

N° d'ordre : 02/2010-M/MT

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE ET DE LE RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE
"HOUARI BOUMEDIEN"
FACULTE DE MATHÉMATIQUES



MÉMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

En : MATHEMATIQUES
Spécialité : RO : Mathématiques Discrètes et Optimisation
Par : TOUATI Hillal

Sujet

**Codes Identifiants dans les Graphes : Cas
de Chemins et de Circuits**

Soutenu publiquement le 30/09/2010, devant le jury composé de :

Mr. KHELLADI Abdelkader	Professeur, à l'USTHB	Président
Mr. SEMRI Ahmed	Maitre de Conférences/A, à l'USTHB	Directeur de Thèse
Mr. AÏDER Méziane	Professeur, à l'USTHB	Examineur
Mme. BENMEZIANE Zineb	Maitre de Conférences/A, à l'USTHB	invité

Remerciements



JE voudrais exprimer mes remerciements et ma gratitude à Mr. Ahmed SEMRI mon directeur de thèse qui m'a encadré durant mon parcours de Magister. Sa disponibilité, malgré ses responsabilités, ses conseils, son orientation ainsi que les discussions fructueuses m'ont permis de mener à bien ce travail.

De même je tiens à remercier Mr. Méziane AÏDER et Mme. Zineb BEN-MEZIANE de m'avoir fait l'honneur d'assister à ma soutenance en tant qu'examineurs

Je remercie aussi Mr. Abdelkader KHELLADI d'avoir bien voulu présider le jury.

Je souhaite également exprimer ma gratitude envers tout mes amis qui m'ont aidé et soutenu, je pense en particulier à Moumene, Nabil, Nassim ; et puis tous les autres...

Pourrai-je oublier tous mes professeurs, que j'ai croisé durant mes 20 ans d'études, auxquels je suis profondément reconnaissant et je que ne remercierai jamais assez.

Un grand merci pour la grande communauté du logiciel libre pour les efforts qu'ils déploient sans contrepartie, en particulier :les linuxiens, la communauté d'inkscape, les latexien, la Kile team et tand d'autre...

Ces dernières lignes vont à ma mère, mon frère et son épouse, mes neveux : Mina , A/Raouf et A/Essalam, pour m'avoir soutenu et appuyé durant ces années... J'ai une pensée pour ceux qui ne sont plus de ce monde et qui me manquent qu'allah le très méricordieux, le très méricordieux les accueille en son vaste paradis.



*A ma mère
A la mémoire de mon père
A mon frère*

Table des matières

Table des matières	2
Introduction	4
1 Généralités sur les Graphes	8
1.1 Introduction	8
1.2 Définitions et concepts de base	9
1.2.1 Graphe non orienté	10
1.2.2 Graphe orienté	10
1.2.3 Entre sommets et arêtes	11
1.2.4 Quelques types de graphes	11
1.3 Opérations sur les graphes	15
1.3.1 Produits de graphes	15
1.3.2 Alphabet, mots et longueur d'un mot	17
1.3.2.1 Lien avec les graphes	18
2 Éléments sur les Codes Identifiants	19
2.1 Préliminaires	20
2.1.1 Code couvrant, code séparateur et code identifiant	20
2.2 Applications des codes identifiants	23
2.3 Généralisations et variantes	25
2.3.1 Identification à distance r	25
2.3.2 Identification d'ensembles de sommets	25
2.3.3 Sommets de code non identifiables	27
2.3.4 Codes identifiants robustes	27
2.3.5 Codes $(r, \leq p)$ -fortement identifiants	28
2.3.6 Codes identifiant et codes discriminants	28
2.4 Codes identifiants et le problème de recouvrement d'ensemble	30

2.5	Existence, complexité et construction de codes identifiants	33
2.5.1	Graphes n'admettant pas de code identifiant	33
2.5.2	Complexité du problème	33
2.5.3	Graphes optimaux	34
2.5.3.1	Graphes admettant un code identifiant de cardinalité égale à $\lceil \log_2(n + 1) \rceil$	35
2.5.3.2	Graphes admettant un code identifiant de cardinalité égale à $n - 1$	39
2.6	Résultats dans certaines classes de graphes	41
2.6.1	Codes identifiants dans les grilles	42
2.6.1.1	Grilles carrées	42
2.6.1.2	Grilles triangulaires	43
2.6.1.3	Grilles hexagonales	44
2.6.1.4	Grilles carrées à deux diagonales	45
2.6.2	Hypercubes	46
2.6.3	Chaînes et cycles	48
3	Codes Identifiants dans les Graphes Orientés	51
3.1	Algorithme linéaire pour les arbres orientés	52
3.1.1	Validité et linéarité de l'algorithme	56
3.2	Modélisation en PLNE du problème de code identifiants	56
3.2.1	Description du problème	56
3.2.2	Identification des variables de décision	57
3.2.3	Fonction objectif	57
3.2.4	Contraintes	58
3.2.5	Exemple	59
4	Codes Identifiants dans les Chemins et les Circuits	61
4.1	Code identifiant dans les chemins	61
4.1.1	Code identifiant	62
4.1.2	Code 2-identifiant	64
4.2	Code identifiant dans les circuits	69
4.2.1	Code 2-identifiant	69
	Bibliographie	76

Introduction

Durant le siècle dernier et jusqu'à aujourd'hui, les mathématiques discrètes sont devenues une branche importante des mathématiques. Depuis l'avènement des calculateurs dans notre vie quotidienne, son développement ne cesse de croître vu l'intérêt qu'elle suscite chez les scientifiques. Traitant des objets dits discrets (nombres entiers) il s'est avéré que ces concepts ont une adaptation très utile à la description des objets et concepts communément utilisés en informatique. Inversement, l'implémentation de modèles de mathématiques discrètes, conçus pour résoudre des problèmes pratiques sur machine, rend la tâche de résolution moins difficile.

Cette branche des mathématiques intervient dans différents domaines, comme la théorie des ensembles, la géométrie, la théorie des nombres, la théorie des graphes, la théorie de l'information,...

Le domaine des codes identifiants, initié en 1998 par Karpovsky et al [44] fait partie des problèmes pratiques dont la modélisation et la résolution fait appel aux concepts de cette branche. Dans leurs travaux, les auteurs ont investi le problème de couverture de sommets d'un graphe de sorte que chaque sommet est identifié de manière unique moyennant ses voisins. Autrement dit, on fait correspondre à chaque sommet du graphe un ensemble unique de ces sommets voisins, appelé ensemble identifiant. Dans la littérature, le problème de recherche de code identifiant¹ de cardinalité minimum est connu pour être un problème appartenant à la classe des problèmes NP-difficiles [21, 23, 37]. Nous verrons dans les prochains chapitres que même dans des graphes simple tels les cycles et les circuits la recherche d'un code identifiant n'est pas évidente [50, 60]. Cependant, dans le cas d'un arbre, il existe un algorithme linéaire [25].

1. Dans certaines références, on l'appelle problème de placement de détecteurs

Bon nombre d'applications utilisant ce type de code pour résoudre certains problèmes existent, à commencer par l'exemple de détection de pannes dans un réseaux de multiprocesseurs [44]. Soit un réseau de multiprocesseur modélisé par un graphe non orienté. Parmi ces processeurs, certains peuvent détecter ceux qui sont défectueux dans leurs voisinage. Si un processeurs tombe en panne alors en analysant son ensemble identifiant, on saura localiser celui-ci. Par la suite, une variante encore plus puissante appelée codes identifiants robustes a vu le jour (voir [45]), et qui a permis d'étendre le champs d'application de ces codes. Ainsi, dans certains travaux [34, 48, 55], les codes identifiants semblent être une très bonne approche pour le problème de détection et de localisation dans un environnement fermé muni de capteurs sans fils. Ces deux applications seront expliquées davantage au deuxième chapitre.

Aujourd'hui les codes identifiants ne cessent de développer un intérêt toujours grandissant, de sorte qu'une littérature abondante est entrain de naître². Depuis 1998, les codes identifiants, devenus un domaine à part entière, ont été étudiés sur plusieurs structures de graphes tel que, les grilles [16, 17, 18, 22, 29, 34, 39, 40, 42, 43, 44, 45], les chaines et les cycles [6, 47, 50, 60], les graphes aléatoires [1] et graphes quelconque [68, 69], quelques produits de graphes [51, 61] et les hypercubes [13, 15, 36, 37, 41, 58, 66].

En s'inspirant de ce qui a été fait pour les cycles et les chaînes, nous avons essayé d'aborder, dans notre travail, le problème de recherche de code identifiant dans un chemin et un circuit. Nous avons pu déterminer la cardinalité minimale des codes 1 et 2-identifiants dans le cas d'un chemin orienté. En outre, nous avons pu exhiber un code 2-identifiant pour un circuit et nous avons pu montrer qu'il est optimal.

Ce manuscrit est structuré comme suit :

Le premier chapitre est consacré à la présentation des généralités sur les graphes, où nous donnons quelques définitions, terminologies ainsi que les notations usuelles relatives à la théorie des graphes.

Dans le chapitre 2, un aperçu est donné sur le domaine des codes identifiants, où les différentes terminologies et définitions sont introduites.

Des liens existants entre les codes identifiants avec d'autres codes, à savoir : les codes localisateur-dominateurs et les codes discriminants sont présentés. Des variantes de codes identifiants sont données et le problème de recherche de code identifiant de cardinalité minimum a été défini.

En outre, la question d'existence de code identifiant ainsi que la construction de graphes

2. jusqu'au mois de février 2010, on dénombre près de 171 références traitant les codes identifiants [2].

optimaux est abordée. Enfin, nous exposons les plus importants résultats obtenus dans certaines classes de graphes.

Le troisième chapitre concerne l'étude des codes identifiants dans les graphes orientés. Bien que très peu de travaux ont été fait sur ce type de graphes, nous présentons les quelques résultats obtenus, à savoir : l'algorithme linéaire sur les arbre orienté et la modélisation du problème de recherche de code identifiant de cardinalité minimum en un programme linéaire en 0-1. Dans le dernier chapitre, on présentera les premiers résultats que nous avons établi dans le cas des chemins et les circuits. Enfin nous terminons le mémoire par une conclusion générale et des perspectives.

Chapitre 1

Généralités sur les Graphes

Fuyez celui qui vous dit que vous pouvez
réussir autrement que par le travail,
l'honnêteté et l'instruction

VICTOR HUGO

1.1 Introduction

Née des recherches d'Euler (1707-1782) au 18^{ème} siècle, la théorie des graphes est devenue une branche des mathématiques au début du 20^{ème} siècle, grâce aux travaux de König, de Kuratowski, de Cayley et plus récemment, Berge [4], Erdős et Harary[10]. Les recherches récentes en informatique et surtout en algorithmique lui donnent un nouveau souffle. La théorie des graphes permet de résoudre efficacement une grande variété de problèmes pratiques en les ramenant à des configurations qui se dessinent simplement à l'aide de points et de liaisons entre ces points.

On fait généralement remonter la théorie des graphes au problème dit "Ponts de Königsberg", Kaliningrad aujourd'hui (voir la figure 2.20). Résolu par Euler, en 1736, ce problème s'énonce ainsi :

Est-il possible, en partant d'un endroit quelconque de la ville et y revenir, en tentant de traverser chacun des sept ponts une et une seule fois ?

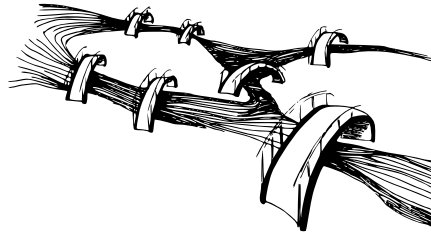


Figure 1.1 – Représentation des ponts Königsberg

Pour sa résolution le problème a été modélisé sous forme d'un graphe (voir la figure 4.3). Ce graphe possède, un ensemble de points représentant les quatre zones de la ville et des segments les reliant représentant les sept ponts.

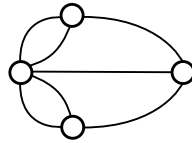


Figure 1.2 – Présentation des ponts de la ville de Königsberg sous forme d'un graphe.

Aujourd'hui, la théorie des graphes connaît un grand engouement, du fait de son utilisation dans la résolution d'une grande variété de problèmes pratiques tels que : les transports, les réseaux, l'ingénierie, l'informatique théorique...etc.

En outre, elle est intimement liée à d'autres branches des mathématiques, à savoir : la théorie des ensembles, le combinatoire, la théorie des matrices, la topologie... En fait, la théorie des graphes est utilisée comme modèle mathématique pour tout système pouvant inclure une relation binaire.

Ce chapitre est consacré à l'introduction aux différentes définitions et à la terminologie utilisées en général en théorie des graphes. Malheureusement, on ne peut présenter toutes les définitions vu leur abondance. On se limitera le plus souvent à celles que nous utiliserons tout au long de ce mémoire. Les lecteurs intéressés sont invités à consulter les références : [3, 4, 7, 10, 11, 52, 53].

1.2 Définitions et concepts de base

D'une manière intuitive un graphe est un schéma constitué d'un ensemble fini de points qu'on relie entre eux par des lignes. Les points sont appelés **sommets** et les lignes sont appelées **arêtes**. Une définition plus formelle est la suivante :

Définition 1 ([4]) *Un graphe $G=(V,E)$ est le couple constitué par :*

1. *un ensemble de sommets $V = \{v_1, v_2, \dots, v_n\}$,*
2. *une famille $E = \{e_1, e_2, \dots, e_m\}$ d'éléments du produit cartésien $V \times V = \{(v_1, v_2) | v_1, v_2 \in V\}$.*

1.2.1 Graphe non orienté

Lors de l'étude de certaines propriétés dans un graphe, il arrive que l'orientation ne joue aucun rôle. On s'intéresse simplement à l'existence d'une ligne entre deux sommets. On appelle alors cette ligne une arête. Elle est représentée souvent par une paire non ordonnée de sommets.

On notera un graphe non orienté $G = (V, E)$ où E désigne l'ensemble des arêtes du graphe. Un *multi-graphe* est un graphe pour lequel il peut exister plusieurs arêtes entre deux sommets.

L'*ordre* d'un graphe représente le nombre de sommets qu'il possède.

Pour une arête $e = (v_1, v_2)$ d'un graphe les sommets v_1 et v_2 sont appelés *extrémités* de l'arête.

On appelle *boucle* une arête dont les deux extrémités coïncident.

Un *graphe simple* est un graphe sans boucle et sans arêtes multiples.

1.2.2 Graphe orienté

Dans bien des situations, on peut avoir une relation entre le sommet v_i et le sommet v_j , sans qu'on ait, nécessairement, une relation entre v_j et v_i . De ce fait, on met une orientation pour désigner le sens de la relation. On représente cette orientation par un *arc*. Pour un arc (v_i, v_j) , v_i est l'extrémité initiale et v_j l'extrémité terminale. On appelle aussi v_i prédécesseur de v_j et v_j successeur de v_i .

On notera $\Gamma^+(v_i)$ l'ensemble des successeurs du sommet v_i et $\Gamma^-(v_i)$ l'ensemble de ses prédécesseurs.

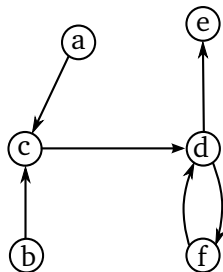


Figure 1.3 – Exemple de graphe orienté d'ordre 6

Remarque 1 *Un graphe infini est un graphe possédant un ensemble infini de sommets.*

1.2.3 Entre sommets et arêtes

Dans cette partie, on introduit quelques définitions relatives aux sommets et aux arêtes. Ces définitions sont fondamentales en théorie des graphes.

Adjacence Deux sommets v_1, v_2 sont dits *adjacents* (ou voisins) s'il existe une arête reliant ces deux sommets.

Arête incidente On dit qu'une arête est *incidente* à un sommet si ce dernier est une extrémité de cette arête.

Degré Le *degré* d'un sommet v , noté $d_G(v)$, est le nombre d'arêtes incidentes à ce sommet. Dans un graphe simple, le degré d'un sommet représente le nombre de sommets adjacents à celui-ci.

Dans le cas orienté, on parle de demi-degré intérieur (resp. extérieur), noté $d_G^-(v_i)$ (resp. $d_G^+(v_i)$), représentant le nombre de prédécesseurs (resp. successeurs) de v_i .

Voisinage On appelle *voisinage* d'un sommet v , l'ensemble de tous ses voisins noté $N(v)$. On note $N[v]$ le *voisinage étendu* du sommet v avec $N[v] = N(v) \cup \{v\}$.

1.2.4 Quelques types de graphes

En théorie des graphes on dénombre plusieurs types ou familles de graphes. nous présentons dans cette partie quelques définitions de graphes particuliers les plus étudiés.

Chaîne On appelle chaîne de longueur n toute suite de sommets (v_1, v_2, \dots, v_n) de G tels que $\forall 1, \dots, k-1 : (v_i, v_{i+1}) \in E$.

Elle est dite *élémentaire* si elle ne rencontre pas deux fois le même sommet, et est dite *simple* si elle ne rencontre pas deux fois la même arête.

Dans le cas d'un graphe orienté, on parle alors de chemin.

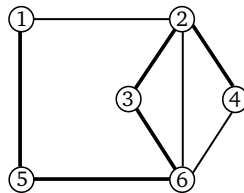


Figure 1.4 – L'ensemble des arêtes en gras représente un exemple de chaîne

Cycle Un cycle est une chaîne telle que v_1 et v_n sont confondus.

Un cycle est dit *eulerien* (respectivement *hamiltonien*) s'il passe par toutes les arêtes (respectivement tous les sommets) une et une seule fois.

Graphe connexe, graphe complet, graphe régulier Un graphe $G = (V, E)$ *connexe* (respectivement *fortement connexe*), c'est un graphe pour lequel il existe une chaîne (respectivement un chemin) de x vers y ou de y vers x entre toute paire (x, y) de sommets. On appelle *graphe complet* tout graphe où toutes les paires de sommets sont adjacentes. En d'autre terme, chaque sommet du graphe est adjacent à tous les autres sommets. On note souvent ce type de graphes K_n .

Un *graphe régulier* est un graphe dont les degrés des sommets sont tous égaux. Ainsi, on appelle *graphe k -régulier* un graphe dont tous les sommets du graphe ont le même degré et est égale à k . Une clique est un sous-graphe complet.

Arbre et forêt On appelle arbre tout graphe connexe acyclique, c'est-à-dire ne contenant pas de cycle. Une *forêt* est graphe sans cycles.

Les sommets incidents à une seule arête sont appelés *sommets pendants* ou *feuilles*.

Une arborescence est un arbre ayant un sommet particulier qu'on appelle racine. Dans le cas d'un arbre orienté la racine est un sommet duquel on peut atteindre, par un chemin, tout autre sommet. Pour une arborescence, on définit :

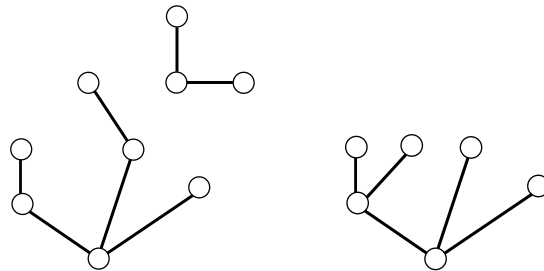


Figure 1.5 – La figure de gauche représente un exemple de forêt et celle de droite un arbre

- Hauteur : La hauteur d'un sommet représente la distance entre celui-ci et la racine. Ainsi, la hauteur d'un arbre est la longueur d'une plus longue chaîne reliant une feuille à la racine.
- Père : Le père d'un sommet v est son prédécesseur.
- Fils : On appelle fils d'un sommet le successeur de celui-ci.

Grilles La grille n -dimensionnelle (infinie) est définie comme étant le graphe ayant pour ensemble de sommets \mathbb{Z}^n et pour ensemble d'arêtes $\{uv | d_1(u, v) = 1\}$, où $d_1(u, v) = \sum_{i=1}^n |u_i - v_i|$ ($d_1(u, v)$ est la longueur d'un plus court chemin entre u et v). La distance d_1 est parfois appelée distance de Manhattan, ou encore distance de Lee (voir plus loin). Nous désignerons simplement par grille la grille bidimensionnelle.

Il existe différents types de grilles, à savoir : les grilles carrées, hexagonales (ou briques), grilles carrées avec une diagonale (triangulaire) et avec deux diagonales (royale).

Hypercube et graphe de Hamming Utilisé dans ce qu'on appelle les architectures de machines parallèles, un hypercube de dimension n , noté Q_n , est le graphe dont l'ensemble des sommets est $\{0, 1\}^n$ et où deux sommets sont adjacents s'ils diffèrent d'une coordonnée.

Une autre définition récursive est la suivante :

Soit $Q_1 = K_2$ une chaîne de longueur 1. Un hypercube de dimension $n \geq 2$ est entièrement défini par la relation de récurrence $Q_{n+1} = Q_n \square K_2$.¹

Un hypercube possède 2^n sommets qui peuvent être libellés par $a_1 a_2 \cdots a_n$ tel que $a_i \in \{0, 1\}$, $\forall 1 \leq i \leq n$.

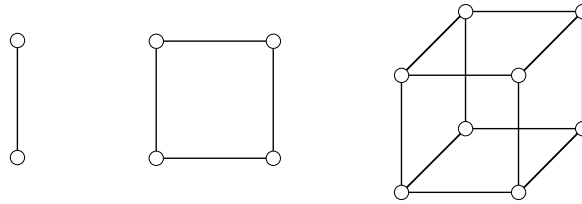


Figure 1.6 – Les hypercubes Q_1 , Q_2 et Q_3

Les graphes de Hamming sont considérés comme une généralisation des hypercubes. Un graphe de Hamming est le produit cartésien de graphes complets K_q . Cette famille de graphes est noté par $H(d, q)$.

Dans un graphe $H(d, q)$ on dispose de q^d sommets où chaque sommet x est noté par x_1, x_2, \dots, x_d avec $x_i \in \{0, 1, 2, \dots, q\}$.

$$H(d, q) = H(d - 1, q) \square K_q = \underbrace{K_q \square K_q \square \cdots \square K_q}_{d \text{ fois}}$$

Lorsque $q = 2$, on tombe sur l'hypercube de dimension d (Q_d).

