Mémoire d'Habilitation à Diriger des Recherches

Université Pierre et Marie Curie, Paris

Contributions Combinatoires au Codage, en Connexion avec la Complexité et la Cryptographie

Antoine Lobstein

Centre National de la Recherche Scientifique, URA 820

Ecole Nationale Supérieure des Télécommunications, Paris

Composition du jury :

Jean-Pierre Barthélemy

Pascale Charpin

Gérard Cohen

Iiro Honkala

Simon Litsyn

Michel Minoux

Patrick Solé

Soutenance le 19 février 2002

Contributions Combinatoires au Codage, en Connexion avec la Complexité et la Cryptographie

A Martine,

à ma famille.

Je me fais l'effet de n'avoir pas été autre chose qu'un garçon jouant sur le rivage, et m'amusant de temps à autre à trouver un caillou poli ou un coquillage plus joli qu'à l'ordinaire ; tandis que le grand océan de la vérité se déroulait devant moi sans que je le connusse.

Isaac Newton

La mathématique est une froide compagne. *Bertolt Brecht*

Il y a trois sortes de gens : ceux qui savent compter et ceux qui ne savent pas. *Anonyme*

Remerciements

Du fond du cœur, je voudrais remercier Pascale Charpin qui a trouvé la force et le courage de mener à bien sa tâche de rapporteur, malgré le deuil terrible qui l'a frappée.

Pascale, veuille trouver ici l'expression de ma profonde sympathie.

Merci à Iiro Honkala et Patrick Solé, qui ont accepté d'être rapporteurs.

Merci à Michel Minoux, qui m'a permis de m'inscrire à l'Université Pierre et Marie Curie, et a bien voulu faire partie du jury.

Merci à Jean-Pierre Barthélemy, Gérard Cohen et Simon Litsyn, qui ont également consenti à figurer dans mon jury.

septembre-octobre 2001

Remerciements

Je voudrais remercier ici toutes celles et tous ceux qui, de près ou de loin, m'ont aidé dans mes travaux de recherche et m'ont permis de travailler dans les meilleures conditions possibles : chercheurs, enseignants, ingénieurs, membres du personnel administratif, techniciens, thésards, ... En repensant à mes jeunes années, je voudrais notamment citer Gérard Cohen bien sûr, sans qui ... etc., et aussi Claude Gueguen, Jean-Claude Bermond, Bernadette Bouchon, Michel Deza, Philippe Godlewski, Neil Sloane, David Chaum, Jack van Lint, Marc Girault, qui ont contribué à guider mes premiers pas hésitants dans la recherche.

Parmi mes co-auteurs, je désire remercier tout particulièrement Olivier (« il m'a montré la voie ») et Irène ; sans leurs pressions gentiment insistantes au sujet de cette Habilitation, j'en serais encore à procrastiner. Pour bien d'autres raisons (chocs élastiques, pithiviers, ...), je me réjouis de leur amitié.

Merci à Gérard et Gilles, vieux complices en recherche (une photo en témoigne).

Merci à Jean-Pierre, pour son art abstrait et du contrepet.

Outre Gérard et Jean-Pierre, j'ai partagé avec Iiro et Simon la dure aventure d'écrire un livre. Je les remercie tous de leur témérité et constate avec plaisir que cela n'a pas réussi à détruire nos relations amicales.

Merci à Skip pour sa critique indulgente de "Covering Codes" dans *Mathematical Reviews*, et pour ses cassettes de jazz.

Merci à Grisha, avec qui j'ai travaillé à Moscou, à l'Institut pour les Problèmes de Transmission de l'Information.

Merci à Faina (pour qui j'ai une pensée toute particulière) et Sergei, avec qui j'ai travaillé à Akademgorodok, à l'Institut Sobolev de Mathématiques.

Merci à Victor pour son hospitalité moscovite.

Merci à Iiro pour de fructueux séjours à Turku.

Mark Karpovsky est venu à l'ENST faire un exposé sur les codes identifiants, point de départ de notre intérêt pour ce sujet. Qu'il en soit remercié ici.

Enfin, comment ne pas remercier Martine d'exister ?

janvier 2001,

octobre 2001.

Table des Matières

PRÉFACE

Ce document se veut le survol d'environ quinze années de recherche. C'est en effet en 1985 que j'ai soutenu ma thèse à l'Ecole Nationale Supérieure des Télécommunications (ENST), sous la direction de Gérard Cohen, et en 1987 (concours de 1986 « gelé » pendant un an) que je suis entré au Centre National de la Recherche Scientifique (CNRS), à l'ENST (Unité de Recherche Associée URA 820). J'y travaille au sein du Département « Informatique et Réseaux », dans l'équipe « Mathématiques de l'Informatique et des Réseaux ».

Tous mes travaux font partie des **Mathématiques Discrètes** et de la **Combinatoire** ; le thème central, le **codage**, y est abordé d'un point de vue multiple mais toujours combinatoire : ma vision des codes est celle d'objets combinatoires flottant dans différents espaces discrets, et leurs liens avec, par exemple, la théorie de la complexité et ses classes structurées de problèmes, me semblent très naturels.

