la figure a droite on a $v_{i+1} \notin C$

- Cas 1 :** Si $v_{i+1} \in C$ alors on aura $I^-(v_{i+1}) = I^-(v_{i+2}) = \{v_{i+1}\}$, c'est à dire que v_{i+1} et v_{i+2} ne seront pas séparés.
- Cas 2 :** Si $v_{i+1} \notin C$ alors on aura forcément $I^-(v_{i+1}) = I^-(v_{i+2}) = \emptyset$, c'est à dire que v_{i+1} et v_{i+2} ne seront pas couverts.

On remarque que les conditions sont nécessaires et suffisantes pour la couverture de tous les sommets de P_n .

Maintenant, on montre que les conditions sont suffisantes pour la séparations.

Deux cas se présentent :

- Cas 1 :** Si les deux sommets sont voisins(c à d v_i et v_{i+1}).
On a vu que la condition (1) permet la séparation de v_1 et v_2 .
Pour les autres paires de sommets consécutifs, d'après la condition (3) on a $v_{i-1} \in C$ ou $v_{i+1} \in C$, ainsi on aura $I^-(v_i) \neq I^-(v_{i+1})$.
Donc v_i et v_{i+1} seront séparés.
- Cas 2 :** Si les deux sommet ne sont pas voisins(c à d v_i et v_{i+l} ($l \geq 2$)).
D'après la condition (2) on a $v_{i+1} \in C$ ou $v_{i+l} \in C$, ainsi an aura $I^-(v_i) \neq I^-(v_{i+l})$.
Donc v_i et v_{i+l} seront séparés.

Alors les conditions sont suffisantes pour la séparations.

De ce qui précède, on conclut que pour chaque trois sommets consécutifs au moins deux sont des mots de code. □

Théorème 3.2.1 (La cardinalité minimum). [2]

La cardinalité minimum d'un code identifiant, dans un chemin orienté est donnée par :

$$\gamma^{Id}(P_n) = \begin{cases} 2p & Si \quad n = 3p. \\ 2p + 1 & Si \quad n = 3p + 1. \\ 2p + 2 & Si \quad n = 3p + 2. \end{cases}$$

La démonstration de ce théorème peut être trouvée dans [2].

Dans la suite, nous allons étudier le problème de recherche d'un code identifiant de cardinalité minimum sur le produit cartésienne $P_m \square P_n$.

Nous considérons, dans un premier temps le cas où $m = 2$ puis le cas où $m = 4$.

3.3 Code identifiant dans $P_2 \square P_n$

Lemme 3.3.1. :

Soit le sous-ensemble suivant :

$$C = \{v_{i,1} : 1 \leq i \leq n\} \cup \{v_{1,2}\}.$$

Alors C est un code identifiant de $P_2 \square P_n$.

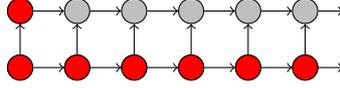


FIGURE 3.4 – Les sommets rouges forment un code identifiant C de $P_2 \square P_n$.

Démonstration. :

Soit $v_{i,j}$ un sommet quelconque de $P_2 \square P_n$, alors on a :

Cas 1 : Pour tout $v_{i,1}$, tel que $i \geq 1$:

On a $I^-(v_{i,1}) \neq \emptyset$ car $v_{i,1} \in C$.

Cas 2 : Pour tout $v_{i,2}$, tel que $i \geq 1$:

Puisque $v_{i,1} \in \Gamma^-[v_{i,2}]$, et $v_{i,1} \in C$. Alors $I^-(v_{i,2}) \neq \emptyset$.

Donc, l'ensemble C est bien un code couvrant.

Montrons maintenant que C est un code séparateur.

Soient $v_{i,j}, v_{i-1,j}, v_{i+1,j}$ et $v_{i,j+1}$ des sommets de $P_2 \square P_n$, alors 5 cas se présentent :

Cas 1 : $v_{1,1} \in C, v_{1,2} \in C$ et $v_{2,1} \in C$ d'après lemme 3.1.1 les 3 sommets sont séparés.

Cas 2 : Pour tout $v_{i,j} \in \{v_{i,1} : 3 \leq i \leq n-1\}$:

$$\begin{aligned} \text{On a : } I^-[v_{i,j}] &= \{v_{i-1,j}, v_{i,j}\}. & I^-[v_{i-1,j}] &= \{v_{i-2,j}, v_{i-1,j}\}. & I^-[v_{i+1,j}] &= \{v_{i,j}, v_{i+1,j}\}. \\ I^-[v_{i,j+1}] &= \{v_{i,j}\}. \end{aligned}$$

donc ils sont séparés, puisqu'ils sont tous différents .

Cas 3 : Pour tout $v_{i,j} \in \{v_{i,2} : 2 \leq i \leq n-1\}$:

De manière similaire, on observe que tous les sommets sont séparés.