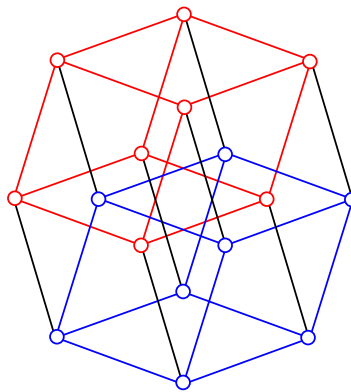


Figure 1.7 – Exemple de graphe de Hamming $H(4, 2)$

1. L'opérateur \square représente le produit cartésien de deux graphes, défini plus loin

Graphe valué ou pondéré On dit qu'un graphe $G = (V, E)$ est valué lorsque pour toute arête, $e_i \in E$, on associe une valeur (un poids) $w(e_i) \in \mathbb{R}$. Ce dernier peut représenter un temps, un coût, une distance, une capacité...

Définition 2 (Distance) *La distance entre deux sommets v_1, v_2 , notée $d(v_1, v_2)$, est la longueur d'une plus courte chaîne reliant v_1 et v_2 .*

Le *diamètre* d'un graphe est la plus grande distance entre deux sommets du graphe.

Définition 3 (boule) *Soit v un sommet quelconque de $G = (V, E)$. On appelle boule de centre v et de rayon r , notée $B_r(v)$, l'ensemble*

$$\{x \in V \mid d(x, v) \leq r\}$$

Définition 4 (Transversal) *On appelle transversal dans un graphe $G = (V, E)$, tout sous ensemble de sommets $T \subset V$ avec la propriété que chaque arête de G est incidente à au moins un sommet de T .*

Définition 5 (Couplage) *Un couplage est une collection (ensemble) d'arêtes indépendantes. En d'autre terme, il n'existe pas d'arêtes qui possèdent un sommet en commun.*

Une définition similaire relative aux sommets est la notion de stable.

Définition 6 (Stable) *un stable est un sous ensemble de sommets $S \subseteq V$ deux à deux non adjacents.*

Définition 7 (Recouvrement) *On appelle recouvrement un sous-ensemble $X \subseteq E$ d'arêtes telle que tous les sommets sont incidents à au moins une arête de X .*

Un couplage qui est un recouvrement est un couplage parfait.

Définition 8 (Graphe partiel) *On appelle graphe partiel de $G = (V, E)$ le graphe $G' = (V, E')$ dont les sommets sont ceux du graphe G et les arêtes est l'ensemble $E' \subset E$.*

Autrement dit, un graphe partiel de G est le même graphe en lui ôtant des arêtes.

Définition 9 (Sous-graphe) *Un sous graphe de $G = (V, E)$ est un graphe $G' = (V', E(V'))$ tel que $V' \subset V$ et $E(V')$ est le sous ensemble d'arêtes induit par*

l'ensemble des sommets V' .

Définition 10 (Graphe biparti) *Un graphe $G = (V, E)$ est dit biparti, si son ensemble de sommets V peut être partitionné en deux sous ensembles V_1 et V_2 de sorte que chaque arête de G relie un sommet de V_1 avec un autre de V_2 , et chaque sous-ensemble forme un stable.*

En outre, on définit un *graphe biparti complet* lorsque tout sommet de V_1 est adjacent à tous les sommets de V_2 et réciproquement. Si $|V_1| = m$ et $|V_2| = n$, on note alors ce graphe $K_{m,n}$. Ainsi, une étoile est un graphe biparti complet $K_{1,n}$.

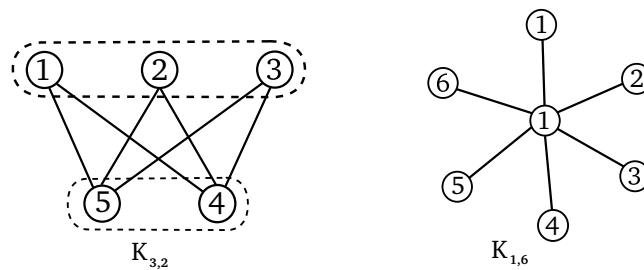


Figure 1.8 – Exemples de graphes bipartis

1.3 Opérations sur les graphes

Dans cette section, on présente quelques opérations qui peuvent être effectuées sur les graphes. Parmi celles-ci, on cite l'union de deux graphes, leur intersection et quelques produits de graphes. Dans notre cas on s'intéresse aux produits de graphes, pour plus de détails le lecteur pourra se référer à [10].

1.3.1 Produits de graphes

De nombreux produits de graphes ont été définis. On dénombre près de 256 types de produits dans la littérature. Dans notre cas on se limite aux produits les plus utilisés.

On définit le produit de deux graphes G et H comme étant le graphe, noté $G \otimes H$, ayant pour ensemble de sommets le produit cartésien des ensembles de sommets de G et H . Cependant, l'ensemble des arêtes est entièrement défini par les relations d'égalité ou d'adjacence entre les sommets de G et de H .

Produit cartésien Le *produit cartésien* (appelé aussi produit carré), 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 :

$$\left\{ \begin{array}{l} u_1 = u_2 \text{ et } v_1 v_2 \in E_2 \\ \text{ou} \\ v_1 = v_2 \text{ et } u_1 u_2 \in E_1 \end{array} \right.$$

Remarque 2 Le degré d'un sommet du produit cartésien est égal à la somme des degrés de ses coordonnées.

$$d_{G_1 \square G_2}(u, v) = d_{G_1}(u) + d_{G_2}(v)$$

Produit croisé ou directe Le *produit directe* $G_1 \times G_2$ 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 :

$$\left\{ \begin{array}{l} u_1 u_2 \in E_1 \\ \text{et} \\ v_1 v_2 \in E_2 \end{array} \right.$$

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

Remarque 3

$$d_{G_1 \times G_2}(u, v) = d_{G_1}(u) \cdot d_{G_2}(v)$$

$$E(G_1 \times G_2) \cap E(G_1 \square G_2) = \emptyset$$

Produit total Appelé aussi *produit fort*, le *produit total* de G_1 et G_2 , noté $G_1 \boxtimes G_2$, est le graphe tel que deux sommets (u_1, v_1) et (u_2, v_2) de $V_1 \times V_2$ sont adjacents si et seulement si :

$$\left\{ \begin{array}{l} u_1 = u_2 \text{ ou } u_1 u_2 \in E_1 \\ \text{et} \\ v_1 = v_2 \text{ ou } v_1 v_2 \in E_2 \end{array} \right.$$

Dans le produit total, le degré d'un sommet vérifie l'égalité :

$$d_{G_1 \boxtimes G_2}(u, v) = d_{G_1}(u) + d_{G_2}(v) + d_{G_1}(u) d_{G_2}(v)$$

Produit lexicographique On appelle le produit lexicographique de G_1 et G_2 le graphe $G_1 \circ G_2$, tel que deux sommets (u_1, v_1) et (u_2, v_2) de $V_1 \times V_2$ sont adjacents si et seulement si :

$$\left\{ \begin{array}{l} u_1 u_2 \in E_1 \\ \text{ou} \\ u_1 = u_2 \text{ et } v_1 v_2 \in E_2 \end{array} \right.$$

Remarque 4 Des définitions précédentes, on peut déduire que le produit $G_1 \boxtimes G_2$ est un sous graphe de $G_1 \circ G_2$. De même, $G_1 \square G_2$ et $G_1 \times G_2$ sont des sous graphes de $G_1 \boxtimes G_2$

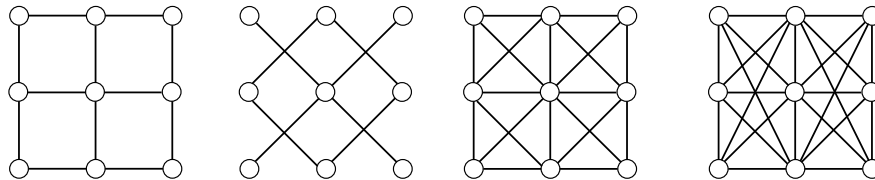


Figure 1.9 – Illustrations de quelques produits de P_2 et P_2

1.3.2 Alphabet, mots et longueur d'un mot

Un alphabet Σ est un ensemble de symboles. Un mot sur cet ensemble est un élément de Σ^k , où k désigne la longueur du mot. L'alphabet le plus courant est l'alphabet binaire. Étant donné deux mots $x = (x_1, x_2, \dots, x_k)$ et $y = (y_1, y_2, \dots, y_k)$ de $\{0, 1\}^k$, on désigne par $x + y$ le mot $x = ((x_1 + y_1), (x_2 + y_2), \dots, (x_k + y_k))$ où $+$ est la somme modulo 2.

Sur un ensemble de mots, on définit une métrique, autrement dit, une règle de calcul des distances. Il existe principalement trois métriques :

Métrique de Hamming La métrique de Hamming est le nombre de caractères par lesquels deux mots de l'alphabet diffèrent.

Si l'alphabet est $\Sigma = \{0, 1, 2\}$, alors les mots $(0,2,1,1)$ et $(0,1,1,2)$ sont à distance 2.

Métrique de Manhattan Si l'alphabet comporte k éléments, alors les éléments sont notés de 0 à $k - 1$. La distance entre deux mots $x = (x_1, x_2, \dots, x_n)$ et $y = (y_1, y_2, \dots, y_n)$ est définie par :

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Par exemple, dans l'alphabet $\sigma = \{0, 1, 2, 3\}$, les mots $(0,2,1,0,3)$ et $(1,3,1,2,2)$ sont à distance $|0 - 1| + |2 - 3| + |1 - 1| + |0 - 2| + |3 - 2| = 5$.

Métrie de Lee Soit $x = (x_1, x_2, \dots, x_n)$ et $y = (y_1, y_2, \dots, y_n)$ deux mots de cet alphabet. La métrie de Lee, noté $d(x, y)$, est définie par :

$$d(x, y) = \sum_{i=1}^n \min(|x_i - y_i|, k - |x_i - y_i|) \pmod n$$

Étant donné l'alphabet $\Sigma = \{0, 1, 2, 3\}$, les mots $(0,3,2,3,1)$ et $(3,3,2,1,1)$ sont à distance 3, puisque $\min(|0 - 3|, 4 - |0 - 3|) + \min(|3 - 1|, 4 - |3 - 1|) = 3 \pmod 4$.

Remarque 5 Dans un alphabet binaire, les trois métriques correspondent. De plus, sur un alphabet non fini, les métriques de Manhattan et de Lee correspondent.

1.3.2.1 Lien avec les graphes

Pour un alphabet Σ , une longueur de mots k et une métrie correspondant à une distance d , on définit le graphe G ayant pour ensemble de sommets $V = \Sigma^k$ et pour arêtes $E = \{(u, v) \in V \mid d(u, v) = 1\}$, où d est une métrie sur Σ .

Si on considère l'alphabet binaire $\{0, 1\}$ une longueur de mots k , dans chacune des trois métriques présentées, le graphe résultant est l'hypercube Q_k .

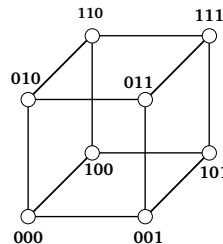


Figure 1.10 – Dans cette figure, on a $\Sigma = \{0, 1\}$, les mots sont de longueur $k = 3$. L'ensemble des sommets est $\{0, 1\}^3$ et deux sommets sont adjacents s'ils diffèrent d'une coordonnée. Ainsi, on a bien un Q_3

Dans ce chapitre on a pu faire un survol sur des notions et des définitions relatives à la théorie des graphes. Nous avons décrit les différentes caractéristiques ainsi que quelques propriétés.

Éléments sur les Codes Identifiants

Il faut tout rendre le plus simple possible, mais pas plus.

Ne vous inquiétez pas de vos difficultés en mathématiques, je vous assure que les miennes sont bien plus grandes.

ALBERT EINSTEIN

Introduit pour la première fois dans les travaux de Karpovsky, Chakrabarty et Levitin dans [44], les codes identifiants ont été utilisés dans l'étude des problèmes de localisation et de détection de pannes dans les réseaux de multi-processeurs.

On lie les codes identifiants à la famille des problèmes de couverture par tests, définit de manière générique comme suit [32] :

Étant donnée une matrice A , quels sont les sous-ensembles de lignes de A de sorte que les colonnes résultantes sont toutes différentes.

La figure 2.1 présente un exemple de ce problème dans le cas d'une matrice à coefficients entiers.

Le problème de couverture par test a été, à l'origine, formulé pour modéliser un problème d'aide au diagnostic médical dans lequel les lignes de la matrice correspondent à des symptômes et les colonnes à des maladies. Les coefficients de cette matrice correspondent à l'intensité de chaque symptôme dans les maladies. Le problème consistait alors à déterminer un sous ensemble de symptômes qui identifiait les maladies de façon unique [32].

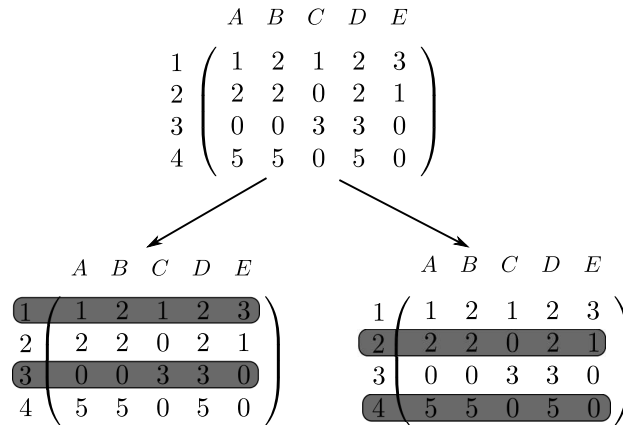


Figure 2.1 – Problème de couverture par tests : il faut déterminer un ensemble de lignes de la matrice permettant de différencier les colonnes entre elles. Dans cet exemple les lignes 2 et 4 ne permettent pas de différencier les colonnes de la matrice. Les lignes 1 et 3, quant a elles, permettent d’identifier les colonnes A, B, C, D, E : on voit bien que les traces des colonnes A, B, C, D, E sur les lignes 1 et 3 sont toutes différentes.

2.1 Préliminaires

Dans cette section, on présentera la terminologie et les notations ayant trait aux codes identifiants dans le but de normaliser la notation utilisée dans ce mémoire. Dans toute la suite, on notera $G = (V, E)$, où V désigne l’ensemble des sommets du graphe et E l’ensemble de ses arêtes. Généralement, ce graphe est connexe et simple.

Dans certains cas, nous utiliserons la première lettre de l’appellation de graphe, par exemple on utilisera \mathcal{G} pour indiquer un graphe sous forme de grille.

2.1.1 Code couvrant, code séparateur et code identifiant

Dans un graphe $G = (V, E)$ donné, un code C est un sous-ensemble de sommets ($C \subseteq V$). Un élément de ce code est appelé *mot de code*.

Définition 11 (Code couvrant) *On dit qu’un sous-ensemble $C \subseteq V$ de sommets est un code couvrant du graphe $G = (V, E)$ si et seulement si : $\forall v \in V$ on a*

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

En d’autre terme, si tout sommet de $V \setminus C$ est voisin à, au moins, un sommet de C . On dira qu’un sommet $c \in C$ *couvre* le sommet v si : $c \in N[v]$.

Les codes couvrants sont très étudiés dans la littérature. Dans [20] on définit un code couvrant, sur l'espace euclidien, comme l'ensemble minimum de cercles identiques pour couvrir le plus large volume d'une surface donnée.

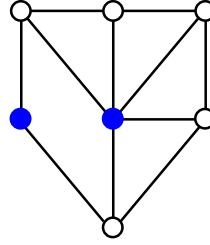


Figure 2.2 – Le sous-ensemble de sommets en noir est un code couvrant minimum du graphe : tout sommet du graphe est voisin à au moins un sommet noir

Définition 12 (Code séparateur) *Un sous-ensemble $C' \subseteq V$ est un code séparateur si et seulement si pour toute paire de sommets distincts $(u, v) \in V^2$, on a*

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

ou encore :

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

Où $A \Delta B$ représente la différence symétrique de A et B qui vaut : $(A \setminus B) \cup (B \setminus A)$

On dira qu'un sommet $c \in C'$ sépare les sommets u et v s'il appartient à la différence symétrique de $(N[u] \cap C') \Delta (N[v] \cap C')$.

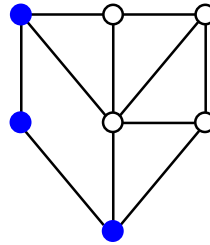


Figure 2.3 – Le sous-ensemble de sommets S en noirs représente un code séparateur, en examinant tous les ensembles $N[v] \cap S$, on trouvera que ce code sépare bien ces ensembles, en revanche on remarque qu'il existe un sommet v' pour lequel $N[v'] \cap S = \emptyset$

Remarque 6 *Dans la littérature anglo-saxon, un code identifiant peu désigner, parfois, un code séparateur.*

Un code dans $G = (V, E)$ qui est à la fois un code couvrant et séparateur est appelé code identifiant dans G . De manière plus formelle :

Définition 13 (Code identifiant) *Étant donné un graphe $G = (V, E)$. On dit qu'un sous-ensemble de sommets $C \subseteq V$ est un code identifiant si et seulement si :*

1. *pour tout sommet $v \in V$, on a : $N[v] \cap C \neq \emptyset$ et*
2. *pour toute paire de sommets $u, v \in V$, on a : $N[u] \cap C \neq N[v] \cap C$*

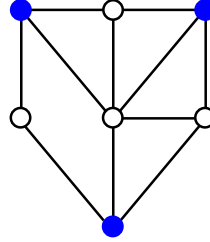


Figure 2.4 – Le sous-ensemble de sommets C en noirs forme un code identifiant, en examinant tous les ensembles $N[v] \cap C$, on trouvera que ce code couvre et sépare tous les sommets

L'ensemble $N[v] \cap C$, qu'on notera $I(v, C)$ (parfois $I(v)$ lorsqu'il n'y a pas d'ambiguïté) est appelé l'*ensemble identifiant* du sommet v . Une autre définition de code identifiant est la suivante [32] :

Le sous ensemble de sommets C est un code identifiant si et seulement si l'application f définie par :

$$\begin{aligned} f : V &\rightarrow \wp(V) \\ v &\mapsto I(v, C) \end{aligned}$$

est injective et dont l'image ne contient aucun ensemble vide.

Lorsque le graphe considéré est infini, on parle de la densité minimum au lieu de la cardinalité minimum d'un code identifiant.

Soit G est un graphe infini, on note $d(C, G)$ la densité d'un code identifiant et $d^*(G)$ la densité minimum d'un code identifiant de G .

On définit la *densité* d'un code identifiant comme suit :

Définition 14 (Densité[43]) *La densité $d(C, G)$ d'un code identifiant C de G est définie comme étant la limite :*

$$d_r(C, G) = \limsup_{n \rightarrow \infty} \frac{|C \cap B_n(v)|}{|B_n(v)|}$$

Lorsque $G = (V, E)$ est fini, on a $d^*(G) = \frac{|C^*|}{|V|}$, où C^* est un code identifiant minimum et $|V|$ est l'ordre de G .

Définition 15 (Code identifiant irréductible[55]) *On dit qu'un code identifiant est irréductible si la suppression d'un élément de celui-ci donne un code qui n'est pas identifiant.*

En d'autre terme, un code identifiant est irréductible s'il est minimal par inclusion.

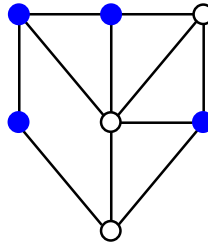


Figure 2.5 – Exemple de code identifiant irréductible, la suppression d'un seul sommet du code entraîne certainement la perte de la propriété d'identification (le code est représenté par les sommets colorés).

2.2 Applications des codes identifiants

Les premiers travaux sur les codes identifiants avaient pour motivation la résolution du problème d'optimisation lié à l'identification de processeurs défectueux dans un réseau de multi-processeurs[44].

Un tel système peut être modélisé sous forme d'un graphe dont les sommets représentent les processeurs et les arêtes modélisent les liens existants entre ces processeurs.

Le travail consiste à placer sur des processeurs, appelés processeurs détecteurs, des routines (algorithmes) permettant le diagnostic des processeurs voisins. Un processeur détecteur renvoie un signal binaire (0 ou 1), égal à 1 lorsqu'un processeur défectueux est voisin à celui-ci et 0 si aucun n'est défectueux.

L'objectif est d'intégrer ces routines de test sur un nombre minimum de processeurs tout en assurant un diagnostic permettant la détection et la localisation (ou tout simplement l'identification) des processeurs défectueux. On appellera l'ensemble des processeurs identifiants un code et un processeurs appartenant à cet ensemble un mot du code.

En d'autre terme, on veut minimiser le nombre de processeurs identifiants.

L'intérêt de ce modèle est de permettre l'identification, moyennant des signaux émis, des

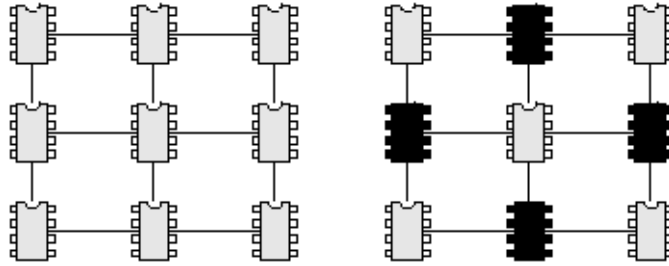


Figure 2.6 – Dans cette figure, on présente un réseau multi-processeurs contenant 9 processeurs (figure à gauche), l'ensemble des processeurs colorés (figure de droite) permettent de couvrir et de séparer l'ensemble des processeurs, ce qui veut dire qu'ils identifient de manière unique tous les processeurs du système.

processeurs défectueux. Ce qui mène à l'utilisation de la notion de code identifiant.