J'ai choisi de faire une présentation thématique plutôt que chronologique, avec la volonté justement de montrer les liens existant entre les différents domaines couverts par mes recherches, et celle également de parfois mettre en relief un résultat me paraissant plus intéressant, plus significatif, ou plus facile à exposer, sans entrer dans de trop longs développements techniques, que d'autres.

D'un point de vue chronologique, disons simplement que mes thématiques ont progressivement évolué, au gré des rencontres, des circonstances, des fréquentes et diverses collaborations et des goûts, à partir du rayon de recouvrement et des codes arithmétiques vers les codes en blocs parfaits puis les codes identifiants, avec très souvent en toile de fond la question de la complexité de ces problèmes, les incursions actives en cryptographie étant rares. Cette trajectoire n'est certes pas rectiligne, mais elle reste cohérente et inscrite dans les grands axes de recherche de l'URA 820, le traitement et la communication de l'information. De ce dernier point témoignent mes collaborations régulières avec plusieurs de mes collègues de l'ENST.

La première section, la plus longue, est consacrée aux codes (voir Figure 1) :
– codes en blocs, avec leurs deux paramètres fondamentaux, distance minimale d et rayon de recouvrement R, qui se rejoignent dans la relation $d = 2R + 1$ pour les codes parfaits ;
– codes arithmétiques, utilisant différents types de représentation des entiers, soulevant des problèmes de métriques et pouvant encore donner naissance à des codes parfaits ;
– codes identifiants, pour la localisation de sommets dans les graphes ; y sont en particulier étudiées les grilles carrée, triangulaire, royale, et hexagonale.

Les liens entre codes et complexité, entre codes et cryptologie, sont exposés ultérieurement, après la deuxième section, consacrée à un exposé succinct de théorie de la complexité, et la troisième section, qui décrit trois cryptosystèmes à clé publique : je m'appuierai sur ces deux sections par la suite.

La quatrième section met en évidence certains liens entre codes et complexité (voir Figure 2) :
– NP- ou Π_2-complétude de problèmes concernant les codes en blocs, étude de certains problèmes où un prétraitement des entrées est possible, utilisation d'heuristiques pour la construction de « bons » codes ;
– complexité et codes arithmétiques : difficulté de calculer le poids modulaire de Clark-Liang ;
– NP-complétude du problème de l'existence de codes identifiants de taille majorée, utilisation d'heuristiques pour la construction de « petits » codes identifiants.

La cinquième section met en évidence certains liens entre codes et cryptologie (voir Figure 3), notamment les relations existant entre représentation modifiée non adjacente, poids modulaire, et exponentiation modulaire accélérée pour le cryptosystème RSA.

Une courte conclusion évoque quelques pistes possibles pour le futur.

J'ai également joint une version légèrement raccourcie, rédigée dans un pidgin assez répandu, à l'intention de certains de mes collègues étrangers.

Enfin, j'ai mis en annexes la liste complète de mes publications, ainsi que quelques articles qui me paraissent significatifs (ces derniers uniquement dans la « version papier » de ce document, disponible auprès de l'auteur).

Mes recherches se sont très souvent faites en collaboration, débouchant sur des articles ou des livres. Ces échanges d'idées constituent toujours d'enrichissantes expériences et je voudrais saisir cette occasion pour ici remercier très vivement tous mes co-auteurs (par ordre d'apparition, comme au cinéma) :

Gérard Cohen (ENST, France),
Neil Sloane (Laboratoires Bell, Etats-Unis d'Amérique),
Gerhard van Wee (Université d'Eindhoven, Pays-Bas),
Jean-Pierre Barthélemy (ENST Bretagne, France),
Patrick Solé (CNRS, France),
Vera Pless (Université d'Illinois, Etats-Unis d'Amérique),
Grigori Kabatianski (Institut pour les Problèmes de Transmission de l'Information [IPPI], Russie),
Irène Charon (ENST, France),
Olivier Hudry (ENST, France),
Simon Litsyn (Université de Tel Aviv, Israël),
Skip Mattson (Université de Syracuse, Etats-Unis d'Amérique),
Iiro Honkala (Université de Turku, Finlande),
Victor Zinoviev (IPPI, Russie),
Gilles Zémor (ENST, France),
David Naccache (Gemplus, France),
Sylvain Gravier (CNRS, France),
Michel Mollard (CNRS, France),
Charles Payan (CNRS, France),
Sergei Avgustinovich (Institut Sobolev de Mathématiques [ISM], Russie),
et Faina Solov'eva (ISM, Russie).

Le mérite de beaucoup de ce qui suit leur revient.

Et donc, le « nous » que j'utiliserai dans la suite de ce document tantôt sera un nous de modestie, tantôt désignera un ensemble d'auteurs.