Cas 4 : Pour $v_{2,2}$: On a : $I^-[v_{2,2}] = \{v_{1,2}, v_{2,1}\}$. $I^-[v_{1,2}] = \{v_{1,2}, v_{1,1}\}$. $I^-[v_{2,1}] = \{v_{2,1}, v_{1,1}\}$. $I^-[v_{3,2}] = \{v_{3,1}\}$.

donc ils sont séparés.

Cas 5 : Pour $v_{n,1}$ et $v_{n-1,1}$: On a : $I^-[v_{n,1}] = \{v_{n,1}, v_{n-1,1}\}$. $I^-[v_{n-1,1}] = \{v_{n-2,1}, v_{n-1,1}\}$. $I^-[v_{n,2}] = \{v_{n,1}\}$. $I^-[v_{n-1,2}] = \{v_{n-1,1}\}$.

donc ils sont séparés.

D'où C est un code identifiant dans $P_2 \square P_n$. □

Théorème 3.3.1 (Cardinalité minimum). :

La cardinalité minimum d'un code identifiant, dans $P_2 \square P_n$ est donnée par :

$$\gamma^{Id}(P_2 \square P_n) = n + 1.$$

Démonstration. :

La cardinalité minimum d'un code couvrant (dominant), notée par $\gamma(P_2 \square P_n)$.

D'après M.Mollard [15] :

$$\gamma(P_2 \square P_n) = n.$$

On sait que $\gamma^{Id}(P_2 \square P_n) \geq \gamma(P_2 \square P_n)$ [12]. Donc $\gamma^{Id}(P_2 \square P_n) \geq n$.
 Puisque $\gamma(P_2 \square P_n) = n$ alors $C = \{v_{i,1}\}$ est un ensemble dominant.

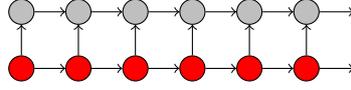


FIGURE 3.5 – $\gamma(P_2 \square P_n) = n$

Or d'après le lemme 3.1.1 pour que C soit un code identifiant, le sommet $v_{1,2}$ doit être dans le code. Alors $C = \{v_{i,1}\} \cup \{v_{1,2}\}$ et on a déjà montré que $C = \{v_{i,1}\} \cup \{v_{1,2}\}$ est un code identifiant .

D'où : $\gamma^{Id}(P_2 \square P_n) = n + 1$. □

3.4 Code identifiant dans $P_4 \square P_n$

Lemme 3.4.1. :

Soit le sous-ensemble de sommets suivant :

$$C = \{v_{i,1} : 1 \leq i \leq n\} \cup \{v_{i,3} : 1 \leq i \leq n\} \cup \{v_{1,2}\}.$$

Alors C est un code identifiant de $P_4 \square P_n$.

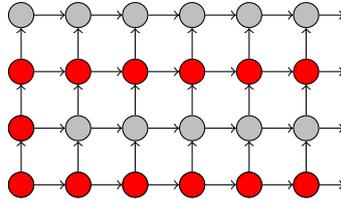


FIGURE 3.6 – Les sommets rouges forment un code identifiant C de $P_4 \square P_n$.

Démonstration. :

Pour montrer que C est un code identifiant on procède par deux étapes : on montre dans un premier temps que C est couvrant, ensuite on montre que C est séparateur.

Soit $v_{i,j}$ un sommet quelconque de $P_4 \square P_n$, alors on a :

Cas 1 : Pour $v_{i,1}$ et $v_{i,3}$ on a $I^-(v_{i,1}) \neq \emptyset$ et $I^-(v_{i,3}) \neq \emptyset$ car $v_{i,1} \in C$ et $v_{i,3} \in C$.

Cas 2 : Pour $v_{i,2}$ et $v_{i,4}$ on a : $v_{i,1} \in \Gamma[v_{i,2}]$ et $v_{i,1} \in C$ alors $I^-(v_{i,2}) \neq \emptyset$; De même pour $v_{i,4}$. Donc pour tout $v_{i,j}$ on a :

$$I^-(v_{i,j}) \neq \emptyset.$$

Alors le code vérifie les conditions nécessaire et suffisantes pour la couverture de tout sommets de $P_4 \square P_n$. On montre que C vérifie la condition de séparation. Soient $v_{i,j}, v_{i,j-1}, v_{i,j+1}, v_{i-1,j}$ et $v_{i+1,j}$ des sommets quelconques de $P_4 \square P_n$, alors 6 cas se présentent :

Cas 1 : $v_{1,1} \in C, v_{1,2} \in C$ et $v_{2,1} \in C$ d'après lemme 3.1.1 les 3 sommets sont séparés.