Une autre application des codes identifiants a été réalisée pour le problème de localisation par un réseau de capteurs à une échelle locale (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, des stations nucléaires, des réseaux de distribution...etc [34, 48, 55, 65].

Considérant, par exemple, le plan d'étage d'un immeuble (voir figure(a) ci-dessous), cet étage est divisé en régions (5 pour notre exemple) susceptibles de contenir un capteur. Ce plan d'étage sera modélisé sous forme de graphe (figure(b)) où les régions sont représentées par des sommets et deux sommets seront reliés si les deux régions sont RF-connexes [55]¹. L'objectif est de construire un réseau de capteurs optimal (le nombre de capteurs à placer est minimum) pouvant identifier d'une manière unique la région de présence d'un éventuel danger. Ceci revient à exhiber un sous-ensemble de sommets correspondant à un code identifiant de cardinalité minimum du graphe. L'emplacement des détecteurs assurera que chaque endroit sera couvert par un unique ensemble de détecteurs. Une description plus détaillée a été donnée dans [55].

Par ailleurs, l'application des codes identifiants ne cesse de croître, comme par exemple dans [34] où on modélise le problème de couverture de grille dans un réseau de capteurs distribué sous forme de programme linéaire en nombres entiers. En outre, dans [48] on utilise ces codes pour l'étude le problème de surveillance dans un réseau sans fil.

1. On parle de RF-connectivité entre deux régions lorsque les ondes radio d'une région peuvent pénétrer sans obstacle dans l'autre région.

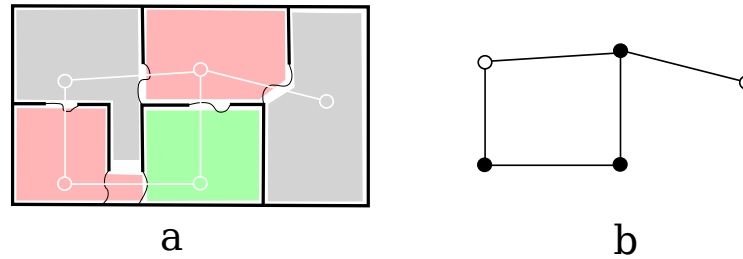


Figure 2.7 – Exemple de plan d'immeuble subdivisé en 5 régions, le graphe correspondant est donné par la figure de droite. Les sommets en noir représente un code identifiant pour le graphe, c'est à dire les endroits ou les capteurs seront placés.

2.3 Généralisations et variantes

Dans cette section, nous donnons quelques généralisations et variantes relatives aux codes identifiants. Parmi celles-ci nous allons voir l'identification à distance r et l'identification d'ensembles de sommets.

2.3.1 Identification à distance r

Lorsqu'on a parlé dans la section précédente de code identifiant, on faisait allusion à un code 1-identifiant c'est à dire en considérant le voisinage d'un sommet ($r = 1$). Cette notion peut être étendue à l'identification de sommets à distance au plus r . Dans la suite, lorsqu'on parlera de code identifiant le lecteur est averti qu'il s'agit du code 1-identifiant.

Définition 16 (code r -identifiant) *On dit qu'un sous-ensemble de sommets $C \subseteq V$ est un code r -identifiant si et seulement si :*

1. *pour tout sommet $v \in V$, on a : $B_r(v) \cap C \neq \emptyset$,*
2. *pour toute paire de sommets $\{u, v\}$, on a : $B_r(u) \cap C \neq B_r(v) \cap C$.*

Dans ce cas on notera l'ensemble identifiant d'un sommet v par $I_r(v, C) = B_r(v) \cap C$.

2.3.2 Identification d'ensembles de sommets

La première définition est adaptée au cas où un seul processeur pourrait tomber en panne. Cependant, en pratique on peut dénombrer plusieurs processeurs défectueux. Pour remédier à une telle éventualité, on définit un code pouvant identifier jusqu'à p processeurs défectueux. On parle alors d'identification d'ensembles de sommets.

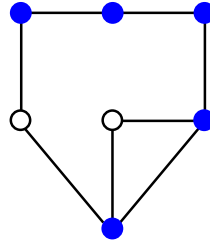


Figure 2.8 – Le sous-ensemble de sommets C colorés représente un code 2-identifiant, car les sous-ensembles $B_r(v)$ de chaque sommet du graphe sont tous non vides et deux à deux distincts

De manière formelle, un code identifiant les ensembles d'au plus p sommets est défini comme suit :

On dit que C est un code identifiant les ensembles d'au plus p sommets si et seulement si les ensembles $I(X, C)$ sont distincts pour tous les sous-ensembles $X \subseteq V$ de cardinalité au plus p , où $I(X, C)$ représente l'union de tous les ensembles identifiant les sommets appartenant au sous-ensemble X , c'est-à-dire

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

Dans ce cas, on parle de code $(1, \leq p)$ -identifiant. Si on ajoute la notion d'identification à distance r alors on parle de code encore plus général qu'est le code $(r, \leq p)$ -identifiant qu'on définit comme suit :

Définition 17 (code $(r, \leq p)$ -identifiant) *On dit qu'un code C est $(r, \leq p)$ -identifiant si et seulement si : les sous-ensembles $I_r(X, C)$ sont tous deux à deux distincts et non vides pour tous les sous-ensembles $X \subseteq V$ avec $|X| \leq p$. Où :*

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

Cependant, lorsque plus de p défaillances surviennent dans le système, ces codes ne sont pas en mesure de nous informer de l'existence de plus des p défaillances déjà existantes. Pour ce faire, Honkala et Laihonen [40] ont introduit les codes $(r, \leq p)^+$ -identifiants. permettant de détecter plus de p pannes dans le réseau.

2.3.3 Sommets de code non identifiables

Une autre variante des codes identifiants est de supposer que les sommets du code n'ont pas à être identifiés. En d'autre terme, on peut considérer que les processeurs identifiants (exécutant les routines de diagnostic) ne tombent jamais en panne. Ce qui veut dire qu'il nous suffit de vérifier que :

$$I(u, C) \neq I(v, C)$$

pour toute paire de sommet distincts $u, v \in V \setminus C$.

Un tel code est appelé *code dominateur-localisateur*.

2.3.4 Codes identifiants robustes

On peut supposer que les processeurs détecteurs peuvent être sujets à des défaillances. Cette situation a été considérée par R.Ungrangsi et al dans [54, 55], où ils ont introduit une nouvelle variante qui sont les codes identifiants r -robustes utilisés dans des environnements hostiles.

Définition 18 ([55]) *Un code identifiant C d'un graphe donné $G = (V, E)$ est dit r -robuste si :*

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

pour tout $u, v \in V$ et $A, B \subseteq V$ avec $|A|, |B| \leq r$.

En d'autre terme, on peut permettre l'ajout ou la suppression d'au plus r mots de code de n'importe quel ensemble identifiant sans pour autant 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 :

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

Définition 19 ([54]) *On dit que le code identifiant C est r -robuste si et seulement si*

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

Par ailleurs, dans [45] les auteurs considèrent deux classes de codes r -robustes, à savoir les codes r -sommets robustes et les codes r -arêtes robustes. La différence est que dans la deuxième variante on ne supprime pas des sommets mais plutôt des arêtes (c'est-à-dire

que les connections entre les sommets qui sont sujet à des défaillances) alors que dans la première classe on supprime des sommets.

Cette deuxième classe de codes r -robustes a été traitée dans [35], où l'auteur donne la densité minimum d'un code 1-arête robuste identifiant² dans une grille triangulaire. Par la suite en 2006, pour certaines valeurs de t et r , T. Laihonen [62] étudie les codes t -arêtes robustes r -identifiants dans les grilles royales, et obtient des densités optimales.

2.3.5 Codes $(r, \leq p)$ –fortement identifiants

Dans les modèles précédents, on s'attend à ce que les informations (rapports) envoyées par les processeurs soient correctes. Il est alors légitime de considérer que les processeurs défectueux peuvent aussi envoyer une information erronée.

Dans [42] une autre variante a été proposée, ce sont les codes $(r, \leq p)$ –fortement identifiants, prenant en considération une telle situation.

Soit $C \subseteq F_2^n$ un code et $r, p \geq 0$ deux entiers, pour tout sous-ensemble $X \subseteq F_2^n$ on définit l'ensemble $\mathcal{I}_r(X, C)$ tel que :

$$\mathcal{I}_r(X, C) = \{U \mid \mathcal{I}_r(X, C) \setminus (X \cap C) \subseteq U \subseteq \mathcal{I}_r(X, C)\}$$

Définition 20 ([42]) *On dit qu'un code C est $(r, \leq p)$ -fortement identifiant si pour toute paire de sous-ensembles $X_1, X_2 \subseteq F_2^n$, distincts et $|X_1|, |X_2| \leq p$ on a :*

$$\mathcal{I}_r(X_1, C) \cap \mathcal{I}_r(X_2, C) = \emptyset$$

Différentes bornes et constructions ont été établies dans les hypercubes.

2.3.6 Codes identifiant et codes discriminants

Les codes discriminants, ou distinguants si on veut une traduction du mot anglais "discriminate", est une nouvelle notion qui a été introduite en 2006 par E. Charbit et al [8]. Nous allons dans un premier temps donner la définition, puis nous énonçons deux théorèmes qui font le lien entre ce type code et le code identifiant.

Soit $G = (V = I \cup A, E)$ un graphe biparti connexe et D un sous-ensemble de A ($D \subseteq A$).

Définition 21 ([8]) *On dit que D est un code discriminant si les ensembles identifiants les sommets de I sont tous distincts et non vides.*

La motivation pour ce type de code émane du fait qu'on peut considérer l'ensemble I (resp. A) comme un ensemble d'individus (resp. d'attributs), avec une arête entre $i \in I$

2. Ici on veut dire 1-identifiant 1-arête robuste

et $a \in A$ si l'individu i possède l'attribut a . Ainsi, un code discriminant est un ensemble d'attributs pouvant distinguer les individus.

Les codes discriminants sont étroitement liés aux codes identifiants. La différence réside dans le fait que les mots du code n'appartiennent qu'au sous-ensemble A et ne doivent identifier de manière unique que les sommets du sous-ensemble I .

Le problème de recherche de code r -discriminant de cardinalité minimum est NP-Difficile [8, 27]. Vu que toute distance entre un sommet de I et un autre de A est impair alors on ne considère que les codes r -discriminants avec r impair [9].

Définition 22 (sommets jumeaux) *On dit que deux sommets, u et v , sont jumeaux si :*

$$B_r(u) = B_r(v)$$

Remarque 7 *Un code r -discriminant n'existe pas toujours. Une condition nécessaire et suffisante pour qu'un graphe $G = (V = I \cup A, E)$ biparti admette un code r -discriminant est qu'il ne possède pas de sommets jumeaux [27].*

Avant de donner le lien, un rappel et quelques notations sont nécessaires. Ce lien n'est établi que dans l'espace de Hamming de dimension n , c'est à dire un n -cube binaire, qui est aussi un graphe biparti régulier. On notera ce graphe $G = (F^n, E)$, où $F = \{0, 1\}$ et $E = \{(x, y) : d(x, y) = 1\}$. Un vecteur est dit pair (respectivement impair) si son poids⁴ est pair (respectivement impair).

Étant donné un vecteur $x \in F^n$, on note $\pi(x)$ la fonction parité, telle que :

$$\pi(x) = \begin{cases} 0 & \text{Si } x \text{ est pair} \\ 1 & \text{Si } x \text{ est impair} \end{cases}$$

Si l'opérateur $|$ représente la concaténation de vecteurs, alors $x|\pi(x)$ est toujours un vecteur de poids pair. On note \mathcal{E}^n (resp. \mathcal{O}^n) l'ensemble des 2^{n-1} vecteurs pairs (resp. impair) dans F^n .

Notons $M_r(n)$ (respectivement $D_r(n)$) la cardinalité minimum d'un code r -identifiant (respectivement discriminant) dans F^n .

Les deux théorèmes suivants montrent que tout code r -identifiant dans F^n peut être étendu à un code r -discriminant dans F^{n+1} , et que tout code r -discriminant dans F^n peut

3. La distance de hamming (ce référer au chapitre précédent).

4. c'est le nombre de coordonnées non nulles dans l'espace de Hamming

être réduit à un code r -identifiant dans F^{n-1} , pour tout $r \geq 1$.

Théorème 1 ([26]) *Soit deux entiers $n \geq 2$, $p \geq 0$ tel que $2p + 1 < n$, et soit $C \subseteq F^n$ un code $(2p + 1)$ -identifiant et*

$$C' = \{c|\pi(c) : c \in C\}$$

Alors C' est un code $(2p + 1)$ -discriminant dans F^{n+1} .

Et de ce fait :

$$D_{2p+1}(n + 1) \leq M_{2p+1}(n)$$

Théorème 2 ([26]) *Soit deux entiers $n \geq 3$, $p \geq 0$ tel que $2p + 2 < n$, et soit $C \subseteq \mathcal{E}^n$ un code $(2p + 1)$ -discriminant et $C' \subseteq F^{n-1}$ un code obtenu par la suppression d'une coordonnée de C . Alors C' est un code $(2p + 1)$ -identifiant dans F^{n-1} .*

De ce fait :

$$M_{2p+1}(n - 1) \leq D_{2p+1}(n)$$

Corollaire 1 ([26]) *Pour tout $n \geq 2$ et $p \geq 0$ tel que $2p + 1 < n$, on a :*

$$D_{2p+1}(n + 1) = M_{2p+1}(n)$$

On notera que le problème de recherche d'un code 1-discriminant est polynômial pour un arbre [27].

2.4 Codes identifiants et le problème de recouvrement d'ensemble

Le problème de recouvrement d'ensemble est un des problèmes d'optimisation des plus anciens et des plus étudiés. Énormément d'applications peuvent être formulées à l'aide d'un problème de recouvrement comme l'assignation de personnel, positionnement d'équipements d'urgence, construction de circuits imprimés.

Il appartient à la classe des problèmes NP-difficile.

Laifenfeld et al ont donné dans [49] une réduction⁵ du problème de recherche de code r -identifiant au problème de recouvrement d'ensemble.

Une redéfinition du code identifiant est nécessaire, nous donnerons dans la suite une autre interprétation du code identifiant et deux lemmes qui nous permettront de faire le rapprochement entre les deux problèmes.

5. Concerant la notion de réduction le lecteur pourra se référer à [64]

Dans la suite on suppose donné un graphe $G = (V, E)$. Le problème de recouvrement d'ensemble (SCP) peut être défini comme suit :

Définition 23 ([49]) *Soit U un ensemble de m éléments et \mathcal{S} une collection de sous-ensembles de U : $\mathcal{S} = \{S_1, \dots, S_n\}$ où $S_i \subseteq U, (1 \leq i \leq n)$ et soit $(c_i)_{1 \leq i \leq n}$ des poids associés aux sous-ensembles de la collection \mathcal{S} .*

Une solution au problème SCP est un sous-ensemble T d'indices de $\{1, 2, \dots, n\}$ tel que tous les éléments de U soient recouverts :

ie, $T \subseteq \{1, \dots, n\} | \bigcup_{i \in T} S_i = U$ et tel que $\sum_{j \in T} c_j$ soit minimale.

Exemple 1 *Soit $U = \{1, 2, 3, 4, 5\}$, $S_1 = \{1, 3, 5\}$, $S_2 = \{1, 2, 4\}$, $S_3 = \{5, 2\}$, $S_4 = \{1, 3, 2\}$.*

On prends $c_j = 1, 1 \leq j \leq 4$.

La solution est $\{1, 2\}$.

Ainsi, le problème de recherche d'un recouvrement d'ensemble minimum est défini comme suit :

NOM : Recouvrement d'ensemble.

INSTANCE : Une collection de sous-ensemble \mathcal{S} de U .

QUESTION : Quel est le sous-ensemble $T \subseteq \{1, \dots, n\}$ tel que :

$$\bigcup_{i \in T} S_i = U \text{ et } \sum_{j \in T} c_j \text{ soit minimum}$$

Une réduction a été établie du problème de code r -identifiant (r -IC) au problème de recouvrement d'ensemble (SCP) et vice versa. Nous n'allons donné qu'une réduction, le lecteur pourra se référer à [49] pour de plus amples informations, puis nous donnons l'algorithme d'approximation basé sur un algorithme glouton [64]. Donnons d'abord quelques notations :

Soit $D(u, v)$ la différence symétrique entre $B_1(u)$ et $B_1(v)$, c'est-à-dire $D(u, v) = B_1(u) \Delta B_1(v)$.

On notera $D_C(u, v)$, l'intersection entre $D(u, v)$ et le code C , i.e : $D_C(u, v) = D(u, v) \cap C$. Ainsi, on peut remarquer que $D_C(u, v)$ n'est autre que la différence symétrique entre les ensembles identifiants des sommets u et v , c'est-à-dire :

$$D_C(u, v) = I(u, C) \Delta I(v, C)$$

Notons L l'ensemble $\{\{x, y\} | x \neq y, x, y \in V\}$.

Définition 24 ([49]) *On appelle ensemble distinguant d'un sommet v , noté δ_v , le sous-ensemble des paires de sommets $\{x, y\}$ de L pour lequel le sommet v appartient à $D(x, y)$:*

$$\delta_v = \{\{x, y\} \in L \mid v \in D(x, y)\}$$

Ainsi, le lemme suivant découle trivialement de la définition d'un code identifiant.

Lemme 1 ([49]) *Le code C est un code identifiant si :*

$$\emptyset \notin \{D_C(x, y) \mid \{x, y\} \in L\}$$

Une autre manière de définir un code identifiant est d'utiliser la notion d'ensemble distinguant défini précédemment.

Lemme 2 ([49]) *On dit qu'un sous-ensemble C est un code identifiant si la famille des ensembles distinguants des sommets le constituant couvre l'ensemble $L = \{\{x, y\} \in V^2 \mid x \neq y\}$*

La construction de code identifiant en utilisant une heuristique résolvant le problème de recouvrement d'ensemble est la suivante :

1. Déterminer les voisins ($B_1(v)$) de chaque sommet de V ,
2. Calculer les ensembles distinguants $M = \{\delta_u \mid u \in V\}$,
3. Calculer le recouvrement minimum \mathbb{C} par un algorithme glouton $Set_cover_greedy(L, M)$,
4. Le code identifiant résultant est $C = \{u \in V \mid \delta_u \in \mathbb{C}\}$, c'est-à-dire les sommets de V pour lesquels leurs ensembles distinguants appartient au recouvrement minimum (\mathbb{C}).

Rappelons que Laifenfeld et Trachtenberg ont pu montrer qu'une réduction polynômiale[64] existe du problème de recouvrement d'ensemble au problème des codes identifiants et vice versa, et ce aussi bien dans le cas non orienté que dans le cas orienté.

D'autres réductions ont été établies et des heuristiques ont été développées pour les codes r -identifiants et les codes r -robustes. Ils montrent aussi que

Théorème 3 ([49]) *Il existe une constante a tel que pour tout graphe G d'ordre n on a :*

$$\frac{C_{greedy}}{C_{min}^*} < a \ln n$$

où C_{greedy} et C_{min}^* représentent respectivement la cardinalité minimum obtenue par l'heuristique et la cardinalité minimum d'un code identifiant dans G . En d'autre terme, le

code obtenu par cette heuristique est proche à un facteur près de l'optimum.

Cependant, pour le cas où $r \geq 1$ la question reste ouverte concernant l'efficacité (que peut-on dire des résultats obtenus par rapport à l'optimum ?) de l'heuristique proposée dans le cas des codes r -robustes identifiants.

2.5 Existence, complexité et construction de codes identifiants

A la différence de certains codes, comme par exemple les codes couvrants ou localisateurs-dominateurs, les graphes n'admettent pas toujours un code identifiant. Nous allons présenter dans cette section les graphes n'admettant pas de code identifiant et nous allons aborder aussi la question de la cardinalité extrême d'un code identifiant en plus de la complexité du problème.

2.5.1 Graphes n'admettant pas de code identifiant

Malheureusement, tous les graphes n'admettent de code identifiant. Étant donné un graphe $G = (V, E)$ et un entier r .

Remarque 8 *On dit que le graphe G admet un code r -identifiant $C \subseteq V$ si et seulement si :*

Pour toute paire de sommets distincts $v_1, v_2 \in V$, on a $B_r(v_1) \neq B_r(v_2)$.

En d'autre terme, le graphe G ne contient pas de sommets jumeaux.

En effet, si pour deux sommets v_1 et v_2 distincts on a $B_r(v_1) = B_r(v_2)$, alors pour tout code $C \subseteq V$ on aura $I_r(v_1, C) = I_r(v_2, C)$ ce qui rend impossible l'existence d'un code identifiant. Un exemple de graphes n'admettant pas de code identifiant sont les graphes complets K_n .

2.5.2 Complexité du problème

Dans cette sous section, nous allons aborder la complexité du problème d'optimisation associé à la recherche de code identifiant de cardinalité minimum. Le problème est défini comme suit :

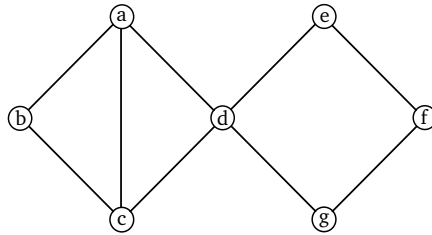


Figure 2.9 – Un exemple de graphe qui n’admet pas de code 1-identifiant, puisque $B_1(a) = B_1(c)$ et par conséquent $I_1(a, C) = I_1(c, C)$ quelque soit l’ensemble C .

NOM : CODE r -IDENTIFIANT :

INSTANCE : Un graphe, $G = (V, E)$, admettant un code r -identifiant

QUESTION : Quelle est le sous-ensemble de sommets de cardinalité minimum représentant un code identifiant du graphe ?

Dans [21, 23] Irène Charon, Olivier Hudry et Antoine Lobstein ont montré que le problème de recherche de code identifiant optimal est NP-difficile. Nous verrons par la suite que même pour un graphe aussi simple que le cycle, le problème de recherche d’un code identifiant optimale n’est pas facile.