COMBINATOIRE

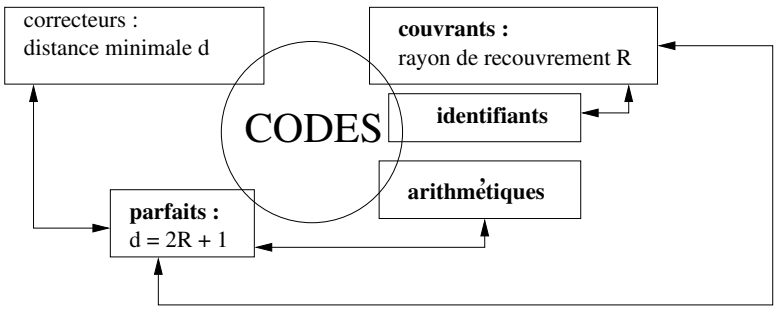

FIG. 1 – *Codage ; en gras, les thèmes que nous avons étudiés.*

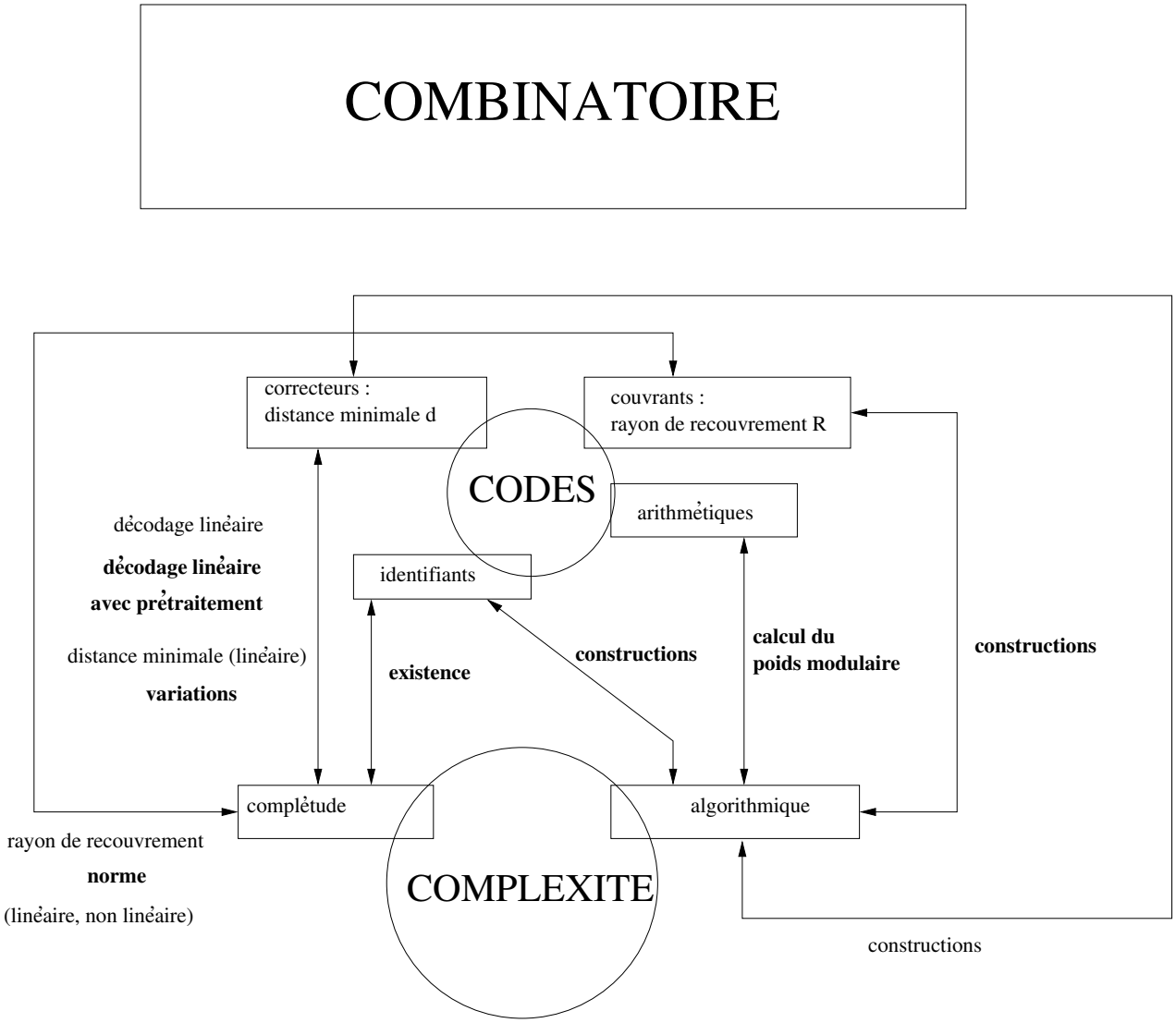

FIG. 2 – *Quelques liens entre codage et complexité ; en gras, les liens que nous avons étudiés.*

