

N°d'ordre : 38/2012-M/MT

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE  
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE  
HOUARI BOUMEDIENE



FACULTE DES MATHEMATIQUES

**MEMOIRE**

Présenté en vue de l'obtention du diplôme de **Magister**

En : MATHEMATIQUES

Spécialité : **Recherche Opérationnelle, Mathématique Discrète et  
Optimisation**

Par Mme : **HISSOUM ANISSA**

Sujet

*Codes  $r$ -identifiants dans les espaces de  
Hamming*

Soutenu publiquement, le 24/06/2012 devant le jury composé de :

M. R. OUAFI	Professeur,	à L'U.S.T.H.B	Président
M. A. SEMRI	M.C.A,	à L'U.S.T.H.B	Directeur de mémoire
M. A. BERRACHEDI	Professeur,	à L'U.S.T.H.B	Examineur

## *Remerciements*

Avant tout Je remercie notre Dieu le tout puissant de m'avoir donné la force d'atteindre mon but et d'accomplir mon travail.

Que mes remerciements les plus sincères s'adressent à mon directeur de thèse Monsieur **A.SEMRI**, maitre de conférence à l'UTSHB, pour sa présence, ses remarques et suggestions qui m'ont aidé à bien mener ce travail.

J'adresse mes vifs remerciements à Mr **R. OUAFI**, Professeur à l'USTHB, pour avoir accepté d'examiner mon travail et de présider le jury de soutenance.

Mr **A. BERRACHEDI**, Professeur à l'USTHB a bien voulu examiner le travail. Ses critiques et ses remarques me seront sans doute très utiles pour la suite. Je tiens à lui exprimer toute ma gratitude.

Je remercie enfin mes parents, mon mari, toute ma famille et tous mes amis pour leurs encouragements et leurs soutiens pendant toutes ces années d'études.

## Résumé

Dans ce mémoire, on s'intéresse au problème des codes identifiant de cardinalité minimum dans certaines classes de graphes. Un code identifiant dans un graphe est un sous ensemble de sommets tel que deux sommets quelconques du graphe ont leurs ensembles de voisinage fermé différents et non vides dans le code.

Les codes identifiants font partie des problèmes de domination et ont été introduits pour modéliser un problème de détection et de localisation de défaillance dans un réseau de multiprocesseurs. Un problème d'optimisation combinatoire associé est celui qui consiste à trouver la cardinalité minimum d'un code identifiant (lorsqu'il existe) dans un graphe . C'est un problème *NP*-difficile dans le cas des graphes non orientés ou orientés.

Dans notre travail, nous avons, dans un premier temps, appliqué une méta-heuristique recherche tabou pour générer de bons codes identifiants à distance  $r \geq 1$  dans l'espace binaire de Hamming de dimension  $n \geq 2$ .

Par la suite, nous avons déterminé une borne supérieure de la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques de dimensions différentes  $K_n \square K_m$  avec  $n \geq 2$  et  $n < m$ .

**Mots clés :** Graphes, Domination, Code identifiant, Graphe de Hamming, Hypercube, Produit Cartésien des graphes.

## Abstract

We consider the problem of minimum identifying codes in some classes of graphs. An identifying code is a subset of vertices in a graph with the property that the closed neighborhood of any vertex has a unique and nonempty intersection with the code.

The initial application for identifying codes was to fault diagnosis in multiprocessor system in 1998 and has been since fundamentally connected to a wide range of applications. The identifying code problem is NP-complete in the case of undirected or directed graphs.

In the first part of our work, we have applied a tabu search metaheuristic to generate good identifying codes in the binary Hamming space of dimension  $n \geq 2$  and  $r \geq 1$ .

In the second part, we determined an upper bound on the cardinality minimum of a code identifying the Cartesian product of two cliques with different dimensions  $K_n \square K_m$ ,  $n \geq 2$  and  $n < m$ .

**Keywords :** Graphs, Domination, identifying code, Hamming graph, Hypercube, Cartesian product graphs.

# Table des matières

<b>Table des matières</b>	<b>4</b>
<b>Table des figures</b>	<b>6</b>
<b>Introduction générale</b>	<b>7</b>
<b>1 Préliminaires</b>	<b>9</b>
1.1 Introduction . . . . .	10
1.2 Quelques notations dans les ensembles . . . . .	10
1.2.1 Notations des opérations usuelles . . . . .	10
1.2.2 Partition . . . . .	10
1.2.3 Ensemble minimal et ensemble minimum . . . . .	10
1.3 Les graphes . . . . .	11
1.3.1 Définitions classiques . . . . .	11
1.3.2 Classes de graphes . . . . .	14
1.3.3 Produit de graphe . . . . .	16
1.4 Graphes de Hamming . . . . .	17
1.4.1 L'hypercube . . . . .	17
1.4.2 Définition . . . . .	19
1.5 Problèmes d'Optimisation Combinatoire et Théorie de la Complexité . . . . .	21
<b>2 Les codes identifiants</b>	<b>23</b>
2.1 Introduction . . . . .	25
2.2 Définition des codes identifiants . . . . .	25

2.2.1	Code couvrant d'un graphe . . . . .	25
2.2.2	Code séparateur d'un graphe . . . . .	26
2.2.3	Code identifiant d'un graphe . . . . .	26
2.3	Application pratique des codes identifiants . . . . .	27
2.3.1	Détection et localisation de défaillance dans les réseaux	28
2.3.2	Systemes de localisation et de détection dans les envi- ronnements fermés munis de capteurs sans fil . . . . .	29
2.3.3	Conservation d'énergie . . . . .	29
2.4	Généralisations et variantes . . . . .	30
2.4.1	Code identifiant à distance $r \geq 1$ . . . . .	30
2.4.2	Code identifiant d'ensemble de sommets . . . . .	30
2.4.3	Code identifiant dans le cas d'un graphe orienté . . . . .	31
2.4.4	Code localisateur dominateur . . . . .	31
2.4.5	Code discriminant . . . . .	32
2.5	Densité d'un code identifiant . . . . .	33
2.6	Premiers résultats . . . . .	34
2.7	Codes identifiants de certains graphes particuliers . . . . .	36
2.7.1	Codes identifiants dans les cycles . . . . .	36
2.7.2	Codes identifiants dans les arbres . . . . .	37
2.7.3	Codes identifiants dans les grilles et les bandes . . . . .	38
2.8	Quelques résultats sur les codes identifiants dans les espaces de Hamming . . . . .	39
2.8.1	Bornes inférieures . . . . .	39
2.9	Conclusion . . . . .	42
<b>3</b>	<b>Construction de codes identifiants par la Recherche Tabou</b>	<b>43</b>
3.1	Introduction . . . . .	45
3.2	Les Méta-Heuristiques . . . . .	45
3.2.1	Présentation générale . . . . .	45
3.2.2	Aperçu sur les métaheuristiques . . . . .	46
3.3	Construction des codes $r$ -identifiants dans $Q_n$ . . . . .	49
3.3.1	Construction théorique . . . . .	50

3.3.2	Heuristiques : la méthode de bruitage et l'algorithme glouton . . . . .	51
3.3.3	Application . . . . .	52
3.4	Générer des codes $r$ -identifiants dans $Q_n$ par une Recherche Tabou . . . . .	52
3.4.1	Principe de la méthode . . . . .	52
3.4.2	Expérimentation et Résultat . . . . .	55
3.5	Conclusion . . . . .	56
<b>4</b>	<b>Codes identifiants dans le produit cartésien de deux cliques de dimensions différentes</b>	<b>57</b>
4.1	Introduction . . . . .	58
4.2	Codes identifiants dans le produit cartésien de deux cliques de même dimension . . . . .	58
4.3	Codes identifiants dans le produit cartésien de deux cliques de dimensions différentes . . . . .	59
4.3.1	Le cas où $b = 0$ et $a \geq 2$ . . . . .	60
4.3.2	Le cas où $b \neq 0$ et $a \geq 2$ . . . . .	62
4.3.3	Le cas où $b \neq 0$ et $a = 1$ . . . . .	66
4.3.3.1	Si $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$ . . . . .	66
4.3.3.2	Si $\lfloor \frac{n}{2} \rfloor + 1 \leq b \leq n - 1$ . . . . .	67
4.4	Conclusion . . . . .	72
	<b>Conclusion</b>	<b>73</b>
	<b>Bibliographie</b>	<b>74</b>

# Table des figures

1.1	Graphes complets $K_n, n \leq 5$ . . . . .	14
1.2	Exemple d'un graphe complémentaire . . . . .	14
1.3	Graphe biparti complet $K_{3,4}$ . . . . .	15
1.4	L'étoile $S_n$ est graphe biparti avec 1 supports et $n - 1$ sommets pendants . . . . .	15
1.5	Exemple de fermeture d'un graphe. . . . .	15
1.6	Illustration des différents produits . . . . .	17
1.7	Le quadrangle généralisé $GQ(2, 1)$ . . . . .	17
1.8	Les hypercubes de dimensions 1,2 et 3 . . . . .	18
1.9	L'hypercube est un graphe biparti . . . . .	18
1.10	Hypercube de dimension 4 ou Tesseract . . . . .	19
2.1	Le code $C = \{000, 111\}$ est un code couvrant de $Q_3$ . . . . .	26
2.2	Le code $C = \{001, 100, 111\}$ est un code séparateur de $Q_3$ . . . . .	26
2.3	Les sommets gris identifient tous les sommets de $Q_3$ . . . . .	27
2.4	L'ensemble des sommets en gris est un code identifiant du graphe orienté mais pas du graphe non orienté $K_5$ . . . . .	32
2.5	Les sommets en gris constituent un code localisateur-dominateur mais pas un code identifiant . . . . .	32
4.1	Produit cartésien de deux cliques $K_n \square K_m$ . . . . .	58
4.2	Le code $D \cup A$ suivant la parité de $n$ . . . . .	59
4.3	Code identifiant de $K_5 \square K_{25}$ . . . . .	61
4.4	Code identifiant de $K_6 \square K_{24}$ . . . . .	61



---

4.5	Codes identifiants de $K_n \square K_{an+b}$ , $n = 4$ , $a = 2$ et $b \in \{1, 2, 3\}$ .	65
4.6	Codes identifiants de $K_n \square K_{an+b}$ , $n = 5$ , $a = 2$ et $b \in \{1, 2, 3, 4\}$	65
4.7	Codes identifiants de $K_n \square K_{n+b}$ , $n = 6$ et $b \in \{1, 2, 3, 4, 5\}$ . . .	70
4.8	Codes identifiants de $K_n \square K_{n+b}$ , $n = 7$ et $b \in \{1, 2, 3, 4, 5, 6\}$ .	71

# Introduction générale

Un graphe est une structure très simple constitué d'un ensemble de sommets et d'une famille de liens orientés ou non, appelés arêtes ou arcs, entre certains couples de sommets.

On considère que le premier résultat formel de la Théorie des graphes est le problème des sept ponts de Königsberg (maintenant Kaliningrad), qui consiste à faire un tour dans la ville en passant une fois exactement sur chaque pont, et dont Euler a démontré qu'une telle promenade est impossible.

La Théorie des graphes présente des liens évidents avec l'algèbre, la topologie et d'autres domaines de la combinatoire. On trouve des applications de la Théorie des graphes et souvent aussi la motivation de nouveaux problèmes en Informatique, Recherche Opérationnelle, Théorie des jeux et Théorie de la décision. Les grands problèmes classiques d'optimisation combinatoire en théorie des graphes sont : flots et connectivité, couplage, parcours eulérien, parcours hamiltonien, coloration, ensemble stable et ensemble dominant. Certains de ces problèmes peuvent être résolus efficacement, mais la plupart sont très difficiles (problèmes  $\mathcal{NP}$ -complets).

Dans ce mémoire, nous nous intéressons à la notion des codes identifiants, qui est un cas particulier de problème de domination. Un code identifiant dans un graphe est un sous ensemble dominant dont deux sommets quelconques du graphe ont leur ensemble de voisinage différents et non vides dans le code. Parmi les variantes qui existent, l'identification à distance  $r$  est étudiée aussi.

Un des problèmes d'optimisation combinatoire associé est celui de trouver un code identifiant, lorsqu'il existe, de cardinalité minimum dans un graphe. C'est un problème  $\mathcal{NP}$ -difficile aussi bien dans les graphes orientés [10] que dans les graphes non orientés [11].

La notion des codes identifiants a été introduite par Karpovsky, Chakrabarty and Levitin en 1998 dans [31] pour modéliser un problème de détection et de localisation de processeurs défectueux dans des réseaux multiprocesseurs. Depuis, les codes identifiants ont été étudiés dans différentes classes de graphes comme les grilles [1, 8, 9, 16, 30, 29], les arbres [4, 7, 12], les chaînes et les cycles [3, 23, 36] et l'hypercube [5, 6, 20, 26, 28, 34].

Le premier chapitre est consacré à la présentation des concepts fondamentaux, l'énoncé des définitions ainsi que les notations utilisées concernant les ensembles, les graphes et la complexité dans les problèmes d'optimisation combinatoire.

Dans le deuxième chapitre, nous donnons d'abord une définition complète des codes identifiants, ses applications pratiques et quelques variantes et généralités. Puis nous citons les principaux résultats dans la littérature, plus particulièrement les codes identifiants à distance  $r$  dans l'hypercube (espace binaire de Hamming).

Après avoir donné les bornes de la cardinalité minimum d'un code identifiant dans l'hypercube, nous appliquons une métaheuristique Recherche Tabou pour essayer de construire de bons codes identifiant à distance  $r \geq 1$  dans l'espace binaire de Hamming de dimension  $n \geq 3$ . Cela fera l'objet du troisième chapitre.

Dans le chapitre quatre, nous étudions le problème des codes identifiants dans le produit cartésien de deux cliques de dimensions différentes.

# 1

## Préliminaires

### Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>10</b>
<b>1.2</b>	<b>Quelques notations dans les ensembles</b>	<b>10</b>
1.2.1	Notations des opérations usuelles	10
1.2.2	Partition	10
1.2.3	Ensemble minimal et ensemble minimum	10
<b>1.3</b>	<b>Les graphes</b>	<b>11</b>
1.3.1	Définitions classiques	11
1.3.2	Classes de graphes	14
1.3.3	Produit de graphe	16
<b>1.4</b>	<b>Graphes de Hamming</b>	<b>17</b>
1.4.1	L'hypercube	17
1.4.2	Définition	19
<b>1.5</b>	<b>Problèmes d'Optimisation Combinatoire et Théorie de la Complexité</b>	<b>21</b>

---

## 1.1 Introduction

Ce premier chapitre est consacré aux définitions des principales notions utilisées dans ce mémoire.

## 1.2 Quelques notations dans les ensembles

### 1.2.1 Notations des opérations usuelles

On note  $\Delta$  la *différence symétrique* de deux ensembles, c'est à dire que  $S\Delta S' = (S\setminus S') \cup (S'\setminus S) = (S \cup S') \setminus (S \cap S')$ .

Soit  $S$  un ensemble,  $X$  un sous ensemble de  $S$ , on note  $\bar{X}$  le complémentaire de  $X$  dans  $S$ , c'est à dire  $S\setminus X$ .

### 1.2.2 Partition

Soit  $S$  un ensemble. Une *partition* de  $S$  est un ensemble de sous ensembles de  $S$ , notés  $S_1, S_2, \dots, S_n$  tels que

- tout élément de  $S$  est dans un sous ensemble  $S_i$ , c'est à dire  $\bigcup S_i = S$ ;
- les sous ensembles sont disjoints :  $\forall i, j, S_i \cap S_j = \emptyset$ ;
- chacun des  $S_i$  est non vide :  $\forall i, S_i \neq \emptyset$ .

### 1.2.3 Ensemble minimal et ensemble minimum

Soit une propriété  $P$ . On dit qu'un ensemble  $S$  est *minimal* pour la propriété  $P$  si aucun sous ensemble strict de  $S$  ne vérifie cette propriété. Un ensemble est dit *minimum* pour la propriété  $P$  si aucun ensemble plus petit (pas nécessairement un sous ensemble) ne vérifie la propriété. Ainsi, un ensemble minimum est nécessairement minimal, mais l'inverse n'est pas vrai en général.

De même, on dit qu'un ensemble  $S$  est *maximal* pour la propriété  $P$  si aucun ensemble contenant  $S$  et différent de  $S$  ne vérifie la propriété  $P$ . Il est *maximum* si aucun ensemble plus grand que  $S$  (sans nécessairement le contenir) ne vérifie  $P$ .

## 1.3 Les graphes

La terminologie adoptée est celle utilisée par Berge dans [2].

### 1.3.1 Définitions classiques

Un *graphe* non orienté est un couple  $G = (V, E)$ , où  $V$  est un ensemble fini d'éléments appelés *sommets* de  $G$ , et  $E \subseteq V \times V$  est un sous ensemble de famille des paires de sommets de  $G$  appelées *arêtes*.

L'*ordre* d'un graphe  $G$  est le nombre de ses sommets, noté souvent  $n$ .

On note  $e = uv \in E$  l'arête joignant les sommets  $u$  et  $v \in V$  et qui sont appelés *extrémités* de l'arête. On dit que l'arête  $e$  est *incidente* à  $u$  et  $v$  et que les deux sommets  $u$  et  $v$  sont *voisins*. Une *boucle* est une arête ayant pour extrémités le même sommet. Un graphe est dit simple s'il est sans boucles et sans arêtes multiples.

#### Degré

On appelle degré d'un sommet  $u$  dans  $G$ , noté  $d_G(u)$  (ou  $d(u)$  quand le contexte est explicite), le nombre de ses voisins. Le degré minimum (respectivement maximum) d'un graphe  $G$ , noté  $\delta(G)$  (respectivement  $\Delta(G)$ ), est le minimum (respectivement maximum) des degrés de ses sommets. On appelle sommet *isolé* un sommet de degré 0. Un sommet de degré 1 est appelé un *sommet pendant*.

#### Sous graphe

Soit  $S$  un sous ensemble de sommets du graphe,  $S \subseteq V$ . Le *sous graphe* de  $G$  *induit* sur  $S$ , noté  $G[S]$ , est le graphe ayant pour ensemble de sommets  $S$  et pour ensemble d'arêtes toutes les arêtes de  $E$  dont les extrémités sont dans  $S$ .

#### Graphe partiel

Un *graphe partiel* de  $G$  est un graphe obtenu en enlevant à  $G$  une ou plusieurs arêtes. Autrement dit, un graphe partiel de  $G$  est un graphe  $H$

ayant le même ensemble de sommets :  $V(H) = V(G)$ , et pour ensemble d'arêtes un sous-ensemble de  $E(G)$  :  $E(H) \subset E(G)$ . Si on enlève à la fois des arêtes et des sommets, on aura un *sous graphe partiel*.