Les mêmes auteurs ont montré que le problème de décision associé au problème d’existence d’un code r -identifiant de cardinalité au plus k dans un graphe $G = (V, E)$, orienté ou non, est NP-Complet pour tout r .

Théorème 4 ([23]) *Soit $r \geq 1$, le problème de décision suivant est NP-Complet :*

NOM : Code r -identifiant.

INSTANCE : Un graphe $G = (V, E)$ connexe et un entier $k \leq |V|$.

QUESTION : Existe-t-il un code r -identifiant $C \subseteq V$ de cardinalité au plus k ?

La preuve est basée sur une réduction polynômiale du célèbre problème de décision 3 – SAT [46] au problème de décision défini ci-dessus.

2.5.3 Graphes optimaux

Dans cette partie, nous présenterons des résultats parus dans [44], [24] et [33] traitant des classes de graphes ayant la propriété de posséder des codes identifiants de cardinalité

optimale⁶.

2.5.3.1 Graphes admettant un code identifiant de cardinalité égale à $\lceil \log_2(n+1) \rceil$

Théorème 5 ([44]) *Soit $G = (V, E)$ un graphe d'ordre n admettant un code r -identifiant C , alors :*

$$|C| \geq \lceil \log_2(n+1) \rceil$$

Pour un code C identifiant un ensemble d'au plus l sommets, c'est-à-dire code $(1, \leq l)$ -identifiant, on a :

Théorème 6 ([44])

$$|C| \geq \left\lceil \log_2 \sum_{i=0}^l \binom{n}{i} \right\rceil$$

Dans la suite nous allons illustrer trois méthodes pour la construction de graphes optimaux décrites, respectivement, dans [33, 44] et [24].

Première méthode Dans cette méthode, Karpovsky et al se sont intéressés à la génération de graphe d'ordre n pour lequel on associe un code identifiant de cardinalité minimum, c'est-à-dire dont la cardinalité est égale à $\lceil \log_2(n+1) \rceil$.

Soit le graphe $G = (V, E)$, avec $|V| = 2^n - 1$. Pour tout sommet de ce graphe, on associe un vecteur de longueur $k = \log_2(n+1)$ dont les composantes appartiennent à l'ensemble $\{0, 1\}$.

$$(1, 0, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (1, 1, \dots, 1)$$

On sélectionne comme mots de code les vecteurs dont les poids⁷ sont égaux à 1.

Ainsi, pour un sommet $B = (b_1, b_2, \dots, b_k)$, où $b_i \in \{0, 1\}$, si $\exists b_i = 1$, alors celui-ci est voisin au mot de code ayant la $i^{\text{ème}}$ composante égale à 1.

Une telle construction assurera que tout sommet soit couvert par un unique ensemble de mots de code et donc tout sommet est identifié de manière unique.

6. L'optimalité dans ce paragraphe est considérée dans les deux sens, c'est-à-dire maximiser et minimiser la cardinalité d'un code r -identifiant, bien que la pratique exige le plus souvent une minimisation

7. Le poids d'un vecteur binaire est égale au nombre de composantes non nulles

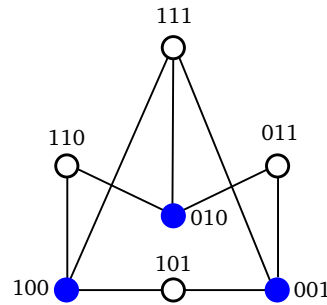


Figure 2.10 – Graphe optimal $G = (V, E)$ avec $|V| = 2^3 - 1$ ayant comme code 1-identifiant optimal l'ensemble $\{(001), (010), (100)\}$.

Deuxième méthode Une autre méthode a été présentée dans [33]. Le but est de donner une construction permettant de déterminer toutes les classes de graphe d'ordre n ayant la propriété que le code 1-identifiant associé est de cardinalité minimum, c'est-à-dire égale à $p = \lceil \log_2(n + 1) \rceil$.

Soit n un entier strictement positif ($n \geq 1$) et $p = \lceil \log_2(n + 1) \rceil$. Soit H un graphe ayant p sommets qu'on note par x_1, x_2, \dots, x_p admettant un code 1-identifiant. A partir de H on construit un autre graphe noté $\mathcal{C}(H)$ d'ordre n et admettant comme code identifiant l'ensemble $C = \{x_1, x_2, \dots, x_p\}$. Ce graphe, dit optimal, est obtenu par la construction suivante :

1. Soit H un graphe admettant l'ensemble $C = \{x_1, x_2, \dots, x_p\}$ comme code 1-identifiant, au fur et à mesure, on rajoute des sommets connectés à H de manière à ce que l'ensemble C reste toujours un code 1-identifiant.

Pour chaque sommet x_j de H , on lui associe un vecteur caractéristique de son ensemble identifiant $I(x_j, C)$. C'est un vecteur $v(x_j)$ dont les composantes appartiennent à $\{0, 1\}$ et tel que $v(x_j)_i = 1$ si et seulement si $x_i \in N[x_j]$. Soit $\mathcal{V} = \{v(x_j) | j = 1, \dots, p\}$ l'ensemble des vecteurs caractéristiques des mots du code. Puisque H est pris de sorte qu'il admet un code 1-identifiant, alors \mathcal{V} contient exactement p vecteurs, et ne contient pas le vecteur $(0, 0, \dots, 0)$.

2. Soit, maintenant, \mathcal{W} un sous-ensemble de $n - p$ éléments de $\{0, 1\}^p \setminus \{\mathcal{V} \cup \{(0, 0, \dots, 0)\}\}$. Pour tout élément $w \in \mathcal{W}$, on ajoute un sommet y_m au graphe $\mathcal{C}(H)$ tel que

$$N(y_m) = \{x_i | w_i = 1\}$$

y_m est tel que w est le vecteur caractéristique de l'ensemble identifiant $I(y_m, C)$.

3. On peut rajouter des arêtes entre deux sommets de $\mathcal{C}(H)$ correspondant au vecteurs $y_m, w \in \mathcal{W}$, ceci ne change pas le fait que $I(y_m, C) = \{x_i | w_i = 1\}$ pour tout $w \in \mathcal{C}(H)$

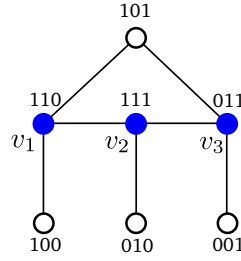


Figure 2.11 – Un exemple de construction de graphe optimal $\mathcal{C}(H)$ d'ordre 7 à partir d'un chemin de longueur 2 dont les sommets sont v_1, v_2 et v_3 . Les sommets du graphe optimal $\mathcal{C}(H)$ sont libellés par les vecteurs caractéristiques de leurs ensembles identifiants. Dans ce graphe $C = \{v_1, v_2, v_3\}, \mathcal{V} = \{(011), (110), (111)\}$ et $\mathcal{W} = \{(001), (010), (100), (101)\}$.

La validation de cette construction a été prouvée.

Théorème 7 ([33]) *Soit $n \geq 1$ et H un graphe à $p = \lceil \log_2(n+1) \rceil$ sommets, admettant un code 1-identifiant. Soit $\mathcal{C}(H)$ le graphe obtenu par la construction décrite ci-dessus. Alors $\mathcal{C}(H)$ est optimal, c'est-à-dire qu'il admet un code 1-identifiant de cardinalité $\lceil \log_2(n+1) \rceil$.*

Troisième méthode Dans cette troisième méthode, on présentera une construction décrite dans le travail de I. Charon et al [24]. Les auteurs donnent la construction d'une classe de graphes d'ordre n (fini) admettant des codes identifiants de cardinalité égale à $p = \lceil \log_2(n+1) \rceil$.

Soit r et n deux entiers tel que $k = \lceil \log_2(n+1) \rceil \geq 2r+2 \geq 4$ (où $2^k - 1 \leq n \leq 2^{k-1}$). Le but est la recherche d'un graphe G d'ordre n admettant un code r -identifiant de cardinalité égale à p .

Considérons le graphe $G_k = (V_k, E_k)$ un k -cycle (cycle de longueur k) tel que :

$$V_k = \{v_{1,1}, v_{2,1}, \dots, v_{k,1}\}, E_k = \{\{v_{1,1}, v_{2,1}\}, \{v_{2,1}, v_{3,1}\}, \dots, \{v_{k-1,k}, v_{k,1}\}, \{v_{k,1}, v_{1,1}\}\}$$

On peut déduire que l'ensemble $C = V_k$ est un code r -identifiant de G_k , car $k \geq 2r+2$.

Pour tout $1 \leq i \leq k$, soit $G'_i = (V'_i, E'_i)$ une chaîne de longueur $r-2$ tel que :

$$V'_i = \{v_{i,2}, \dots, v_{i,r}\}, E'_i = \{\{v_{i,2}, v_{i,3}\}, \dots, \{v_{i,r-1}, v_{i,r}\}\}$$

Ainsi, chaque chaîne G'_i est reliée au sommet $v_{i,1}$.

Maintenant, on construit le graphe $G_k^* = (V_k^*, E_k^*)$, d'ordre kr , dont l'ensemble des sommets et des arêtes sont respectivement :

$$V_k^* = V_k \cup \left(\bigcup_{i=1}^k V'_i \right), \quad E_k^* = E_k \cup \left(\bigcup_{i=1}^k E'_i \right) \cup \left(\bigcup_{i=1}^k \{v_{i,1}, v_{i,2}\} \right)$$

où l'ensemble $C = V_k$ demeure toujours un code 1-identifiant du nouveau graphe G_k^* , pour les deux raisons suivantes :

- a-** Tous les sommets de V'_i sont r -couvert par C .
- b-** Soient deux sommets distincts $v_{i,j}$ et $v_{m,l}$ de V_k^* . Si $j < l$, alors $v_{i,j}$ est r -couvert par plus de mots du code que $v_{m,l}$, et si $j = l$ alors $v_{i,j}$ et $v_{m,l}$ ne peuvent être r -couvert par les même mots du code.

Notons que $|V_k^*| = kr \leq k(k-2)/2 < 2^{k-1} \leq n$.

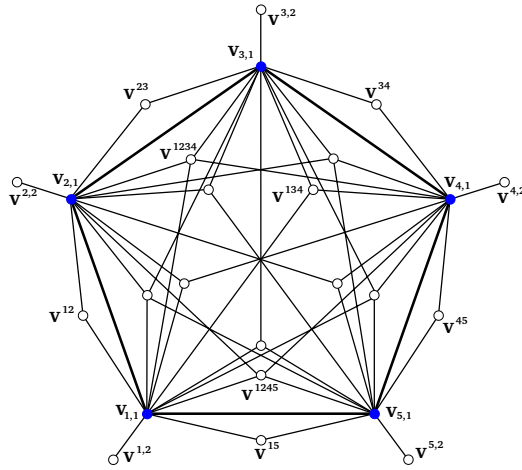


Figure 2.12 – Un exemple du cas où $r = 1$ et $k = 5$, les sommets colorés constituent un code identifiant. Dans ce graphe le 5-cycle est $C = \{v_{1,1}, v_{2,1}, v_{3,1}, v_{4,1}, v_{5,1}\}$, on n'a construit que 25 sommets, en lui ajoutant les six sommets suivants : $v^{13}, v^{14}, v^{24}, v^{25}, v^{35}, v^{12345}$, on obtiendra le graphe optimal.

Par ailleurs, des sommets peuvent être rajouter au graphe toute en gardant l'ensemble C comme étant un code r -identifiant. En considérant un sous ensemble de V_k , de p éléments. Soit $\{v_{i_j,1} : 1 \leq i_j \leq k, 1 \leq j \leq p\}$ un ensemble non vide et différent de tous les sous-ensembles r -couvrant les sommets de V_k^* . A cet ensemble, on associe les nouveaux sommets v^{i_1, \dots, i_p} , qu'on rajoute au graphe G_k^* , et qui sont reliés aux sommets $v_{i_1, r}, \dots, v_{i_p, r}$. De ce fait, il existe $2^k - 1 - |V_k^*| = 2^k - 1 - kr$ sous-ensembles possibles de ce type de sommets et puisque $2^k - 1 \geq n$, on peut rajouter $n - kr$ de ce type de sommets au graphe G_k^* et on obtient, ainsi, le graphe G d'ordre n . Puisque les sommets v^{i_1, \dots, i_p} sont r -couvert par $v_{i_1,1}, \dots, v_{i_p,1}$, cette construction montre que l'ensemble C , choisi au départ

(le k -cycle), demeure un code r -identifiant pour le nouveau graphe G .

Ainsi, cette construction permet d'affirmer le résultat donné par le théorème suivant :

Théorème 8 ([24]) *Soit r et n deux entiers tel que $n \geq 2^{2r+1}$. Il existe un graphe connexe d'ordre n admettant un code r -identifiant de cardinalité égale à $\lceil \log_2(n+1) \rceil$.*

Ce théorème est valide pour $r = 1$ avec $n = 1$ et $n \geq 3$, et pour $r = 2$ est valide si et seulement si $n \geq 6$.

2.5.3.2 Graphes admettant un code identifiant de cardinalité égale à $n - 1$

Dans cette partie, on présentera des graphes dits "mauvais" ([24]). Cette nomination est un peu logique, et ceci vient du fait que lorsqu'on est amené à minimiser la cardinalité d'un code identifiant, il se trouve que tous les codes identifiants possibles ont une cardinalité proche de l'ordre du graphe considéré, à savoir $n - 1$ ou n .

Avant de présenter la méthode permettant la construction de tels graphes, on évoquera le théorème suivant montrant l'existence de tels graphes (d'ordre infini) :

Théorème 9 ([59]) *Soit $G = (V, E)$ un graphe d'ordre fini et admettant un code 1-identifiant (1-identifiable). Alors il existe $x \in V$ tel que $V \setminus \{x\}$ est un code 1-identifiant de G .*

Dans ce théorème, on exclue le cas où G est lui même un stable.

Une généralisation immédiate pour un code r -identifiant est donnée par le corollaire suivant :

Corollaire 2 ([59]) *Soit r un entier strictement positif et $G = (V, E)$ un graphe r -identifiable, non stable et de multiplicité fini. Alors il existe $x \in V$ tel que $V \setminus \{x\}$ est un code r -identifiant de G .*

Dans [24], I.Charon et al exhibent des graphes d'ordre n pour lesquels tout code r -identifiant minimum est de cardinalité égale à $n - 1$ sommets.

Théorème 10 ([24]) *Pour tout $n \geq 3$, il existe un graphe G_n connexe d'ordre n , tel que tout code 1-identifiant est de cardinalité égale à $n - 1$.*

Au moins trois graphes d'ordre n admettent un code 1-identifiant de cette cardinalité. Le premier de ces graphes est le plus simple, c'est le graphe étoile qu'on note par $K_{1,n-1}$. Le deuxième graphe a un nombre pair de sommets, tel que :

$$V_n = \{0, 1, \dots, n-1 = 2p-1\}, \text{ et}$$

$$E_n = \{\{i, j\} : i, j \in V_n, i \neq j, i \neq j + p \pmod{n}\}$$

où $p \geq 2$. Ce graphe n'est autre qu'un graphe complet (K_{2p}) en lui ôtant le couplage parfait (voir la figure 2.13).

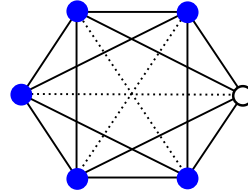


Figure 2.13 – Exemple de graphe ayant un code identifiant de cardinalité minimum égale à $n-1$, les sommets colorés représentent le code. Le couplage est désigné par les lignes discontinues.

Ainsi, tout sous-ensemble de $n-1$ sommets est un code 1-identifiant. La preuve se base sur l'impossibilité de construire un code possédant moins de $n-1$ sommets. Car on a pour tout sommet $i \in V_n$,

$$B_1(i) = V_n \setminus \{i + p \pmod{n}\}$$

et pour toute paire de sommets $(i, j) \in V_n^2$, $i \neq j$ on a :

$$B_1(i) \Delta B_1(j) = \{i + p \pmod{n}, j + p \pmod{n}\}$$

Ceci traduit le fait qu'au moins un sommet de la différence symétrique ci-dessus doit appartenir au code C pour séparer les deux sommets i et j . En supposant, sans perdre de généralité, que $\{0\} \neq C$. Alors pour tout sommets $j \neq p$,

$$B_1(p) \Delta B_1(j) = \{0, j + p \pmod{n}\} \text{ et}$$

$$\emptyset \neq (B_1(p) \Delta B_1(j)) \cap C \subseteq \{j + p \pmod{n}\}.$$

De ce fait, tous les $n-1$ sommets $j + p \pmod{n}$ appartiennent nécessairement au code, à l'exception du sommet 0.

Le troisième graphe possède un nombre impair de sommets, tel que :

$$V_n = \{0, 1, \dots, n-1 = 2p\}, \text{ est l'ensemble de ses sommets, et}$$

$$E_n = \{\{i, j\} : i, j \in V_n \setminus \{n-1\}, i \neq j, i \neq j+p \pmod{n-1}\} \cup \{\{n-1, j\} : j \in V_n \setminus \{n-1\}\}$$

l'ensemble des arêtes de ce graphe.

A la différence avec le deuxième graphe, on rajoute, pour ce graphe, un sommet adjacent à tous les autres sommets (voir la figure 2.14).

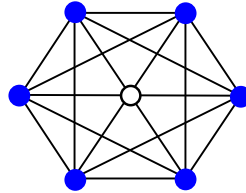


Figure 2.14 – Code identifiant (sommets colorés) ayant pour cardinalité minimum $n - 1$.

Puisque le sommet $n - 1$ est à distance 1 de tous les autres, on a pour tout sommet $i, 0 \leq i \leq n - 2, B_1(i) \Delta B_1(n - 1) = \{j + p \pmod{n - 1}\}$. De ce fait, les $n - 1$ ^{ème} premiers sommets doivent appartenir au code.

Une généralisation a été donnée par les deux théorèmes qui suivent, et leurs démonstrations reposent sur l'utilisation du deuxième et du troisième graphes su-cités.

Théorème 11 ([24]) *Soit r et n deux entiers, où $r \geq 2$. Alors, pour tout n pair, avec $n \geq 3r^2$, il existe un graphe G_n connexe d'ordre n tel que tout code r -identifiant de G_n de cardinalité minimum possède $n - 1$ sommets.*

Théorème 12 ([24]) *Soit r et n deux entiers, où $r \geq 2$. Alors, pour tout n impair, avec $n \geq 3r^2$, il existe un graphe, G_{n+1} connexe d'ordre $n + 1$ tel que tout code r -identifiant de G_{n+1} de cardinalité minimum possède n sommets.*

Il faut noter que les théorèmes 11 et 12 traitent, respectivement, le cas pair et impair de l'ordre du graphe.

2.6 Résultats dans certaines classes de graphes

Dans cette section, nous présentons les plus importants résultats sur les codes r -identifiants. Pour commencer, nous donnons la borne générale pour le cas des graphes réguliers :

Théorème 13 ([44]) *Soit un entier $r \geq 1$. La cardinalité minimum d'un code r -identifiant pour un graphe, $G = (V, E)$, régulier d'ordre n est :*

$$M_r(G) \geq \frac{2n}{b_r + 1}$$

où b_r est la cardinalité d'une boule quelconque de rayon r centrée sur un sommet arbitraire.

Cette borne a été prouvée d'une autre manière dans [17], ce qui a permis d'obtenir des bornes plus fines.

2.6.1 Codes identifiants dans les grilles

2.6.1.1 Grilles carrées

Plusieurs auteurs ont étudié le problème des codes r -identifiants dans cette classe de graphes, en particulier [17, 44, 67] pour le cas d'un code identifiant ($r = 1$). Ce cas a été complètement résolu et nous avons un code identifiant optimal de densité égale à $\frac{7}{20}$ (voir la figure 2.15) :

Théorème 14 ([67]) *La densité minimum d'un code identifiant dans \mathbb{Z}^2 est $\frac{7}{20}$*

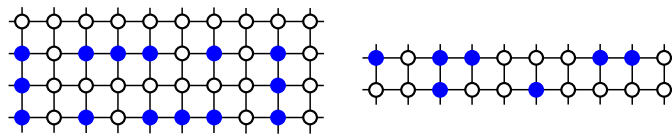


Figure 2.15 – Deux codes identifiants de densité $\frac{7}{20}$

Dans le cas des codes r -identifiants nous avons :

$$\frac{3}{8r + 1} \leq d_r^*(\mathcal{G}_C) \leq \begin{cases} \frac{2}{5r} & \text{si } r \text{ est pair,} \\ \frac{2r}{5r^2 - 2r + 1} & \text{si } r \text{ est impair.} \end{cases}$$

où les bornes inférieures et supérieures sont obtenues respectivement dans [29] et [38].

Cependant pour de petites valeurs de r , il peut exister des constructions de densité inférieure à celles données dans la construction générale (voir la figure 2.16). Ainsi, pour $3 \leq r \leq 6$ de nouvelles bornes supérieures ont été établies dans [22], les différents résultats sont listés dans le tableau suivant :

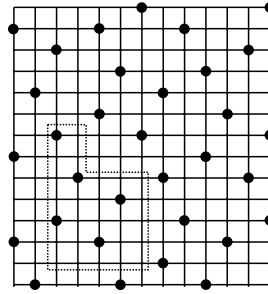


Figure 2.16 – Exemple de code 2-identifiant de densité $\frac{5}{29}$

r	borne inf	nouvelles borne sup
3	$\frac{3}{28} \approx 0.1072$	$\frac{1}{8} \approx 0.125$
4	$\frac{1}{12} \approx 0.0833$	$\frac{8}{85} \approx 0.0941$
5	$\frac{3}{44} \approx 0.0682$	$\frac{2}{25} \approx 0.08$
6	$\frac{3}{52} \approx 0.0577$	$\frac{3}{46} \approx 0.0652$

Par ailleurs, les grilles carrées finies ont été aussi étudiées dans [43]. Les constructions obtenues se basent, essentiellement, sur les résultats optimaux déjà obtenus dans les grilles carrées infinies, en apportant un nombre de corrections (ajout ou soustraction de sommets au code).