Cas 2 : Pour $v_{i,j} \in \{v_{i,1} : 3 \leq i \leq n-1\}$ on a :

$$I^-(v_{i,j}) = \{v_{i,j}, v_{i-1,j}\}, I^-(v_{i+1,j}) = \{v_{i,j}, v_{i+1,j}\}, I^-(v_{i-1,j}) = \{v_{i-2,j}, v_{i-1,j}\} \text{ et } I^-(v_{i,j+1}) = \{v_{i,j}\}.$$

Donc ils sont séparés, puis qu'ils sont tous différents .

Cas 3,4 et 5 : Pour $v_{i,j} \in \{v_{i,2} : 2 \leq i \leq n-1\}, v_{i,j} \in \{v_{i,3} : 2 \leq i \leq n-1\}$ et $v_{i,j} \in \{v_{i,4} : 2 \leq i \leq n-1\}$:

De manière similaire à le premier cas, on observe que tous les sommets sont séparés.

Cas 6 : Dans ce cas on va vérifié la séparation des sommets qui sont sur les bords, alors 6 sommets se présentent :

• Pour $v_{1,3}$ et $v_{1,4}$ on a : $I^-(v_{1,3}) = \{v_{1,2}, v_{1,3}\}, I^-(v_{2,3}) = \{v_{1,3}, v_{2,3}\}, I^-(v_{1,2}) = \{v_{1,1}, v_{1,2}\}, I^-(v_{1,4}) = \{v_{1,3}\}$ et $I^-(v_{2,4}) = \{v_{2,3}\}$.

Donc ils sont séparés .

• Pour $v_{n,1}, v_{n,2}, v_{n,3}$ et $v_{n,4}$ on a : $I^-(v_{n,1}) = \{v_{n-1,1}, v_{n,1}\}, I^-(v_{n-1,1}) = \{v_{n-2,1}, v_{n-1,1}\}, I^-(v_{n,2}) = \{v_{n,1}\}, I^-(v_{n-1,2}) = \{v_{n-1,1}\}, I^-(v_{n,3}) = \{v_{n-1,3}, v_{n,3}\}, I^-(v_{n-1,3}) = \{v_{n-2,3}, v_{n-1,3}\}, I^-(v_{n,4}) = \{v_{n,3}\}$ et $I^-(v_{n-1,4}) = \{v_{n-1,3}\}$.

Donc ils sont séparés .

Donc C est vérifié les conditions de couverture et de séparation. Alors C est un code identifiant dans $P_4 \square P_n$.

□

Remarque 3.4.1. :

Dans le code C , nous remarquons qu'à partir de la deuxième colonne, toutes les colonnes contiennent deux mots de code.

Lemme 3.4.2. :

Soit C un code identifiant dans G Soient c_i et c_{i+1} deux colonnes successives de G . Alors on a :

$$|(c_{i+1} \cup c_i) \cap C| \geq 4.$$

Démonstration. :

On suppose qu'une colonne c_i de $P_4 \square P_n$ avec $2 \leq i \leq n$ peut avoir deux mots de code, un seul mot ou aucun mot c'est à dire on a 3 cas possibles :

On commence par :

1. S'il existe $i : c_i \cap C = \emptyset$

• On a $c_{i,1}$ n'est pas un mot de code donc d'après lemme 3.2.1 les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code.

• Le sommet $c_{i-1,2}$ doit être un mot de code pour couvrir $c_{i,2}$.

• Les deux sommets $c_{i-1,3}$ et $c_{i-1,4}$ sont des mots de codes d'après lemme 3.2.1 .

Alors le min de sommets qui doivent être dans C sur c_{i-1} et c_i égale à 4.

• Le sommet $c_{i+1,2}$ doit être dans le code pour séparé $c_{i+1,1}$.

• D'après lemme 3.1.2 l'un des deux sommets $c_{i+1,4}$ ou $c_{i+1,3}$ doit être dans le code .

- Si $c_{i+1,3} \in C$:

On a d'après lemme 3.1.2 l'un des trois sommets $c_{i+2,3}$ ou $c_{i+2,4}$ ou $c_{i+1,4}$ doit être un mot de code donc on peut prend $c_{i+1,4}$, donc le min de sommets entre c_i et c_{i+1} égale à 4.

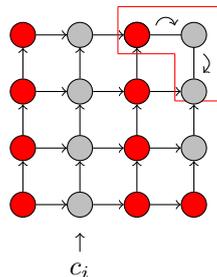


FIGURE 3.7 – Si $c_{i+1,3} \in C$.

- Si $c_{i+1,4} \in C$:

On a d'après lemme 3.1.2 l'un des trois sommets $c_{i+2,3}$ ou $c_{i+2,2}$ ou $c_{i+1,3}$ doit être un mot de code donc on peut prendre $c_{i+1,3}$. Alors le min de sommets entre c_i et c_{i+1} égale à 4.

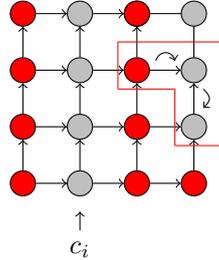


FIGURE 3.8 – Si $c_{i+1,4} \in C$.