### Clique

Un sous graphe complet est dit une *clique*.

### Stable

Un sous ensemble de sommets  $S \subseteq V$  est dit *stable* de  $G$  s'il n'y a pas d'arêtes entre deux sommets quelconques de  $S$ .

### Transversal

Un sous ensemble  $T \subseteq V$  est un *transversal* de  $G$  si toute arête de  $G$  a au moins une extrémité dans  $T$ .

### Couplage

Un sous ensemble d'arêtes  $M \subseteq E$  est dit un *couplage* si les arêtes de  $M$  sont deux à deux non adjacentes, il est dit *couplage parfait* si tout sommet de  $G$  est incident à une arête de  $M$ .

### Recouvrement

Un sous ensemble d'arêtes  $F \subseteq E$  est dit *recouvrement* si tout sommet de  $G$  est l'extrémité d'au moins une arête de  $F$ . Un couplage qui est un recouvrement est un couplage parfait.

### Parcours

Une *chaîne* de *longueur*  $q$  est une séquence alternée de sommets et d'arêtes  $[v_1e_1v_2, v_2e_2v_3, \dots, v_qe_qv_{q+1}]$ , telle que, pour  $1 \leq i \leq q$ , les extrémités de  $e_i$  sont  $v_i$  et  $v_{i+1}$ . Une chaîne simple fermée est un *cycle*.

Le terme de *parcours* regroupe les chaînes et les cycles. Un parcours est élémentaire s'il n'emprunte qu'une fois ses sommets.

Un parcours est *eulérien* s'il passe exactement une fois par chaque arête de graphe.

Un parcours est *hamiltonien* s'il passe exactement une fois par chaque sommet du graphe.

### Connexité

Un graphe est dit *connexe* si toute paire de sommets du graphe est reliée par au moins une chaîne. Si le graphe est non connexe, une composante connexe est un sous ensemble de sommets  $C$  maximal tel que le sous-graphe induit  $G[C]$  est connexe.

### Distance

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

### Voisinage

On appelle *voisinage ouvert* du sommet  $u$ , noté  $N_G(u)$  l'ensemble des sommets  $v$  adjacents à  $u$  dans  $G$ . Le plus souvent, le contexte étant explicite, on notera simplement  $N(u)$ . On a  $|N(u)| = d(u)$ . Le *voisinage fermé* ou étendu du sommet  $u$  désigne l'ensemble  $N[u] = N(u) \cup \{u\}$ .

On parle aussi du voisinage ouvert d'un sous-ensemble  $S \subseteq V$ , noté  $N(S)$ , pour désigner l'union des voisinages (ouverts) des sommets de  $S$ . De même, le voisinage fermé  $N[S]$  de  $S$  est l'union des voisinages fermés des sommets de  $S$ , autrement dit  $N[S] = N(S) \cup S$ .

On note par  $B_r(v)$  la *boule* de centre  $v \in V$  et de rayon  $r$  le sous ensemble de sommets à distance au plus  $r$  de  $v$ , c'est à dire  $B_r(v) = \{u \in V : d(u, v) \leq r\}$ . On remarque que  $B_1(v) = N[v]$ . Les deux sommets  $u$  et  $v$  sont dits *jumeaux* si  $B_r(u) = B_r(v)$ .



### 1.3.2 Classes de graphes

#### Graphe complet

Un graphe à  $n$  sommets est dit *complet* si toute paire de ses sommets est reliée par une arête, il est noté par  $K_n$ .

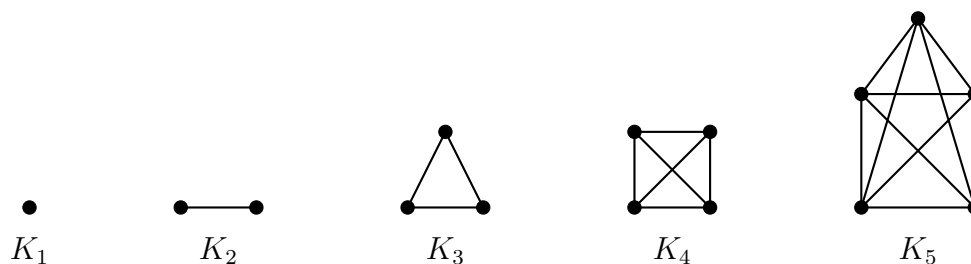


FIG. 1.1 – Graphes complets  $K_n, n \leq 5$

#### Graphe complémentaire

Le graphe *complémentaire* d'un graphe  $G = (V, E)$  est le graphe  $\overline{G} = (V, V^2 - E)$ , dont les arêtes sont tous les couples de sommets qui ne sont pas dans  $E$ .

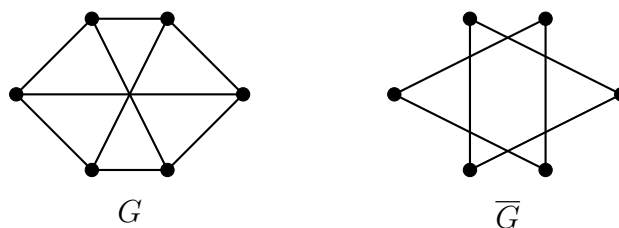


FIG. 1.2 – Exemple d'un graphe complémentaire

#### Graphe biparti

$G = (V, E)$  est *biparti* si on peut partitionner ses sommets en deux sous ensembles  $V_1$  et  $V_2$  avec aucune arête entre deux sommets de  $V_1$  ou deux de  $V_2$ . On le note  $G = (V_1 \cup V_2, E)$ . Un graphe simple biparti complet avec  $|V_1| = a$  et  $|V_2| = b$  est noté par  $K_{a,b}$ .

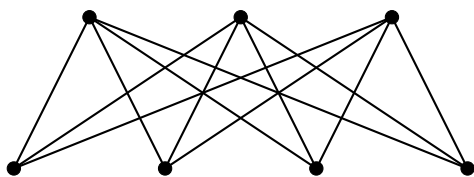


FIG. 1.3 – Graphe biparti complet  $K_{3,4}$

### Arbre

Un *arbre* est un graphe  $G$  connexe sans cycle (donc il est biparti). Il a une arête de moins que de sommets :  $|E| = |V| - 1$  et admet au moins deux sommets pendants. Le voisin d'un sommet pendent est dit un *support*.

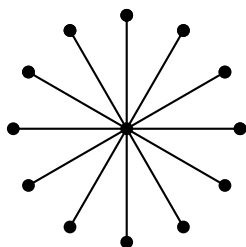


FIG. 1.4 – L'étoile  $S_n$  est graphe biparti avec 1 supports et  $n - 1$  sommets pendants

### Fermeture $t$ -transitive

Soit  $G$  un graphe et soit  $t \geq 1$  un entier. La *fermeture  $t$ -transitive* de  $G$ , notée  $G^t$ , est le graphe ayant le même ensemble de sommets que  $G$  tel que deux sommets  $u$  et  $v$  sont adjacents dans  $G^t$  si et seulement si ils sont à distance inférieure ou égale à  $t$  dans  $G$ .

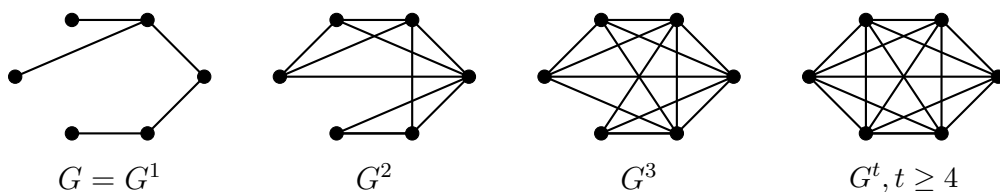


FIG. 1.5 – Exemple de fermeture d'un graphe.

### 1.3.3 Produit de graphe

#### Produit Cartésien

Étant donnés deux graphes  $G$  et  $H$ , le *produit cartésien*  $G \square H$  est le graphe ayant pour ensemble de sommets  $V(G) \times V(H)$  et dont deux sommets  $(x_1, y_1)$  et  $(x_2, y_2)$  sont reliés par une arête si et seulement si, soit  $x_1 x_2 \in E(G)$  et  $y_1 = y_2$ , soit  $y_1 y_2 \in E(H)$  et  $x_1 = x_2$ .

#### Produit Croisé

Étant donnés deux graphes  $G$  et  $H$ , le *produit croisé*  $G \times H$  est le graphe ayant pour ensemble de sommets  $V(G) \times V(H)$  et dont deux sommets  $(x_1, y_1)$  et  $(x_2, y_2)$  sont reliés par une arête si et seulement si  $x_1 x_2 \in E(G)$  et  $y_1 y_2 \in E(H)$ . On remarque que

$$E(G \square H) \cap E(G \times H) = \emptyset.$$

#### Produit Total

Étant donnés deux graphes  $G$  et  $H$ , le *produit total*  $G \boxtimes H$  est le graphe ayant pour ensemble de sommets  $V(G) \times V(H)$  et dont deux sommets  $(x_1, y_1) \neq (x_2, y_2)$  sont reliés par une arête si et seulement si :

$$\text{i) } x_1 = x_2 \text{ ou } x_1 x_2 \in E(G) \quad \text{et} \quad \text{ii) } y_1 = y_2 \text{ ou } y_1 y_2 \in E(H),$$

c'est à dire si les sommets sont adjacents ou égaux dans chaque coordonnée.

On remarque que

$$E(G \boxtimes H) = E(G \square H) \cup E(G \times H)$$

#### Produit Lexicographique

Étant donnés deux graphes  $G$  et  $H$ , le *produit lexicographique*  $G \bullet H$  est le graphe ayant pour ensemble de sommets  $V(G) \times V(H)$  et dont deux sommets  $(x_1, y_1) \neq (x_2, y_2)$  sont reliés par une arête si et seulement si :

i)  $x_1 x_2 \in E(G)$  ou ii)  $x_1 = x_2$  et  $y_1 y_2 \in E(H)$ , c'est à dire quand les sommets sont adjacents pour la première coordonnée ou quand ils sont égaux pour la première coordonnée et adjacents pour la seconde. On a  $E(G \boxtimes H) \subseteq E(G \bullet H)$ .

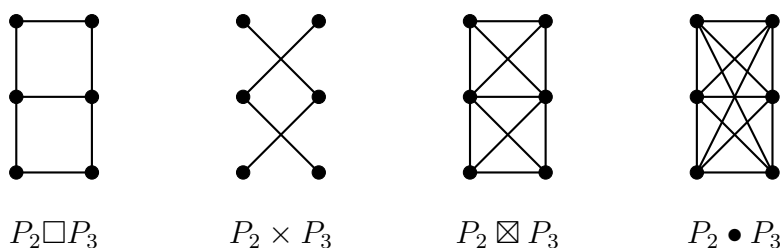


FIG. 1.6 – Illustration des différents produits

## 1.4 Graphes de Hamming

Le *graphe de hamming*, noté par  $H(n, p)$ ,  $n, p \geq 1$ , est le produit cartésien de  $n$  copies du graphe complet  $K_p$  :  $H(n, p) = \square^n K_p$ .

Des cas particuliers sont résumés dans le tableau suivant :

$H(1, p)$	le graphe complet $K_p$
$H(2, 3)$	le quadrangle généralisé $GQ(2, 1)$
$H(2, p)$	la grille $L_{pp}$
$H(n, 1)$	le graphe $K_1$
$H(n, 2)$	l'hypercube $Q_n$

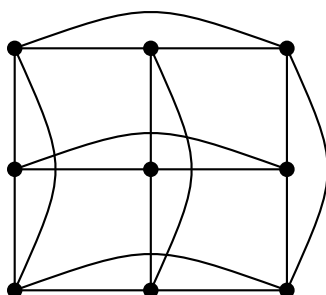


FIG. 1.7 – Le quadrangle généralisé  $GQ(2, 1)$

### 1.4.1 L'hypercube

L'hypercube de dimension  $n$ , noté  $Q_n$ , est le graphe dont tous les sommets représentent les  $n$ -uplets de  $V = \{0, 1\}^n$ , tels que deux vecteurs de  $V$  sont

adjacents si et seulement si ils diffèrent en exactement une coordonnée. Il est noté aussi par l'espace binaire de Hamming de dimension  $n$ .

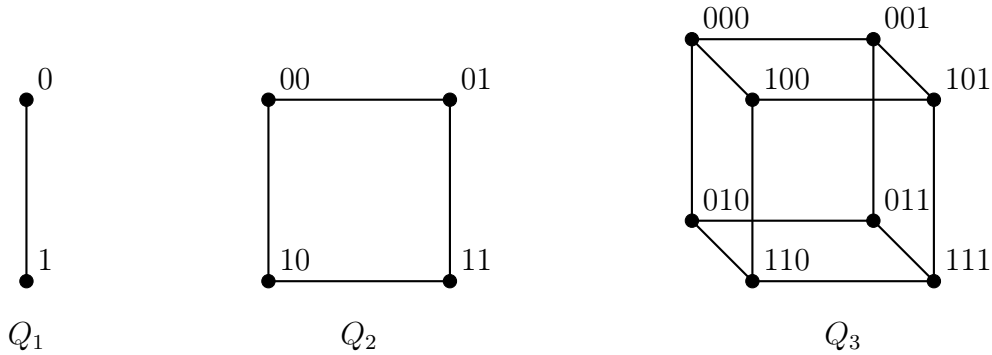


FIG. 1.8 – Les hypercubes de dimensions 1,2 et 3

Par définition des graphes de Hamming, l'hypercube de dimension  $n$  est le produit cartésien de  $K_2$  par lui même :  $Q_n = \square^n K_2$

### Quelques propriétés sur l'hypercube

1.  $\forall n \geq 1, Q_n$  est un graphe  $n$ -régulier,
2.  $\forall n \geq 1, Q_n$  est un graphe hamiltonien,
3.  $\forall n \geq 1, Q_n$  est un graphe biparti tel que l'ensemble des sommets est partitionné en deux ensembles : l'ensemble des vecteurs de poids pair et l'ensemble des vecteurs de poids impairs.

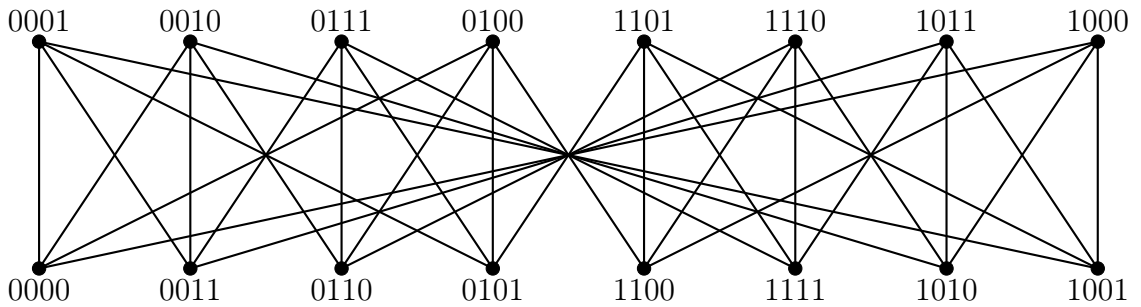


FIG. 1.9 – L'hypercube est un graphe biparti

4.  $\forall n \geq 2, Q_n = Q_{n-1} \square K_2$ .

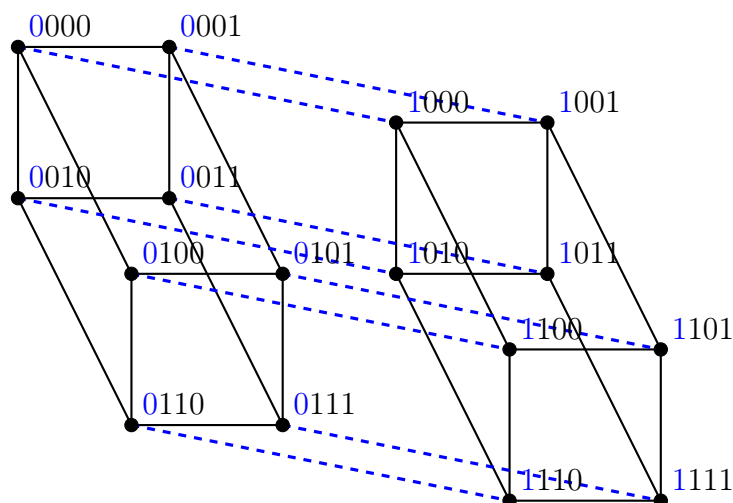


FIG. 1.10 – Hypercube de dimension 4 ou Tesseract

## 1.4.2 Définition

### Distance de Hamming

Soit  $\mathbb{F}_q$  le corps de Galois à  $q$  éléments, et soit  $\mathbb{F}^n$  un espace vectoriel sur le corps  $\mathbb{F}_q$ . La *distance de hamming* entre deux vecteurs  $x = (x_1, \dots, x_n)$ ,  $y = (y_1, \dots, y_n) \in \mathbb{F}^n$  est le nombre des coordonnées qui se diffèrent, c'est à dire,

$$d(x, y) = |\{i/x_i \neq y_i\}|.$$

### Support et Poids

L'ensemble des coordonnées non nuls d'un vecteur  $x \in \mathbb{F}^n$  est dit *support* de  $x$ , et il est noté par  $supp(x)$ .

$$supp(x) = \{i \in \{1, \dots, n\}/x_i \neq 0\}$$

Le poids  $w(x)$  d'un vecteur  $x \in \mathbb{F}^n$  est la cardinalité de son support  $supp(x)$ .

$$w(x) = |supp(x)| = d(x, 0)$$

Dans la suite, on prend  $q = 2$ .

### Fonction de parité

Notant par  $\pi(x)$  la fonction de parité tel que  $\pi(x) = 0$  si  $x$  est de poids pair et  $\pi(x) = 1$  si  $x$  est de poids impair.