2.6.1.2 Grilles triangulaires

Ce sont les grilles carrées avec une diagonale, c'est une autre famille de graphes réguliers. Dans le cas d'un code identifiant, nous avons une construction donnée dans [44] qui coïncide avec la borne inférieure générale (voir la figure 2.17).

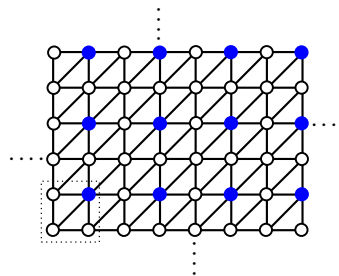


Figure 2.17 – code identifiant (les sommets colorés) de densité optimale égale à $\frac{1}{4}$.

Dans [29], en posant des conditions sur r , les auteurs ont donné des constructions générales de codes r -identifiants. Les résultats obtenus correspondent au cas où r est impair, $r \geq 4$ et divisible par 4 et pour r paire et non divisible par 4.

Par la suite, les travaux parus dans [22], I. Charon et al ont pu établir de nouvelles bornes

moyennant une heuristique, pour de petites valeurs de r ($2 \leq r \leq 6$). Le tableau suivant représente un récapitulatif des résultats obtenus :

r	borne inf	borne sup
2	$\frac{2}{15} \approx 0.1333$	$\frac{1}{6} \approx 0.1667$
3	$\frac{2}{21} \approx 0.0952$	$\frac{2}{17} \approx 0.1177$
4	$\frac{2}{27} \approx 0.0741$	$\frac{1}{12} \approx 0.0833$
5	$\frac{2}{33} \approx 0.0606$	$\frac{1}{13} \approx 0.0769$
6	$\frac{2}{39} \approx 0.0513$	$\frac{1}{14} \approx 0.0714$

Pour le cas le plus général, c'est-à-dire les codes $(r, \leq l)$ -identifiants, des travaux ont été réalisés, on peut citer par exemple le travail de Honkala et Laihonen [39], où les auteurs montrent que la densité minimum d'un code $(1, \leq 2)$ -identifiant, dans une grille triangulaire infinie est au moins $\frac{9}{16}$, et donnent une construction atteignant cette borne.

De plus ils montrent que pour tout $n \geq 16$ et divisible par 4, le code $(1, \leq 2)$ -identifiant optimal possède au moins $\frac{9}{16}n^2$ mots de code. En outre, les codes identifiants robustes ont été aussi considérés dans [45] pour cette famille de graphes.

2.6.1.3 Grilles hexagonales

Connues aussi sous le nom de grilles briques. Pour le cas d'un code identifiant nous avons

$$\frac{12}{29} \leq d_1^*(\mathcal{G}_H) \leq \frac{3}{7}$$

où les bornes inférieures et supérieures sont donnée respectivement dans [5] et [16].

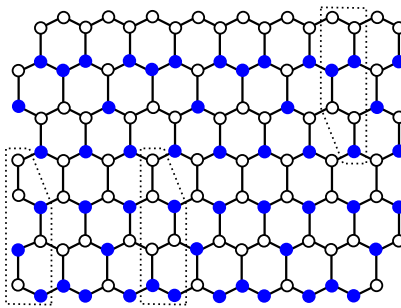


Figure 2.18 – Exemple de code identifiant de densité $\frac{3}{7}$

Pour les cas où $r \geq 2$, le tableau suivant résume quelques résultats donnés dans [22] :

r	borne inf	nouvelle borne inf
2	$\frac{2}{11} \approx 0.1818$	$\frac{4}{19} \approx 0.2105$
3	$\frac{2}{17} \approx 0.1176$	$\frac{1}{6} \approx 0.1667$
4	$\frac{2}{23} \approx 0.087$	$\frac{1}{9} \approx 0.1111$
5	$\frac{2}{27} \approx 0.0741$	$\frac{4}{35} \approx 0.1143$
6	$\frac{2}{33} \approx 0.0606$	$\frac{1}{11} \approx 0.0909$
7	$\frac{2}{37} \approx 0.0541$	$\frac{1}{12} \approx 0.0833$
8	$\frac{2}{43} \approx 0.0456$	$\frac{1}{13} \approx 0.0769$

Notons que les résultats obtenus sont pour des valeurs allant jusqu'à $r = 30$.

2.6.1.4 Grilles carrées à deux diagonales

Tout comme les grilles précédentes, une grille royale est aussi un graphe régulier. En 2001, G. Cohen et al [19] ont montré que pour une grille royale infinie la densité minimale d'un code identifiant est au moins égale à $\frac{2}{9}$. Puis en 2002, l'heuristique développée dans [22] montre que la nouvelle borne supérieure coïncide parfaitement avec les bornes inférieures déjà établies dans [29, 30]. Le tableau suivant résume quelques résultats obtenus pour la densité minimale de codes r -identifiant ($r = 1, \dots, 4$) :

r	borne inf	la nouvelle borne sup
1	$\frac{2}{9} \approx 0.2222$	$\frac{2}{9} \approx 0.2222$
2	$\frac{1}{8} \approx 0.125$	$\frac{1}{8} \approx 0.125$
3	$\frac{1}{12} \approx 0.0833$	$\frac{1}{12} \approx 0.0833$
4	$\frac{1}{16} \approx 0.0625$	$\frac{1}{16} \approx 0.0625$

Dans [30], I. Charon et al ont traité le problème de recherche de code r -identifiant de densité minimum pour tout $r \geq 2$, et ont donné la cardinalité minimum.

Théorème 15 ([30]) *Pour tout entier r positif, il existe un code r -identifiant de densité égale à $\frac{1}{4r}$*

Soit $k \in \mathbb{Z}$, et C_k l'ensemble de sommets tel que

$$C_k = \{(2kr + \alpha, \alpha) \mid \alpha \in \mathbb{Z}, \alpha \text{ pair}\}$$

Le code C représentant l'union de tous les ensembles C_k pour tout $k \in \mathbb{Z}$ est un code r -identifiant.

En outre, en montrant que tout code r -identifiant a une densité au moins égale à $\frac{1}{4r}$, alors

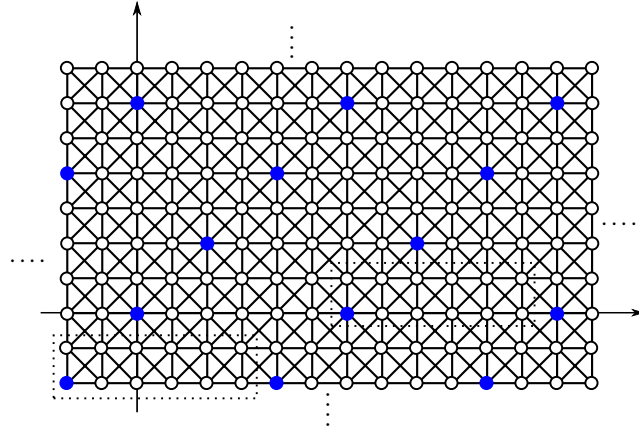


Figure 2.19 – Exemple de construction de code 3-identifiant de densité optimale égale à $\frac{1}{12}$

Théorème 16 ([30]) *La densité minimum d'un code r -identifiant est telle que :*

$$d_r^*(\mathcal{G}_K) = \begin{cases} \frac{2}{9} & \text{Si } r = 1 \\ \frac{1}{4r} & \text{Pour tout } r \geq 2 \end{cases}$$

2.6.2 Hypercubes

Beaucoup de chercheurs se sont penchés sur le problème des codes identifiants dans les hypercubes, ce qui a engendré une littérature abondante. Parmi les références traitant de cette classe de graphes, nous citons [12, 13, 14, 15, 28, 31, 44, 56, 57, 58, 63, 66].

Dans le cas des hypercubes nous avons un résultat asymptotique donné par le théorème suivant :

Théorème 17 ([57]) *Soit $l \geq 1$ un entier fixe et $\rho \in [0, \frac{1}{2}[$. Supposons que $\frac{r}{n} \rightarrow \rho$, alors :*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log_2 M_r^{\leq l}(n) = 1 - h(\rho)$$

où $M_r^{\leq l}(n)$ et $h(\cdot)$ représentent respectivement la cardinalité minimum d'un code $(r, \leq l)$ -identifiant dans Q_n et la fonction entropie binaire, tel que $h(x) = -x \log_2 x - (1-x) \log_2(1-x)$.

On peut remarquer que lorsque $l \geq 2$, il n'existe pas de code $(r, \leq l)$ -identifiant pour tout $r \geq \lfloor \frac{1}{2} \rfloor$. Ce qui explique le fait qu'on suppose que $\rho < \frac{1}{2}$ dans le théorème ci-dessus.

U. Blass, I. Honkala, S. Litsyn ont posé la conjecture suivante :

Conjecture 1 ([66]) *Pour tout r et $n \geq 2r + 1$ on a*

$$M_r(n) \leq M_r(n+1)$$

Cette conjecture a été démontrée dans le cas où $r = 1$ par J. Moncel [31]. De manière encore plus générale, nous avons la conjecture suivante :

Conjecture 2 ([66]) *Pour tout r et $n \geq 2r + 1$ on a*

$$M_{r_1+r_2}(n_1 + n_2) \leq M_{r_1}(n_1)M_{r_2}(n_2)$$

Dans [15], cette conjecture a été prouvée partiellement dans le cas général pour les codes $(r, \leq l)$ -identifiants. Ainsi, nous avons

Théorème 18 ([15]) *Pour tout $l \geq r + 3$, on a*

$$M_{r+1}^{\leq l}(n + m) \leq M_1^{\leq l}(n)M_r^{\leq l}(m)$$

Dans la même optique, G. Exoo et al ont montré le résultat suivant :

Théorème 19 ([12]) *pour tout $n \geq 2$, on a*

$$M_1(n + 1) \leq \left(2 + \frac{1}{n + 1}\right) M_1(n)$$

Les résultats présentés ici ont fait l'objet de plusieurs améliorations.

Dans le tableau suivant, on récapitule les résultats les plus récents obtenus dans les hypercubes [14, 28] :

n	$M_1(n)$		$M_2(n)$		$M_3(n)$	
	borne inf	borne sup	borne inf	borne sup	borne inf	borne sup
2	3	3	–	–	–	–
3	4	4	7	7	–	–
4	7	7	6	6	15	15
5	10	10	6	6	10	10
6	19	19	8	8	7	7
7	32	32	14	14	8	8
8	56	61	17	21	10	13
9	101	112	27	32	13	17
10	183	208	43	57	18	25
11	337	352	71	100	25	36
12	623	684	118	177	39	67
13	1158	1280	199	318	61	109
14	2164	2550	341	566	95	180
15	4063	4787	590	1020	151	305
16	7654	9494	1033	1844	241	530
17	14469	18558	1824	3476	383	901
18	27434	35604	3244	6430	608	1628
19	52155	65536	5809	12458	974	2846
20	99392	131072	10465	25401	1656	5813

L'équivalence existante entre les codes $(1, \leq l)$ -identifiants et les codes μ -fold m -couvrants⁸ a été à l'origine de la construction de codes identifiants optimaux. De ce fait, nous avons :

Corollaire 3 ([63]) *Soit un entier $l \geq 3$. Alors*

$$M_1^{(\leq l)}(n) = (2l - 1) \frac{2^n}{n + 1}$$

si et seulement si il existe deux entiers i et μ_0 tel que $\mu_0 \equiv 0 \pmod{2l - 1}$, $2l - 1 \leq 2^i \mu_0$ et $n = \mu_0 2^i - 1$

Corollaire 4 ([56]) *Pour tout $k \geq 1$:*

$$M_1^{(\leq 2)}(3 \cdot 2^k - 1) = 2^{3 \cdot 2^k - k - 1}$$

et pour tout $k \geq 3$:

$$M_1^{(\leq 2)}(2^k - 1) = 3 \cdot 2^{2^k - k - 1}$$

2.6.3 Chaînes et cycles

Dans le cas des chaînes (\mathcal{CH}) et des cycles (\mathcal{CY}), le problème de recherche d'un code r -identifiant de cardinalité minimum est complètement résolu.

Dans les chaînes infinies, nous avons la densité optimale d'un code r -identifiant qui est donnée par le théorème suivant :

Théorème 20 ([50]) *Pour tout $r \geq 1$, on a :*

$$d_r^*(\mathcal{CH}_\infty) = \frac{1}{2}$$

Dans les premiers travaux N. Bertrand et al [50] ont obtenu les résultats suivants dans le cas où la longueur du circuit est paire :

- $M_r(\mathcal{CY}_n) = \frac{n}{2}$ pour tout $r \geq 1$ et $n \geq 2r + 4$,
- $M_r(\mathcal{CY}_{2r+2}) = 2r + 1$ pour tout $r \geq 1$;

8. Un code μ -fold m -couvrant est un code m -couvrant où chaque sommet est couvert par au moins μ mots du code [20].

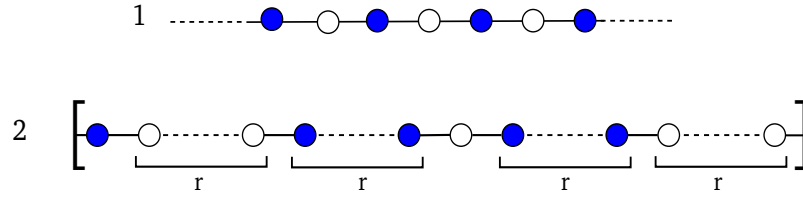


Figure 2.20 – Exemple de deux codes r -identifiants périodiques de densité minimum égale à $\frac{1}{2}$ dans \mathcal{CH}_n , les mots du codes sont colorés

Il ne restait que le cas où la longueur est impaire. Gravier et al ont étudié ce cas dans [60]. Pour ce faire, les auteurs ont défini un graphe $C'_{(n,r)}$ ayant pour ensemble de sommets $\{v_i : i \in \mathbb{Z}_n\}$ tel que $v_{i-r}v_{i+r+1}$ soit une arête de $C'_{(n,r)}$. Ainsi, ils ont obtenu des bornes générales pour la cardinalité minimum d'un code r -identifiant dans les cycles :

Théorème 21 ([60]) *Pour tout $r \geq 1$ et $n \geq 2r + 3$ et impair, on a :*

$$\frac{n+1}{2} + \frac{\text{pgcd}(2r+1, n) - 1}{2} \leq M_r(\mathcal{CY}_n) \leq \frac{n+1}{2} + r$$

Les auteurs ont donné les conditions sous lesquelles ces bornes sont atteintes. Cependant, le cas où $\text{pgcd}(2r+1, n) = 1$, et $n \geq 3r + 2$ et impair n'a été traité que plus tard dans [47].

Théorème 22 ([47]) *Soient $r \geq 1$ et $n \geq 3r + 2$ et impair, et $\text{pgcd}(2r+1, n) = 1$.*

$$M_r(\mathcal{CY}_n) = \begin{cases} \frac{n+1}{2} + 1 & \text{si } n = 2m(2r+1) + 1 \text{ ou } n = (2m+1)(2r+1) + 2r \\ \frac{n+1}{2} & \text{sinon.} \end{cases}$$

D.L Roberts et al ont traité le cas où $r = 2$ en donnant la cardinalité minimum d'un code 2-identifiant aussi bien dans une chaîne que dans un cycle [6].

Conclusion

Dans ce chapitre, nous avons donné les différentes terminologies et notations utilisées pour les codes identifiants.

Nous avons évoqué les liens existants entre les codes identifiants avec d'autres types de codes, à savoir : les codes localisateurs-dominateurs et les codes discriminants. Des variantes des codes identifiants sont données et le problème NP-dur de recherche de code identifiant de cardinalité minimum a été défini dans les graphes non-orientés.

En outre, la question d'existence de code identifiant a été évoquée ainsi que la construction de graphes optimaux. Enfin, nous avons exposé les plus importants résultats obtenus

pour certaines classes de graphes tels que les grilles, les chaînes, les cycles et les hypercubes.

Codes Identifiants dans les Graphes Orientés

Chaque homme passe sa vie entre deux meules. S'il est homme, elle l'usent jusqu'à la perfection. Sinon, jusqu'à sa destruction. Les meules sont le ciel et la terre.

PROVERBE CHINOIS.

En considérant des liens directionnels (orientés) entre les processeurs du système, le graphe correspondant à ce réseau sera orienté. Dans ce cas, un processeur p_1 ne peut tester un autre processeur p_2 que s'il existe un arc allant de p_1 vers p_2 .

Ainsi, on dira qu'un sous-ensemble de sommets C est un code identifiant d'un graphe orienté $G_D = (V_D, E_D)$ si et seulement si :

1. pour tout sommet $v \in V_D$, on a : $\Gamma^-[v] \cap C \neq \emptyset$ et
2. pour toute paire de sommets $u, v \in V_D$, on a : $\Gamma^-[u] \cap C \neq \Gamma^-[v] \cap C$

où $\Gamma^-[v]$ représente l'ensemble des prédécesseurs du sommet v y compris ce dernier.

Dans l'exemple donné par la figure (3.1) on a :

$$\Gamma^-[1] = \{1, 4\}, \quad \Gamma^-[2] = \{1, 2\}, \quad \Gamma^-[3] = \{1, 2, 3, 4\}, \quad \Gamma^-[4] = \{4\}$$

Ainsi, l'ensemble $C = \{1, 2, 3\}$ représente un code identifiant dans ce graphe.

Peu de travaux traitant le problème de recherche de codes identifiants de cardinalité minimum dans le cas de graphes orientés existent, mis à part le cas d'arbre orienté où un

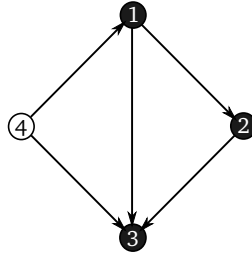


Figure 3.1 – Exemple de code identifiant d'un graphe orienté.

algorithme linéaire a été proposé [25] et une modélisation du problème en un programme linéaire en nombres entiers (PLNE).

Dans [21], l'étude de la complexité du problème en question montre que pour un graphe G asymétrique et un entier k , le problème de décision relatif à l'existence d'un code r -identifiant de cardinalité au plus k dans G est NP-Complet pour tout $r \geq 1$.

3.1 Algorithme linéaire pour les arbres orientés

Cette section est dédiée à la présentation de l'algorithme linéaire pour la recherche d'un code identifiant dans un arbre orienté [25].

Avant de commencer la description de l'algorithme, nous présentons d'abord la terminologie utilisée dans cette algorithme. Pour ne pas tomber dans la confusion, oublions pour le moment l'orientation.

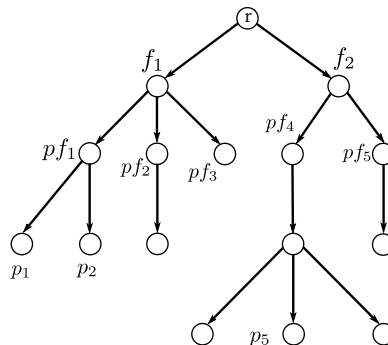


Figure 3.2 – Exemple d'arbre orienté, $T = (V, A)$, de profondeur égale à 4 et de racine r

Dans la figure ci-dessus, le graphe étant un arbre, noté par $T(V, A)$, et admettant le sommet r comme racine. On appelle aussi r le père des sommets f_1 et f_2 et grand-père de

pf_1, pf_2, \dots, pf_5 . Les sommets dont le degré est égale à 1 sont appelé des feuilles, comme par exemple les sommets p_1, p_2 et p_5 (voir la figure 3.2).

Enfin, on appelle *profondeur* (parfois appelée aussi *hauteur*) la plus longue distance entre la racine r et une feuille de l'arbre T (dans notre exemple elle est égale à 3).

Lorsqu'on tient compte des orientations on parle aussi de voisin-entrant et voisin-sortant. Ainsi, dans la figure ci-dessus la racine r est un voisin entrant de f_1 et f_2 .

Le premier appel de l'algorithme récursif ID-TREE commence par un arbre $T_0 = (V, E)$ non vide à partir duquel on construit un code 1-identifiant C de T_0 , et un sous-ensemble de sommets $C_0 = \emptyset$.

Dans la suite T et C désignent, respectivement, l'arbre et le code identifiant d'un appel de l'algorithme. A l'étape initiale on a l'arbre et le code initiaux T_0, C_0 .

L'idée de l'algorithme est qu'à chaque étape on réduit l'arbre T et on augmente l'ensemble C . En d'autre terme, l'algorithme ne fait qu'ôter des sommets de T et rajouter des sommets à C .

ALGORITHME ID-TREE(T,C)

Cas 1 : La profondeur de T est au plus 1

- **Si** f a des voisins sortant appartenant au code C , alors on ôte ces sommets de V (voir figure 3.3.0).
- **Si non** on opère selon ces cinq cas :
 1. Si f n'a pas de voisins entrants, alors à C on rajoute f et tous ses voisins sortants(voir figure 3.3.1).
 2. Si f n'a qu'un seul voisin entrant s et au moins un voisin sortant, alors à C on rajoute s, f et tous ses voisins sortants à l'exception d'un seule(voir figure 3.3.2).
 3. Si f n'a qu'un seul voisin entrant s mais pas de voisin sortant, alors $C \leftarrow C \cup \{f, s\}$ (voir figure 3.3.3).
 4. Si f a au moins deux voisins entrants et au moins un voisin sortant, alors à C on rajoute f et tous ses voisins entrants et tous ses voisins sortants sauf un(voir figure 3.3.4).
 5. Si f a au moins deux voisins entrants et aucun voisin sortant, alors à C on rajoute tous les voisins entrants de f (voir figure 3.3.5).
- **Stop :** fin de l'algorithme.

Cas 2 : L'arbre a une profondeur au moins égale à 2.

Étape 1 : On choisit un $x \in V$ tel que tous ses fils sont des feuilles, on appellera ce sommet x un *porte-feuille*.

Étape 2 : Cette étape englobe deux opérations α et β , tel que :

Opération α : On ôte de V tous les voisins sortants de x qui sont des mots du code (appartenant au code C).

Opération β : Si x a au moins un fils, alors on opère suivant les neuf cas suivant :

1. Si x n'a aucun fils entrant et a un père entrant y , alors on rajoute à C les sommets x, y et tous les fils de x sauf un, puis on enlève x et tous ses fils de V (voir figure 3.4.1).