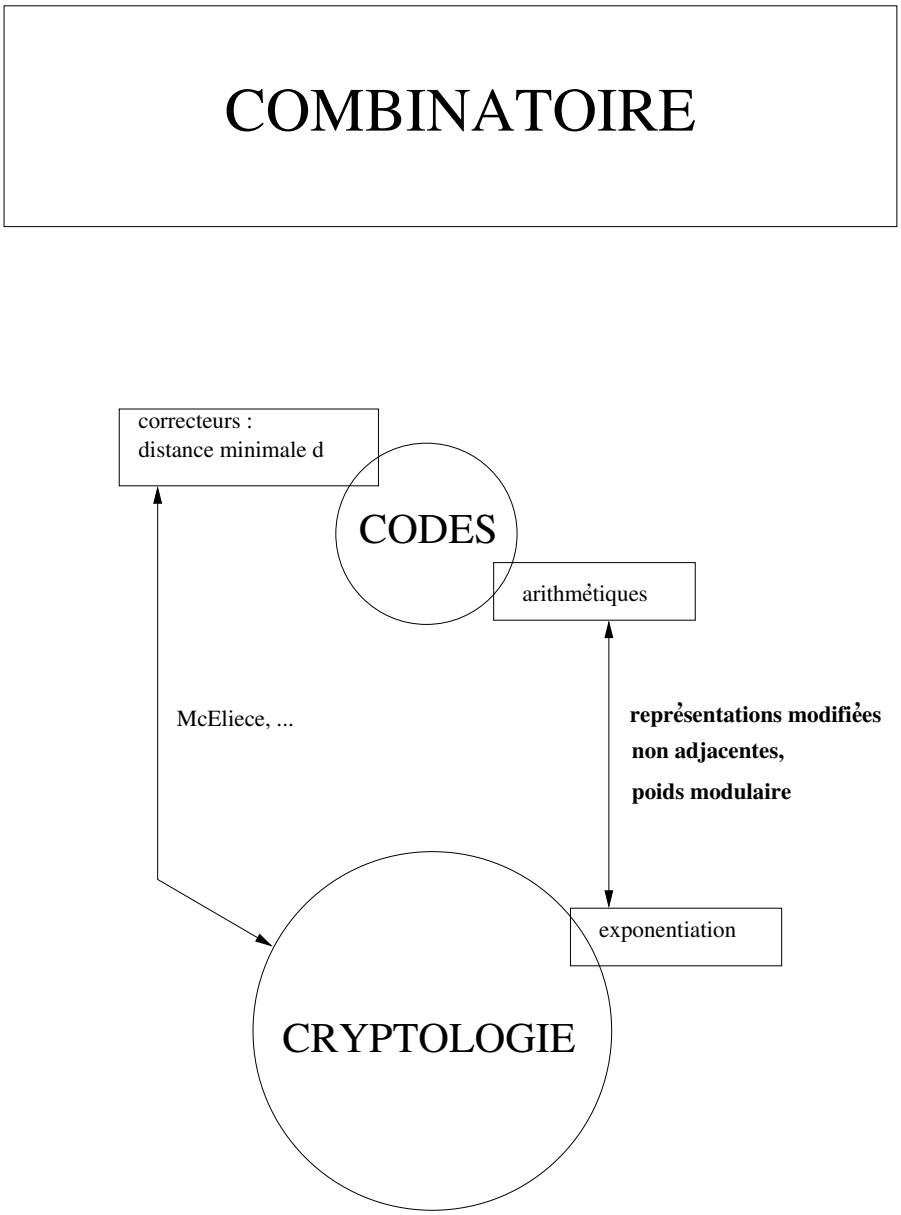

FIG. 3 – *Quelques liens entre codage et cryptologie ; en gras, les liens que nous avons étudiés.*

1 Éléments de codage

L'erreur naissait tantôt d'un élément dont il n'avait pas suspecté la présence, tantôt d'une bévue dans la supputation du temps, qui s'était avéré plus rétractile ou plus extensible que sur les horloges. *Marguerite Yourcenar*

Notre problématique peut s'énoncer de manière très générale : on se place dans un espace discret (espace vectoriel sur un corps fini, anneau d'entiers, graphe), muni d'une métrique (distance de Hamming, de Rao-Garcia, de Clark-Liang, du plus court chemin) et on étudie certaines propriétés de certains sous-ensembles (appelés *codes*) de cet espace, propriétés relatives à la métrique qu'on s'est donnée.

Nous avons divisé cette section en trois sous-sections. La première concerne les codes en blocs dans les espaces de Hamming, considérés sous deux angles différents, celui de la distance minimale (*codes correcteurs*) et celui du rayon de recouvrement (*codes couvrants*). La deuxième traite des *codes arithmétiques*, qui, pratiquement toujours vus sous l'angle de leur capacité de correction, nous paraissent présenter des caractéristiques suffisamment particulières pour être étudiés séparément. La troisième étudie les *codes identifiants*, qui constituent mon sujet de recherche prédominant depuis environ deux ans. Ces codes peuvent être vus comme un cas particulier des codes couvrants, mais nous les considérons dans d'autres graphes que les espaces de Hamming (ou n-cubes), et c'est pourquoi nous en faisons une sous-section à part.

1.1 Codes en blocs

Nous travaillerons le plus souvent sur l'alphabet constitué du corps à deux éléments $F = F_2 = \{0, 1\}$, et, par souci de simplicité, les définitions, notations, et notions de base que nous allons maintenant présenter ne le seront que dans ce cadre, leur généralisation au cas d'un corps fini F_q, où q est une puissance de nombre premier, se faisant sans difficulté. En particulier, tous les calculs ci-dessous sont effectués modulo 2.