Alors le min de sommets entre deux colonnes s'existe $i : c_i \cap C = \emptyset$ égale à 4.

2. S'existe $i : c_i \cap C = 1$, alors 4 cas se présentent :

Cas 1 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,1}\}$, alors d'après lemme 3.2.1 l'un des deux sommets $c_{i-1,1}$ ou $c_{i+1,1}$ doit être un mot de code, donc on aura deux sous-cas possibles :

Cas 1.1 : Si le sommet $c_{i-1,1}$ est un mot de code C :

- Le sommet $c_{i-1,2}$ est un mot de code pour séparer $c_{i,2}$ et $c_{i,1}$.
- Les deux sommets $c_{i-1,3}$ et $c_{i-1,4}$ sont des mots de code d'après lemme 3.1.2. Donc le min de sommets entre c_{i-1} et c_i égale à 5.
- Le sommet $c_{i+1,2}$ doit être dans le code pour couvrir lui même, le sommet $c_{i+1,3}$ pour séparer $c_{i+1,2}$.
- D'autre part, d'après lemme 3.1.2 l'un des trois sommets $c_{i+1,4}$ ou $c_{i+2,4}$ ou $c_{i+2,3}$ doit être un mot de code, alors on peut prendre $c_{i+1,4}$. Donc le min de sommets entre c_i et c_{i+1} égale à 4.

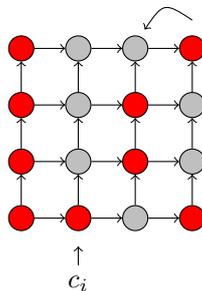


FIGURE 3.9 – le premier sous-cas 1.1

Alors le min de sommets sans ce sous-cas égale à 4.

Cas 1.2 : Si le sommet $c_{i+1,1}$ est un mot de code C :

- Les trois sommets $c_{i-1,2}$, $c_{i-1,3}$ et $c_{i-1,4}$ doit être des mots de code pour la même raison. Donc le min de sommets entre c_i et c_{i-1} égale à 4.
- L'un des deux sommets $c_{i+1,2}$ ou $c_{i+1,3}$ doit être un mot de code d'après lemme 3.1.2 .
- Le sommet $c_{i+1,4}$ doit être un mot de code :

- Si le sommet $c_{i+1,2}$ est mot de code, alors le sommet $c_{i+1,4}$ doit être un mot de code pour séparé $c_{i+1,3}$,
- Si le sommet $c_{i+1,3}$ est mot de code, alors le sommet $c_{i+1,4}$ doit être un mot de code d'après lemme 3.1.2.

Donc le min de sommets entre c_i et c_{i+1} égale à 4.

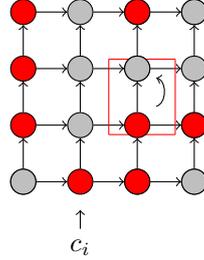


FIGURE 3.10 – Le deuxième sous-cas 1.2.

Alors le min de sommets sans ce sous-cas égale à 4.

Donc le min de sommets entre deux colonne s'existe $i : c_i \cap C = \{c_{i,1}\}$ égale à 4.

Cas 2 : Si $c_i \cap C = \{c_{i,2}\}$ on a :

- Les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code d'après lemme 3.2.1.
- L'un des deux sommets $c_{i-1,2}$ ou $c_{i-1,3}$ pour séparé de $c_{i,2}$ et $c_{i,3}$.
- Le sommet $c_{i-1,4}$ est mot de code d'après lemme 3.1.2.

Donc le min de sommets entre c_{i-1} et c_i égale à 4.

- L'un des deux sommets $c_{i+1,2}$ ou $c_{i+1,3}$ doit être dans le code d'après lemme 3.1.2.
- Le sommet $c_{i+1,4}$ est mot de code pour séparé $c_{i+1,3}$.

Donc le min de sommets entre c_i et c_{i+1} égale à 4.

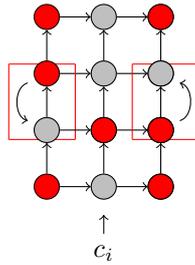


FIGURE 3.11 – Le deuxième cas : $c_i \cap C = \{c_{i,2}\}$

Alors le min de sommets entre deux colonne s'existe $i : c_i \cap C = \{c_{i,2}\}$ égale à 4.

Cas 3 : Si $c_i \cap C = \{c_{i,3}\}$ on a :

- Les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code d'après lemme 3.1.2.
- Le sommet $c_{i-1,2}$ doit être un mot de code pour couvrir $c_{i,2}$.
- L'un des deux sommets $c_{i-1,3}$ ou $c_{i-1,4}$ doit être un mot de code pour la séparation de $c_{i,3}$ et $c_{i,4}$.

Donc le min de sommets entre c_i et c_{i-1} égale à 4.