### Boule et sphère

Pour tout  $x \in \mathbb{F}^n$ , on note

$$B_r(x) = \{y \in \mathbb{F}^n \mid d(x, y) \leq r\},$$

$$S_r(x) = \{y \in \mathbb{F}^n \mid d(x, y) = r\}$$

L'ensemble  $B_r(x)$  est dit boule (*de Hamming*) de rayon  $r$  centré en  $x$ . Pour  $X \in \mathbb{F}^n$   $B_r(X) = \bigcup_{x \in X} B_r(x)$ . La cardinalité d'une boule de rayon  $r$  dans  $\mathbb{F}^n$  ne dépend pas du choix de centre et il est noté par  $V(n, r)$ , et elle vaut

$$V(n, r) = |B_r(x)| = \sum_{i=0}^r \binom{n}{i}.$$

### Somme directe

Pour deux ensembles  $X \subseteq \mathbb{F}^{n_1}, Y \subseteq \mathbb{F}^{n_2}$ , la somme directe de  $X$  et  $Y$ , noté par  $X \oplus Y$ , est défini par  $X \oplus Y = \{x|y \in \mathbb{F}^{n_1+n_2} : x \in X, y \in Y\}$ , où  $|$  désigne la concaténation des vecteurs.

## 1.5 Problèmes d'Optimisation Combinatoire et Théorie de la Complexité

### Définition d'un problème d'optimisation Combinatoire

Etant donné un couple  $(S, f)$  où

- $S$  est un ensemble fini de solutions réalisables
- $f : S \rightarrow \mathbb{R}$  une fonction objectif ( ou de coût)

Une solution réalisable  $x$  peut être donc décrite par un vecteur de décision  $(x_1, x_2, \dots, x_n)$  à laquelle une valeur objectif  $f(x)$  est associée . Un problème d'optimisation combinatoire revient à trouver la ou les solutions réalisables de coût minimum (resp maximum) de  $S$ , c'est à dire trouver  $x^* \in S$  tel que  $f(x^*) \leq f(x)$  (resp  $f(x^*) \geq f(x)$ ) pour tout  $x \in S$ .

La difficulté majeure de résolution des problèmes d'optimisation combinatoire réside dans l'augmentation exponentielle de la taille de l'espace de recherche lorsque la taille des données augmente.

### Problème de décision

Un problème de décision est un problème qui peut se formuler comme une question à laquelle la réponse est soit "Oui" soit "Non".

### Notions de la théorie de la complexité

Certains problèmes d'optimisation combinatoires disposent depuis longtemps d'algorithmes polynômiaux, tandis que d'autres n'en ont toujours pas. Existe-t-il réellement une classe de problèmes combinatoire pour lesquels on ne trouvera jamais d'algorithmes polynomiaux, ou est-ce que les problèmes difficiles ont en fait de tels algorithmes, mais non encore découverts ?

La Théorie de la complexité a été développée vers 1970 pour répondre à cette question, par des logiciens et des spécialistes en informatique théorique. A cause des difficultés rencontrées, la théorie de la complexité ne traite que des problèmes de décision, car leur formalisme à réponse Oui-Non a permis de les étudier avec les outils de la logique mathématique, et un problème



d'optimisation est donc au moins aussi difficile que son problème de décision associé.

### Problèmes $\mathcal{P}$ et $\mathcal{NP}$

La classe  $\mathcal{P}$  contient tous les problèmes de décision pour lesquels on connaît des algorithmes polynomiaux déterministe pour les résoudre. La Classe  $\mathcal{NP}$  contient les problèmes de décision pour lesquels on connaît des algorithmes non-déterministes polynômiaux (problèmes de décision pour lesquels n'importe quel certificat peut être vérifié en temps polynomial pour une réponse *oui*).

Tout problème de décision avec un algorithme polynomial est dans  $\mathcal{P}$ , et donc dans  $\mathcal{NP}$ . Et on conjecture jusqu'à présent que  $\mathcal{P} \neq \mathcal{NP}$ .

### Problèmes $\mathcal{NP}$ – complet

Parmi l'ensemble des problèmes appartenant à  $\mathcal{NP}$ , il en existe un sous ensemble qui contient les problèmes les plus difficiles : on les appelle les problèmes  $\mathcal{NP}$ -complet. Un problème  $\mathcal{NP}$ -complet possède la propriété que tout problème dans  $\mathcal{NP}$  peut être transformé en celui-ci en temps polynômial. L'existence de problèmes  $\mathcal{NP}$ -complets a été montrée par S. A. Cook dans les années 70 [17].

Signalons que le terme  $\mathcal{NP}$ -difficile est différent du terme  $\mathcal{NP}$ -complet. En effet, un problème d'optimisation combinatoire est  $\mathcal{NP}$ -difficile si le problème de décision associé est  $\mathcal{NP}$  – complet.

Le principal résultat est que tous les problèmes difficiles sont liés : si on arrive à déterminer un algorithme polynomial pour un seul problème  $\mathcal{NP}$ -complet, on pourra en déduire un autre pour tout autre problème difficile de  $\mathcal{NP}$ .

# 2

## Les codes identifiants

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>25</b>
<b>2.2</b>	<b>Définition des codes identifiants</b>	<b>25</b>
2.2.1	Code couvrant d'un graphe	25
2.2.2	Code séparateur d'un graphe	26
2.2.3	Code identifiant d'un graphe	26
<b>2.3</b>	<b>Application pratique des codes identifiants</b>	<b>27</b>
2.3.1	Détection et localisation de défaillance dans les réseaux	28
2.3.2	Systèmes de localisation et de détection dans les environnements fermés munis de capteurs sans fil	29
2.3.3	Conservation d'énergie	29
<b>2.4</b>	<b>Généralisations et variantes</b>	<b>30</b>

2.4.1	Code identifiant à distance $r \geq 1$ . . . . .	30
2.4.2	Code identifiant d'ensemble de sommets . . . . .	30
2.4.3	Code identifiant dans le cas d'un graphe orienté . .	31
2.4.4	Code localisateur dominateur . . . . .	31
2.4.5	Code discriminant . . . . .	32
<b>2.5</b>	<b>Densité d'un code identifiant . . . . .</b>	<b>33</b>
<b>2.6</b>	<b>Premiers résultats . . . . .</b>	<b>34</b>
<b>2.7</b>	<b>Codes identifiants de certains graphes particuliers</b>	<b>36</b>
2.7.1	Codes identifiants dans les cycles . . . . .	36
2.7.2	Codes identifiants dans les arbres . . . . .	37
2.7.3	Codes identifiants dans les grilles et les bandes . .	38
<b>2.8</b>	<b>Quelques résultats sur les codes identifiants dans</b>	
	<b>les espaces de Hamming . . . . .</b>	<b>39</b>
2.8.1	Bornes inférieures . . . . .	39
<b>2.9</b>	<b>Conclusion . . . . .</b>	<b>42</b>

---

## 2.1 Introduction

La notion de codes identifiants est apparue en 1998 par Karpovsky, Chakrabarty et Levitin [31], comme un cas particulier des codes dominants, pour résoudre le problème de détection et de la localisation des défaillances dans un système de microprocesseurs dans l'espace binaire de Hamming.

Dans ce chapitre, nous présentons les codes identifiants en détaillant leurs domaines d'application pratique et nous présentons certaines variantes et généralisations.

Nous terminerons ce chapitre en donnant un aperçu des principaux résultats concernant les codes identifiants. Ces résultats parviennent sur la détermination de bornes inférieures de la cardinalité minimum d'un code identifiant dans des structures particulières tel que les cycles, les arbres, l'hypercube, et sur la densité dans les grilles et les bandes.

## 2.2 Définition des codes identifiants

Soit  $G = (V, E)$  un graphe simple non orienté d'ordre  $n$ . Un sous ensemble non vide  $C$  de  $V$  est dit *un code* et les sommets de  $C$  sont appelés les mots de code.

### 2.2.1 Code couvrant d'un graphe

Si un code  $C$  est tel que tout sommet de  $v$  de  $V \setminus C$  est voisin d'au moins un sommet de  $C$ , alors on dit que  $C$  est un *code couvrant* de  $G$ . Un code couvrant est aussi appelé *dominant* du graphe et on a

$$\forall v \in V, N[v] \cap C \neq \emptyset$$

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

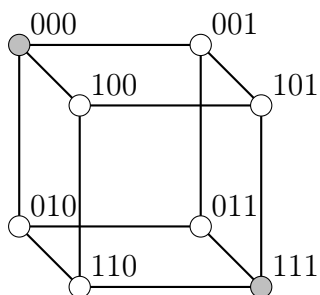


FIG. 2.1 – Le code  $C = \{000, 111\}$  est un code couvrant de  $Q_3$

### 2.2.2 Code séparateur d'un graphe

Un code  $C$  de  $G$  est un *code séparateur* de  $G$  si et seulement si pour toute paire de sommets distincts  $u, v$  de  $G$  on a :

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

ou, d'une façon équivalente

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

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

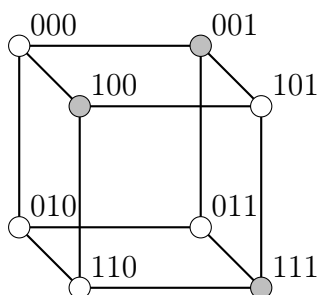


FIG. 2.2 – Le code  $C = \{001, 100, 111\}$  est un code séparateur de  $Q_3$

### 2.2.3 Code identifiant d'un graphe

Un sous ensemble de sommets de  $G$  est appelé un *code identifiant* de  $G$  s'il est à la fois un code couvrant et un code séparateur.

Ainsi, un sous ensemble de sommets  $C$  d'un graphe  $G$  est un code identifiant de  $G$  si et seulement si tous les sommets de  $G$  sont couverts et séparés par  $C$ .

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

**Exemple 2.2.1.** *Le code  $C = \{010, 001, 100, 111\}$  est un code identifiant de  $Q_3$ , et on a :*

$$\begin{aligned}
 I_1(000) &= \{010, 100, 001\} & I_1(001) &= \{001\} & I_1(100) &= \{100\} \\
 I_1(011) &= \{001, 111, 010\} & I_1(101) &= \{001, 100, 111\} & I_1(111) &= \{111\} \\
 I_1(010) &= \{010\} & I_1(110) &= \{010, 100, 111\}
 \end{aligned}$$

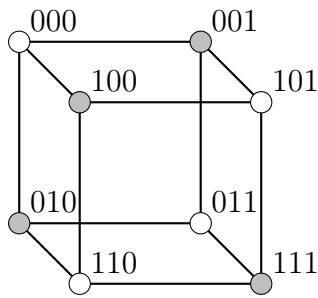


FIG. 2.3 – Les sommets gris identifient tous les sommets de  $Q_3$

## 2.3 Application pratique des codes identifiants

Les codes identifiants ont été définies par M. Karpovsky et al. [31] pour modéliser un problème de détection de défaillance dans les réseaux qui fait partie de la grande famille des problèmes de couverture par tests. Ces problèmes possèdent de nombreuses applications pratiques, dans des domaines variés comme la reconnaissance de formes, de séquençage d'ADN, d'aide au diagnostic médical ou encore en communication dans des réseaux multi-utilisateurs.

### 2.3.1 Détection et localisation de défaillance dans les réseaux

Les techniques de détection traditionnelles modélisent le système de multiprocesseurs par un graphe orienté, dit le graphe test, dont les sommets sont les processeurs constituant le réseau et un arc, dit un lien de test,  $(p_i, p_j)$  de processeur  $p_i$  vers le processeur  $p_j$  indiquent que  $p_i$  teste  $p_j$ . Un arc  $(p_i, p_j)$  est étiqueté par 1 si  $p_i$  détecte la défaillance de  $p_j$  et par 0 sinon. On aura ainsi un vecteur binaire d'informations dont le nombre de bits est égal au nombre d'arcs dans le graphe test qui peut être extrêmement grand dans un réseau avec des milliers de processeurs.

On modélise un réseau de multiprocesseurs par un graphe non orienté  $G = (V, E)$ , où  $V$  est l'ensemble des processeurs et  $E$  est l'ensemble des liaisons dans le réseau. Supposant que chaque processeur  $p$  soit capable de tester le bon fonctionnement de  $p$  et de ses voisins à l'aide d'une procédure  $test(p)$  en retournant une information binaire vaut 1 si une défaillance a été détecté sur le voisinage fermé de  $p$  et 0 sinon. En supposant qu'à tout moment, au plus un processeur soit défectueux, le problème est de trouver un sous ensemble de processeurs  $C$  tel que :

1. si tous les processeurs de  $C$  retournent 0 après l'exécution de test, alors tous les processeurs du système sont en bon état de marche.
2. si au moins un des processeurs de  $C$  renvoie 1 après l'exécution de test, alors il existe un seul processeur défectueux dans le réseau, qu'on est en mesure d'identifier d'après les résultats de test sur  $C$ .

Le problème revient donc à déterminer un code identifiant  $C$  dans l'ensemble des processeurs. En effet, la première condition garantie que  $C$  est un code couvrant du graphe associé au réseau, et la deuxième condition équivaut à ce que  $C$  est un code séparateur.

### 2.3.2 Systèmes de localisation et de détection dans les environnements fermés munis de capteurs sans fil

En 2004, une équipe de l'Université de Boston a utilisé les codes identifiants pour la conception de systèmes de localisation dans des environnements fermés munis de capteurs sans fil. Dans des environnement fermés, les systèmes de localisation basés sur des systèmes de type GPS sont inefficaces du fait de la mauvaise transmission des ondes depuis l'intérieur d'un bâtiment vers l'extérieur (phénomène de réverbération). Une façon d'y remédier est d'équiper les individus de badges émetteurs et disséminer des capteurs sans fil à l'intérieur du bâtiment afin de localiser les porteurs de badges à tout moment. De tel systèmes sont pertinents dans des environnements à risque (usines chimiques, centrales nucléaires, etc) où, en cas de problème, l'ensemble de problème, l'ensemble du personnel doit pouvoir être localisé en vue de l'évacuation du bâtiment.

Le système est basé sur une discrétisation de l'environnement par les zones de portée de capteurs sans fil disséminés dans le bâtiment. Comme pour le problème de détection de défaillance décrit au paragraphe précédent, les capteurs renvoient une information de type binaire : soit le porteur d'un badge est détecté par un capteur, soit il est hors de portée du capteur.

Dans ce contexte, les codes identifiants sont utilisés pour déterminer le placement optimal des capteurs afin de minimiser leur nombre.

### 2.3.3 Conservation d'énergie

Les codes identifiants disjoints, une variante des codes identifiants, à été proposé et développé dans [32] pour la conservation d'énergie dans les réseaux munis de capteurs alimentés par batterie.

Dans cette approche, une zone à couvrir est partitionnée en un ensemble fini de régions et des balises sont placées dans certaines régions telle que chaque région reçoit un ensemble unique de signaux de balises. On peut formuler ce problème par un graphe dont les sommets sont les régions, les balises sont les mots de code identifiant et une arête entre deux sommets si



les deux régions correspondantes sont dans la marge de réception.

Comme les balises sont typiquement alimentées par batterie, le fonctionnement d'un tel système est lié à la durée de vie d'une seule balise. Une approche raisonnable pour prolonger la durée de vie d'un tel système est de balancer les balises actives entre  $\mathcal{K}$  codes identifiants différents. Si de plus ces codes identifiants sont disjoints, alors la durée du vie de système est multiplié par environ  $\mathcal{K}$ .

## 2.4 Généralisations et variantes

### 2.4.1 Code identifiant à distance $r \geq 1$

Soit un entier  $r \geq 1$ . Notons par  $B_r(v)$  la boule de centre  $v \in V$  et de rayon  $r$ .

On dit qu'un code  $C$  est *r-couvrant* si et seulement si tout sommet de  $G$  est à distance inférieur ou égale à  $r$  à au moins un mot de code  $C$ , c'est à dire :

$$\forall v \in V, B_r(v) \cap C \neq \emptyset$$

On dit qu'un code  $C$  est *r-séparateur* si et seulement si pour toute paire de sommets distincts  $u, v$  de  $V$ , il existe un mot de code  $c$  appartenant à l'un des ensembles  $B_r(u) \cap C$  ou  $B_r(v) \cap C$ . D'une autre façon, un code  $C$  est *r-séparateur* si

$$\forall u, v \in V, (B_r(u) \cap C) \Delta (B_r(v) \cap C) \neq \emptyset$$

On dit que le code  $C$  est *r-identifiant* si et seulement s'il est à la fois *r-couvrant* et *r-séparateur*. L'ensemble  $B_r(v) \cap C$  est appelée *ensemble r-identifiant* de sommet  $v$ , noté souvent  $I_r(v)$ .

### 2.4.2 Code identifiant d'ensemble de sommets

Soit  $k \geq 1$  un entier, on dit qu'un sous ensemble de sommets  $C$  est un code identifiant les ensembles d'au plus  $k$  sommets de  $G = (V, E)$  si et seulement si les ensembles identifiants  $I(X, C)$  sont distincts pour tous les

sous ensembles  $X$  de cardinalité au plus  $k$  de  $V$ , où  $I(X, C)$  est défini comme l'union des ensembles identifiants des sommets de  $X$  :

$$I(X, C) = \bigcup_{x \in X} I(x, C) = \bigcup_{x \in X} N[x] \cap C$$

On parle alors de code  $(1, \leq k)$  – *identifiant*.

La combinaison de l'identification à distance  $r \geq 1$  et l'identification d'ensemble de sommets de cardinalité au plus  $k$  donne un code  $(r, \leq k)$  – *identifiant* : tel que un sous ensemble de sommets  $C$  est dit un code  $(r, \leq k)$  – *identifiant* si les sous ensembles  $I_r(X, C)$  sont distincts pour tous les sous ensembles de sommets  $X$  de cardinalité au plus  $k$ , où :

$$I_r(X, C) = \bigcup_{x \in X} B_r(x) \cap C$$

### 2.4.3 Code identifiant dans le cas d'un graphe orienté

Soit  $G = (V, U)$  un graphe simple orienté. Pour un sommet  $v$  de  $V$ , on note par  $\Gamma^-[v]$  l'ensemble fermé des voisins entrants de  $v$  tel que  $\Gamma^-[v] = \Gamma^-(v) \cup \{v\}$ .

La notion de code identifiant est définie pour le graphe orienté  $G$  en remplaçant  $N[v]$  par  $\Gamma^-[v]$ . Ainsi, un sous ensemble de sommets  $C$  de  $V$  est un code identifiant de  $G$  si et seulement si :

$$\forall v \in V, \Gamma^-(v) \cap C \neq \emptyset \quad \text{et} \quad \forall u, v \in V, \Gamma^-(u) \cap C \neq \Gamma^-(v) \cap C$$

On peut également identifier des sommets à distances  $r \geq 1$  ou identifier des ensembles d'au plus  $k$  sommets dans les graphes orientés.

### 2.4.4 Code localisateur dominateur

On parle de code *localisateur dominateur* dans le cas où les sommets du code n'ont pas à être identifiés. Ainsi, un code  $C$  est localisateur dominateur (respectivement  $r$ -localisateur dominateur) si et seulement s'il est couvrant (respectivement  $r$ -couvrant) et  $N[u] \cap C \neq N[v] \cap C$  (respectivement  $I_r(u) \neq I_r(v)$ ) pour toute paire de sommets distincts de  $V \setminus C$ .