On note F_2^n $(= F^n)$, et on appelle espace de Hamming, l'ensemble des vecteurs binaires

de longueur n, que l'on munit d'une métrique d, la *distance de Hamming*, définie entre deux vecteurs $\mathbf{x} = x_1x_2\ldots x_n \in F^n$ et $\mathbf{y} = y_1y_2\ldots y_n \in F^n$ comme le nombre de coordonnées sur lesquelles ils diffèrent : $d(\mathbf{x},\mathbf{y}) = |\{i \in \{1,2,\ldots,n\} \ : x_i \neq y_i\}|$. Le *poids* (de Hamming), $w(\mathbf{x})$, du vecteur $\mathbf{x}$ est sa distance (de Hamming) au vecteur nul et représente donc le nombre de ses composantes non nulles : $w(\mathbf{x}) = d(\mathbf{x},\mathbf{0})$. La distance entre le vecteur $\mathbf{x}$ et un sous-ensemble non vide $Y \subseteq F^n$ est définie par $d(\mathbf{x},Y) = \min\{d(\mathbf{x},\mathbf{y}) \ : \mathbf{y} \in Y\}$. On appelle *sphère* de centre $\mathbf{x}$ et de rayon t, et on note $B_t(\mathbf{x})$, l'ensemble des vecteurs de F^n à distance t ou moins de $\mathbf{x}$:

$$B_t(\mathbf{x}) = \{\mathbf{y} \in F^n : d(\mathbf{x},\mathbf{y}) \leq t\}.$$

Son volume, indépendant de son centre, sera noté $V(t)$. On dit que $\mathbf{x}$ est *t-couvert* (ou *couvert* s'il n'y a pas d'ambiguïté) par un vecteur $\mathbf{y} \in F^n$ si $d(\mathbf{x},\mathbf{y}) \leq t$, ou, de manière équivalente, $\mathbf{x} \in B_t(\mathbf{y})$ ou $\mathbf{y} \in B_t(\mathbf{x})$. Le vecteur $\mathbf{x}$ est dit couvert par un sous-ensemble non vide $Y \subseteq F^n$ s'il est couvert par au moins un élément de Y.

Un *code binaire* C de *longueur* n et de taille K ($K \geq 2$) est un ensemble de K vecteurs binaires de longueur n : $C \subseteq F^n$ et $|C| = K > 1$. Ses éléments sont appelés *mots de code*.

Lorsque C est un sous-espace vectoriel de dimension k de F^n, on parle de code *linéaire* de *dimension* k. On peut alors définir C par une *matrice génératrice*, $\mathbf{G}$, matrice de dimensions $k \times n$ dont les lignes forment une base de C : l'ensemble des 2^k combinaisons linéaires, à coefficients dans F, des lignes de $\mathbf{G}$ est égal à C. On appelle *code dual* de C, et on note $C^\perp$, l'ensemble des vecteurs *orthogonaux* à tous les vecteurs de C :

$$C^\perp = \{\mathbf{x} = x_1x_2\ldots x_n \in F^n : \forall \mathbf{c} = c_1c_2\ldots c_n \in C, < \mathbf{x},\mathbf{c} >= \sum_{1\leq i\leq n} x_ic_i = 0\}.$$

Le code $C^\perp$ est alors un sous-espace vetoriel de F^n, de dimension $n-k$, dont toute matrice génératrice $\mathbf{H}$, de dimensions $(n-k)\times n$, caractérise le code C de la manière suivante :

$$\mathbf{c} \in C \Longleftrightarrow \mathbf{c}\mathbf{H}^T = \mathbf{0},$$

où $\mathbf{0}$ désigne maintenant le vecteur nul de longueur $n-k$, et T le symbole de transposition des matrices. La matrice $\mathbf{H}$ est appelée *matrice de parité* du code C. On appelle *syndrome*

d'un vecteur $\mathbf{y} \in F^n$ le produit $\mathbf{y}\mathbf{H}^T \in F^{n-k}$. De ce qui précède on voit qu'un vecteur est mot de code si et seulement si son syndrome est nul (le code est le noyau d'une certaine application linéaire).

Les deux paramètres essentiels d'un code C, que nous allons étudier dans une partie de ce document, sont sa *distance minimale*, notée $d(C)$ ou d, et son *rayon de recouvrement*, noté $R(C)$ ou R. On dit alors que C est un code de paramètres $(n, K, d)R$, et $[n, k, d]R$ s'il est linéaire. On utilise aussi les notations (n, K), $[n, k]$, (n, K, d), $[n, k, d]$, $(n, K)R$, ou $[n, k]R$ si le besoin de spécifier la distance minimale ou le rayon de recouvrement ne se fait pas sentir.
Définition. La distance minimale d'un code C est la plus petite distance existant entre deux mots de code distincts :

$$d = d(C) = \min\{d(\mathbf{c}_1, \mathbf{c}_2) : \mathbf{c}_1 \in C, \mathbf{c}_2 \in C, \mathbf{c}_1 \neq \mathbf{c}_2\}.$$

Si l'on pose $e = \lfloor \frac{d-1}{2} \rfloor$, les sphères de rayon e centrées sur les mots de code ont deux à deux une intersection vide, et e est le plus grand entier ayant cette propriété.
Lorsque le code est linéaire,

$$d = d(C) = \min\{w(\mathbf{c}) : \mathbf{c} \in C, \mathbf{c} \neq \mathbf{0}\}.$$

Toujours dans le cas linéaire, la distance minimale de C peut être caractérisée à l'aide de la matrice de parité de la manière suivante : c'est le plus petit entier positif d tel que le vecteur $\mathbf{0}^T$ (de longueur $n-k$) peut s'exprimer comme somme de d colonnes d'une matrice de parité (de dimensions $(n-k) \times n$) de C.
Définition. Le rayon de recouvrement d'un code C est la plus grande distance existant entre le code et un vecteur de F^n :

$$R = R(C) = \max\{d(\mathbf{x}, C) : \mathbf{x} \in F^n\}.$$

En d'autres termes, les mots de code R-couvrent tout l'espace F^n, et R est le plus petit entier ayant cette propriété.