2. Si x n'a aucun fils entrant et a un père sortant y , alors on ajoute x et tous ses fils à C et on enlève tous les fils de x de V (voir figure 3.4.2).
3. Si x a au moins un fils sortant et au moins un fils entrant, alors on ajoute x et tous ses fils sauf un fils sortant à C . Puis on enlève tous les fils de x de V (voir figure 3.4.3).
4. Si x n'a aucun fils sortant, a au moins deux fils entrants, et a un père entrant, alors on ajoute tous les fils de x à C , et on enlève x et tous ses fils de V (voir figure 3.4.4).
5. Si $x \in C$, x n'a aucun fils sortant, a au moins deux fils entrants, et a un père sortant y , alors : on ajoute tous les fils de x à C , on enlève tous les fils de x sauf un t de V , on enlève l'arc (t, x) de A et on ajoute l'arc (t, y) à A (voir figure 3.4.5).
6. Si $x \notin C$, x n'a aucun fils sortant, a au moins deux fils entrants, et a un père sortant, alors : on ajoute tous les fils de x à C , on enlève x et tous ses fils de V (voir figure 3.4.6).
7. Si $x \notin C$, x n'a aucun fils sortant, a un seul fils entrant, et a un père entrant y , alors : on ajoute x et y à C , on enlève x et le fils de x de V (voir figure 3.4.7).
8. Si $x \in C$, x n'a aucun fils sortant, a un seul fils entrant, et a un père entrant, alors : on ajoute le fils de x à C , on enlève x et le fils de x de V (voir figure 3.4.8).
9. Si x n'a aucun fils sortant, a un seul fils entrant t , et a un père sortant y , alors : on ajoute x et t à C , on enlève l'arc (t, x) de A et on ajoute l'arc (t, y) à A (voir figure 3.4.9).

Étape 3 : Appel récursif : ID-Tree(T, V) avec les nouveaux paramètres.

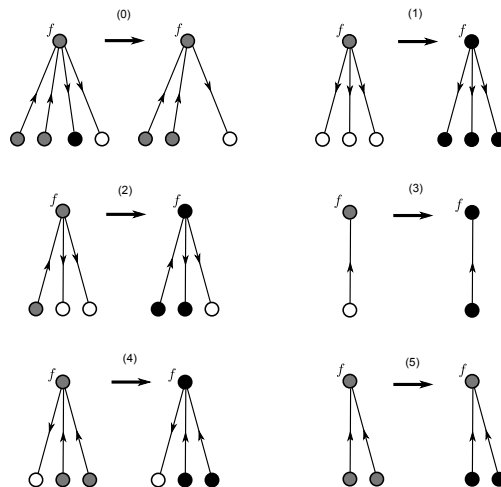


Figure 3.3 – Cas où T est de profondeur 1 dans l'algorithme. Les sommets noirs appartiennent au code C , les blancs dans $V \setminus C$, et le statut des sommets gris est indéterminé. Un sommet gris restant gris après la transformation n'a pas changé de statut.

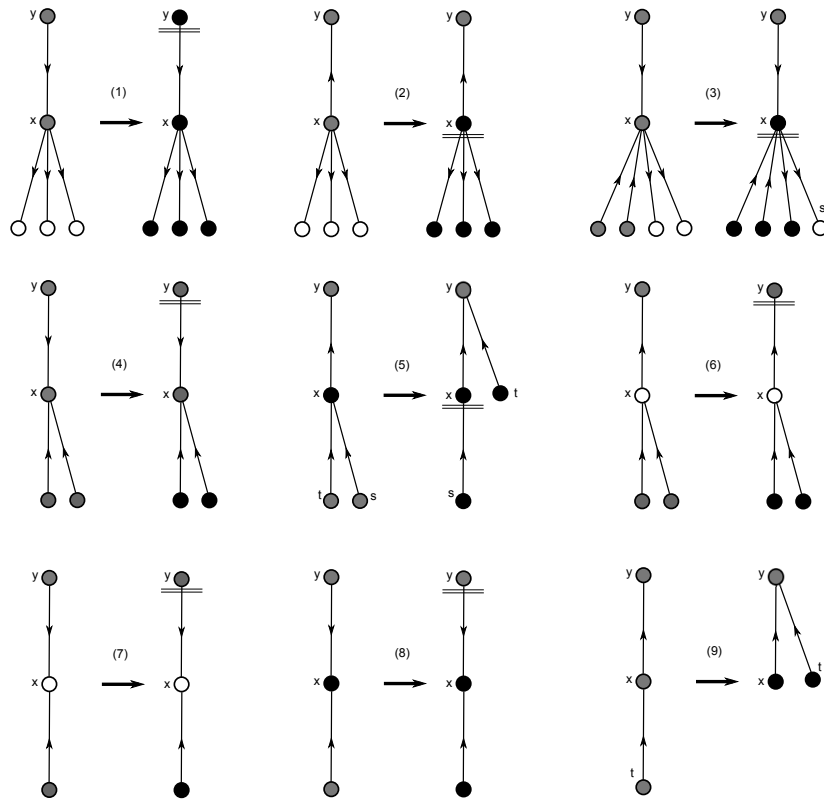


Figure 3.4 – Les cas $\beta.1$ à $\beta.9$ de l'algorithme. Les sommets noirs sont dans C , les blancs dans $V \setminus C$, et le statut des sommets gris est indéterminé. Un sommet gris restant gris après la transformation n'a pas changé de statut. Les sommets situés au-deçà de la double barre sont enlevés de V .

3.1.1 Validité et linéarité de l'algorithme

On remarque qu'un sommet n'est choisi qu'au plus une seule fois comme porte-feuille (sommet dont le fils est un sommet pendant) ce qui implique que l'algorithme se termine. De plus chaque opération effectuée réduit soit le nombre de sommets, soit le nombre de sommets de profondeur maximale de l'arbre. Ainsi, on peut affirmer que l'algorithme réduit la taille du problème initial.

La validité de l'algorithme a été prouvée par induction sur le nombre d'appels de l'algorithme récursif, d'une part.

D'autre part, la linéarité de l'algorithme a été aussi prouvée. En effet, le parcours de l'arbre se fait en largeur car l'algorithme décrit l'arbre en donnant pour chaque sommet son père et ses fils ainsi que la direction des arcs.

Ainsi, ce parcours se fait en un temps linéaire par rapport au nombre de sommets de l'arbre. En outre, les opérations effectuées ne traitent que des arcs et ne nécessitent qu'un nombre élémentaire d'opérations borné par une constante. Ce qui montre que la complexité de l'algorithme est linéaire par rapport au nombre d'arcs.

3.2 Modélisation en PLNE du problème de code identifiants

Dans cette section, nous donnons une formulation du problème d'optimisation combinatoire lié à la recherche de code identifiant de cardinalité minimum dans un graphe orienté. Notons que ce travail est une extension des résultats dans le cas de graphes non-orientés au cas de graphes orienté [68, 69].

3.2.1 Description du problème

Rappelons que le problème de recherche de code identifiant de cardinalité minimum est défini comme suite :

NOM : CODE r -IDENTIFIANT :

INSTANCE : Un graphe, $G = (V, E)$, admettant un code r -identifiant

QUESTION : Quelle est le sous-ensemble de sommets de cardinalité minimum représentant un code r -identifiant du graphe ?

Comme données, nous avons un graphe identifiable et dans notre cas on aura besoin d'une certaine matrice.

Les auteurs ont défini un code identifiant comme suit :

Étant donné un graphe orienté $G = (V, E)$, d'ordre n , modélisant un système de diagnostic de pannes, où les sommets représentent les composants du système et la propagation de failles entre deux composants est représentée par un arc. C'est à dire qu'un composant peut devenir la source de la faille qui peut se propager à d'autres composants. Il s'agit alors de placer un nombre minimum d'alarmes pour détecter la source de la faille.

Soit u un sommet de V et $T(u) = \{u\} \cup \{v \mid \text{il existe un chemin allant de } u \text{ à } v\}$. Un code identifiant C dans G est tel que

$$(T(u) \cap C) \neq \emptyset \text{ et}$$

$$(T(u) \cap C) \Delta (T(v) \cap C) \neq \emptyset, \text{ pour tout } u \neq v$$

Un tel code n'existe que si le graphe considéré est acyclique (ne contient pas de cycle) [68].

3.2.2 Identification des variables de décision

Soit X un sous-ensemble de sommets de V . On note \tilde{X} le vecteur caractéristique $(x_1, x_2, \dots, x_n)^t$, où

$$x_i = \begin{cases} 1, & \text{si le sommet } i \in X, \\ 0 & \text{sinon.} \end{cases}$$

Dans le cas du problème de code identifiant nous nous demandons pour un sommet donné, s'il fait partie du code ou non.

Si C est un code identifiant auquel on associe le vecteur caractéristique $(c_1, c_2, \dots, c_n)^t$, alors les variables c_i , $i \in \{1, 2, \dots, n\}$, représentent les variables de décision de notre modèle indiquant si le sommet i de V est un mot de code ou non.

3.2.3 Fonction objectif

L'objectif est de choisir un sous-ensemble C minimum de sommets de sorte qu'à chaque sommet du graphe on lui associe un sous-ensemble de C unique et non vide.

En d'autre terme, il s'agit de choisir un sous ensemble de cardinalité minimum C , ce qui revient à minimiser :

$$\sum_{i=1}^n c_i \tag{3.1}$$

De manière plus générale, la fonction objectif s'écrit comme suit :

$$\sum_{i=1}^n f_i c_i \tag{3.2}$$

où f_i représente le coût lié au choix du sommet i comme mot de code.

3.2.4 Contraintes

Quand on fait le choix du sous-ensemble de sommets, ce dernier doit satisfaire les conditions (contraintes) suivantes :

- la couverture de tous les sommets
- la séparation.

les deux contraintes ci-dessus seront redéfinies en utilisant la notation vectorielle :

Soit $R = [r_{ij}]_{n \times n}$ la matrice définie comme suit :

$$r_{ij} = \begin{cases} 1 & \text{si le sommet } j \text{ peut être atteint en partant du sommet } i, \\ 0 & \text{sinon.} \end{cases}$$

Notons par \vec{v} un vecteur colonne constant, par exemple :

$$\vec{1} = \underbrace{(1, 1, \dots, 1)}_{n \text{ composantes}}$$

Etant donné deux vecteurs X et Y de $\{0, 1\}^n$, alors

$$X \neq Y \Leftrightarrow (|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|)^t \neq \vec{0}$$

et

$$X \neq \vec{0} \Leftrightarrow \sum_{k=1}^n x_k \geq 1$$

L'ensemble identifiant de C d'un sommet x_i est donné par $I_1^-(x_i, C) = (r_{i1}c_1, r_{i2}c_2, \dots, r_{in}c_n)^t$.

Ainsi

$$\begin{aligned} I_1^-(x_i, C) \neq I_1^-(x_j, C) &\Leftrightarrow (r_{i1}c_1, r_{i2}c_2, \dots, r_{in}c_n) \neq (r_{j1}c_1, r_{j2}c_2, \dots, r_{jn}c_n) \\ &\Leftrightarrow \sum_{k=1}^n |r_{ik} - r_{jk}|c_k \geq 1 \end{aligned} \quad (3.3)$$

Par conséquent, dans le programme linéaire on aura $m = n(n-1)/2$ inégalité de type (3.3).

Théorème 23 ([68]) *Soit $L = [l_{ij}]_{m \times n}$ une matrice telle que*

$$l_{ij} = \sum_{\substack{k=1 \\ i \neq j}}^n |r_{ik} - r_{jk}|.$$

Ainsi, C est un code identifiant si et seulement si

$$\begin{cases} RC & \geq \vec{1} & (3.4) \\ LC & \geq \vec{1} & (3.5) \\ C \in \{0, 1\}^n \end{cases}$$

L'inégalité 3.4 assurera que $I_1^-(x_i, C) \neq \emptyset$ pour tout sommet $x_i \in V$, et l'inégalité 3.5 assurera que tout les sommets soit séparés.

Pour déterminer le problème d'optimisation associé au code C on associe à chaque sommet x_i du graphe un coût f_i . Notons par $\vec{F} = (f_1, f_2, \dots, f_n)$ le vecteur coût, alors le PLNE associé au problème de recherche de code identifiant minimum dans un graphe orienté est

$$(PLNE) \begin{cases} \min & FC \\ \text{sous les contraintes.} & LC \geq \vec{1} \\ & RC \geq \vec{1} \\ & c_i \in \{0, 1\} \forall 1 \leq i \leq n \end{cases}$$

3.2.5 Exemple

Étant donné le graphe illustré dans la figure 3.5-a

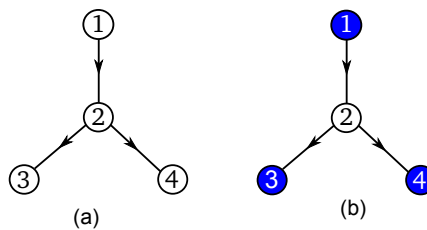


Figure 3.5 – Exemple de code identifiant obtenu en résolvant le programme linéaire associé au graphe donné dans (a)

La matrice R associé à ce graphe est

$$R_{4 \times 4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Par conséquent la matrice L est égale

$$L_{6 \times 4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

La résolution du programme linéaire en nombres entiers suivant :

$$\left\{ \begin{array}{ll} \min & \sum_{i=1}^4 c_i \\ \text{sc.} & LC \geq \vec{1} \\ & RC \geq \vec{1} \\ & c_i \in \{0, 1\} \forall 1 \leq i \leq 4 \end{array} \right.$$

nous donne le vecteur solution $C = (1, 0, 1, 1)$ qui correspond à l'ensemble des sommets $\{1, 3, 4\}$ (voir la figure 3.5-b).

Remarque 9 *Dans cet exemple, nous avons pour vecteur coût le vecteur $F = (1, 1, 1, 1)$, ce qui veut dire que tous les sommets du graphe sont sujets au même coût.*

Chapitre 4

Codes Identifiants dans les Chemins et les Circuits

L'expérience est une lanterne accrochée
dans le dos qui éclaire le passé.

CONFICIUS.

Dans ce chapitre nous présentons les premiers résultats obtenus dans le cas des chemins et des circuits.

Nous énonçons les différents résultats ainsi que les démonstrations.

Pour les graphes orientés, un sous-ensemble $C \subseteq V$ est un code r -identifiant si et seulement si :

- i. $I_r^-(i) \neq \emptyset$, pour tout sommet $i \in V$
- ii. $I_r^-(i) \Delta I_r^-(j) \neq \emptyset$ pour toute paire de sommets (i, j) , $i \neq j$.

où $I_r^-(i) = \Gamma_r^-[i] \cap C$ Dans le cas d'un chemin nous avons :

$$I_r^-(i) \Delta I_r^-(i+1) = \{i-r, i-r+1\}$$

Ainsi, on remarque que pour ce type de graphes, un code identifiant doit nécessairement couvrir tous les sommets et séparer toute paire de sommets consécutifs.

En exploitant cette propriété, nous avons pu obtenir les premiers résultats sur la cardinalité minimum de codes 1 et 2-identifiant.

4.1 Code identifiant dans les chemins

Soit $\mathcal{P}_n = \{x_0, x_1, \dots, x_n\}$ un chemin de longueur n . Nous allons d'abord étudier le cas d'un code 1-identifiant puis le cas 2-identifiant.

4.1.1 Code identifiant

Lemme 3 *L'ensemble C est un code 1-identifiant de \mathcal{P}_n si et seulement si :*

1. Les deux sommets x_0 et x_1 sont des mots du code C ,
2. Pour toute paire de sommets consécutifs, x_i et x_{i+1} , $i \in \{2, 3, \dots, n-1\}$, x_i ou x_{i+1} est un mot du code C ,
3. Pour tout triplet de sommets consécutifs, x_i, x_{i+1} et x_{i+2} , $i \in \{0, 3, \dots, n-2\}$, on a $x_i \in C$ ou $x_{i+2} \in C$.

Démonstration. Pour (1), si $x_0 \notin C$ alors $\mathcal{I}^-(x_0) = \emptyset$, c'est à dire que x_0 ne sera pas couvert par le code. Donc $x_0 \in C$. Si $x_1 \notin C$ alors $\mathcal{I}^-(x_0) = \mathcal{I}^-(x_1) = \{x_0\}$, c'est-à-dire que x_0 et x_1 ne seront pas séparés. Donc $x_1 \in C$.

Pour (2), supposons que $x_i \notin C$ et que $x_{i+1} \notin C$. Alors $\mathcal{I}^-(x_{i+1}) = \emptyset$. Donc $x_i \in C$ ou $x_{i+1} \in C$.

Pour (3) supposons qu'aucun des deux sommets x_i et x_{i+2} n'appartient au code. Alors deux cas se présentent (voir la figure 4.1) :

cas 1 : Si $x_{i+1} \in C$, alors nous aurons $\mathcal{I}^-(x_{i+1}) = \mathcal{I}^-(x_{i+2}) = \{x_{i+1}\}$, c'est-à-dire que les deux sommets x_{i+1} et x_{i+2} ne seront pas séparés.

Cas 2 : Si $x_{i+1} \notin C$, forcément les deux sommets x_{i+1} et x_{i+2} ne seront pas couverts puisque $\mathcal{I}^-(x_{i+1}) = \mathcal{I}^-(x_{i+2}) = \emptyset$.

Donc, dans les deux cas l'un des deux sommets x_i et x_{i+2} doit appartenir au code.

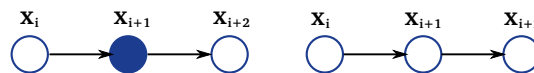


Figure 4.1 – Dans la figure de gauche on peut remarquer que $\mathcal{I}^-(x_{i+1}) = \mathcal{I}^-(x_{i+2}) = x_{i+1}$, et dans le figure de droite on peut voir que $\mathcal{I}^-(x_{i+1}) = \mathcal{I}^-(x_{i+2}) = \emptyset$

On remarque que les conditions sont nécessaires et suffisantes pour la couverture de tous les sommets de \mathcal{P}_n , car chaque demi-boule¹ est constitué de deux sommets consécutifs.

Montrons que (1), (2) et (3) sont suffisantes pour la séparation.

Soient x_i et x_j deux sommets. Alors deux cas se présentent :

Cas 1 : Les deux sommets sont voisins, sans perdre de généralité, on prend $j = i + 1$.

Nous avons vu que la condition (1) permet de séparer les deux sommets x_0 et x_1 .

Pour les autres paires de sommets consécutifs; en vertu de la condition (3), on sait

1. La demi-boule de rayon r et de centre v représente l'ensemble $\Gamma_r^-[v]$

que $x_{i-1} \in C$ ou $x_{i+1} \in C$, ainsi nous aurons $\mathcal{I}^-(x_i) \neq \mathcal{I}^-(x_j)$. Donc x_i et x_j seront séparés.

Cas 2 : x_i et x_j ne sont pas voisins, en d'autre terme, la distance $d(x_i, x_j) \geq 2$.

Supposons, sans perdre de généralité, que $j = i + 2$. Ainsi, nous avons $\Gamma_1^-[x_i] = \{x_{i-1}, x_i\}$ et $\Gamma_1^-[x_j] = \{x_{i+1}, x_{i+2}\}$, or en vertu de la condition (2), nous aurons $\mathcal{I}^-(x_i) \neq \mathcal{I}^-(x_j)$. Donc x_i et x_j seront séparés.

D'où la suffisance des conditions (1), (2) et (3) pour la séparation.

□

De ce qui précède, on peut déduire que pour chaque trois sommets consécutifs au moins deux sont des mots du code. Alors, si on note par d_1^* la densité optimale d'un code 1-identifiant dans \mathcal{P}_∞ , on a

Théorème 24 Soit C un code identifiant dans \mathcal{P}_∞ , alors :

$$d^*(\mathcal{P}_\infty) = \frac{2}{3}$$

Démonstration. Il suffit de remarquer que ce résultat est une conséquence directe des conditions (2) et (3) du lemme 3, donc on a $d^*(\mathcal{P}_n) \geq \frac{2}{3}$. Et pour conclure il suffit d'exhiber un code ayant cette densité (voir la figure 4.2).

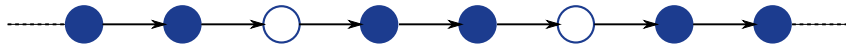


Figure 4.2 – Code 1-identifiant de densité optimale égale à $\frac{2}{3}$

□

A présent, en se basant sur la propriété de couverture d'ensemble, nous allons donner la cardinalité minimum d'un code 1-identifiant dans un chemin, \mathcal{P}_n , de longueur finie.

Théorème 25 La cardinalité minimum d'un code identifiant, notée par $M_1^-(\mathcal{P}_n)$ dans un chemin orienté de longueur $n - 1$ (possédant n sommets) est donnée par :

$$M_1^-(\mathcal{P}_n) = \begin{cases} 2p & \text{Si } n=3p \\ 2p + 1 & \text{Si } n=3p+1 \\ 2p + 2 & \text{Si } n=3p+2 \end{cases}$$

Démonstration. Soit V l'ensemble des sommets de \mathcal{P}_n . Notons par L , l'ensemble des sommets qui sont identifiés par un seul mot de code (couvert par un seul mot du code). Alors on aura $|V| - |L|$ dont leurs ensembles identifiants sont de cardinalité égale à deux. En d'autre terme C couvre de manière double ces sommets. Ainsi, en utilisant le fait que $|L| \leq |C|$ (au plus $|C|$ sont couvert par un seul mot du code), alors

$$2(|V| - |L|) + |L| \leq \sum_{x_i \in C} |\Gamma_1^-[x_i]| \leq 2 \cdot |C|$$

d'où

$$\begin{aligned} 2|V| - |L| \leq 2|C| &\Leftrightarrow 2|V| - |C| \leq 2|C| \\ &\Leftrightarrow \frac{2}{3}|V| \leq |C| \end{aligned}$$

Donc

$$|C| \geq \left\lceil \frac{2n}{3} \right\rceil$$

Posons $n = 3p + q$, $q \in \{0, 1, 2\}$. Ainsi on aura :

$$\left\lceil \frac{2n}{3} \right\rceil = \left\lceil \frac{2(3p + q)}{3} \right\rceil = 2p + \left\lceil \frac{2q}{3} \right\rceil$$

Donc : Si $q = 0$, alors $\lceil \frac{2q}{3} \rceil = 0$. Si $q = 1$, alors $\lceil \frac{2q}{3} \rceil = 1$. Et Si $q = 2$, alors $\lceil \frac{2q}{3} \rceil = 2$.