- Le sommet $c_{i+1,2}$ doit être un mot de code pour séparé $c_{i+1,1}$.
- L'un des deux sommets $c_{i+1,3}$ ou $c_{i+1,4}$ doit être un mot de code d'après lemme 3.1.2.

Donc le min de sommets entre c_i et c_{i+1} égale à 4.

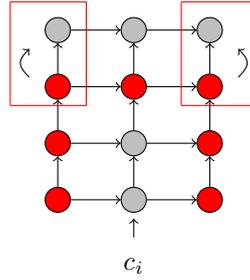


FIGURE 3.12 – Le deuxième cas : $c_i \cap C = \{c_{i,3}\}$

Alors le min de sommets entre deux colonne si $\exists i : c_i \cap C = \{c_{i,3}\}$ égale à 4.

Cas 4 : Si $c_i \cap C = \{c_{i,4}\}$ on a :

- Les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code d'après lemme 3.1.2.
- • Les deux sommets $c_{i-1,2}$ et $c_{i-1,3}$ sont des mot de code pour la couverture de $c_{i,2}$ et $c_{i,3}$.

Donc le min de sommets entre c_i et c_{i-1} égale à 4.

- Le sommet $c_{i+1,2}$ doit être un mot de code pour séparé $c_{i+1,1}$.
- L'un des deux sommets $c_{i+1,3}$ ou $c_{i+1,4}$ doit être un mot de code pour séparé $c_{i,4}$ et $c_{i+1,4}$.

Donc le min de sommets entre c_i et c_{i-1} égale à 4.

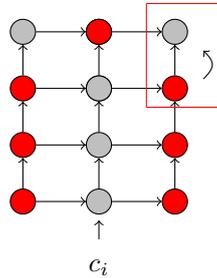


FIGURE 3.13 – Le deuxième cas : $c_i \cap C = \{c_{i,4}\}$

Alors le min de sommets entre deux colonne s'existe $i : c_i \cap C = \{c_{i,4}\}$ égale à 4.

Donc dans s'existe $i : c_i \cap C = 1$ le min de sommets entre deux colonne égale à 4.

3. s'existe $i : c_i \cap C = 2$ alors on a 6 cas :

Cas 1 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,1} \cup c_{i,2}\}$, alors d'après lemme 3.2.1 l'un des deux sommets $c_{i-1,1}$ ou $c_{i+1,1}$ doit être un mot de code, donc on aura deux sous-cas possibles :

Cas 1.1 : Si le sommet $c_{i-1,1}$ est un mot de code C :

- Le sommet $c_{i-1,4}$ est un mot de code d'après lemme 3.1.2.
Donc le min de sommet entre c_i et c_{i-1} égale à 4.
- Le sommet $c_{i+1,3}$ est mot de code d'après lemme 3.1.2.
- Le sommet $c_{i+1,4}$ doit être un mot de code pour séparé $c_{i+1,3}$.
Donc le min de sommet entre c_i et c_{i+1} égale à 4.

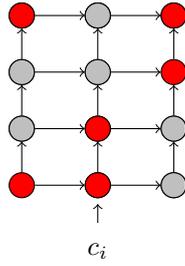


FIGURE 3.14 – Le premier sous-cas : $c_i \cap C = \{c_{i,1} \cup c_{i,2}\}$

Cas 1.2 : Si le sommet $c_{i+1,1}$ est un mot de code C :

- Le sommet $c_{i-1,4}$ est un mot de code d'après lemme 3.1.2.

Dans c_{i-1} on a juste $c_{i-1,4}$ qui doit être un mot de code .

Mais dans la colonne c_{i-2} tout les sommet doit être des mots de code .

- D'après lemme 3.1.2 l'un des trois sommet suivants $c_{i-1,2}$ ou $c_{i-1,3}$ ou $c_{i-2,3}$ doit être dans le code donc on peut prend $c_{i-1,3}$ ou $c_{i-1,4}$ à la place de $c_{i-2,3}$ et C reste un code identifiant .

Donc le min de sommet entre c_{i-1} et c_i égale à 4.

- L'un des deux sommets $c_{i+1,2}$ ou $c_{i+1,3}$ doit être un mot de code d'après lemme 3.1.2 .

- Le sommet $c_{i+1,4}$ est un mot de code pour séparé $c_{i+1,3}$.

Donc le min de sommets entre c_i et c_{i+1} égale à 5.

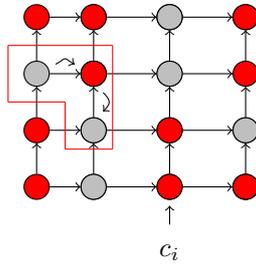


FIGURE 3.15 – Le deuxième sous-cas : $c_i \cap C = \{c_{i,1} \cup c_{i,2}\}$

Alors le min de sommets entre deux colonne s'existe i : $c_i \cap C = \{c_{i,1} \cup c_{i,2}\}$ égale à 4.