De cette définition, on déduit qu'un code  $r$ -identifiant est un code  $r$ -localisateur dominateur mais la réciproque est fautive (voir la figure 2.5).

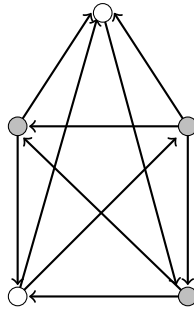


FIG. 2.4 – L'ensemble des sommets en gris est un code identifiant du graphe orienté mais pas du graphe non orienté  $K_5$

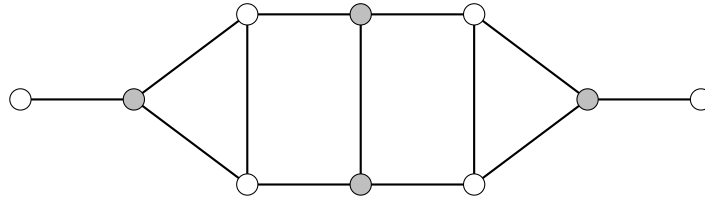


FIG. 2.5 – Les sommets en gris constituent un code localisateur-dominateur mais pas un code identifiant

### 2.4.5 Code discriminant

Soit  $G = (I \cup A, E)$  un graphe biparti non orienté. Un sous ensemble  $C$  de  $A$  est dit un *code discriminant* de  $G$  si et seulement si :

$$\forall i \in I : N(i) \cap C \neq \emptyset \quad \text{et} \quad \forall i \neq j \in I : N(i) \cap C \neq N(j) \cap C$$

L'ensemble  $I$  peut être vue comme un ensemble d'*individus* et  $A$  comme l'ensemble des *attributs*, où une arête entre  $i \in I$  et  $a \in A$  existe si l'individu  $i$  possède l'attribut  $a$ . Un code discriminant est alors un ensemble des attributs qui permet de différencier tous les individus.

#### Code $r$ -discriminant dans un espace de Hamming

Notant par  $\mathcal{E}^n$  (respectivement,  $\mathcal{O}^n$ ) l'ensemble des vecteurs binaires de longueur  $n$  de poids pair (respectivement, impair). Un code  $C \in \mathcal{E}^n$  est dit  $r$ -discriminant si les ensembles  $B_r(x) \cap C, x \in \mathcal{O}^n$  sont tous différents et non vides.

### Relation entre les codes $r$ -discriminant et $r$ -identifiant dans l'hypercube [13]

Les deux notions des codes  $r$ -identifiant et  $r$ -discriminant sont équivalentes dans le cas de l'espace de Hamming, dans le sens où pour tout entier impair  $r$ , il existe une bijection entre l'ensemble des codes  $r$ -identifiant de  $\mathbb{F}_2^n$  et l'ensemble des codes  $r$ -discriminant dans  $\mathbb{F}_2^{n+1}$ .

Notant par  $\pi(x)$  la fonction de parité tel que  $\pi(x) = 0$  si  $x$  est de poids pair et  $\pi(x) = 1$  si  $x$  est de poids impair, alors le vecteur  $x|\pi(x)$  est de poids pair. Notant par  $D_r(n)$  la cardinalité minimum d'un code  $r$ -discriminant dans  $\mathbb{F}_2^{n+1}$ .

**Théorème 2.4.1** ([13]). *Soient  $n \geq 2, p \geq 0$  tels que  $2p + 1 < n$ , et soit  $C \subseteq \mathbb{F}_2^n$  un code  $2p + 1$ -identifiant. Alors le code*

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

*est un code  $(2p + 1)$ -discriminant dans  $\mathbb{F}_2^{n+1}$ . De même,*

$$D_{2p+1}(n + 1) \leq M_{2p+1}(n) \tag{2.1}$$

**Théorème 2.4.2** ([13]). *Soient  $n \geq 3, p \geq 0$  tels que  $2p + 2 < n$ . Soit  $C \subseteq \mathcal{E}^n$  un code  $(2p + 1)$ -discriminant et soit  $C' \subseteq \mathbb{F}_2^{n-1}$  un code obtenu par la suppression de n'importe quelle coordonnée de  $C$ . Alors  $C'$  est un code  $(2p + 1)$ -identifiant de  $\mathbb{F}_2^{n-1}$ . De même,*

$$M_{2p+1}(n - 1) \leq D_{2p+1}(n) \tag{2.2}$$

**Corollaire 2.4.1** ([13]). *Pour tout  $n \geq 2$  et  $p \geq 0$  tels que  $2p + 1 < n$ , on a :*

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

## 2.5 Densité d'un code identifiant

Soit  $G$  un graphe et soit  $(v_0)$  un sommet quelconque de  $G$ . Pour tout entier positif  $t$ , soit  $B_t(v_0)$  la boule de rayon  $t$  centrée en  $v_0$ . La densité

$d(C, G)$  d'un code  $r$ -identifiant  $C$  de  $G$  est définie comme étant la limite :

$$d(C, G) = \limsup_{t \rightarrow \infty} \frac{|C \cup B_t(v_0)|}{|B_t(v_0)|}$$

On note  $d_r^*(G)$  la densité minimum d'un code  $r$ -identifiant de  $G$ . Dans le cas où  $G$  est fini, on a

$$d_r^*(G) = \frac{|C^*|}{|G|}$$

où  $C^*$  est un code  $r$ -identifiant de cardinalité minimum de  $G$ .

## 2.6 Premiers résultats

### Existence d'un code identifiant

Une condition nécessaire et suffisante pour qu'un graphe  $G = (V, E)$  admette un code  $r$ -identifiant est qu'il ne contient pas de sommets jumeaux (deux sommets avec le même ensemble de voisinage fermé à distance  $r \geq 1$ ). Dans ce cas, l'ensemble des sommets  $V$  est toujours un code  $r$ -identifiant de  $G$ . Le problème est donc de trouver le nombre minimum des mots de codes nécessaires pour  $r$ -identifier chaque sommet de  $G$ . On note par  $M_r(G)$  la cardinalité minimum d'un code  $r$ -identifiant, lorsqu'il existe, dans un graphe  $G = (V, E)$  lorsqu'une boule de rayon  $r \geq 1$  est utilisée.

Un sous ensemble de sommets  $C$  est un code  $r$ -identifiant dans un graphe  $G = (V, E)$  si et seulement si  $C$  est un code 1-identifiant du graphe  $G^r$  la fermeture  $r$ -transitive de  $G$ .

Il est à noter qu'un graphe admettant un code  $r$ -identifiant admet aussi un code  $r'$ -identifiant pour tout  $r' \leq r$ .

### Complexité

Le problème d'optimisation associé est le suivant :

**Données** Un graphe  $G$  admettant un code  $r$ -identifiant.

**Question** Quelle est la cardinalité minimum d'un code  $r$ -identifiant de  $G$  ?

Il a été montré dans [11] que ce problème était *NP*-difficile pour tout  $r \geq 1$ , et ce même lorsque l'on se restreignait aux graphes bipartis. Dans le

cas des graphes orientés, le problème est aussi  $NP$ -difficile pour tout  $r \geq 1$  [10].

### Bornes inférieures

Soit  $V_r(i)$  la taille de boule de rayon  $r$  et de centre  $v_i, i \in \{1, \dots, n\}$ . Les premières bornes inférieures sur  $M_r(G)$  ont été donné par le théorème 1 de [31]. La première borne inférieure vient de fait qu'avec les éléments de  $C$ , il est possible de construire  $2^{|C|-1}$  mots distincts. Pour pouvoir identifier l'ensemble des sommets du graphe, on doit avoir  $n \leq 2^{|C|-1}$ .

**Théorème 2.6.1** (Karpovsky et al.[31]). *Soit un graphe  $G = (V, E)$  pour un graphe à  $n$  sommets, on a :*

- i)  $M_r(G) \geq \lceil \log_2(n+1) \rceil$
- ii) *Soit  $n/2 \geq V_r(1) \geq V_r(2) \geq \dots \geq V_r(n)$  alors  $M_r(G) \geq K$ , où  $K$  est le plus petit entier tel que :*

$$\sum_{i=1}^K h\left(\frac{V_r(i)}{n+1}\right) \geq \log_2(n+1)$$

où  $h(x) = -x \log_2 x - (1-x) \log_2(1-x)$  est la fonction binaire d'entropie.

- iii) *Soit  $K$  le plus petit entier tel que pour un certain  $l(1 \leq l \leq \min(K, V_r(1)))$ , les conditions suivantes sont satisfaites :*

$$n \leq \sum_{j=1}^{l-1} \binom{K}{j} + \left\lfloor \frac{1}{l} \left( \sum_{i=1}^K V_r(i) - \sum_{j=1}^{l-1} j \binom{K}{j} \right) \right\rfloor \quad (2.3)$$

$$\sum_{j=1}^{l-1} j \binom{K}{j} < \sum_{i=1}^K V_r(i) \leq \sum_{j=1}^l j \binom{K}{j} \quad (2.4)$$

Alors  $M_r(G) \geq K$ .

## 2.7 Codes identifiants de certains graphes particuliers

### 2.7.1 Codes identifiants dans les cycles

Le problème de minimisation d'un code  $r$ -identifiant dans un cycle  $C_n$  a été quasiment résolu. Notant qu'un cycle  $C_n$  admet un code  $r$ -identifiant si et seulement si  $n \geq 2r + 1$ . Pour le cas où  $n$  est pair, Bertrand et al. ont donné le Théorème suivant [3] :

**Théorème 2.7.1** (Bertrand et al. [3]). *Pour  $r \geq 1$ , on a  $M_r(C_{2r+2}) = 2r + 1$  et  $M_r(C_n) = \frac{n}{2}$  pour  $n \geq 2r + 4$  pair.*

Dans [23], Gravier et al. ont défini le graphe  $C'_{(n,r)}$  dont l'ensemble des sommets est  $\{v_i : i \in \mathbb{Z}_n\}$  où tous les indices sont pris modulo  $n$  tel que, pour tout  $i \in \mathbb{Z}_n$ ,  $v_{i-r}v_{i+r+1}$  est une arête de  $C'_{(n,r)}$ . Par l'utilisation de graphe  $C'_{(n,r)}$ , ils ont prouvé le Théorème suivant :

**Théorème 2.7.2** (Gravier et al. [23]). *Pour tout  $r \geq 1$  et  $n \geq 2r + 3$ ,  $n$  impair, on a :*

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

De ce théorème, des valeurs exactes de  $M_r(n)$  sont trouvés pour  $n \geq 3r + 2$  et  $\text{pgcd}(2r+1, n) \neq 1$  (voir le tableau 2.1). Dans [36], Min Xu et al. ont traité le cas où  $n \geq 3r + 2$  et  $\text{pgcd}(2r+1, n) = 1$  :

**Théorème 2.7.3** (Min Xu et al. [36]). *Pour  $r \geq 1$ ,  $n$  un entier impair tel que  $n \geq 3r + 2$  et  $\text{pgcd}(2r+1, n) = 1$ . Si  $n = 2m(2r+1) + 1$  ou  $n = (2m+1)(2r+1) + 2r$  pour  $m \geq 1$ , alors  $M_r(C_n) = \frac{n+1}{2} + 1$  ; sinon  $M_r(C_n) = \frac{n+1}{2}$*

$n$ impair	$\text{pgcd}(2r + 1, n) \neq 1$	$\text{pgcd}(2r + 1, n) = 1$
$n = 2r + 3$	$\lfloor \frac{2n}{3} \rfloor$	$\lfloor \frac{2n}{3} \rfloor$
$3r + 3 \leq n \leq 4r + 1$	$\frac{n+1}{2} + \frac{\text{pgcd}(2r+1, n)-1}{2}$	$\frac{n+1}{2}$
$n = 4r + 3$	$2r + 3$	$2r + 3$
$4r + 5 \leq n \leq 8r + 1$	$\frac{n+1}{2} + \frac{\text{pgcd}(2r+1, n)-1}{2}$	$\frac{n+1}{2}$
$n \geq 8r + 3$	$\frac{n+1}{2} + \frac{\text{pgcd}(2r+1, n)-1}{2}$	$\left\{ \begin{array}{ll} \frac{n+1}{2} + 1 & \text{si } n = 2m(2r + 1) + 1 \\ & \text{ou } n = (2m + 1)(2r + 1) + 2r, m \geq 1 \\ \frac{n+1}{2} & \text{sinon.} \end{array} \right.$

 TAB. 2.1 – Les valeurs de  $M_r(C_n)$  dans le cas où  $n$  est impair

## 2.7.2 Codes identifiants dans les arbres

Dans [4], Bertrand, Charon, Hudry et Lobstein ont établi la borne inférieure suivante sur la cardinalité minimum d'un code identifiant dans les arbres non orientés (on peut se référer à [12] pour le cas des arbres orientés enracinés) :

**Théorème 2.7.4** (Bertrand et al. [4]). *Si  $T$  est un arbre d'ordre  $n \geq 3$ , alors*

$$M(T) \geq \frac{3(n+1)}{7}$$

Cette borne inférieure a été améliorée par Blidia et al. dans le théorème 3.1 de [7] en introduisant le nombre de supports et le nombre de sommets pendants associés à l'arbre :

**Théorème 2.7.5** (Blidia et al. [7]). *Si  $T$  est un arbre d'ordre  $n \geq 4$ , avec  $l$  sommets pendants et  $s$  supports, alors :*

$$M(T) \geq \frac{3(n+l-s+1)}{7}$$



### 2.7.3 Codes identifiants dans les grilles et les bandes

#### Grille infinie

La grille  $n$ -dimensionnelle (infinie) est le produit cartésien de chemins infinis  $P_\infty$ , dont l'ensemble de sommets est  $\mathbb{Z}^n$  et l'ensemble d'arêtes  $\{uv, d_1(u, v) = \sum_{i=1}^n |v_i - u_i| = 1\}$  où  $d_1(u, v)$  (distance de Manhattan) représente la longueur d'un plus court chemin entre  $u$  et  $v$ . Le mot grille désigne généralement la grille bidimensionnelle  $\mathbb{Z}^2$ .

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

**Théorème 2.7.6** (Cohen et al. [15], Ben-Haim et Litsyn [1]).

$$d_1^*(\mathbb{Z}^2) = \frac{7}{20}$$

Ce résultat a été prouvé en deux temps. En 1999, Cohen et al. [15] ont prouvé que  $d_1^*(\mathbb{Z}^2) \leq \frac{7}{20}$ , et en 2005, Ben-Haim et Litsyn [1] ont montré que  $d_1^*(\mathbb{Z}^2) \geq \frac{7}{20}$ .

#### Bande

On désigne par *bande* de hauteur  $k$  le sous graphe  $\mathcal{S}_k$  de la grille induit par le sous ensemble de sommets  $\{1, \dots, k\} \times \mathbb{Z}$ , et par *demi-bande* de hauteur  $k$  le sous graphe  $\mathcal{S}_k^+$  de la grille induit par le sous ensemble de sommet  $\{1, \dots, k\} \times \mathbb{N}$ .

Seules les bandes ont été étudiées car les résultats sur les demi bandes se déduisant trivialement de la proposition suivante :

**Proposition 2.7.1** ([3]). *Pour tout  $k \geq 1$ , on a :*

$$d_1^*(\mathcal{S}_k) = d_1^*(\mathcal{S}_k^+)$$

Daniel, Gravier et Moncel (2004) ont prouvé que  $d_1^*(\mathcal{S}_1) = \frac{1}{2}$  et  $d_1^*(\mathcal{S}_2) = \frac{3}{7}$  [18]. Plus généralement, on a :

**Théorème 2.7.7** (Daniel et al. [18]). *Pour tout  $k \geq 3$ , on a :*

$$\frac{7}{20} - \frac{1}{2k} \leq d_1^*(\mathcal{S}_k) \leq \min\left(\frac{2}{5}, \frac{7}{20} - \frac{2}{k}\right)$$

### grille finie

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

Pour les grilles finies, on a les bornes suivantes :

**Théorème 2.7.8** (Daniel et al. [18]). *Pour tout  $k, n \geq 2$ , on a :*

$$\frac{7}{20} - \frac{1}{2} \frac{k+n}{kn} \leq d_1^*(\mathcal{G}_{k \times n}) \leq \frac{7}{20} + 2 \frac{k+n-2}{kn}$$

## 2.8 Quelques résultats sur les codes identifiants dans les espaces de Hamming

Dans cette section, on va présenter quelques résultats sur les codes  $r$ -identifiants dans les espaces de Hamming.

**Conjecture 2.8.1** (U. Blass, I. Honkala et S. Litsyn, 2001). *Pour tout  $n \geq 2$  et  $r \geq 1$ , on a  $M_r(n) \leq M_r(n+1)$ .*

Cette conjecture n'a pas été prouvée que pour  $r = 1$  dans [34].

### 2.8.1 Bornes inférieures

Une première borne inférieure pour les graphes réguliers est donné par [31]. En effet, il existe au plus  $|C|$  vecteurs  $r$ -couverts par exactement un mot de code  $C$ . Tous les autres vecteurs sont  $r$ -couverts par au moins deux mots de code. Ainsi, en comptant le nombre des paires  $\{x, c\}$  où  $x \in \mathbb{F}^n$ ,  $c \in C$  et  $d(c, x) \leq r$  par deux manière, on aura  $M_r(n)V(n, r) \geq M_r(n) + 2(2^n - M_r(n))$ , d'où :

**Théorème 2.8.1** (Karpovsky et al. [31]).

$$M_r(n) \geq \frac{2^{n+1}}{V(n, r) + 1}$$

Cette borne a été améliorée dans le Théorème 2 de [20] :

**Théorème 2.8.2** (Exoo et al. [20]). *Soit  $C \subseteq \mathbb{F}^n$  un code  $r$ -identifiant et  $m = \max\{|I_r(x)| : x \in \mathbb{F}^n\}$ . Soit*

$$f_r(x) = \frac{(x-2)\binom{2r}{r} - 1}{\binom{2r}{r} + \binom{x}{2} - 1}.$$

On a

$$|C| \geq \frac{2^n(2 + f_r(v))}{V(n, r) + f_r(v) + 1}, \text{ où } v = \begin{cases} m & \text{si } m \geq 2 + 2\binom{2r}{r} \\ 3 & \text{sinon.} \end{cases}$$

On dit qu'un ensemble  $A$   $r$ -identifie un autre ensemble  $B$  si pour tous  $x, y \in B, x \neq y$ , on a  $A \cap B_r(x) \neq A \cap B_r(y)$ .