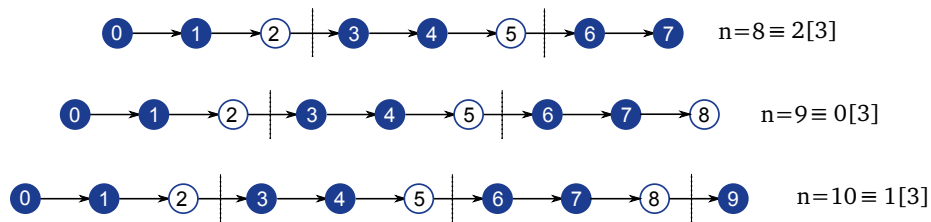


Figure 4.3 – Exemple de code identifiant dans des chemins de longueurs égales à 7,8,9

Pour conclure on exhibe un code identifiant atteignant la borne pour chaque cas. Ainsi, nous prendrons $C = \{x_i | i \equiv 0[3] \text{ et } i \equiv 1[3]\}$ pour tous les cas (voir la figure 4.3).

□

4.1.2 Code 2-identifiant

Dans la preuve de notre théorème nous avons besoin du résultat suivant : Soit $\mathcal{P}_n = \{x_0, x_1, \dots, x_n\}$ un chemin de longueur n . Et C un code dans \mathcal{P}_n .

Lemme 4 Soit C un sous-ensemble de sommets de \mathcal{P}_n . On dit que C est un code 2-identifiant dans \mathcal{P}_n si et seulement si :

1. Les trois sommets x_0, x_1 et x_2 appartiennent au code,
2. Pour tout triplet de sommets consécutifs, $x_i, x_{i+1}, x_{i+2}, i \in \{3, 4, \dots, n-2\}$, au moins un appartient au code,
3. Pour tout quadruplet de sommets consécutifs, $x_i, x_{i+1}, x_{i+2}, x_{i+3}, i \in \{3, 4, \dots, n-3\}$, on ne peut avoir $x_i \notin C$ et $x_{i+3} \notin C$.

Démonstration. Pour la condition (1), si $x_0 \notin C$, alors $\Gamma_2^-[x_0] = \emptyset$ (le sommet ne sera pas couvert). Donc $x_0 \in C$.

Supposons que $x_1 \notin C$ alors $\mathcal{I}_2^-(x_0) = \mathcal{I}_2^-(x_1)$, donc les sommets x_0 et x_1 ne sont pas séparés ce qui contredit le fait que C soit un code identifiant. Donc $x_1 \in C$.

Enfin, supposons que $x_2 \notin C$, alors les deux sommets x_1 et x_2 ne sont pas séparés ($\mathcal{I}_2^-(x_1) = \mathcal{I}_2^-(x_2)$). Donc $x_2 \in C$.

Pour la condition (2), si aucun des trois sommets x_i, x_{i+1}, x_{i+2} n'appartient au code alors $\mathcal{I}^-(x_{i+2}) = \emptyset$ ce qui contredit le fait que le code C est couvrant.

Pour (3) supposons qu'aucun des deux sommets x_i et x_{i+3} n'est dans C alors les sommets x_{i+2} et x_{i+3} ne seront pas séparés du fait que $\mathcal{I}_2^-(x_{i+2}) = \mathcal{I}_2^-(x_{i+3}) = \{x_{i+1}, x_{i+2}\}$. donc $x_i \in C$ ou $x_{i+3} \in C, i \in \{3, 4, \dots, n-2\}$.

On remarque que les conditions (1) et (2) sont nécessaires et suffisantes pour que $\Gamma_2^-[x_i] \cap C \neq \emptyset$ pour tout $i \in \{0, 1, \dots, n\}$.

Montrons que les conditions (1), (2) et (3) sont suffisantes pour la séparation. Soient x_i et x_j deux sommets distincts. Ainsi, deux cas se présentent :

Cas 1 Les deux sommets sont voisins, $j = i + 1$. Dans ce cas, en vertu de (1) nous avons les paires $(x_0, x_1), (x_1, x_2), (x_2, x_3)$ qui sont séparés, et en vertu de (2), nous avons $x_{i-3} \in C$ et $x_i \in C$ pour tout paire (x_i, x_{i+1}) , où $i \in \{3, 4, \dots, n-1\}$. Ainsi nous aurons $\mathcal{I}_2^-(x_i) \neq \mathcal{I}_2^-(x_{i+1})$ pour toute les paires de sommets consécutifs. Donc les deux sommets x_i et x_j sont séparés par le code C .

Cas 2 x_i et x_j sont à distance au moins égale à 2, c'est-à-dire $d(x_i, x_j) \geq 2$. Dans ce cas, si $d(x_i, x_j) > 2$ ($j > i + 2$), alors en vertu de (2) nous aurons $\mathcal{I}_2^-(x_i) \neq \mathcal{I}_2^-(x_j)$, et si $d(x_i, x_j) = 2$ en vertu des conditions (2) et (3) nous aurons aussi $\mathcal{I}_2^-(x_i) \neq \mathcal{I}_2^-(x_j)$. Donc les sommets x_i et x_j sont séparés par le code C .

□

Pour une question d'ambéguïté nous avons jugé plus convenable de noter tout sommet x_i par son indice i .

Nous savons que tout code 2-identifiant est 2-séparateur dans \mathcal{P}_n . Or pour tout sommet

$i \in \mathcal{P}_n$ nous avons $\Gamma_2^-[i]\Delta\Gamma_2^-[i+1] = \{i-2, i+1\}$, alors $\forall i \in \mathcal{P}_n$ l'un des sommets $i-2$ et $i+1$ doit appartenir à C (condition (3) du lemme 4). De ce fait on aura $i-2 \in C$ ou $i+1 \in C$ pour tout sommet $i \in \{2, n-1\}$ de \mathcal{P}_n (voir la figure 4.4). On appellera alors cette disjonction une *contrainte élémentaire* (CE) et on l'abrégera dans la suite par $i-2 \vee i+1$.

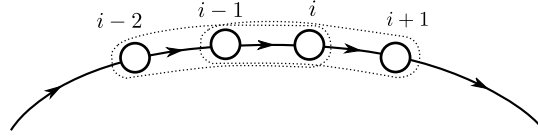


Figure 4.4 – Un des deux sommets i ou $i+3$ appartient au code pour séparer les deux sommets $i+2$ $i+3$

Nous allons présenter un exemple qui nous permettra d'éclaircir quelques notations que nous allons utiliser dans les deux prochains paragraphes (étude de code 2-identifiants dans le cas de chemin et de circuit).

Exemple 2 *Etant donné un chemin, $\mathcal{P}_{10} = x_0, x_1, \dots, x_9$, de longueur 9. Ainsi, pour ce chemin on aura neuf paires de sommets consécutifs qu'on doit séparer, puisque tout code 2-identifiant doit 2-séparer chaque deux sommets consécutifs. Donc on aura neuf contraintes élémentaires (CEs) à satisfaire que nous énumérons comme suit : $0 \vee 3, 3 \vee 6, 6 \vee 9, 1 \vee 4, 4 \vee 7, 2 \vee 5, 5 \vee 8$, notons que nous avons omi les deux CEs relatives à la séparation des paires $(0,1)$ et $(1,2)$, puisque, en vertu de la condition (1) du lemme 4, elles sont séparées. L'ensemble de ces contrainte sera appelé *contrainte générale* (CG). Partionnant cette ensemble de contraintes élémentaires (ou la contrainte générale) en trois sous-ensembles de contraintes, que nous appellerons *contraintes partielles* (CP). Ainsi, nous avons*

$$\begin{aligned} &0 \vee 3, 3 \vee 6, 6 \vee 9 \\ &1 \vee 4, 4 \vee 7 \\ &2 \vee 5, 5 \vee 8 \end{aligned}$$

Cependant, une adaptation de la notation est nécessaire, dans le but d'en tirer la formulation pour le cas général. Ainsi, ces contraintes partielles peuvent s'écrire comme suit :

$$\begin{aligned} &0 \vee 0 + 1 \times 3, 0 + 1 \times 3 \vee 0 + 2 \times 3, 0 + 2 \times 3 \vee 0 + 3 \times 3 \quad (c_0) \\ &1 \vee 1 + 1 \times 3, 1 + 1 \times 3 \vee 1 + 2 \times 3 \quad (c_1) \\ &2 \vee 2 + 1 \times 3, 2 + 1 \times 3 \vee 2 + 2 \times 3 \quad (c_2) \end{aligned}$$

On appellera c_i , $i = 0, 1, 2$, la contrainte partielle i .

Dans le cas générale, si le nombre de sommets du chemin considéré est n , alors posons $n = 3p + q$, avec $p \in \{1, 2, \dots, \lfloor \frac{n}{3} \rfloor\}$ et $q \in \{0, 1, 2\}$. On appellera la contrainte partielle i , la contrainte donnée par :

$$i \vee i + 1 \times 3, i + 1 \times 3 \vee i + 2 \times 3, \dots, i + (s_i - 1) \times 3 \vee i + s_i \times 3$$

où s_i est le plus grand entier pour lequel l'inégalité suivante est vérifiée :

$$i + s_i \times 3 \leq n$$

Dans l'exemple précédent nous avons, pour la contrainte partielle c_0 , $s_0 = 3$.

Notons par V_i l'ensemble des sommets de la contrainte partielle i . Nous remarquons que $V_i \cap V_j = \emptyset$, pour tout $i \neq j | i, j \in \{0, 1, 2\}$. En d'autre terme, toutes les contraintes partielles sont indépendantes les unes des autres.

Ainsi, satisfaire la contrainte générale, pour obtenir un code 2-identifiant, revient à satisfaire toutes les contraintes partielles.

En utilisant cette notation, nous aboutissons au résultat suivant :

Théorème 26 Soit $n = 3p + q$, où $q \in \{0, 1, 2\}$ et C un code 2-identifiant d'un chemin \mathcal{P}_n , et $M_2^-(\mathcal{P}_n)$ sa cardinalité minimum. Alors :

$$\begin{aligned} \text{(1) Si } q = 0, p \geq 1, M_2^-(\mathcal{P}_n) &= \begin{cases} \frac{3p}{2} + 1 & \text{si } p \text{ est pair} \\ \frac{3(p+1)}{2} & \text{si } p \text{ est impair} \end{cases} \\ \text{(2) Si } q = 1, p \geq 1, M_2^-(\mathcal{P}_n) &= \begin{cases} \frac{3p}{2} + 2 & \text{si } p \text{ est pair} \\ \frac{3(p+1)}{2} & \text{si } p \text{ est impair} \end{cases} \\ \text{(3) Si } q = 2, p \geq 1, M_2^-(\mathcal{P}_n) &= \begin{cases} \frac{3p}{2} + 2 & \text{si } p \text{ est pair} \\ \frac{3(p+1)}{2} + 1 & \text{si } p \text{ est impair} \end{cases} \end{aligned}$$

Démonstration. Pour le premier cas, c'est à dire $p \geq 1$ et $q = 0$, en vertu de la condition (1) du lemme 4, on sait, d'un côté, que les sommets 0, 1 et 2 sont du code, ce qui veut dire que la première CE des CPs 0, 1 et 2 est satisfaite.

D'un autre côté et en vertu de la condition (2) du lemme 4 on aura forcément un mot du code parmi les sommets 3, 4 et 5, donc on aura une CE parmi les CPs 0, 1 et 2 dont les deux sommets sont du code. Sans perte de généralité, soit 3 ce sommet, alors on aura deux CEs de la CP 0 satisfaites, dans ce cas il reste $(p - 3)$ CEs non encore satisfaites, alors nous aurons besoin d'au moins $\lceil \frac{p-3}{2} \rceil$ mots du code pour satisfaire les CEs restantes de CP 0.

Par ailleurs, pour les CPs 1 et 2 on a qu'une seule contrainte qui est satisfaire (puisque $1, 2 \in C$), alors pour chacune des CP il reste $(p - 2)$ CEs à satisfaire, d'où la nécessité

d'avoir au moins $\lceil \frac{p-2}{2} \rceil$ mots du code pour satisfaire les CPs 1 et 2.

On conclut qu'au total il faut au moins :

$$4 + 2 \left\lceil \frac{p-2}{2} \right\rceil + \left\lceil \frac{p-3}{2} \right\rceil$$

mots du code pour satisfaire la contrainte générale.

Si p est pair, alors $M_2^-(\mathcal{P}_n) \geq \frac{3p}{2} + 1$. Sinon, $M_2^-(\mathcal{P}_n) \geq \frac{3(p+1)}{2}$.

Il ne reste qu'à exhiber un code atteignant cette borne pour conclure. En effet, en utilisant la construction suivante, on atteint cette borne :

On prend tout les sommets $i \in V$, tel que i est pair en ajoutant les sommets 0 et 1.



Figure 4.5 – Code 2-identifiant $C = \{2, 4, 6, 8\} \cup \{0, 1\}$ pour un chemin de longueur égale à 8 ($n = 3 \times 3 + 0$ sommets)

Pour le cas où $p \geq 1$ et $q = 1$, le raisonnement est similaire. En effet, concernant la première CE de la première CP on aura $(p-1)$ CEs à satisfaire puisque $0 \in C$. Ainsi, on aura besoin d'au moins $\lceil \frac{p-1}{2} \rceil$ mots du code. Pour les CPs 1 et 2 on aura besoin, respectivement, d'au moins $\lceil \frac{p-3}{2} \rceil$ et $\lceil \frac{p-2}{2} \rceil$ mots du code pour satisfaire les contraintes élémentaires restantes. Au total on aura besoin d'au moins :

$$4 + \left\lceil \frac{p-3}{2} \right\rceil + \left\lceil \frac{p-2}{2} \right\rceil + \left\lceil \frac{p-1}{2} \right\rceil$$

mots du code.

Ainsi, si p est pair, alors $M_2^-(\mathcal{P}_n) \geq \frac{3p}{2} + 2$. Sinon, $M_2^-(\mathcal{P}_n) \geq \frac{3(p+1)}{2}$.



Figure 4.6 – Code 2-identifiant dans un chemin ayant $n = 3 \times 3 + 1$ sommets ($p=3, q=1$)

Pour conclure nous exhibons un code atteignant cette borne. En effet, nous avons le code $C = \{i | i \text{ pair}\} \cup \{0, 1\}$ atteignant cette borne (voir la figure 4.6).

Enfin dans le dernier cas, c'est-à-dire $p \geq 1$ et $q = 2$, par un un raisonnement analogue, on aura $(p-1)$ contraintes élémentaires à satisfaire pour les CPs 0 et 1, et on aura besoin respectivement, d'au moins $\lceil \frac{p-1}{2} \rceil$ et $\lceil \frac{p-2}{2} \rceil$ mots de code. Et pour la CP 2 on a $(p-1)$ CE

et puisque $2 \in C$ il reste $(p - 2)$ CEs à satisfaire, donc au moins $\lceil \frac{p-2}{2} \rceil$ mots du code. Ainsi, au total on aura besoin de :

$$4 + 2 \left\lceil \frac{p-2}{2} \right\rceil + \left\lceil \frac{p-1}{2} \right\rceil$$

mots du code.

Si p est pair, alors $M_2^-(\mathcal{P}_n) \geq \frac{3p}{2} + 2$. Et si p est impair, alors $M_2^-(\mathcal{P}_n) \geq \frac{3(p+1)}{2}$. Pour conclure il suffit de considérer la même construction que dans les deux cas précédents pour exhiber un code 2-identifiant atteignant cette borne.



Figure 4.7 – Exemple de code 2-identifiant dans un chemin ayant $3 \times 3 + 2 = 11$ sommets ($p=3, q=2$)

□

4.2 Code identifiant dans les circuits

Dans le cas d'un circuit, on a établi des résultats optimaux dans le cas de codes 2-identifiants.

4.2.1 Code 2-identifiant

Dans le cas d'un circuit, les deux conditions du lemme 4 restent valides. Ainsi :

Lemme 5 Soit $\mathcal{CI}_n = \{1, 2, \dots, n, 1\}$ un circuit de longueur n . On dit que C est un code 2-identifiant pour \mathcal{CI}_n si et seulement si les deux conditions suivantes sont vérifiées :

1. Pour tout triplet de sommets consécutifs x_i, x_{i+1} et x_{i+2} au moins un appartient au code,
2. Pour tout quadruplet de sommets consécutifs x_i, \dots, x_{i+3} on ne peut avoir $x_i \notin C$ et $x_{i+3} \notin C$.

Démonstration. La preuve est similaire au lemme 4 dans le cas d'un chemin orienté. En ajoutant la condition que la distance séparant les deux sommets i et j n'est pas plus grande que celle séparant j de i pour montrer la suffisance et la nécessité des conditions (1),(2) et (3). □

Pour le cas d'un circuit, bien que le raisonnement est analogue à celui d'un chemin, il existe, cependant, quelques différences au niveau de la structure et de la formulation. Ainsi, on définit une contrainte partielle i ($i \in \{1, 2, 3\}$) par :

$$i \vee i + 1 \times 3, i + 1 \times 3 \vee i + 2 \times 3, \dots, i + (s_i - 1) \times 3 \vee i + s_i \times 3, i + s_i \times 3 \vee h_i$$

où s_i représente le plus grand entier tel que :

$$i + s_i \times 3 \leq n$$

et h_i est tel que $i + (s_i + 1) \times 3 \equiv h_i \pmod{[n]}$, (ie $h_i \in \{1, 2, 3\}$).

Exemple 3 Soit $\mathcal{C}_n = \{1, 2, \dots, n, 1\}$ un circuit de longueur n . Supposons que $n=10$, ainsi $p = 3$ et $q = 1$. Alors les contraintes partielles (CPs), 1, 2 et 3, seront données comme suit :

$$1 \vee 4, 4 \vee 7, 7 \vee 10, 10 \vee 3, \quad (i = 1)$$

$$2 \vee 5, 5 \vee 8, 8 \vee 1 \quad (i = 2)$$

$$3 \vee 6, 6 \vee 9, 9 \vee 2 \quad (i = 3)$$

Notons par $i|j$ la contrainte élémentaire $i \vee j$. Alors la contrainte générale (CG) sera de la forme :

$$1|4|7|10|3|6|9|2|5|8|1$$

Si, par exemple, $n = 12$, on aura alors une CG de la forme :

$$1|4|7|10|1, 2|5|8|11|2, 3|6|9|12|3$$

Pour le cas d'un circuit, nous avons pu déterminer la cardinalité minimum d'un code 2-identifiant. Ce résultat est donnée par le théorème suivant :

Théorème 27 Pour un circuit de longueur n on a :

- i. Un chemin de longueur 3 n'admet pas de code 2-identifiant,
- ii. $M_2^-(\mathcal{C}_4) = 3$,
- iii. $M_2^-(\mathcal{C}_{2k}) = k$, pour tout $k \geq 3$,
- iv. $M_2^-(\mathcal{C}_{2k+1}) = k + 2$, pour tout $k \geq 2$

Démonstration. Pour (i), il est clair que pour un chemin de longueur 3 on ne peut obtenir un code 2-identifiant du fait de la présence de sommets jumeaux.

Pour (ii), nous montrons qu'il ne peut y'avoir de code 2-identifiant de cardinalité deux dans un circuit de longueur 4. En effet, supposons qu'on a deux sommets seulement appartenant au code. Sans perdre de généralité, soient x_1 et x_3 ces deux sommets, alors $I_2^-(x_1) = I_2^-(x_3) = \{x_1, x_3\}$. Donc au moins 3 sommets sont du code et il suffit d'exhiber un code ayant une cardinalité égale à 3 pour conclure (voir la figure 4.8).

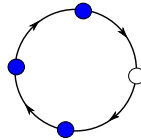


Figure 4.8 – Exemple de code 2-identifiant dans C_4

Concernant (iii), c'est-à-dire le cas où la longueur du circuit est paire ($n = 2k$), on sait qu'on a $\frac{n}{2} = k$ contraintes élémentaires (CEs) à satisfaire donc on aura besoin d'au moins k mots de code.

Il suffit alors d'exhiber un code 2-identifiant de cardinalité égale à k pour conclure. Or on peut prendre comme code l'ensemble $C = \{i | i \text{ pair}, 1 \leq i \leq n\}$ (voir la figure 4.9) d'où le résultat.

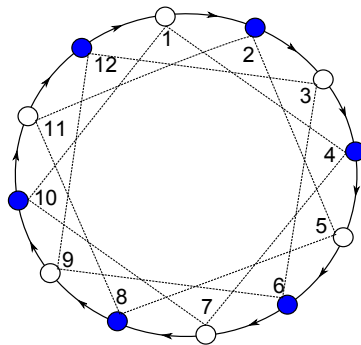


Figure 4.9 – Exemple de code 2-identifiant dans un circuit de longueur 12. On voit que dans cette exemple qu'il suffit de 6 mots du code pour couvrir et séparer tous les sommets du circuit.

Pour les mêmes raisons que pour le cas pair, lorsque $n = 2k + 1$ on aura besoin d'au moins $k + 1$ mots de code pour satisfaire toutes les CEs.

Ainsi, on aura au moins une CE dont les deux sommets sont du code. On veut montrer que $|C| > k + 1$. Pour ce faire on supposera qu'on peut trouver un code 2-identifiant C

de cardinalité égale à $k + 1$ pour un circuit de longueur n et dont les sommets sont dans $\{1, 2, \dots, n\}$, et on aboutira à une contradiction.

Puisqu'on a $k + 1$ sommets qui sont du code, alors forcément on a deux sommets adjacents qui sont du code. Sans perte de généralité soient 1 et 2 ces deux sommets or l'un des deux sommets n et 3 doit nécessairement appartenir au code en vertu de la condition (2) du lemme 5. Ainsi, pour tout code 2-identifiant on aura au moins trois sommets consécutifs qui sont du code.