Cas 2 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,1} \cup c_{i,3}\}$, alors d'après lemme 3.2.1 l'un des deux sommets $c_{i-1,1}$ ou $c_{i+1,1}$ doit être un mot de code, donc on aura deux sous-cas possibles :

Cas 2.1 : Si le sommet $c_{i-1,1}$ est un mot de code C :

- L'un des deux sommets $c_{i-1,3}$ ou $c_{i-1,4}$ doit être un mot de code pour séparé $c_{i,3}$ et $c_{i,4}$.

Donc le min de sommets entre c_{i-1} et c_i égale à 4 .

- Le sommet $c_{i+1,2}$ est un mot de code pour couvert lui même.

- L'un des deux sommets suivants $c_{i+1,3}$ ou c_{i+4} d'après lemme 3.1.2.

Donc le min de sommets entre c_{i+1} et c_i égale à 4 .

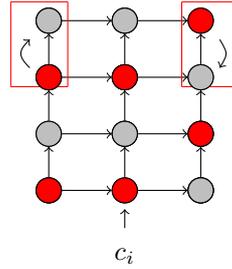


FIGURE 3.16 – Le premier sous-cas : $c_i \cap C = \{c_{i,1} \cup c_{i,3}\}$

Cas 2.2 : Si le sommet $c_{i+1,1}$ est un mot de code C :

bullet De même manière que le cas précédent on trouve que le min des sommet entre deux colonne égale à 4.

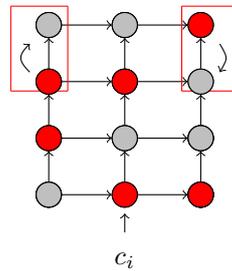


FIGURE 3.17 – Le deuxième sous-cas : $c_i \cap C = \{c_{i,1} \cup c_{i,3}\}$

Alors, le min de sommets entre deux colonne s'existe $i : c_i \cap C = \{c_{i,1} \cup c_{i,3}\}$ égale à 4.

Cas 3 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,1} \cup c_{i,4}\}$, alors d'après lemme 3.2.1 l'un des deux sommets $c_{i-1,1}$ ou $c_{i+1,1}$ doit être un mot de code, donc on aura deux sous-cas possibles :

Cas 3.1 : Si le sommet $c_{i-1,1}$ est un mot de code C :

- Le sommet $c_{i-1,3}$ est un mot de code d'après lemme 3.1.2. Donc le min de sommet entre c_i et c_{i-1} égale à 4.
- Le sommet $c_{i+1,2}$ doit être un mot de code pour couvert lui même.
- Le sommet $c_{i+1,3}$ doit être un mot de code pour séparé $c_{i+1,2}$ et $c_{i,4}$. Donc le min de sommet entre c_i et c_{i-1} égale à 4.

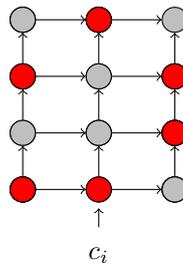


FIGURE 3.18 – Le premier sous-cas : $c_i \cap C = \{c_{i,1} \cup c_{i,4}\}$

Cas 3.2 : Si le sommet $c_{i+1,1}$ est un mot de code C :

bullet De même manière que le cas précédent on trouve que le min des sommet entre deux colonne égale à 4.

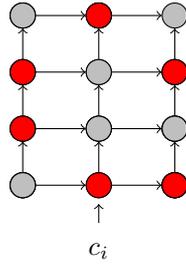


FIGURE 3.19 – Le deuxième sous-cas : $c_i \cap C = \{c_{i,1} \cup c_{i,4}\}$

Alors, le min de sommets entre deux colonne si $\exists i : c_i \cap C = \{c_{i,1} \cup c_{i,4}\}$ égale à 4.

Cas 4 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,2} \cup c_{i,3}\}$ alors on a :

- Les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code d'après lemme 3.1.1.
- L'un des trois sommets suivant $c_{i-2,3}$ ou $c_{i-1,2}$ ou $c_{i-1,3}$ doit être un mot , donc on peut prend $c_{i-1,2}$ ou $c_{i-1,3}$.

Donc le min de sommets entre c_{i-1} et c_i égale à 4.

- L'un des deux sommets $c_{i+1,3}$ ou $c_{i+1,4}$ doit être un mot de code . Donc le min de sommets entre c_{i+1} et c_i égale à 4.

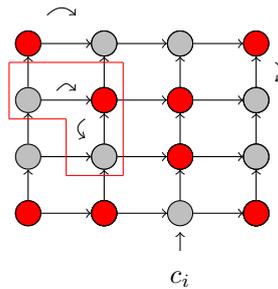


FIGURE 3.20 – Le quatrième cas : $c_i \cap C = \{c_{i,2} \cup c_{i,3}\}$

Alors, le min de sommets entre deux colonne s'il existe $i : c_i \cap C = \{c_{i,2} \cup c_{i,3}\}$ égale à 4.