Dans le cas linéaire, le rayon de recouvrement de C peut être caractérisé à l'aide de la matrice de parité de la manière suivante : c'est le plus petit entier R tel que tout vecteur

transposé de longueur $n-k$ peut s'exprimer comme somme d'au plus R colonnes d'une matrice de parité (de dimensions $(n-k)\times n$) de C.

Un code de paramètres $(n,K,d)R$ vérifie les inégalités suivantes :

$$K\cdot V\Big(\Big\lfloor\frac{d-1}{2}\Big\rfloor\Big)\leq 2^n \quad \text{et} \quad K\cdot V(R)\geq 2^n. \tag{1.1}$$

Ces deux bornes sont très usuellement appelées (en français !) "sphere-packing bound" ou « borne de Hamming » pour la première, "sphere-covering bound" pour la seconde.

1.1.1 Codes correcteurs

Les codes correcteurs ont été inventés pour corriger des erreurs lors de la transmission sur un canal de communication bruité. Adoptons le modèle suivant, dit du canal *binaire symétrique sans mémoire*: on envoie des '0' et des '1' qui sont en général correctement transmis, mais occasionnellement (avec une probabilité $p<1/2$) un '1' se transforme en '0', ou un '0' en '1'. On va coder un bloc de k symboles d'information $\mathbf{u}=u_1u_2\ldots u_k$ par un mot de code $\mathbf{c}=c_1c_2\ldots c_n\in C$, avec $n\geq k$.

Pour illustrer cette procédure de la façon la plus simple possible, plaçons-nous dans le cas d'un code linéaire C de paramètres $[n,k,d]$, de matrice génératrice $\mathbf{G}$, et de matrice de parité $\mathbf{H}$. La capacité de détection et de correction d'erreurs de C est liée à sa distance minimale de manière directe : après codage du vecteur $\mathbf{u}$ par calcul du vecteur, de longueur n, $\mathbf{c}=\mathbf{uG}\in C$, et après transmission du mot de code $\mathbf{c}$, le destinataire reçoit le vecteur $\mathbf{z}=\mathbf{c}+\mathbf{e}$, où $\mathbf{e}\in F^n$ est le vecteur d'erreur. En rappelant que $e=\lfloor\frac{d-1}{2}\rfloor$, on constate que si le poids de l'erreur, $w(\mathbf{e})$, est au plus e, alors $\mathbf{c}$ est l'unique mot de code le plus proche de $\mathbf{z}$. Le paramètre e est appelé la *capacité de correction* de C, et C est appelé *code e-correcteur*; on dira indifféremment que C peut corriger e erreurs ou que C peut corriger une erreur de poids e.

Décoder (retrouver $\mathbf{c}$) peut s'effectuer à l'aide de la matrice $\mathbf{H}$: on calcule le syndrome de $\mathbf{z}$, $\mathbf{y}=\mathbf{zH}^T$, puis $\mathbf{c}^*=\mathbf{z}+\mathbf{x}$, où $\mathbf{x}$ est solution, de poids minimal, de l'équation $\mathbf{xH}^T=\mathbf{y}$. En effet, $\mathbf{c}^*\mathbf{H}^T=\mathbf{zH}^T+\mathbf{xH}^T=\mathbf{y}+\mathbf{y}=\mathbf{0}$: $\mathbf{c}^*$ est le mot de code le plus proche de $\mathbf{z}$.

On se trouve maintenant face à deux problèmes absolument cruciaux en théorie du codage :
1) Trouver des codes « grands » et « courts », ayant une « grande » distance minimale. Ces codes peuvent être linéaires ou non, mais la mise en œuvre pratique des codes linéaires est beaucoup plus simple.

Le plus souvent, on se fixe n et d et on cherche un code (n, K, d) (ou $[n, k, d]$) ayant la plus grande taille K (ou dimension k) possible, ou on se fixe n et k et on cherche un code $[n, k, d]$ ayant la plus grande distance minimale d possible, enfin on peut se fixer d et k et chercher la plus petite longueur n possible pour un code $[n, k, d]$.
2) Trouver des algorithmes de décodage performants.

Ainsi que nous le verrons en Section 4.1, ce sont là des problèmes difficiles (on peut d'ailleurs en profiter pour les utiliser en cryptographie, cf. Section 3.3). Cela n'a heureusement pas empêché la construction et la mise en pratique de vastes sous-classes de codes pour lesquels existent des algorithmes de décodage performants (pour ne citer qu'eux, les codes BCH, ou bien les codes de Goppa — voir page 44), mais cet aspect-ci des codes correcteurs, qui a suscité énormément de recherches, se situe en dehors de notre propos.