D'après le théorème 4 de [31], on a  $S_1(0)$  1-identifie  $B_2(0)$ . En substituant  $r = 1$  et  $m = |S_1(0)|$  dans le Théorème précédent, on aura la meilleure borne inférieure connue pour les codes 1-identifiants donnée par le théorème 3 de [31].

**Corollaire 2.8.1** (Karpovsky et al.[31]).

$$M_1(n) \geq \frac{n2^{n+1}}{2 + n + n^2} = \frac{n2^n}{V(n, 2)}$$

Blass, Honkala et Litsyn ont prouvé dans la construction 3 de [5] que tous les vecteurs de poids deux, sauf ceux de support  $\{i, i + 1\}, i = 1, \dots, n - 1$  et  $\{1, n\}$  (où  $n \geq 7$ ), 2-identifient tous les vecteurs de  $B_4(\mathbf{0})$ . Ainsi,  $\binom{n}{2} - n$  vecteurs dans  $S_2(\mathbf{0})$  2-identifient les vecteurs dans  $B_4(\mathbf{0})$  pour  $n \geq 7$ . Posons  $m = \binom{n}{2} - n$  dans le Théorème 2.8.2, on obtient alors le résultat suivant [20].

**Corollaire 2.8.2** ([20]). *pour  $n \geq 7$ , on a :*

$$M_2(n) \geq \frac{2^{n+2}(n^3 - 6n^2 + 17n - 24)}{n^5 - 5n^4 + 5n^3 - 11n^2 + 114n - 56}$$

D'après le théorème 5 de [31], l'ensemble  $S_r(\mathbf{0})$   $r$ -identifie tous les vecteurs de  $B_{2r}$  pour  $r < n/2$ . Le choix de  $m = \binom{n}{r}$  implique les résultats suivants [20].

**Corollaire 2.8.3** ([20]). *Si  $\binom{n}{r} \geq 2 + 2\binom{2r}{r}$  et  $r < n/2$ , alors*

$$M_r(n) \geq \frac{2^n (2 + f_r(\binom{n}{r}))}{V(n, r) + f_r(\binom{n}{r}) + 1}.$$

**Corollaire 2.8.4** ([20]). *Si  $2r - 1$  divise  $n$ ,  $r \geq 3$  et  $R = \binom{n}{r} - \binom{2r-1}{r} \left(\frac{n}{2r-1}\right)^r$ , on a :*

$$M_r(n) \geq \frac{2^n (2 + f_r(V(n, r - 1) + R))}{V(n, r) + f_r(V(n, r - 1) + R) + 1}.$$

Pour  $n$  assez grand, Honkala et Lobstein [28] ont trouvé le résultat asymptotique suivant :

**Théorème 2.8.3** (I.Honkala et A.Lobstein [28]). *Soit  $0 \leq \rho < 1$ . Alors*

$$\lim_{n \rightarrow \infty} \frac{\log_2 M_{\lfloor \rho n \rfloor}}{n} = 1 - H(\rho),$$

où  $H(x) = -x \log_2 x - (1 - x) \log_2(1 - x)$

Dans le cas où  $\rho = 1/2$ . Le théorème suivant donne une famille de codes identifiants de petites cardinalités.

**Théorème 2.8.4** (I.Honkala et A.Lobstein [28]). *Pour tout  $n \geq 3$ ,*

$$M_{\lfloor n/2 \rfloor}(n) \leq \begin{cases} \frac{n^2 - n + 2}{2}, & \text{si } n \text{ est impair;} \\ \frac{n^2 - 4}{2}, & \text{si } n \text{ est pair.} \end{cases}$$

Nous terminons cette section par le résultat suivant :

**Théorème 2.8.5** (U. Blass, I. Honkala et S. Litsyn[5]).

$$M_r(r + 1) = 2^{r+1} - 1, r \geq 1.$$

## 2.9 Conclusion

Dans ce deuxième chapitre, nous avons présenté un état de l'art sur le domaine des codes identifiants. Les principaux résultats sur la détermination de bornes inférieures de la cardinalité minimum d'un code identifiant dans des structures particulière nous ont permis de situer le thème de notre recherche et de circonscrire les objectifs de notre travail.

# 3

## Construction de codes identifiants par la Recherche Tabou

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>45</b>
<b>3.2</b>	<b>Les Méta-Heuristiques</b>	<b>45</b>
3.2.1	Présentation générale	45
3.2.2	Aperçu sur les métaheuristiques	46
<b>3.3</b>	<b>Construction des codes <math>r</math>-identifiants dans <math>Q_n</math></b>	<b>49</b>
3.3.1	Construction théorique	50
3.3.2	Heuristiques : la méthode de bruitage et l'algorithme glouton	51
3.3.3	Application	52
<b>3.4</b>	<b>Générer des codes <math>r</math>-identifiants dans <math>Q_n</math> par une Recherche Tabou</b>	<b>52</b>

3.4.1 Principe de la méthode . . . . . 52

3.4.2 Expérimentation et Résultat . . . . . 55

**3.5 Conclusion . . . . . 56**

---

## 3.1 Introduction

Lorsqu'un graphe  $G$  admet un code  $r$ -identifiant, alors  $V(G)$  est toujours un code  $r$ -identifiant de  $G$ , ainsi le problème d'optimisation associé est :

**Instance :** Un graphe  $G$  admettant un code  $r$ -identifiant.

**Question :** Quelle est la cardinalité minimum d'un code  $r$ -identifiant de  $G$ ?

Charon, Hudry et Lobstein ont prouvé que ce problème est  $\mathcal{NP}$ -difficile, aussi bien dans les graphes orientés [10] que dans les graphes non orientés [11], et ce même lorsque l'on se restreignait aux graphes bipartis.

De ce fait, cela nous amène à dire que nous ne pouvons espérer guère mieux qu'un algorithme d'énumération exhaustive pour résoudre ce problème dans le cas général.

En s'inspirant du travail de I.Charon et al. dans [14] sur les codes identifiants dans  $Q_n$ , on va appliquer une métaheuristique *Recherche Tabou* afin d'essayer de construire de bons codes  $r$ -identifiants dans un espace binaire de Hamming de dimension  $n$ .

## 3.2 Les Méta-Heuristiques

### 3.2.1 Présentation générale

Une méthode approchée ou heuristique (heuristic, approximation method) pour un problème d'optimisation combinatoire est un schéma à adopter au données de problème et qui a pour but de trouver une solution réalisable, tenant compte de la fonction objectif, mais sans garantie d'optimalité. On oppose les méthodes approchées aux méthodes exactes qui trouvent toujours l'optimum si on leur en laisse le temps (énumération complète, méthodes arborescentes, programmation dynamique, etc...).

Les méthodes exactes ont une complexité exponentielle sur les problèmes  $NP$ -difficiles, et seules les heuristiques peuvent résoudre des cas de grande taille.



Il existe un très grand nombre d'heuristiques selon les problèmes à traiter. On distingue cependant trois grands types non exhaustifs car certaines heuristiques sont inclassables ou peuvent combiner plusieurs types.

**Méthodes constructives** : construisant une seule solution par une suite de choix partiels et définitifs, c'est à dire sans retours en arrière. On les appelle méthodes gloutonnes quand elles cherchent à chaque itération à faire le choix le plus avantageux.

**Recherches locales** : On part d'une solution initiale et, par transformations successives, on construit une suite de solutions de coût décroissant. Le processus s'arrête quand on ne peut plus améliorer la solution courante.

**Métaheuristiques** : Une recherche locale peut être piégée dans un minimum local. Les métaheuristiques peuvent échapper à ces minima en construisant aussi une suite de solutions, mais dans laquelle la fonction économique peut temporairement augmenter.

### 3.2.2 Aperçu sur les métaheuristiques

Dans cette partie nous allons présenter quelques métaheuristiques les plus utilisées dans la littérature pour les problèmes d'optimisation combinatoire.

#### Recherche local

La recherche locale démarre d'une solution initiale et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage de la solution courante. Le processus de la recherche est réitéré jusqu'à ce qu'aucune amélioration de la solution courante ne pourrait être faite. Le principal défaut des méthodes de descente est qu'elles s'arrêtent au premier minimum local rencontré. La méthode de recherche locale la plus élémentaire est la méthode de descente. Elle peut être décrite comme suit.

Pour éviter d'être bloqué au premier minimum local rencontré, on peut décider d'accepter, sous certaines conditions, de se déplacer d'une solution  $s$  vers une solution  $s' \in N(s)$  telle que  $f(s') \geq f(s)$ . C'est ce que font les méthodes que nous décrivons ci-dessous.

---

**Algorithm 1** Méthode de descente

---

- 1: choisir une solution réalisable  $s \in S$
  - 2: Déterminer une solution  $s'$  qui minimise  $f$  dans  $N(s)$ .
  - 3: if  $f(s') < f(s)$ , alors poser  $s := s'$  et retourner à 2, sinon STOP.
- 

**Recuit simulé**

Le recuit simulé est une technique d'optimisation qui est inspirée des méthodes de simulation de la mécanique statistique dans les années 1950. La méthode de recuit simulé considère une solution initiale et recherche dans son voisinage une autre solution de façon aléatoire. L'originalité de cette méthode est qu'il est possible de se diriger vers une solution voisine de moins bonne qualité avec une probabilité non nulle, ceci permet d'échapper aux optima locaux. Le processus du recuit simulé répète une procédure itérative qui cherche des configurations de coût le plus faible tout en acceptant de manière contrôlée des configurations qui dégradent la fonction coût.

---

**Algorithm 2** Recuit simulé

---

- 1: Choisir une solution  $s \in S$  ainsi qu'une température initiale  $T$ .
  - 2: Tant qu'aucun critère d'arrêt n'est satisfait faire
  - 3:     Choisir aléatoirement  $s' \in V(s)$ ;
  - 4:     Générer un nombre réel aléatoire  $r$  dans  $[0,1]$ ;
  - 5:     Si  $r < p(T, s, s')$  alors poser  $s := s'$ ;
  - 6:     Mettre à jour  $T$ ;
  - 7: Fin du tant que
- 

La fonction  $p(T, s, s')$  est généralement choisie comme étant égale à la distribution de Boltzmann  $\exp \frac{f(s)-f(s')}{T}$ . Ainsi,

- Si  $f(s') < f(s)$  alors  $\exp \frac{f(s)-f(s')}{T} > 1$ , ce qui signifie que  $r < p(T, s, s')$  et on accepte donc  $s'$ .
- Si  $T$  a une très grande valeur alors  $\exp \frac{f(s)-f(s')}{T} \approx 1$ , et on est donc presque sûr d'accepter  $s'$ .
- Si  $T$  a une très petite valeur et si  $f(s') > f(s)$  alors  $\exp \frac{f(s)-f(s')}{T} \approx 0$  et on va donc probablement refuser  $s'$ .

### Recherche tabou

Les méthodes tabou ont été introduites par Glover en 1989 [21, 22]. Elles n'ont aucun caractère stochastique et elles ont trois caractéristiques fondamentales :

1. A chaque itération, on examine complètement le voisinage  $V(s)$  de la solution actuelle  $s$ , et on va sur la meilleur solution  $s'$ , même si le coût remonte.
2. On s'interdit de revenir sur une solution visitée dans un passé proche grâce à une liste tabou  $T$  de longueur limité  $NT$ , qui stocke de manière compacte les  $NT$  dernières solutions visitées. On cherche donc  $s'$  dans  $V(s) - T$ .
3. On conserve en cours de route la meilleure solution trouvée  $s^*$ . On s'arrête après un nombre maximal  $NMax$  d'itérations, ou après un nombre maximal d'itérations sans améliorer la meilleure solution, ou quand  $V(s) - T = \emptyset$ . Ce dernier cas, très rare, ne peut se produire que sur de très petits problèmes pour lesquels le voisinage tout entier peut être contenu dans  $T$ .

La Recherche Tabou peut être décrite comme suit

---

**Algorithm 3** Recherche Tabou

---

- 1: Choisir une solution  $s \in S$ , poser  $T := \emptyset$  et  $s^* := s$ ;
  - 2: Tant qu'aucun critère d'arrêt n'est satisfait faire
  - 3:     Déterminer une solution  $s'$  qui minimise  $f(s')$  dans  $V(s)$
  - 4:     Si  $f(s') < f(s^*)$  alors poser  $s^* := s'$ ;
  - 5:     Poser  $s := s'$  et mettre à jour  $T$ ;
  - 6: Fin du tant que
- 

### Algorithmes génétiques

Les algorithmes génétiques sont inspirés de la théorie de l'évolution et des processus biologiques qui permettent à des organismes de s'adapter à leur environnement. Ils ont été introduits dans des années 60 par Holland.

Les Algorithmes génétiques font évoluer un ensemble de solutions candidates, appelé une *population d'individus*. Un individu n'est autre qu'une solution possible du problème à résoudre. Chaque individu de cette population se voit attribuer une fonction appelée fonction d'adaptation qui permet de mesurer sa qualité ou son poids ; cette fonction d'adaptation peut représenter la fonction objectif à optimiser. Ensuite, les meilleurs individus de cette population sont sélectionnés, subissent des croisements et des mutations et une nouvelle population de solutions est produite pour la génération suivante. Ce processus se poursuit, génération après génération, jusqu'à ce que le critère d'arrêt soit atteint, comme par exemple le nombre maximal de générations.

---

**Algorithm 4** Algorithme génétique standard

---

- 1: Population initiale de la génération  $t = 0$ .
  - 2: Tant qu'aucun critère d'arrêt n'est satisfait faire
  - 3:      $t := t + 1$  ;
  - 4:     Evaluation de la fonction d'adaptation de chaque individu ;
  - 5:     Selection des individus ;
  - 6:     Opérateurs de croisement et de mutations ;
  - 7:     Création de la nouvelle population  $P(t)$  ;
  - 8: Fin du tant que
- 

### 3.3 Construction des codes $r$ -identifiants dans

$Q_n$

Dans cette section on présentera les méthodes de constructions des codes  $r$ -identifiants dans  $Q_n$  données par I. Charon, G. Cohen, O. Hudry et A. Lobstein dans [14]. Ils ont donné des bornes supérieures numérique sur  $M_r(n)$ ,  $n \leq 21$  en utilisant des heuristiques et des constructions théoriques.

### 3.3.1 Construction théorique

On utilise la notation  $(r, n)$  ou  $(r, n)K$  pour désigner un code  $r$ -identifiant de  $\mathbb{F}^n$  et qui admet  $K$  éléments. Le théorème suivant montre comment à partir d'un  $(r, n)$  code  $C$ , la somme directe  $C \oplus \mathbb{F}^p$  peut être utilisée pour la construction d'un  $(r, n + p)$  code [14].

**Théorème 3.3.1** (Charon et al. [14]). *Soient  $r \geq 1, p \geq 1$ , et  $k \in \{0, 1, \dots, p-1\}$ ; soit  $C$  un  $(r, n)$  code et*

$$X_p = \{x \in \mathbb{F}^n : \forall c \in C, d(x, c) \leq r - p \text{ ou } d(x, c) > r\}$$

*Construction  $\mathcal{C}_1$  : Soit  $Y_p \subseteq \mathbb{F}^n$  un ensemble tel que pour tout  $x \in X_p$ , il existe  $y \in Y_p$  avec  $r - p + 1 \leq d(x, y) \leq r$ . Alors*

$$C' = (C \oplus \mathbb{F}^p) \cup (Y_p \oplus (\mathbb{F}^p \setminus \{0^p\})) \quad (3.1)$$

*est un  $(r, n + p)$ .*

*Construction  $\mathcal{C}_2$  : Soit  $Y_{p,k} \subseteq \mathbb{F}^n$  un ensemble tel que pour tout  $x \in X_p$ , il existe  $y \in Y_{p,k}$  avec  $d(x, y) = r - k$ , et soit  $C_{p,k}$  un code  $k$ -séparateur dans  $\mathbb{F}^p$ . Alors :*

$$C' = (C \oplus \mathbb{F}^p) \cup (Y_{p,k} \oplus C_{p,k}) \quad (3.2)$$

*est un  $(r, n + p)$ .*

En préférence, il faut choisir  $Y_p$  (pour la construction  $\mathcal{C}_1$ ) et  $Y_{p,k}, C_{p,k}$  (pour la construction  $\mathcal{C}_2$ ) avec les cardinalités les plus petites possibles.

Les constructions précédentes ont été généralisée en changeant la dimension (de  $n$  à  $n + p$ ) et le rayon (de  $r_1$  vers  $r_1 + r_2$ ). Du même, il faut choisir les ensembles  $Y_{p,r_2}, Y_{p,r_2,k}$  et  $C_{p,k}$  avec les cardinalités les plus petites possibles :

**Théorème 3.3.2** (Charon et al. [14]). *Soient  $r_1 \geq p \geq r_2 \geq 0$ , et  $k \in \{0, 1, \dots, p-1\}$ ; soit  $C$  un  $(r_1, n)$  code et*

$$X_{p,r_2} = \{x \in \mathbb{F}^n : \forall c \in C, d(x, c) \leq r_1 - p + r_2 \text{ ou } d(x, c) > r_1 + r_2\}$$

*Construction  $\mathcal{C}_1$  : Soit  $Y_{p,r_2} \subseteq \mathbb{F}^n$  un ensemble tel que pour tout  $x \in X_{p,r_2}$ , il existe  $y \in Y_{p,r_2}$  avec  $r_1 - p + r_2 + 1 \leq d(x, y) \leq r_1 + r_2$ . Alors*

$$C' = (C \oplus \mathbb{F}^p) \cup (Y_{p,r_2} \oplus (\mathbb{F}^p \setminus \{0^p\}))$$

*est un  $(r_1 + r_2, n + p)$ .*

*Construction  $\mathcal{C}_2$  : Soit  $Y_{p,r_2,k} \subseteq \mathbb{F}^n$  un ensemble tel que pour tout  $x \in X_{p,r_2}$ , il existe  $y \in Y_{p,r_2,k}$  avec  $d(x, y) = r_1 + r_2 - k$ , et soit  $C_{p,k}$  un code  $k$ -séparateur de  $\mathbb{F}^p$ . Alors :*

$$C' = (C \oplus \mathbb{F}^p) \cup (Y_{p,r_2,k} \oplus C_{p,k})$$

*est un  $(r_1 + r_2, n + p)$ .*

### 3.3.2 Heuristiques : la méthode de bruitage et l'algorithme glouton

Les heuristiques utilisées sont la méthode de bruitage et l'algorithme glouton, afin de construire de bons codes identifiants dans l'espace binaire de hamming .