Deux cas se présentent :

Cas 1 : prenons le cas où la longueur du circuit est $n = 4p + 3$ ($k = 2p + 1$). Puisqu'au moins trois sommets consécutifs sont du code (d'après ce qui a été dit précédemment), alors il nous reste encore $4p$ sommets à couvrir et à séparer. Or en vertu des conditions (1) et (2) du lemme 5, on sait que pour chaque 4 sommets consécutifs, au moins deux sont du code. Ainsi on aura besoin d'au moins $2p$ sommets qui sont du code, donc au total $2p + 3 = k + 2$ sommets sont du code (voir l'exemple de la figure 4.10). D'où la contradiction.

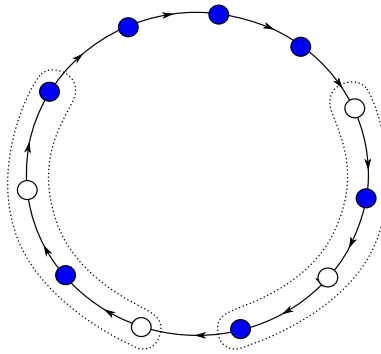


Figure 4.10 – Exemple de code 2-identifiant de cardinalité optimale égale à $2 \times 2 + 3$ dans un circuit de longueur $4 \times 2 + 3$

Cas 2 : pour $n = 4p + 1$ ($k = 2p$). En observant attentivement ce cas, on verra qu'il s'agit d'un cas similaire. Nous avons $n = 4p + 1 = [4(p - 1) + 3] + 2$ ($k = 2p$). Ainsi nous aurons $2(p - 1) + 3 = k + 1$ sommets du code parmi les $4(p - 1) + 3$ que nous avons (En vertu de la condition (2) du lemme 5) plus les 3 sommets consécutifs appartenant au code. Dans le cas où aucun des deux sommets restants ne sont du code, sans perdre de généralité disant $n - 1$ et $n - 2$, alors on aura nécessairement les trois sommets $n - 3, n - 4$ et $n - 5$ appartenant au code (voir la représentation de gauche de la figure 4.11). De ce fait nous avons $4(p - 2)$ sommets qui sont couverts et séparés par $2(p - 2)$ mots de code plus les sommets $n - 3, n - 4$ et $n - 5$ et les trois sommets consécutifs. Ce qui fera au total $2(p - 2) + 6 = 2p + 2 = k + 2$ mot de code.

Si, l'un des deux sommets $n - 1$ ou $n - 2$ appartient au code alors, nous aurons ce dernier plus les $2(p - 1)$ mots de code pour couvrir et séparer les $4(p - 1)$ sommets (condi-

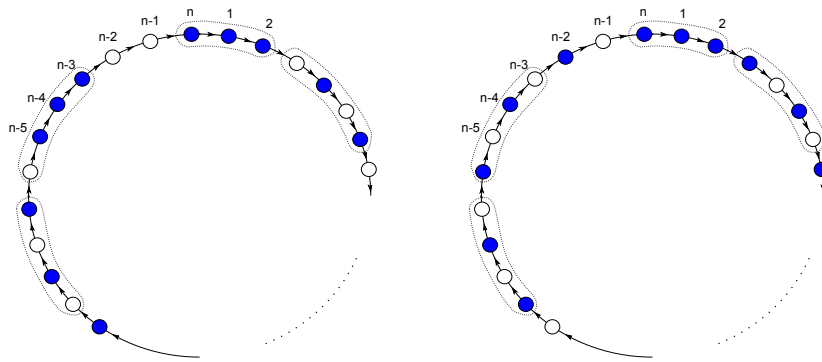


Figure 4.11 – Exemple de deux codes 2-identifiants de cardinalité minimum

tion (2) et (3) du lemme 5) plus les trois sommets consécutifs (voir la représentation de droite de la figure 4.11). Ainsi nous aurons un total de $1 + 2(p-1) + 3 = 2p + 2 = k + 2$ mots de code.

Donc, dans les deux cas, on aura au moins un total de $k + 2$ mots de code.

On conclut avec l'exhibition d'un code atteignant cette borne. Ce dernier est constitué de l'ensemble de sommets $C = \{2\} \cup \{i \equiv 1[2], 1 \leq i \leq n\}$.

□

Conclusion et perspectives

Au terme de ce mémoire, nous avons réalisé que beaucoup de travaux sur les codes identifiants ont été entrepris et ce pour différentes familles de graphes (une bibliographie de près de 180 articles est recensé sur le site d'Antoine Lobstein [2]). Cet intérêt, sans cesse croissant, est du à l'importance de cette notion et de ses applications pratiques, notamment dans les systèmes de détection de pannes dans les réseaux et de dangers dans des environnements hostiles.

Plusieurs variantes ont été étudiées dans, le plus souvent, des graphes réguliers comme les grilles et les hypercubes.

Les derniers travaux réalisés dans ce domaine intègrent, de plus en plus, des heuristiques pour la résolution du problème. Ceci s'explique par le fait que le problème de recherche de code identifiant est NP-difficile.

Nous avons abordé dans ce mémoire le problème de recherche de code identifiant de cardinalité minimum dans les graphes. La recherche bibliographique effectuée nous a permis d'approfondir nos connaissances dans cette thématique en plein essor.

Nous nous sommes fixé pour objectif, d'une part, d'analyser et de comprendre les différents travaux traitant le problème des codes identifiants, et d'autre part, d'essayer de répondre à la question suivante : Qu'elle est la cardinalité minimum d'un code identifiant dans les chemins et les circuits ?

En effet, peu de résultats ont été réalisés dans le cas des graphes orientés. Les seuls résultats existants sont l'algorithme linéaire récursif pour déterminer un code 1-identifiant dans un arbre orienté et la modélisation du problème des codes identifiants par un programme linéaire en nombre entiers.

En s'inspirant des travaux réalisés dans le cas de chaînes et de cycles, et en exploitant le fait qu'un code identifiant dans un chemin doit, à la fois, couvrir tous les sommets de ce dernier et séparer toute paire de sommets consécutifs, nous avons pu obtenir quelques résultats

dans les chemins et les circuits. Nous avons, ainsi, pu déterminer les cardinalités minimum de code 1 et 2-identifiants dans ces deux graphes. Tous les résultats ont été explicités dans le chapitre 4.

Perspectives

Les codes identifiants est un domaine relativement nouveau, ce qui nous laisse à penser que beaucoup de travaux peuvent être entrepris dans ce sens.

Plusieurs questions restent ouvertes notamment dans le cas des graphes réguliers (finis ou infinis) tels que les grilles et les hypercubes vu que dans la plupart des cas, les travaux n'ont abouti qu'à des bornes. En particulier dans les hypercubes, les derniers travaux ont permis d'obtenir des cardinalités exactes de codes r -identifiants ($r \leq 3$) pour seulement les n -hypercubes avec $n \leq 7$. Il y a encore moins de résultats concernant la variante la plus générale que sont les codes $(r, \leq p)$ -identifiants.

Comme perspectives, en plus d'une généralisation des résultats obtenus dans le cas de chemins et de circuits, nous pensons à développer des heuristiques, telles que la recherche tabou, les algorithmes génétiques,... pour la résolution des instances de grandes tailles.

Bibliographie

- [1] Alan Frieze, Ryan Martin, Julien Moncel, Miklos Ruszinko et Cliff Smyth. **Codes identifying Sets of Vertices in Random Networks.** *Discret Mathematics*, 307 :1094–1107, 2007.
- [2] Antoine Lobstein. <http://www.infres.enst.fr/lobstein/debutBIBidetlocdom.pdf>.
- [3] Béla Bollobàs. *Random Graphs*. Cambridge University Press, 2nd edition edition, 2001.
- [4] Claude Berge. *Graphes et hypergraphes*. DUNOD, 1990.
- [5] Daniel W. Cranston et Gexin Yu. **A New Lower Bound on the Density of Vertex Identifying Codes for the Infinite Hexagonal Grid.** *The Electronic Journal of Combinatorics*, #R113(16), 2009.
- [6] David L. Roberts et Fred S. Roberts. **Locating sensors in paths and cycles : The case of 2-identifying codes.** *European Journal of Combinatorics*, 29 :72–82, 2008.
- [7] Edgar G. Goodaire et Michael M. Parmenter. *Discret Mathematics with Graph Theory*. Pearson Prentice Hall, third edition, 2006.
- [8] Emmanuel Charbit, Irène Charon, Gérard Cohen et Olivier Hudry. **Discriminating codes in bipartite graphs.** *Electronic Notes in Discrete Mathematics*, 26 :29–35, 2006.
- [9] Emmanuel Charbit, Irène Charon, Gérard Cohen, Olivier Hudry et Antoine Lobstein. **Discriminating codes in bipartite graphs : Bounds, Extremal cardinalities, Complexity.** *Advances in Mathematics of communications*, 2(4) :403–420, 2008.
- [10] Frank Harary. *Graph theory*. Addison-Wesley, 1969.

-
- [11] Fred S. Roberts. *Graph Theory and Its Applications to Problems of Society*. Regional conference series in applied mathematics ; 29. SIAM, 1978.
- [12] Geoffrey Exoo, Tero Laihonen et Sanna Ranto. **Improved Upper bounds in Binary Identifying Codes**. *IEEE Transactions On Information Theory*, 53(11) :4255–4260, November 2007.
- [13] Geoffrey Exoo, Tero Laihonen et Sanna Ranto. **New Bounds on Binary Identifying Codes**. *Discret Applied Mathematics*, (156) :2250–2263, 2008.
- [14] Geoffrey Exoo, Tero Laihonen, Sanna Ranto et Ville junnila. **Improved Bounds on Identifying Codes in Binary Hamming Spaces**. *European Journal of Combinatorics*, 2009.
- [15] Geoffrey Exoo, Tero Laihonen, Sanna Ranto et Ville junnila. **Upper Bounds For Binary Identifying Codes**. *Advances in Applied Mathematics*, (42) :277–289, 2009.
- [16] Gérard Cohen, Liro Honkala, Antoine Lobstein et Gilles Zémor. **Bounds for Codes Identifying Vertices in the Hexagonal grid**. *SIAM Journal on Discrete Mathematics*, 13 :492–504, 2000.
- [17] Gérard Cohen, Liro Honkala, Gilles Zémor et Antoine Lobstein. **New Bounds for Codes Identifying Vertices in Graphs**. *Electronic Journal of Combinatorics*, 6(1), 1999.
- [18] Gérard Cohen, Liro Honkala, Sylvain Gravier, Antoine Lobstein, Michel Mollard et Charles Payan. **Improved Identifying Codes For the Grid**. *Electronic Journal of Combinatorics*, 1999.
- [19] Gérard Cohen, Liro Honkala, Antoine Lobstein et Gilles Zémor. **On Codes Identifying Vertices in the Two-Dimensional Square Lattice with Diagonals** . *IEEE Transactions On Computers*, 50(2), February 2001.
- [20] Gérard Cohen, Antoine Lobstein, Liro Honkala et Simon Litsyn. *Covering Codes*. Elsevier, 1997.
- [21] Irène Charon, Olivier Hudry et Antoine Lobstein. **Identifying and Locating-Dominating Codes : NP-Completeness Results for Directed Graphs**. *IEEE Transactions On Information Theory*, 48(8), AUGUST 2002.
- [22] Irène Charon, Olivier Hudry et Antoine Lobstein. **Identifying Codes with Small Radius in Some Infinite Regular Graphs**. *Electronic Journal of Combinatorics*, 9 :R11, 2002.

-
- [23] Irène Charon, Olivier Hudry et Antoine Lobstein. **Minimizing the size of an identifying or locating-dominating code in graph is NP-Hard.** *Theoretical Computer Science*, 290 :2109–2120, 2003.
- [24] Irène Charon, Olivier Hudry et Antoine Lobstein. **Extremal cardinalities for identifying and locating-dominating codes in graphs.** *Discret Mathematics*, 307 :356–366, 2007.
- [25] Irène Charon, Sylvain Gravier, Olivier Hudry, Antoine Lobstein, Michel Mollard et Julien Moncel. **A linear algorithm for minimum 1-identifying codes in oriented trees.** *Discret Applied Mathematics*, 154 :1246–1253, 2006.
- [26] Irène Charon, Gérard Cohen, Olivier Hudry et Antoine Lobstein. **Links Between Discriminating and Identifying Codes in the Binary Hamming Space.** *Lect. Notes Compu. Sci*, 4851 :267–270, 2007.
- [27] Irène Charon, Gérard Cohen, Olivier Hudry et Antoine Lobstein. **Discriminating codes in (bipartite) planar graphs.** *European Journal of Combinatorics*, pages 1353–1364, 2008.
- [28] Irène Charon, Gérard Cohen, Olivier Hudry et Antoine Lobstein. **New Identifying Codes in the Binary Hamming Space.** *European Journal of Combinatorics*, 2009.
- [29] Irène Charon, Liro Honkala, Olivier Hudry et Antoine Lobstein. **General Bounds for Identifying Codes in Some Infinite Regular Graphs .** *The Electronic Journal of Combinatorics*, 8 :R39, 2001.
- [30] Irène Charon, Liro Honkala, Olivier Hudry et Antoine Lobstein. **The Minimum Density of an Identifying Code in the King Lattice.** *Discret Mathematics*, 276 :95–109, 2004.
- [31] Julien Moncel. **Monotonicity of the Minimum Cardinality of an Identifying Code in the Hypercube.** *Discret applied mathematics*, 154 :898–899, 2005.
- [32] Julien Moncel. *Codes identifiants dans les graphes.* PhD thesis, Université Joseph FOURIER, 2006.
- [33] Julien Moncel. **On graphs on n vertices having an identifying code of cardinality $\lceil \log_2(n + 1) \rceil$.** *Discret Applied Mathematics*, 154 :2032–2039, 2006.
- [34] K. Chakrabarty, S. Sitharama Iyengar, Hairong Qi et Eungchun Cho. **Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks.** *IEEE Transactions On Computers*, 51(12) :1448–1453, 2002.

-
- [35] Liro Honkala. **An Optimal Edge Robust Identifying Code in the Triangular Lattice.** *Annals of Combinatorics*, 8 :303–323, 2004.
- [36] Liro Honkala et Antoine Lobstein. **On Identifying Codes in Binary Hamming Spaces.** *Journal of Combinatorial Theory, Series A* 99, pages 232–243, 2002.
- [37] Liro Honkala et Antoine Lobstein. **On the Complexity of the Identification Problem in Hamming Spaces.** *Acta Informatica*, 38 :839–845, 2002.
- [38] Liro Honkala et Antoine Lobstein. **On the Density of Identifying Codes in the Square Lattice.** *Journal of Combinatorial Theory, Series B* 85, pages 297–306, 2002.
- [39] Liro Honkala et Tero Laihonen. **On identification in the triangular grid.** *Journal of Combinatorial Theory, Series B* 91, pages 67–86, 2004.
- [40] Liro Honkala et Tero Laihonen. **On new class of identifying codes in graphs.** *Information Processing Letters*, 102 :92–98, 2007.
- [41] Liro Honkala, Tero Laihonen et Sanna Ranto. **On Codes Identifying Sets of Vertices in Hamming Spaces.** Technical Report 331, Turku Centre for Computer Science, February 2000.
- [42] Liro Honkala, Tero Laihonen et Sanna Ranto. **On strongly identifying codes.** *Discret Mathematics*, 254 :191–205, 2002.
- [43] Marc Daniel, Sylvain Gravier et Julien Moncel. **Identifying code in some subgraphs of the square lattice.** *Theoretical Computer Science*, 319 :411–421, 2004.
- [44] Mark. G. Karpovsky K. Chakrabarty et Lev B. Levitin. **On New Class of Codes for Identifying Vertices in Graphs.** *IEEE Transactions On Information Theory*, 44(2) :599–611, 1998.
- [45] Mark. G. Karpovsky, Liro Honkala et Lev. B. Levitin. **On robust and dynamic identifying codes.** *IEEE Transactions On Information Theory*, 52(2) :599–612, Fevrier 2006.
- [46] Michael R. Garey et David S. Johnson. **Computer and Intractability A guide to the theory of NP-Completeness.** W.H Freeman and Co, San Francisco, 1979.
- [47] Min Xu, Krishnaiyan Thulasiraman et Xiao-Dong Hu. **Identifying codes of cycles with odd orders.** *European Journal of Combinatorics*, 29 :1717–1720, 2008.
- [48] Moshe Laifenfeld, Ari Trachtenberg, Reuven Cohen et David Starobinski. **Joint Monitoring and Routing in Wireless Sensor Network Using Robust Identifying Codes.** *Mobile networks and applications*, 14(4) :415–432, 2009.

-
- [49] M. Laifenfeld, A. Trachtenberg et T.Y. Berger-Wolf. **Identifying Codes and the Set Cover Problem**. *Annual Allerton Conf. on Comm, Ctrl and Comput 44th*, 2006.
- [50] Nathalie Bertrand, Irène Charon, Olivier Hudry et Antoine Lobstein. **Identifying and locating-dominating codes on chains and cycles**. *European Journal of Combinatorics*, 25 :969–987, 2004.
- [51] Petri Rosendahl. **On the Identification Problems in Products of Cycles** . *Discret Mathematics*, 275 :277–288, 2004.
- [52] Reinhard Diestel. *Graph theory*. Springer-Verlag Heidelberg, 2005.
- [53] Robin J. Wilson. *Introduction to graph theory*. Addison-Wesley Longman, 1998.
- [54] Saikat Ray, David Starobinski, Ari Trachtenberg et Rachanee Ungrangsi. **Robust Location Detection With Sensor Networks**. *IEEE Journal on selected areas in communications*, 22(6), 2004.
- [55] Saikat Ray, David Starobinski, Ari Trachtenberg, Rachanee Ungrangsi et Fransesco De Pellegrini. **Robust Location Detection in Emergency Sensor Networks**. *IEEE INFOCOM*, pages 1044–1053, 2003.
- [56] Sanna Ranto, Liro Honkala et Tero Laihonen. **Two Families of Optimal Identifying Codes in Binary Hamming Spaces**. *IEEE Transactions On Information Theory*, 48(3) :1200–1202, 2002.
- [57] Svante Janson et Tero Laihonen. **An Optimal Result for Codes Identifying Sets of Words** . *Proceedings of the 2009 IEEE international conference Symposium on Information Theory*, 4 :2547–2551, 2009.
- [58] Svante Janson et Tero Laihonen. **On the Size of Identifying Codes in Binary Hypercubes**. *Journal of Combinatorial Theory Series A*, (116) :1087–1096, 2009.
- [59] Sylvain Gravier et Julien Moncel. **On graphs having a $V \setminus \{x\}$ set as an identifying code**. *Discret Mathematics*, 307 :432–434, 2007.
- [60] Sylvain Gravier, Julien Moncel et Ahmed Semri. **Identifying codes of cycles**. *European Journal of Combinatorics*, 27 :767–776, 2006.
- [61] Sylvain Gravier, Julien Moncel et Ahmed Semri. **Identifying Codes of Cartesian of two Cliques of the Same Size**. *The Electronic Journal of Combinatorics*, 15, #N4, 2008.

- [62] Tero Laihonen. **Optimal t -Edge-Robust r -Identifying Codes in the King Lattice.** *Graphs and Combinatorics*, 22 :487–496, 2006.
- [63] Tero Laihonen. **Sequences of Optimal Identifying Codes.** *IEEE Transactions On Information Theory*, 48(3), March 2002.
- [64] Thomas H. Corman, Charles E. Leiserson, Ronald L. Revest et Clifford Stein. **Introduction to Algorithms.** McGraw-Hill, second edition, 2001.
- [65] T.Y. Berger-Wolf, W.E. Hart et J. Saia. **Discrete Sensor Placement Problems in Distribution Networks .** *Mathematical and Computer Modelling*, 42 :1385–1396, 2005.
- [66] Uri Blass, Liro Honkala et Simon Litsyn. **Bounds on Identifying Codes.** *Discret Mathematics*, 241 :119–128, 2001.
- [67] Yael Ben-Haim et Simon Litsyn. **Exact Minimum Density of Codes Identifying Vertices in the Square Grid.** *SIAM Journal on Discrete Mathematics*, 19(1) :69–82, 2005.
- [68] Yi-Chun Xu et Ren-Bin Xiao. **Identifying Code for Directed Graph.** *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, 2 :97–101, 2007.
- [69] Yi-Chun Xu et Ren-Bin Xiao. **Solving the Identifying Code Problem by a Genetic Algorithm .** *IEEE Transactions On Systems, Man, and Cybernetics-Part A : Systems and Humans*, 37(1), January 2007.

Résumé La notion de codes identifiants a été définie pour la première fois par M. Karpovsky, K. Chakrabarty et L. Levitin en 1998 pour modéliser un problème de détection de processeurs défectueux dans des réseaux multiprocesseurs. Ce concept a depuis été repris par de nombreuses équipes de recherche engendrant une littérature abondante.

Dans ce mémoire, nous avons abordé le problème de recherche de code identifiant de cardinalité minimum un problème appartenant à la classe des problèmes NP-difficiles. Peu de résultats sont connus dans le cas de graphes orientés, ceci nous a amené à étudier les codes identifiants dans ce type de graphes.

Dans notre travail, nous nous sommes intéressé à l'étude de ce problème dans le cas de chemins et de circuits. Nous avons pu déterminer la cardinalité minimum d'un code 1 et 2-identifiant aussi bien pour un chemin que pour un circuit.

Abstract Identifying codes were first introduced by Karpovsky, Chakrabarty, Levitin. Their motivation for identification comes from locating faulty processors in a multi-processor system. In general, finding an optimal identifying codes is known to be an NP-complete problem.

Nowadays, this concept is taken by numerous research team, what produced plentiful literature.

In this thesis we reach the problem of identifying code. Few results are known in the case of directed graphs, this led us to study this problem in this type of graphs.

In our work, we investigate the identifying code problem in paths and circuits and we could determine the smallest cardinality of 1 and 2-identifying codes.