1.1.2 Codes couvrants

Les problèmes de recouvrement, mathématiquement intéressants par eux-mêmes, se prêtent à des applications dans le domaine de la transmission d'information (compression de données, décodage d'effacements, réseaux de diffusion, mémoires à écriture irréversible, codage de la parole, télécommunications cellulaires), ou, de manière plus légère, au loto sportif et au jeu de Berlekamp-Gale.

Ici on s'intéresse principalement, de manière plus ou moins directe, au problème suivant : trouver des codes « petits » et « longs », ayant un « petit » rayon de recouvrement ; usuellement, on cherche à déterminer $K(n, R)$, la plus petite taille K possible pour un code $(n, K)R$, ou bien $t[n, k]$, le plus petit rayon de recouvrement R possible pour un code $[n, k]R$. A codimension $m = n - k$ fixée, les valeurs dans une table de $t[n, k]$ se trouvent sur une diagonale parallèle à la diagonale principale. Lorsque n augmente, on descend sur cette dia-

gonale et $t[n, n-m]$ reste constante ou décroît. Typiquement, $t[n, n-m]$ reste constante pour plusieurs valeurs consécutives de n, puis diminue. Ces points de variation signalent une valeur de la *fonction-longueur* ℓ : si $t_0 = t[n, n-m] < t[n-1, n-1-m]$, alors $\ell(m, t_0) = n$. En d'autres termes, $\ell(m, R)$ est définie comme étant la plus petite longueur n pour laquelle il existe un code binaire linéaire $[n, n-m]R$. Pour les petites longueurs, on constate que la fonction ℓ constitue une manière plus compacte de représenter les valeurs de $t[n, k]$.

•▷ Parmi les travaux menés sur ce sujet depuis notre thèse et publiés entre autres dans les articles ou ouvrage suivants : [30], [31], [62], [28], [23] (voir aussi Section 4.1 pour des résultats tirés de [15] et [46]), nous avons choisi de mettre en relief les points ci-dessous :

Dans le cas des codes de petite taille ou de petite longueur, on peut utiliser des inégalités linéaires sur la distribution des poids des mots de code, des récurrences fondées sur la notion de code « équilibré » (code ayant, à un près, autant de '0' que de '1' dans chacune de ses colonnes), les propriétés de 2-surjectivité (existence des couples '00', '01', '10', et '11' sur deux colonnes quelconques du code), et des partitionnements de codes permettant la construction de codes plus longs ; par exemple, on peut ainsi montrer que $K(2p+3, p) = 7$ pour tout entier $p \geq 1$. En particulier, le code de longueur $2p+3$

$$C = \begin{array}{ccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \ldots & 1 & 1 \end{array}$$

a 7 mots et son rayon de recouvrement vaut p.

L'emploi de codes R-correcteurs, de distance minimale $2R+1$, emboîtés les uns dans les autres, a permis d'établir des bornes inférieures sur $K(n, R)$. Si $A(n, d)$ dénote le plus grand nombre possible d'éléments dans un code de longueur n et distance minimale d (avec

la convention que $A(n,d) = 1$ lorsque $d > n$), on obtient des résultats du type

$$K(n,R) \geq \frac{2^n - A(n,2R+1)\binom{2R}{R}}{\sum_{i=0}^{R}\binom{n}{i} - \binom{2R}{R}}, \tag{1.2}$$

ou

$$K(n,R) \geq \frac{2^n - 2A(n,2R+1)\binom{2R}{R}}{\sum_{i=0}^{R}\binom{n}{i} - \frac{3}{2}\binom{2R}{R}}, \tag{1.3}$$

à condition que les dénominateurs soient positifs. Aucune des deux inégalités ci-dessus n'est toujours meilleure que l'autre.

En utilisant des codes « constants par morceaux » et un système de Steiner (sS), on peut construire un code de longueur 11, rayon de recouvrement 1 et taille 192, montrant que $K(11,1) \leq 192$. Un code C est constant par morceaux s'il vérifie la propriété suivante : on partitionne sa longueur n en $n = n_1 + n_2 + \ldots + n_t$ et ses éléments $\mathbf{c}$ en $\mathbf{c} = \mathbf{c}_1|\mathbf{c}_2|\ldots|\mathbf{c}_t$, où | représente la concaténation et $\mathbf{c}_i$ a pour longueur n_i ; alors, si C contient un vecteur $\mathbf{c}$ tel que $w(\mathbf{c}_1) = w_1, w(\mathbf{c}_2) = w_2, \ldots, w(\mathbf{c}_t) = w_t$, il contient les

$$\binom{n_1}{w_1} \times \binom{n_2}{w_2} \times \cdots \times \binom{n_t}{w_t}$$

vecteurs de ce type. Rappelons qu'un sS $S(t,k,v)$ est un design particulier : c'est une collection de blocs (sous-ensembles de taille k) d'un ensemble S de taille v telle que tout sous-ensemble de taille t de S soit contenu dans exactement un bloc. Il existe un sS $S(4,5,11)$, contenant 66 blocs. Ces 66 blocs et leurs compléments (i.e., 66 vecteurs de poids 5 et 66 vecteurs de poids 6) couvrent tous les vecteurs de poids 4 à 7. Ecrivons $11 = 6 + 5$ et $(w_1, w_2) = (0,1), (0,2)$, ou $(2,0)$. Ceci décrit un code constant par morceaux, de taille 30, qui couvre tous les vecteurs de poids 3 ou moins, tandis que son complément couvre tous les vecteurs de poids 8 ou plus. Au total, on obtient un code $(11, 192)1$, et cette construction demeure la meilleure connue à ce jour — on sait par ailleurs que $K(11,1) \geq 180$ (Blass et Litsyn [9]).