La méthode de bruitage (Noising method) est une généralisation de recuit simulé dont la recherche locale est appliquée avec une fonction d'évaluation  $f$  bruitée ou sur les données bruitées du problème. // Une fois  $r, n$  et  $M$  sont fixés, on prend un code  $C \subseteq F^n$  de cardinalité  $M$  , et on définit  $NC(C)$  le nombre des vecteurs qui ne sont pas  $r$ -couverts par  $C$ ,  $NS(C)$  le nombre des paires des vecteurs qui ne sont pas  $r$ -séparés par  $C$  et la fonction d'évaluation

$$f(C) = NC(C) + NS(C)$$

A chaque itération, on modifie le code courant  $C$  en utilisant une transformation élémentaire qui consiste à remplacer un mot de code par un autre qui n'est pas de code tout en gardant  $|C| = M$ .

Dans l'algorithme glouton, on départ d'un code vide  $C$ , et à chaque étape on ajoute un mot de code  $m$  qui maximise  $f(C) - f(C \cup \{m\})$ .

### 3.3.3 Application

Par l'utilisation des heuristique et des constructions théoriques, I.Charon et al. ont amélioré les bornes supérieures sur  $M_r(n), n \leq 21$ . Premièrement, ils ont utilisé les heuristiques pour générer des codes  $r$ -identifiants dans  $\mathbb{F}^n$ ; ensuite ils ont employé certains de ces codes pour construire d'autre codes en appliquant le théorème 3.3.2.

## 3.4 Générer des codes $r$ -identifiants dans $Q_n$ par une Recherche Tabou

Dans cette section on applique la méthode de recherche tabou, une métaheuristique parmi les plus étudiées dans la littérature, pour construire de bons codes  $r$ -identifiant dans l'espace binaire de Hamming.

Nous avons choisi la recherche tabou car c'est une méthode basée sur des idées simples, mais elle est néanmoins très efficace. Elle a été appliquée avec succès pour résoudre de nombreux problèmes difficiles d'optimisation combinatoire : problèmes de routage de véhicule, problèmes d'affectation quadratique, problèmes d'ordonnancement, problèmes de coloration de graphes, etc.

### 3.4.1 Principe de la méthode

On fixe d'abord les valeurs de rayon  $r$ , la dimension de l'hypercube  $n$  et le nombre des mots de code  $M$ . Ensuite, on génère un code initial  $C \in \mathbb{F}^n$  avec  $M$  mots de code, et on calcule  $NC(C)$  le nombre de sommets non  $r$ -couverts par  $C$ ,  $NS(C)$  le nombre de sommets non  $r$ -séparés par  $C$  et la fonction d'évaluation

$$f(C) = NC(C) + NS(C)$$

qu'on essaye de rendre nulle.

A chaque itération, on modifie le code courant  $C$  par une *transformation élémentaire* qui consiste à remplacer un mot de code par un sommet hors du code tout en gardant  $|C| = M$ . Pour se faire, on passe cycliquement par

tous les mots de code, c'est à dire après la visite du dernier mot de code, on recommence avec le premier mot de code.

Supposons qu'on visite un mot de code  $m$ , pour tout vecteur  $s$  de  $\mathbb{F}^n \setminus C$ , si  $s$  n'est pas dans la liste tabou  $T$  on pose

$$C_{m,s} = C \setminus \{m\} \cup \{s\}$$

On garde le sommet  $s^*$  qui minimise la fonction  $f(C_{m,s})$ ,  $s \in \mathbb{F}^n \setminus C$ ,  $s \notin T$ , on effectue une transformation élémentaire tel que :

$$C := C_{m,s} = C \setminus \{m\} \cup \{s\}$$

Après chaque transformation élémentaire, on vérifie la fonction d'évaluation pour le code courant :

- Si  $f(C) \neq 0$ , on ajoute  $s^*$  à la fin de la liste tabou, on enlève le sommet en tête de la liste tabou et on passe au mot de code suivant.
- Si  $f(C) = 0$ , alors  $C$  est un code  $r$ -identifiant de dimension  $M$ , on initialise le processus en enlevant de  $C$  le mot de code  $m$  qui minimise  $f(C \setminus \{m\})$ , et on applique la procédure encore une fois sur un code de dimension  $M - 1$ . Les étapes de la méthode sont résumés par l'algorithme suivant :



**Algorithm 5** Recherche Tabou pour trouver un code  $r$ -identifiant dans un espace binaire de Hamming de dimension  $n$ .

---

**Require:**  $n \geq 3$  et  $1 \leq r \leq n - 1$

Fixer la dimension de l'espace de Hamming  $n$ , et le rayon  $r$

Fixer le nombre de mots de code  $M$ , et générer un code initial  $C \subseteq \mathbb{F}^n$  de taille  $M$

Calculer  $NC(C)$  le nombre de vecteurs de  $\mathbb{F}^n$  non  $r$ -couverts par le code  $C$

Calculer  $NS(C)$  le nombre de vecteurs de  $\mathbb{F}^n$  non  $r$ -séparés par le code  $C$

Calculer la fonction d'évaluation  $f(C) = NC(C) + NS(C)$

Initialiser  $MaxIter$  le nombre d'itérations maximums

$T := \emptyset$  {Initialiser la liste tabou}

$NIter := 0$

$j := 1$  {sélectionner le premier mot de code  $C$ }

$arret := false$

**while** ( $j \leq M$ ) **and** ( $arret = false$ ) **do**

**for all** vecteur  $s \in \mathbb{F}^n \setminus C$  **do**

**if**  $s \notin T$  **then**

$C_{j,s} := C \setminus \{j\} \cup \{s\}$

**end if**

**end for**

  Soit  $s^*$  tel que  $C_{j,s^*} = \min_{s \notin T, s \in \mathbb{F}^n \setminus C} C_{j,s}$ ,

$C := C_{j,s^*}$  {Mise à jour de code}

**if**  $f(C) = 0$  **then**

$arret := true$

**else**

**if** la liste tabou n'est pas pleine **then**

      Ajouter  $s^*$  à la fin de la liste

**else**

      Enlever un vecteur en tête de  $T$

      Ajouter le vecteur  $s^*$  en fin de  $T$

**end if**

$NIter := NIter + 1$

**if**  $NIter > MaxIter$  **then**

$arret := true$

**end if**

**end if**

**if**  $j = M$  **then**

$j := 1$

**else**

$j := j + 1$

**end if**

**end while**

---

### 3.4.2 Expérimentation et Résultat

La méthode a été implémenté à l'aide d'un langage de programmation orienté objet Delphi 7.0. Cette application nous a permis de générer des codes  $r$ -identifiant dans l'espace binaire de hamming pour  $2 \leq n \leq 21$  et  $r \in \{1, \dots, 5\}$ , mais on a pas pu améliorer les bornes déjà existants dans la littérature :

$n$	$r=1$		$r=2$		$r=3$		$r=4$		$r=5$	
	Bor Inf	Bor Sup	Bor Inf	Bor Sup	Bor Inf	Bor Sup	Bor Inf	Bor Sup	Bor Inf	Bor Sup
2	a 3	B 3	-	-	-	-	-	-	-	-
3	b 4	A 4	f 7	B 7	-	-	-	-	-	-
4	d 7	C 7	g 6	G 6	f 15	B 15	-	-	-	-
5	b 10	A 10	a 6	G 6	l 9	H 10	f 31	B 31	-	-
6	m 19	D 19	a 8	G 8	a 7	H 7	l 18	H 18	f 63	B 63
7	e 32	E 32	h 14	F 14	a 8	H 8	l 13	H 14	l 31	H 32
8	c 56	H 61	a 17	F 21	a 10	H 13	a 9	H 13	l 19	H 21
9	c 101	H 112	a 26	H 32	a 13	H 17	a 10	H 14	l 12	H 17
10	c 183	H 208	i 41	H 57	a 18	H 25	a 12	H 16	a 11	H 16
11	c 337	H 352	i 67	H 100	a 25	H 36	a 15	H 20	a 12	H 17
12	c 623	H 684	i 112	H 177	a 39	H 67	a 19	H 33	a 14	H 22
13	c 1158	H 1280	i 190	H 318	a 61	H 109	a 27	H 47	a 17	H 26
14	c 2164	H 2550	i 326	H 566	a 95	H 180	a 38	H 76	a 21	H 43
15	c 4063	H 4787	i 567	H 1020	a 151	H 305	a 54	H 123	a 28	H 64
16	c 7654	H 9494	i 995	H 1844	a 241	H 530	a 77	H 192	a 37	H 94
17	c 14169	H 18558	i 1761	H 3476	a 383	H 901	a 121	H 305	a 53	H 136
18	c 27434	H 35604	i 3141	H 6430	a 608	H 1628	a 190	H 511	a 77	H 210
19	c 52155	H 65536	i 5638	H 12458	a 959	H 2846	a 304	H 835	a 112	H 326
20	c 99329	H 131072	i 10179	H 25401	k 1593	H 5813	a 489	H 1710	a 161	H 663
21	c 189829	H 262144	i 18471	H 50342	j 2722	H 11477	a 792	H 3358	a 229	H 1310

Borne inférieure			Borne supérieure	
a [31, Th 1(iii)]	f [5, Th 5]	j [33, Cor 5]	A [31]	F [19, Tableau 3 et 4]
b [31, Th 1]	g [5, Th 6]	k [33, Cor 7]	B [5, Th 5]	G [5, Th 6]
c [31, Th 3]	h [19, Table 4]	l [14]	C [6, Th 4]	H [14]
d [6, Th 4]	i [33, Cor 4]	m [20, Th 11]	D [6, Th 5]	I [20]
e [6, Th 11]			E [6, Th 6]	

TAB. 3.1 – Bornes inférieures et supérieures de  $M_r(n)$  pour  $n \in \{1, \dots, 21\}$  et  $r \in \{1, \dots, 5\}$

### 3.5 Conclusion

Dans [14], I. Charon et al. ont combiné deux méthodes heuristiques (la méthode de bruitage et l'algorithme glouton) et les constructions théoriques afin de générer de bons codes  $r$ -identifiant dans l'espace binaire de hamming et par la suite améliorer les bornes existants dans la littérature.

En s'inspirant de cet article, on a appliqué la méthode de recherche tabou, une métaheuristique parmi les plus étudiées dans la littérature, pour construire de bons codes  $r$ -identifiant dans l'espace binaire de Hamming pour  $n \in \{1, \dots, 21\}$  et  $r \in \{1, \dots, 5\}$ , mais sans avoir pu améliorer les bornes déjà existants. Comme perspective, on propose d'appliquer et tester d'autres heuristiques comme les algorithmes génétiques, recherche dispersée ...

# 4

## Codes identifiants dans le produit cartésien de deux cliques de dimensions différentes

### Sommaire

---

<b>4.1</b>	<b>Introduction</b> . . . . .	<b>58</b>
<b>4.2</b>	<b>Codes identifiants dans le produit cartésien de deux cliques de même dimension</b> . . . . .	<b>58</b>
<b>4.3</b>	<b>Codes identifiants dans le produit cartésien de deux cliques de dimensions différentes</b> . . . . .	<b>59</b>
4.3.1	Le cas où $b = 0$ et $a \geq 2$ . . . . .	60
4.3.2	Le cas où $b \neq 0$ et $a \geq 2$ . . . . .	62
4.3.3	Le cas où $b \neq 0$ et $a = 1$ . . . . .	66
4.3.3.1	Si $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$ . . . . .	66
4.3.3.2	Si $\lfloor \frac{n}{2} \rfloor + 1 \leq b \leq n - 1$ . . . . .	67
<b>4.4</b>	<b>Conclusion</b> . . . . .	<b>72</b>

---

## 4.1 Introduction

Etant donnés deux graphes  $G$  et  $H$ , le *produit cartésien*  $G \square H$  est le graphe ayant pour ensemble de sommets  $V(G) \times V(H)$  et dont deux sommets  $(x_1, y_1)$  et  $(x_2, y_2)$  sont reliés par une arête si et seulement si, soit  $x_1 x_2 \in E(G)$  et  $y_1 = y_2$ , soit  $y_1 y_2 \in E(H)$  et  $x_1 = x_2$ .

Le produit cartésien de deux cliques  $K_n \square K_m$ , est une matrice de  $n * m$  sommets tel que chaque ligne  $R_x, x \in \{1, \dots, n\}$  (resp colonne  $C_y, y \in \{1, \dots, m\}$ ) est une clique  $K_m$  (resp  $K_n$ ).

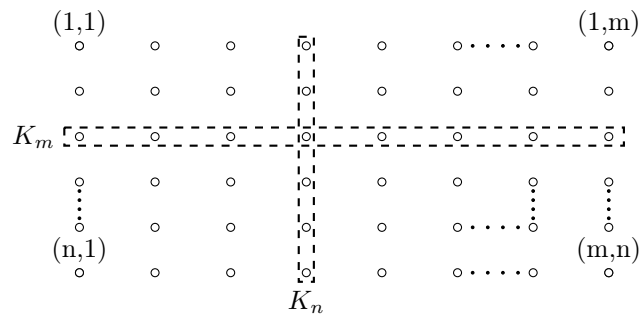


FIG. 4.1 – Produit cartésien de deux cliques  $K_n \square K_m$

Dans [24], il a été démontré que la cardinalité minimum d'un code identifiant de produit cartésien  $K_n \square K_n$  est égale à  $\lfloor \frac{3n}{2} \rfloor$ . Dans ce chapitre, on s'intéresse au cardinalité minimum des codes identifiants dans le produit cartésien de deux cliques de tailles différentes  $K_n \square K_m$ .

## 4.2 Codes identifiants dans le produit cartésien de deux cliques de même dimension

S. Gravier, J. Moncel et A. Semri ont donnée la cardinalité minimum d'un code 1-identifiant dans le produit cartésien de deux cliques de même dimension par le théorème suivant :

**Théorème 4.2.1** (Gravier et al.[24]). *Si  $C$  un code identifiant de cardinalité*

minimum de  $K_n \square K_n$ , alors

$$|C| = \left\lfloor \frac{3n}{2} \right\rfloor$$

De plus, si  $n \geq 5$  est impair alors il existe un unique (à une permutation des lignes et des colonnes) code identifiant de cardinalité  $\left\lfloor \frac{3n}{2} \right\rfloor$

La preuve de ce théorème est basée sur la détermination d'un code identifiant de cardinalité  $\left\lfloor \frac{3n}{2} \right\rfloor$ . Soit  $\{1, \dots, n\}$  l'ensemble des sommets de graphe complet  $K_n$  et soient

$$D = \{(x, x) | x = 1, \dots, n\} \text{ et } A = \begin{cases} \{(n-x+1, x) | x = 1, \dots, \frac{n-1}{2}\}, & \text{si } n \text{ est impair} \\ \{(n-x+1, x) | x = 1, \dots, \frac{n}{2}\}, & \text{si } n \text{ est pair} \end{cases}$$

L'ensemble des sommets  $D \cup A$  est un code identifiant de  $K_n \square K_n$ . C'est un code couvrant car chaque ligne contient un mot de code, et un code séparateur car chaque paire de sommets  $(x, y), (a, b) \in V(K_n \square K_n)$  est séparée par au moins un sommet de  $D \cup A$  (voir [24] pour la preuve complète).

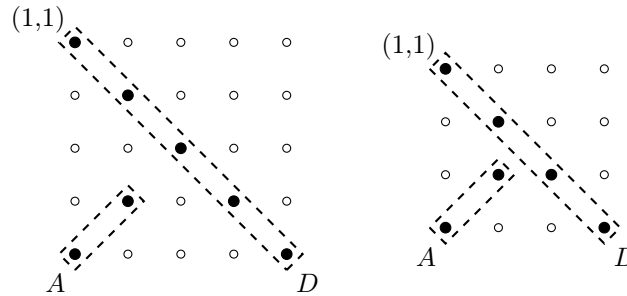


FIG. 4.2 – Le code  $D \cup A$  suivant la parité de  $n$

### 4.3 Codes identifiants dans le produit cartésien de deux cliques de dimensions différentes

Soient deux cliques  $K_n$  et  $K_m$  tels que  $m > n \geq 2$ , et soient  $a$  et  $b$  le quotient et le reste de la division euclidienne de  $m$  sur  $n$  respectivement.

Dans cette section, on propose des bornes supérieures sur la cardinalité minimum des codes identifiant dans le produit cartésien de deux cliques de tailles différentes  $K_n \square K_{an+b}$  suivant les valeurs de  $a$  et  $b$ .