Cas 5 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,2} \cup c_{i,4}\}$ alors on a :

- Les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code d'après lemme 3.1.1.
- L'un des deux sommets $c_{i-1,2}$ ou $c_{i-1,3}$ doit être un mot de code pour séparé $c_{i,2}$ et $c_{i,3}$.

Donc le min de sommets entre c_{i-1} et c_i égale à 4.

- Le sommet $c_{i+1,3}$ est un mot de code pour séparé $c_{i,4}$ et $c_{i+1,4}$. Donc le min de sommets entre c_{i+1} et c_i égale à 4.

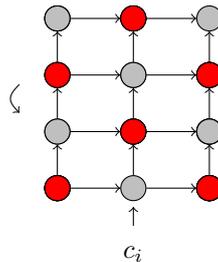


FIGURE 3.21 – Le cinquième cas : $c_i \cap C = \{c_{i,2} \cup c_{i,4}\}$

Alors, le min de sommets entre deux colonne s'il existe $i : c_i \cap C = \{c_{i,2} \cup c_{i,4}\}$ égale à 4.

Cas 6 : Dans ce cas on suppose que $c_i \cap C = \{c_{i,3} \cup c_{i,4}\}$ alors on a :

- Les deux sommets $c_{i-1,1}$ et $c_{i+1,1}$ sont des mots de code d'après lemme 3.1.1.
- Le sommet $c_{i-1,2}$ doit être un mot de code pour couvrir $c_{i,2}$.
Donc le min de sommets entre c_{i-1} et c_i égale à 4.
- Le sommet $c_{i+1,2}$ doit être un mot de code pour séparer $c_{i+1,1}$.
Donc le min de sommets entre c_{i+1} et c_i égale à 4.

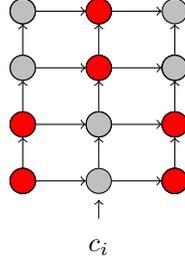


FIGURE 3.22 – Le sixième cas : $c_i \cap C = \{c_{i,3} \cup c_{i,4}\}$

Alors, le min de sommets entre deux colonnes s'il existe $i : c_i \cap C = \{c_{i,3} \cup c_{i,4}\}$ égale à 4.

Donc, dans tous les cas possibles de $P_4 \square P_n$ on a $|(c_{i+1} \cup c_i) \cap C| \geq 4$. □

Théorème 3.4.1 (Cardinalité minimum). :

La cardinalité minimum d'un code identifiant, dans $P_4 \square P_n$ est donnée par :

$$\gamma^{Id}(P_4 \square P_n) = 2n + 1.$$

Démonstration. :

D'après le lemme 3.4.2, on a la cardinalité minimum entre deux colonnes successives est supérieur ou égale à 4.

Par conséquent, le code C que nous avons déjà montré dans le lemme 3.4.1, vérifie le lemme 3.4.2. Alors, la cardinalité de C est minimale.

Or dans $P_4 \square P_n$ il y a $(n - 1)$ colonnes après la première colonne. D'après la Remarque 3.4.1 on a 2 mots de code dans toutes les colonnes qui sont après la première colonne, donc on a $2(n - 1)$ mots de code après la première, et puisqu'on a 3 mots de code, dans la première colonne de $P_4 \square P_n$ (lemme 3.4.1). Alors : $\gamma^{Id}(P_4 \square P_n) = 3 + 2(n - 1)$.

Donc :

$$\gamma^{Id}(P_4 \square P_n) = 2n + 1.$$

□

Conclusion Générale

Dans ce mémoire, nous avons abordé divers problèmes concernant les codes identifiants dans les graphes. Le problème de code identifiant a tout d'abord été lié à les problèmes de couverture par tests.

Nous avons étudié dans ce mémoire le problème de recherche de code identifiant de cardinalité minimum dans le produit cartésien de deux chemins, nous avons commencé notre étude sur ce problème avec un petit rappel sur les graphes, que nous avons nommés dans le premier chapitre avec < Généralités sur les graphes > .

Puis, nous nous sommes fixé pour objectif, d'une part, d'analyser et de comprendre le problème de code identifiant, en répondant à ces questions : Qu'est-ce qu'un code identifiant ? , Quelles sont leur variantes possibles ? ; Est-ce que le code identifiant existe dans tout les types de graphes ? (nous avons répondu à ces questions dans le deuxième chapitre).

D'autre part, d'essayer de répondre à la question suivante :
Quelle est la cardinalité minimum d'un code identifiant dans les produit cartésien de deux chemins ?

En effet, peu de résultats ont été obtenus dans le cas des graphes orientés en particulier le cas de produit cartésien de deux chemins. Parmi ces quelques résultats, nous avons trouvé le cas de chemin et circuit.