### 4.3.1 Le cas où $b = 0$ et $a \geq 2$

**Proposition 4.3.1.** *Si  $C$  est un code identifiant de cardinalité minimum de  $K_n \square K_{an}$  avec  $n, a \geq 2$ , alors :*

$$|C| \leq n(2a - 1)$$

*Preuve.* Le code  $C = A \cup B \cup C$  est un code identifiant de  $K_n \square K_{an}$ ,  $n, a \geq 2$  de cardinalité  $n(2a - 1)$  tel que :

$$A = \{(i, i + kn) / i \in \{1, \dots, n\} \text{ et } k \in \{0, \dots, a - 1\}\}$$

$$B = \{(i, kn - i + 1) / i \in \{1, \dots, n\} \text{ et } k \in \{1, \dots, a - 1\}\}$$

$$C = \begin{cases} \{(n, \lceil \frac{n}{2} \rceil + jn) / j \in \{0, \dots, a - 2\}\}, & \text{si } n \text{ est impair} \\ \emptyset & \text{si } n \text{ est pair} \end{cases}$$

$C$  est un code couvrant car chaque ligne de  $K_n \square K_{an}$  contient au moins un sommet de  $C$ , donc il reste à prouver que  $C$  est un code séparateur. Soient deux sommets distincts  $(x, y), (x', y') \in K_n \square K_{an}$ ,  $a \geq 2$ , avec  $y \leq y'$ . On distingue trois cas :

1. Les sommets  $(x, y)$  et  $(x', y')$  ne sont ni dans la même ligne ni dans la même colonne :
  - (a) Si  $y \leq (a - 1)n$  alors  $(x', (a - 1)n + x') \in N[(x', y')] \setminus N[(x, y)]$
  - (b) Si  $y \geq (a - 1)n + 1$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$
2. Les sommets  $(x, y)$  et  $(x', y')$  sont dans la même colonne :
  - (a) Si  $y \leq (a - 1)n$  alors  $(x, (a - 1)n + x) \in N[(x, y)] \setminus N[(x', y)]$ .
  - (b) Si  $y \geq (a - 1)n + 1$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y)]$ .
3. Les sommets  $(x, y)$  et  $(x', y')$  sont dans la même ligne
  - (a) Si  $y \leq (a - 1)n$ , soient  $j \in \{1, \dots, n - 1\}$  et  $k \in \{0, \dots, a - 1\}$  tel que  $y = kn + j$  :
    - i. Si  $x \leq \lceil \frac{n}{2} \rceil$  alors
      - A. Si  $j = 0$  alors  $(n, y) \in N[(x, y)] \setminus N[(x', y')]$

- B. Si  $n$  est impair et  $j = \lceil \frac{n}{2} \rceil$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$
- C. Si  $j \leq \lfloor \frac{n}{2} \rfloor$  alors  $(n - j + 1, y) \in N[(x, y)] \setminus N[(x, y')]$
- D. Si  $j \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$
- ii. Si  $x \geq \lceil \frac{n}{2} \rceil + 1$  alors
  - A. Si  $j = 0$  alors  $(1, y) \in N[(x, y)] \setminus N[(x, y')]$
  - B. Si  $n$  est impair et  $j = \lceil \frac{n}{2} \rceil$  alors  $(\lceil \frac{n}{2} \rceil, y) \in N[(x, y)] \setminus N[(x, y')]$
  - C. Si  $j \leq \lfloor \frac{n}{2} \rfloor$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$
  - D. Si  $j \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(n - j + 1, y) \in N[(x, y)] \setminus N[(x, y')]$
- (b) Si  $y \geq (a-1)n + 1$ , dans ce cas  $y = (a-1)n + j$  et  $y' = (a-1)n + j'$   
 où  $1 \leq j < j' \leq n$  :
  - i. Si  $j' \leq x$  alors  $(j, (a-1)n + j) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. Si  $j' > x$  alors  $(j', (a-1)n + j') \in N[(x, y)] \setminus N[(x, y')]$

□

**Exemples suivant la parité de  $n$**

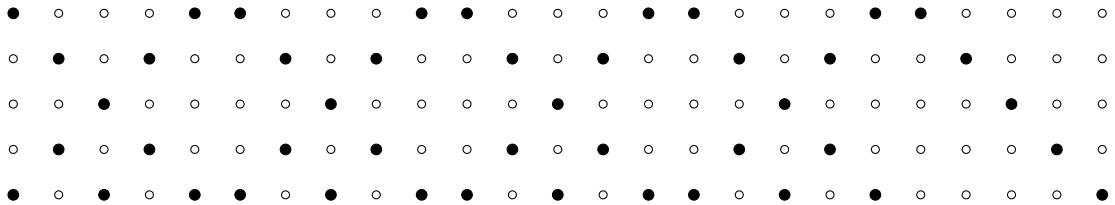


FIG. 4.3 – Code identifiant de  $K_5 \square K_{25}$

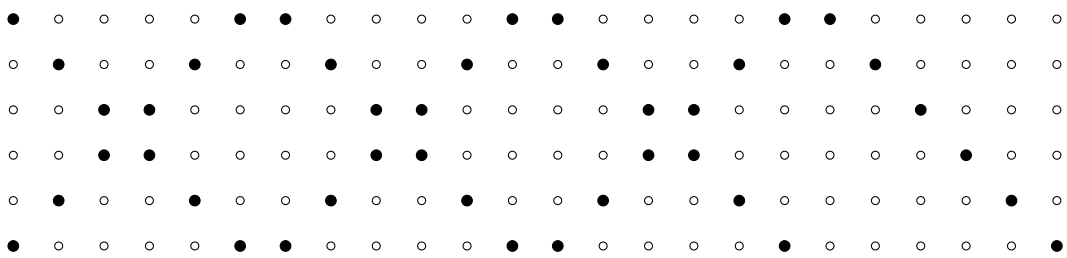


FIG. 4.4 – Code identifiant de  $K_6 \square K_{24}$



### 4.3.2 Le cas où $b \neq 0$ et $a \geq 2$

**Proposition 4.3.2.** *Si  $C$  est un code identifiant de cardinalité minimum de  $K_n \square K_{an+b}$  avec  $n, a \geq 2$  et  $b \neq 0$ , alors :*

$$|C| \leq n(2a - 1) + 2b$$

*Preuve.* Le code  $C = A \cup B \cup C$  est un code identifiant de  $K_n \square K_{an+b}$ ,  $n, a \geq 2$  et  $b \neq 0$  de cardinalité  $n(2a - 1) + 2b$  tel que :

$$A = \{(i, i + kn) / i \in \{1, \dots, n\}, k \in \{0, \dots, a - 1\}\} \cup \{(i, i + an) / i \in \{1, \dots, b\}\}$$

$$B = \{(i, kn - i + 1) / i \in \{1, \dots, n\}, k \in \{1, \dots, a - 1\}\} \cup \{(n - k + 1, (a - 1)n + k) / k \in \{1, \dots, b\}\}$$

$$C = \begin{cases} \{(n, \lfloor \frac{n}{2} \rfloor + jn) / j \in \{0, \dots, a - 2\}\} & \text{si } 1 \leq b \leq \lfloor \frac{n}{2} \rfloor \text{ et } n \text{ est impair} \\ \{(n, \lfloor \frac{n}{2} \rfloor + jn) / j \in \{0, \dots, a - 1\}\} & \text{si } \lfloor \frac{n}{2} \rfloor \leq b \leq n - 1 \text{ et } n \text{ est impair} \\ \emptyset & \text{si } n \text{ est pair} \end{cases}$$

$C$  est un code couvrant car chaque ligne de  $K_n \square K_{an+b}$  contient au moins un sommet de  $C$ , donc il reste à prouver que  $C$  est un code séparateur. Soient deux sommets distincts  $(x, y), (x', y') \in K_n \square K_{an+b}$  tel que  $b \neq 0$  et  $a \geq 2$ . Sans perte de généralités, on suppose que  $y \leq y'$  :

1. Si  $(x, y)$  et  $(x', y')$  ne sont ni dans la même colonne ni dans la même ligne :
  - (a) Si  $y \leq n$  alors
    - i. si  $y' \leq n$  alors  $(x, (a - 1)n + x) \in N[(x, y)] \setminus N[(x', y')]$
    - ii. si  $y' \geq n + 1$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$
  - (b) Si  $y \geq n + 1$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$
2. Si  $(x, y)$  et  $(x', y')$  sont dans la même colonne :
  - (a) si  $y \leq (a - 1)n$  alors  $(x, (a - 1)n + x) \in N[(x, y)] \setminus N[(x', y)]$
  - (b) si  $y \geq (a - 1)n + 1$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y)]$
3. Si  $(x, y)$  et  $(x', y')$  sont dans la même ligne. Soient  $k, k' \in \{0, \dots, a\}$  et  $j, j' \in \{0, \dots, n - 1\}$  tel que  $y = kn + j$  et  $y' = k'n + j'$

- (a) si  $y \leq (a-1)n$
- i. si  $x \leq \lceil \frac{n}{2} \rceil$ 
    - A. si  $j = 0$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$
    - B. si  $n$  est impair et  $j = \lceil \frac{n}{2} \rceil$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$
    - C. si  $j \leq \lfloor \frac{n}{2} \rfloor$  alors  $(n-j+1, y) \in N[(x, y)] \setminus N[(x, y')]$
    - D. si  $j \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. si  $x \geq \lceil \frac{n}{2} \rceil + 1$ 
    - A. si  $j = 0$  alors  $(1, y) \in N[(x, y)] \setminus N[(x, y')]$
    - B. si  $n$  est impair et  $j = \lceil \frac{n}{2} \rceil$  alors  $(\lceil \frac{n}{2} \rceil, y) \in N[(x, y)] \setminus N[(x, y')]$
    - C. si  $j \leq \lfloor \frac{n}{2} \rfloor$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$
    - D. si  $j \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(n-j+1, y) \in N[(x, y)] \setminus N[(x, y')]$
- (b) si  $(a-1)n+1 \leq y < y' \leq an$ , soient  $j$  et  $j'$  tel que  $y = (a-1)n+j$  et  $y' = (a-1)n+j'$  :
- i. si  $j' \leq x$  alors  $(j, (a-1)n+j) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. si  $j' > x$  alors  $(j', (a-1)n+j') \in N[(x, y')] \setminus N[(x, y)]$
- (c) si  $an+1 \leq y < y' \leq an+b$ , soient  $j$  et  $j'$  tel que  $y = an+j$  et  $y' = an+j'$
- i. si  $j' \leq x$  alors  $(j, an+j) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. si  $j' > x$  alors  $(j', an+j') \in N[(x, y')] \setminus N[(x, y)]$
- (d) si  $1 \leq y \leq an$  et  $an+1 \leq y' \leq an+b$ . Soient  $k \in \{0, \dots, a\}$ ,  $j \in \{0, \dots, n-1\}$  et  $j' \in \{1, \dots, b\}$  tels que  $y = kn+j$  et  $y' = an+j'$
- i. si  $x \geq b+1$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$
  - ii. si  $x \leq b$  alors
    - A. si  $x \leq \lceil \frac{n}{2} \rceil$ 
      - si  $j = 0$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$
      - si  $n$  est impair et  $j = \lceil \frac{n}{2} \rceil$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$
      - si  $j \leq \lfloor \frac{n}{2} \rfloor$  alors  $(n-j+1, y) \in N[(x, y)] \setminus N[(x, y')]$

– si  $j \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$

B. si  $x \geq \lceil \frac{n}{2} \rceil + 1$

– si  $j = 0$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$

– si  $n$  est impair et  $j = \lceil \frac{n}{2} \rceil$  alors  $(\lceil \frac{n}{2} \rceil, y) \in N[(x, y)] \setminus N[(x, y')]$

– si  $j \leq \lfloor \frac{n}{2} \rfloor$  alors  $(j, y') \in N[(x, y')] \setminus N[(x, y)]$

– si  $j \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(n - j + 1, y) \in N[(x, y)] \setminus N[(x, y')]$

□

Exemples suivant la parité de  $n$

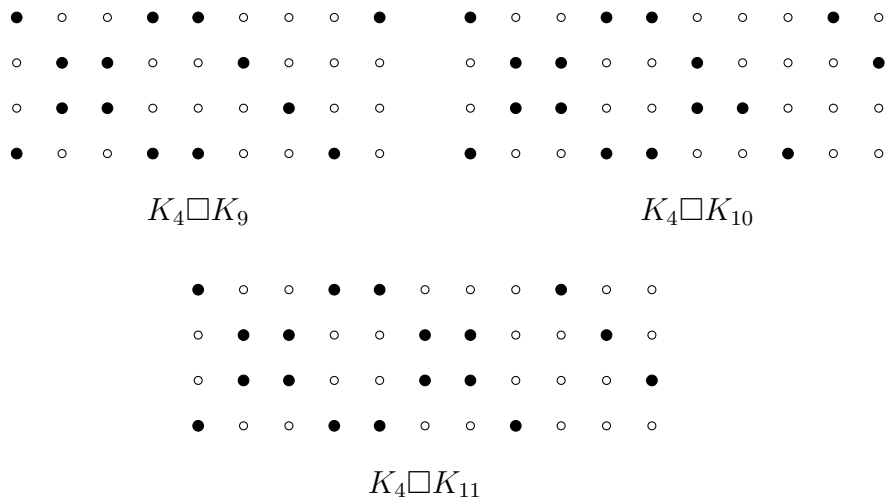


FIG. 4.5 – Codes identifiants de  $K_n \square K_{an+b}$ ,  $n = 4, a = 2$  et  $b \in \{1, 2, 3\}$

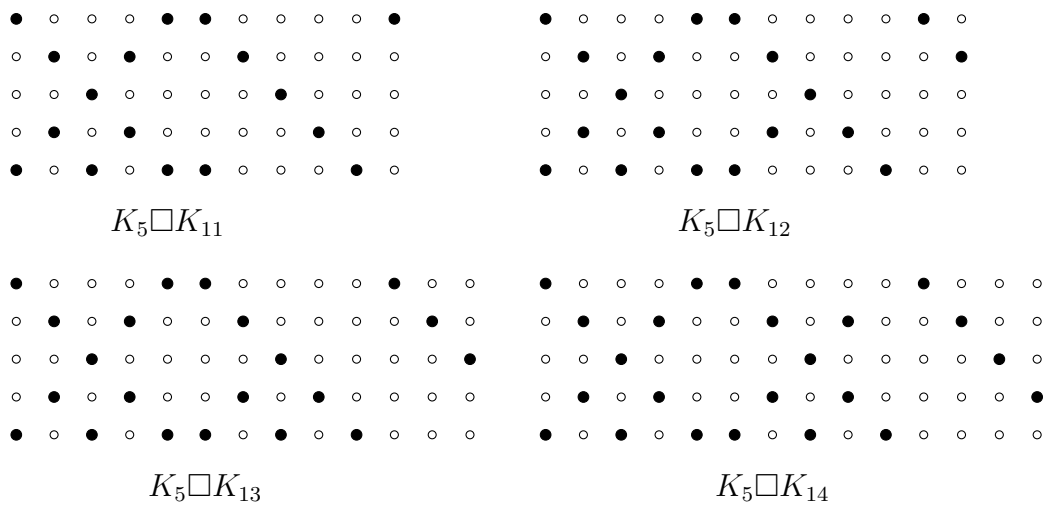


FIG. 4.6 – Codes identifiants de  $K_n \square K_{an+b}$ ,  $n = 5, a = 2$  et  $b \in \{1, 2, 3, 4\}$

### 4.3.3 Le cas où $b \neq 0$ et $a = 1$

Dans ce cas, on propose deux codes identifiants suivant la valeur de  $b$

#### 4.3.3.1 Si $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$

**Proposition 4.3.3.** *Si  $C$  est un code identifiant de cardinalité minimum de  $K_n \square K_{n+b}$  avec  $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$ , alors :*

$$|C| \leq \left\lfloor \frac{3n}{2} \right\rfloor + b$$

*Preuve.* Si  $1 \leq b \leq \lfloor \frac{n}{2} \rfloor$ , alors le code  $C = A \cup B$  est un code identifiant de  $K_n \square K_{n+b}$ ,  $n \geq 2$  de cardinalité  $\lfloor \frac{3n}{2} \rfloor + b$  tel que :

$$A = \{(i, i)/i \in \{1, \dots, n\}\} \cup \{(i, n+i)/i \in \{1, \dots, b\}\}$$

$$B = \{(n-i+1, i)/i \in \{1, \dots, \lfloor \frac{n}{2} \rfloor\}\}$$

$C$  est couvrant on montre qu'il est séparable. Soient deux sommets distincts  $(x, y), (x', y') \in K_n \square K_{an}$ ,  $a \geq 2$ , avec  $y \leq y'$ . On distingue trois cas :

1. Les sommets  $(x, y)$  et  $(x', y')$  ne sont ni dans la même ligne ni dans la même colonne, On distingue 3 positions de  $(x, y)$  et  $(x', y')$  par rapport à la première diagonal :
  - (a) Si  $x \geq y$  et  $x' \geq y'$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$
  - (b) Si  $x \geq y$  et  $x' < y'$  :
    - i. Si  $x \leq \lfloor \frac{n}{2} \rfloor$  alors  $(n-y+1, y) \in N[(x, y)] \setminus N[(x', y')]$
    - ii. Si  $x \geq \lfloor \frac{n}{2} \rfloor + 1$  alors  $(x, n-x+1) \in N[(x, y)] \setminus N[(x', y')]$
  - (c) Si  $x < y$  et  $x' < y'$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$
2. Les sommets  $(x, y)$  et  $(x', y')$  sont dans la même colonne : sans perte de généralités, on suppose que  $x < x'$ 
  - (a) Si  $x \geq y$  alors  $(x', x') \in N[(x', y)] \setminus N[(x, y)]$
  - (b) Si  $x < y$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y)]$
3. Les sommets  $(x, y)$  et  $(x', y')$  sont dans la même ligne :

- (a) Si  $x \leq \lfloor \frac{n}{2} \rfloor$
- i. Si  $y \leq \lfloor \frac{n}{2} \rfloor$  alors  $(n - y + 1) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. Si  $y \geq \lfloor \frac{n}{2} \rfloor + 1$  et  $y' \leq n$  alors  $(y', y') \in N[(x, y')] \setminus N[(x, y)]$
  - iii. Si  $\lfloor \frac{n}{2} \rfloor + 1 \leq y \leq n$  et  $y' \geq n + 1$ 
    - A. Si  $x \leq b$  alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$
    - B. Si  $x \geq b + 1$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$  où  $j' = y' - n$
  - iv. Si  $n + 1 \leq y < y' \leq n + b$ , soient  $j = y - n$  et  $j' = y' - n$ 
    - A. Si  $x \geq j'$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$
    - B. Si  $x < j'$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$
- (b) si  $x \geq \lfloor \frac{n}{2} \rfloor + 1$
- i. Si  $y \leq \lfloor \frac{n}{2} \rfloor$  alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. Si  $y \geq \lfloor \frac{n}{2} \rfloor + 1$  et  $y' \leq n$  alors
    - A. Si  $x \geq y'$  alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$
    - B. Si  $x < y'$  alors  $(y', y') \in N[(x, y')] \setminus N[(x, y)]$
  - iii. Si  $y \geq \lfloor \frac{n}{2} \rfloor + 1$  et  $y' \geq n + 1$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$   
où  $j' = y' - n$

□

#### 4.3.3.2 Si $\lfloor \frac{n}{2} \rfloor + 1 \leq b \leq n - 1$

**Proposition 4.3.4.** *Si  $C$  est un code identifiant de cardinalité minimum de  $K_n \square K_{n+b}$  avec  $\lfloor \frac{n}{2} \rfloor + 1 \leq b \leq n - 1$ , alors :*

$$|C| \leq n + 2b$$

*Preuve.* Si  $\lfloor \frac{n}{2} \rfloor + 1 \leq b \leq n - 1$ , alors le code  $C = A \cup B \cup C$  est un code identifiant de  $K_n \square K_{n+b}$ ,  $n \geq 2$  de cardinalité  $n + 2b$  tel que :

$$A = \{(i, i)/i \in \{1, \dots, n\}\} \cup \{(i, n+i)/i \in \{1, \dots, b\}\}$$

$$B = \{(n-i+1, i)/i \in \{1, \dots, b\}\}$$

$$C = \begin{cases} \{(n, \lceil \frac{n}{2} \rceil)\} & \text{si } n \text{ est impair} \\ \emptyset & \text{si } n \text{ est pair} \end{cases}$$

$C$  est couvrant on va montrer qu'il est séparateur. Soient deux sommets distincts  $(x, y), (x', y') \in K_n \square K_{an}, a \geq 2$ , avec  $y \leq y'$ . On distingue trois cas :

1. Les sommets  $(x, y)$  et  $(x', y')$  ne sont ni dans la même ligne ni dans la même colonne :

(a) Si  $x \geq y$  et  $x' \geq y'$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$

(b) Si  $x \geq y$  et  $x' < y'$  :

i. Si  $x \leq \lceil \frac{n}{2} \rceil$  alors  $(n-y+1, y) \in N[(x, y)] \setminus N[(x', y')]$

ii. Si  $x \geq \lceil \frac{n}{2} \rceil + 1$  alors  $(x, n-x+1) \in N[(x, y)] \setminus N[(x', y')]$

(c) Si  $x < y$  et  $x' < y'$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y')]$

2. Les sommets  $(x, y)$  et  $(x', y')$  sont dans la même colonne : sans perte de généralités, on suppose que  $x < x'$

(a) Si  $x \geq y$  alors  $(x', x') \in N[(x', y)] \setminus N[(x, y)]$

(b) Si  $x < y$  alors  $(x, x) \in N[(x, y)] \setminus N[(x', y)]$

3. Les sommets  $(x, y)$  et  $(x', y')$  sont dans la même ligne :

(a) Si  $x \leq \lceil \frac{n}{2} \rceil$

i. Si  $y \leq \lfloor \frac{n}{2} \rfloor$  alors  $(n-y+1) \in N[(x, y)] \setminus N[(x, y')]$

ii. Si  $n$  est impair et  $y = \lceil \frac{n}{2} \rceil$  alors  $(n, y) \in N[(x, y)] \setminus N[(x, y')]$

iii. Si  $y \geq \lfloor \frac{n}{2} \rfloor + 1$  et  $y' \leq n$  alors  $(y', y') \in N[(x, y')] \setminus N[(x, y)]$

iv. Si  $\lfloor \frac{n}{2} \rfloor + 1 \leq y \leq n$  et  $y' \geq n+1$  alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$

v. Si  $n+1 \leq y < y' \leq n+b$ , soient  $j = y - n$  et  $j' = y' - n$

A. Si  $x \geq j'$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$

B. Si  $x < j'$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$

- (b) si  $x \geq \lceil \frac{n}{2} \rceil + 1$
- i. Si  $y \leq \lfloor \frac{n}{2} \rfloor$  alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$
  - ii. Si  $y = \lfloor \frac{n}{2} \rfloor$  et  $n$  est impair alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$
  - iii. Si  $y \geq \lceil \frac{n}{2} \rceil + 1$  et  $y' \leq n$ 
    - A. Si  $x \geq y'$  alors  $(y, y) \in N[(x, y)] \setminus N[(x, y')]$
    - B. Si  $x < y'$  alors  $(y', y') \in N[(x, y')] \setminus N[(x, y)]$
  - iv. Si  $y \geq \lceil \frac{n}{2} \rceil + 1$  et  $y' \geq n + 1$ 
    - A. Si  $x > b$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$  où  $j' = y' - n$ .
    - B. Si  $x \leq b$  et  $y \leq b$  alors  $(n - y + 1, y) \in N[(x, y)] \setminus N[(x, y')]$
    - C. Si  $x \leq b$  et  $y = n$  alors  $(n, n) \in N[(x, y)] \setminus N[(x, y')]$
    - D. Si  $x \leq b$  et  $y \geq n + 1$ ; soient  $j = y - n$  et  $j' = y' - n$ 
      - Si  $x \geq j'$  alors  $(j, y) \in N[(x, y)] \setminus N[(x, y')]$
      - Si  $x < j'$  alors  $(j', y') \in N[(x, y')] \setminus N[(x, y)]$

□



Exemples suivant la parité de  $n$

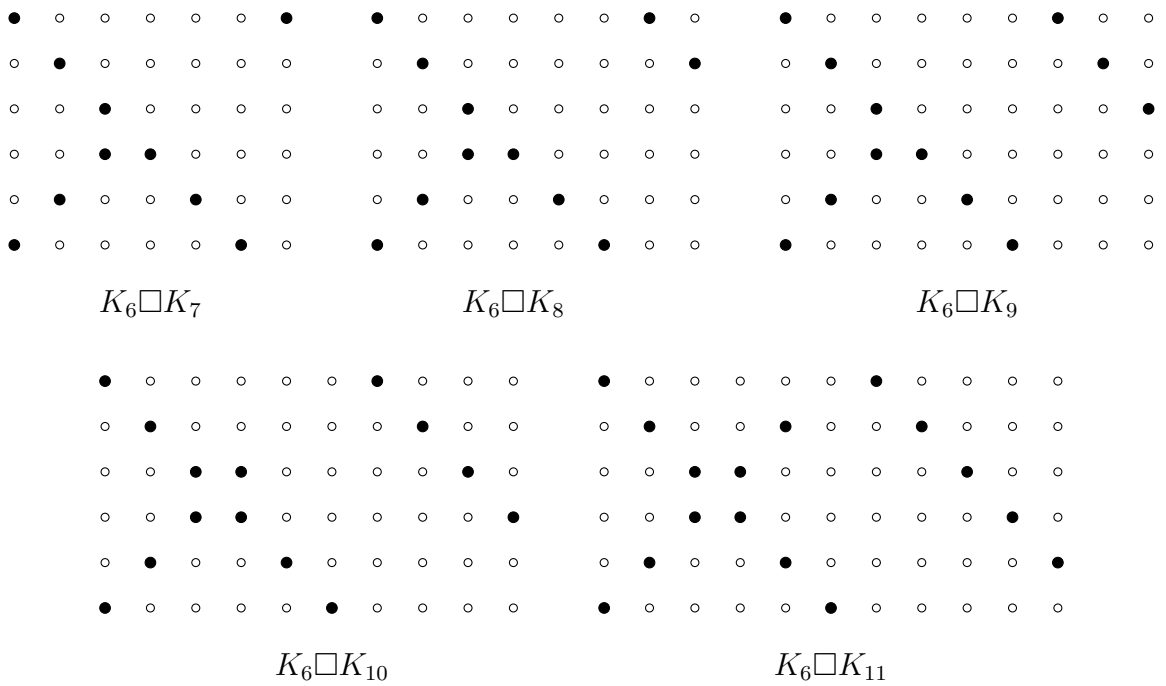
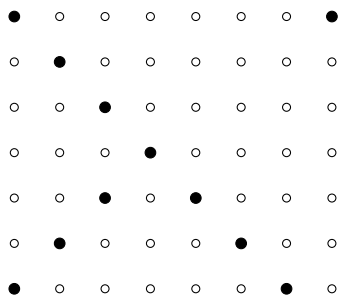
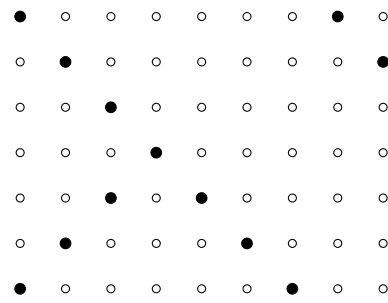


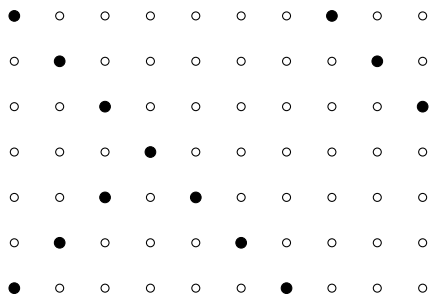
FIG. 4.7 – Codes identifiants de  $K_n \square K_{n+b}$ ,  $n = 6$  et  $b \in \{1, 2, 3, 4, 5\}$



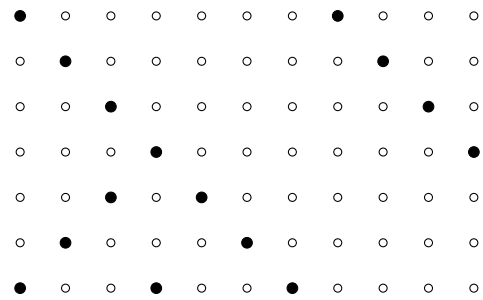
$K_7 \square K_7$



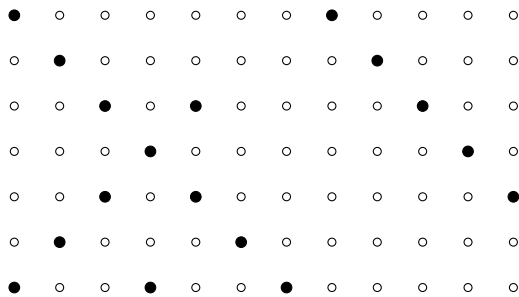
$K_7 \square K_8$



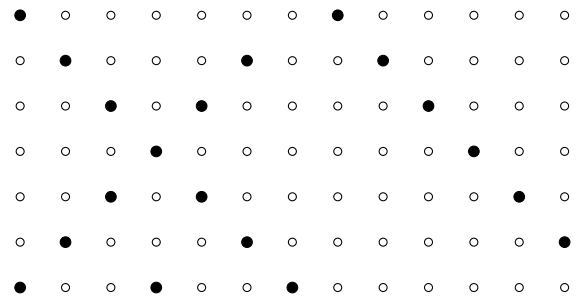
$K_7 \square K_9$



$K_7 \square K_{10}$



$K_7 \square K_{11}$



$K_7 \square K_{12}$

FIG. 4.8 – Codes identifiants de  $K_n \square K_{n+b}$ ,  $n = 7$  et  $b \in \{1, 2, 3, 4, 5, 6\}$

## 4.4 Conclusion

Dans la section précédente, on a donné des bornes supérieures sur la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques de tailles différentes  $K_n \square K_m$  suivant les valeurs de  $a$  et  $b$  où  $m = an + b$  avec  $b \leq n - 1$ . Tous les cas possibles sont résumés dans le tableau suivant.

$a$	$b$	Borne supérieure
$a \geq 2$	$0 \leq b \leq n - 1$	$n(2a - 1) + 2b$
$a = 1$	$1 \leq b \leq \lfloor \frac{n}{2} \rfloor$	$\lfloor \frac{3n}{2} \rfloor + b$
$a = 1$	$\lceil \frac{n}{2} \rceil \leq b \leq n - 1$	$n + 2b$
$a = 1$	$b = 0$	$\lfloor \frac{3n}{2} \rfloor$ (valeur exacte <sup>1</sup> )

TAB. 4.1 – Récapitulatif

---

<sup>1</sup>S. Gravier, J. Moncel, A. Semri, Identifying codes of Cartesian product of two cliques of the same size, The Electronic Journal of Combinatorics, 15 N4, 2008.

# Conclusion

Les résultats de NP-complétude du problème de minimisation de cardinalité des codes identifiant dans les graphes présentés par Honkala et Lobstein [28] et Charon et al [11] nous ont amené à nous intéresser au problème de codes identifiants de certaines classes de graphes notamment l'hypercube et le produit cartésien de deux cliques de dimensions différentes.

Dans la première partie de notre travail, nous avons utilisé une métaheuristique recherche tabou, assez appliqué dans la littérature, pour construire de bons codes identifiant à distance  $r \geq$  dans l'espace binaire de Hamming de dimension  $n \geq 3$ .

La deuxième partie de notre travail a traité la détermination des bornes supérieures de la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques  $K_n \square K_m$  avec  $n \neq m$ .

Ce travail demande à être poursuivi et complété. Plus précisément, on propose comme perspective :

- Appliquer d'autres métaheuristiques pour construire d'autres codes identifiant dans l'hypercube afin d'améliorer les bornes existantes ou d'appliquer des méthodes exactes pour trouver les valeurs exactes, plus particulièrement pour  $n \geq 20$ .
- Rechercher la valeur exacte de la cardinalité minimum d'un code identifiant dans le produit cartésien de deux cliques  $K_n \square K_m$  pour  $n \neq m$ .

# Bibliographie

- [1] Y. Ben-Haim, S. Litsyn. *Exact minimum density of codes identifying vertices in the square grid*, SIAM J. Discrete Math, Vol 19 N°1, pp. 69-82, 2005.
- [2] C. Berge. *Graphes*, Gauthier-Villars, 1985.
- [3] N. Bertrand, I. Charon, O. Hudry, A. Lobstein. *Identifying and locating-dominating codes on chains and cycles*, European J. Combin., 25(7) :pp. 969-987, 2004.
- [4] N. Bertrand, I. Charon, O. Hudry, A. Lobstein. *1-identifying codes on trees*, Australas. J. Combin., 31 : 21-35, 2005.
- [5] U. Blass, I. Honkala, S. Litsyn. *On binary Codes for Identification*, Journal of Combinatorial Designs, Vol. 8, pp. 151-156, 2000.
- [6] U. Blass, I. Honkala, S. Litsyn. *Bounds on identifying codes*. Discrete Math., 241(1-3) :119.128, 2001.
- [7] M. Blidia, M. Chellali, F. Maffray, J. Moncel and A. Semri. *Locating-dominating and identifying codes in trees*. Australasian Journal of Combinatorics, vol. 39, pp.219.232, 2007.
- [8] I. Charon, I. Honkala, O. Hudry, A. Lobstein. *General bounds for identifying codes in some infinite regular graphs*, Electron. J. Combin., 8(1) : Research Paper 39,21 pp., 2001.
- [9] I. Charon, I. Honkala, O. Hudry, A. Lobstein. *Identifying codes with small radius in some infinite regular graphs*, Electron. J. Combin., 9(1) : Research Paper 11, 25 pp., 2002.

- [10] I. Charon, O. Hudry, A. Lobstein, *Identifying and Locating-Dominating Codes : NP-completeness Results for Directed Graphs*, IEEE Transactions on Information Theory 48(8), 2192–2200, 2002.
- [11] I. Charon, O. Hudry, A. Lobstein. *Minimizing the Size of an Identifying or Locating-Dominating Code in a Graph is NP-Hard*, Theoretical Computer Science, vol. 290. Issue 3, pp. 2109-2120, 2003.
- [12] I. Charon, S. Gravier, O. Hudry, A. Lobstein, M. Mollard, J. Moncel, *A linear algorithm for minimum 1-identifying codes in oriented trees*, Discrete Appl. Math., 154 (8) :1246-1253, 2006.
- [13] I. Charon, G. Cohen, O. Hudry, A. Lobstein. *Links Between Discriminating and Identifying Codes in the Binary Hamming Space*, AAECC 2007, LNCS 4851, pp. 267-270, 2007.
- [14] I. Charon, G. Cohen, O. Hudry, A. Lobstein. *New identifying codes in the binary Hamming space*, European Journal of Combinatorics, vol 31, pp 491-501, 2010.
- [15] G. Cohen, S. Gravier, I. Honkala, A. Lobstein, M. Mollard, C. Payan, G. Zémor. *Improved Identifying Codes for the Grid*, Electronic Journal of Combinatorics, comments to 6(1) R19 (1999).
- [16] G. Cohen, I. Honkala, A. Lobstein, G. Zémor. *New bounds for codes identifying vertices in graphs*, Electronic Journal of Combinatorics, 6 : Research Paper 19, 14 pp., 1999.
- [17] S. A. Cook , *The Complexity of Theorem-Proving Procedures* Proceedings of the Third Annual ACM Symposium on the Theory of Computing , 151–158, 1971.
- [18] M. Daniel, S. Gravier, J. Moncel. *Identifying Codes in some Subgraphs of the Square Lattice*, Theoretical Computer Science 319 , 411–421, 2004.
- [19] G. Exoo, T. Laihonen, S. Ranto. *Improved upper bounds on binary identifying codes*, IEEE Transactions on Information Theory, vol 53 pp. 4255-4260, 2007.
- [20] G. Exoo, T. Laihonen, S. Ranto. *New Bounds on Binary Identifying Codes*, Discrete Applied Mathematics, vol 156 pp. 2250-2263, 2008.

- [21] F. Glover. *Tabu Search*. Part I, ORSA Journal on Computing, vol 1, 190-206, 1989.
- [22] F. Glover. *Tabu Search*. Part II, ORSA Journal on Computing, vol 2, 4-32, 1990.
- [23] S. Gravier, J. Moncel, A. Semri. *Identifying codes of cycles*, European Journal of Combinatorics, 27 (5) :767-776, 2006.
- [24] S. Gravier, J. Moncel, A. Semri. *Identifying codes of Cartesian product of two cliques of the same size*, The Electronic Journal of Combinatorics, 15 N4, 2008.
- [25] J. H Holland. *Adaptation In Natural And Artificial Systems*, University of Michigan Press, 1975.
- [26] I. Honkala, M.G Karpovsky, S. Litsyn. *Cycles identifying vertices and edges in binary hypercubes and 2-dimensional tori*, Discrete Applied Mathematics Vol 129 , pp. 409-419, 2003.
- [27] I. Honkala, A. Lobstein. *On the complexity of the identification problem in Hamming spaces*, Acta Informatica, Vol 38, pp 839–845, 2002.
- [28] I. Honkala, A. Lobstein. *On identifying codes in binary Hamming spaces*, Journal of Combinatorial Theory, Ser. A, Vol. 99, pp. 232-243, 2002.
- [29] I. Honkala, T. Laihanen. *On identification in the triangular and square grids*, SIAM J. Comput., 33(2) : pp. 304-312, 2004.
- [30] I. Honkala, T. Laihanen. *On identification in the triangular grid*, J. Combin. Theory Ser. B, 91(1) : pp. 67-86, 2004.
- [31] M.G. Karpovsky, K. Chakrabarty, L.B. Levitin. *On a new class of codes for identifying vertices in graphs*, IEEE Transactions on Information Theory, Vol. 44(2), pp. 599-611, 1998.
- [32] M. Laifenfeld, A. Trachtenberg. *Disjoint Identifying Codes for Arbitrary Graphs*, Boston university, 2005.
- [33] T. Laihonon, S. Ranto. *Codes identifying sets of binary words with large radii*, in : Proc. Workshop on Coding and Cryptographie 2007, Versailles, France, pp. 215-224, 2007.

- [34] J. Moncel. *Monotonicity of the minimum cardinality of an identifying code in the hypercube*, Discrete Applied Mathematics, Vol. 154, pp. 898-899, 2006.
- [35] S. Ray, D. Starobinski, A. Trachtenberg, R. Ungrangsi. *Robust Location Detection with Sensor Networks*, IEEE Journal on Selected Areas in Communications 22(6), 1016–1025, 2004.
- [36] M Xu, K. Thulasiraman, X-D Hu. *Identifying codes of cycles with odd orders*, European Journal of Combinatorics 29,pp. 1717-1720,2008.