D'après les travaux réalisés dans le cas chemin, nous avons pu déterminer la cardinalité minimum d'un code identifiant dans le produit cartésien de deux chemins ($P_2 \square P_n$ et $P_4 \square P_n$). Tous les résultats ont été donnés dans le troisième chapitre.

Bibliographie

- [1] A.Parreau. *Problèmes d'identification dans les graphes*. PhD thesis, Université de Grenoble, <https://tel.archives-ouvertes.fr/tel-00745054>, 2012.
- [2] H.touati A.Semri. Optimal identifying codes in oriented paths and circuits. *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES*, 03 2021.
- [3] F. Foucaud. Les codes identifiants dans certaines classes de graphes. Master's thesis, Université de Bordeaux 1, 2009.
- [4] A. Lobstein G. Cohen, I. Honkala and G. Zémor. New bounds for codes identifying vertices in graphs. *The Electronic Journal of Combinatorics*, 1999.
- [5] S. Gravier and J. Moncel. On graphs having a $v \setminus \{x\}$ set as an identifying code. *Discrete Mathematics* 307, page 432–434, 2007.
- [6] Frank Harary. *Graph theory*. Addison-wesley, 1969.
- [7] H.Touati. Codes identifiants dans les graphes :cas de chemins et de circuits. Master's thesis, Université de USTHB, 2010.
- [8] H.Touati. *Cours éléments de Théorie des Graphes*, chapter 1 et 2. 2018-2019.
- [9] O. Hudrey I. Charon and A.Lobstein. Possible cardinalities for locating-dominating codes in graphs. *AUSTRALASIAN JOURNAL OF COMBINATORICS*, pages 23–31, 2006.
- [10] M.G.Karpovsky I.Honkala and L.B. Levitin. On dynamic identifying codes. *Information Theory,IEEE Transactions on*, 03 2004.
- [11] M.G.Karpovsky I.Honkala and L.B. Levitin. On robust and dynamic identifying codes. *Information Theory,IEEE Transactions on*, pages 599–612, 03 2006.
- [12] J.Moncel. *Codes Identifiants dans les Graphes*. PhD thesis, l'Université Joseph Fourier, <https://tel.archives-ouvertes.fr/tel-00010293v1>, 2006.
- [13] K. Chakrabarty M. G. Karpovsky and L. B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE TRANSACTIONS ON INFORMATION THEORY*, pages 599–611, 1998.
- [14] R.L.Gulagula M.D.Cruz, A.M.Gino and J.P.Onoya. Characterization of cartesian product of cycle graphs and path in maximal and optimal 1-planar graph. 2015.
- [15] M.Mollard. The domination number of cartesian product of two directed paths. *Journal of Combinatorial Optimization*, pages 144–151, 2014.
- [16] S. Ray, R. Ungrangsi, F. De Pellegrini, A. Trachtenberg, and D. Starobinski. Robust location detection in emergency sensor networks. *IEEE INFOCOM*, pages 1044–1053, 2003.
- [17] W.Imrich R.H.Hammack and S.Klavzar. *Handbook of Product Graphs*. CRC Press, 2011.

- [18] R. TRAYLOR. *Cartesian Product of Graphs*. <https://www.themathcitadel.com/the-cartesian-product-of-two-graphs>, 2019.
- [19] A. Semri S. Gravier, J. Moncel. Identifying codes of cycles. *European Journal of Combinatorics* 27, page 767–776, 2006.
- [20] J. Moncel S. Gravier. Construction of codes identifying sets of vertices. *the electronic journal of combinatorics* 12, 2005.
- [21] S.N.Ndiaye. *Quelques rappels sur la théorie des graphes*. <https://perso.liris.cnrs.fr/sambandojh.ndiaye/fichiers/App-Graphes.pdf>, 2011-2012.
- [22] Marion Vandermeer. Codes identifiants de graphes. Master's thesis, Université de Liège, <http://hdl.handle.net/2268.2/5008>, 2017-2018.
- [23] S.Klavzar W.Imrich and D.F.Rall. *Topics in graph theory :Graphs and their Cartesian product*. CRC Press, 2008.

Résumé :

Nous nous intéressons dans ce travail, au problème des codes identifiants dans les graphes. Ce dernier consiste à trouver un sous-ensemble de sommets C , de cardinalité minimum, où chaque sommet du graphe considéré a un voisinage unique dans C par.

On s'intéresse particulièrement aux codes identifiant dans le produit cartésien de deux chemins ($P_2 \square P_n$ et $P_4 \square P_n$).

Les mots clés : Code identifiant, produit cartésien, chemin, dominant.

Abstract :

In this work, we are interested in the problem of identifying codes in graphs. This problem consists in finding a subset of vertices C , of minimum cardinality, where each vertex of the considered graph has a unique neighbor in C .

We are particularly interested in finding minimum in the cartesian product of two paths ($P_2 \square P_n$ and $P_4 \square P_n$).

Keywords : Identifying code, Cartesian product, path, domination.