La notion de *normalité* a été imaginée par Graham et Sloane [45] pour les codes linéaires. Nous l'avons étendue aux codes non linéaires ainsi qu'aux codes non binaires. Nous la présentons ici dans le cas binaire. Soit C un code $(n,K)R$. Pour i variant entre 1 et n, on note

$C_0^{(i)}$ (respectivement, $C_1^{(i)}$) l'ensemble des mots de code dont la i-ième composante vaut '0' (respectivement, '1'). L'entier

$$N^{(i)} = \max\{d(\mathbf{x}, C_0^{(i)}) + d(\mathbf{x}, C_1^{(i)}) : \mathbf{x} \in F^n\}$$

est appelé la *norme* de C relativement à i, et $N_{\min} = \min\{N^{(i)} : i = 1, 2, \ldots, n\}$ est appelé la *norme minimale* de C (nous utilisons la convention $d(\mathbf{x}, \emptyset) = \infty$). Enfin le code C est dit *normal* si sa norme minimale vaut au plus $2R + 1$.

L'intérêt des codes normaux est qu'ils se prêtent à d'efficaces constructions de codes couvrants. En particulier, l'existence d'un code normal $(n, K)R$ permet la construction de codes $(n + 2p, K)R + p$ pour tout entier positif p et rattache la normalité des codes à la conjecture $K(n + 2, R + 1) \leq K(n, R)$ (pour $R < n$) et sa contrepartie linéaire $t[n + 2, k] \leq t[n, k] + 1$ (pour $n \geq k \geq 1$).

Nous avons montré que la première inégalité est vraie, à R fixé, pour n assez grand, et étudié plus spécifiquement les cas $R = 1$ et $R = 2$: nous avons établi que $K(n + 2, 2) \leq K(n, 1)$ pour tout $n \geq 2$, sauf peut-être pour $n = 9$ et $n = 16$ (depuis, il a été montré que cette inégalité est vraie pour tout $n \geq 2$), et que $K(n + 2, 3) \leq K(n, 2)$ pour tout n appartenant à l'ensemble $\{1\} \cup \{3, \ldots, 7\} \cup \{20, \ldots, 28\} \cup \{43, 44\} \cup \{91, \ldots, 127\} \cup \{187, \ldots, 361\}$ et $n \geq 379$.

Nous avons prouvé qu'un code *linéaire* est normal si l'une des conditions suivantes est vérifiée : sa longueur est inférieure ou égale à 12 ; sa dimension est inférieure ou égale à 2 ; sa distance minimale est inférieure ou égale à 3 ; son rayon de recouvrement est inférieur ou égal à 2 (les records actuels sont de 15, 5, 4, et 3, respectivement). Les méthodes utilisées sont assez techniques et *ad hoc*, et consistent notamment à observer le nombre d'occurrences de colonnes identiques dans un code, et à étudier son code contracté, obtenu en prenant une seule copie des colonnes multiples.

Dans le cas binaire, la normalité des codes linéaires a fait couler beaucoup d'encre, et aucun code linéaire anormal n'est connu.

Dans le cas non binaire, la généralisation directe de la définition de la normalité (pour chaque coordonnée i, on définit q sous-codes $C_a^{(i)}$ selon la valeur a de la i-ième composante

des mots de code, et le code C est normal si sa norme minimale vaut au plus $qR + q - 1$) se révèle moins efficace ; il en est de même pour une généralisation (la *sous-normalité*) où l'on considère une partition *quelconque* du code en q sous-codes. Par exemple, alors que dans le cas binaire, tous les codes parfaits sont normaux et l'on ne connaît aucun code qui ne soit pas sous-normal, dans le cas q-aire aucun code parfait n'est sous-normal.

Nous avons montré que la conjecture $t[n,k] \leq t[n+1,k+1]+1$, pour $n \geq k \geq 1$, constatée pour de petites longueurs (≤ 64), n'est en général pas vérifiée. Il en résulte que pour de grandes longueurs, dans une table de valeurs de $\ell(m,R)$, il existe des suites arbitrairement longues de valeurs pouvant s'exprimer avec seulement deux valeurs de la fonction $t[n,k]$.

Sur ce vaste sujet, nous avons écrit, Gérard Cohen, Iiro Honkala, Simon Litsyn, et moi-même, une monographie, intitulée "Covering Codes" [23], parue en 1997. Elle compte xxii+542 pages, vingt chapitres (1. Introduction 2. Basic Facts 3. Constructions 4. Normality 5. Linear Constructions 6. Lower Bounds 7. Lower Bounds for Linear Codes 8. Upper Bounds 9. Reed-Muller Codes 10. Algebraic Codes 11. Perfect Codes 12. Asymptotic Bounds 13. Weighted Coverings 14. Multiple Coverings 15. Football Pools 16. Tilings 17. Writing on Constrained Memories 18. Subset Sums and Constrained Memories 19. Heterodox Coverings 20. Complexity), 714 références et 24 tables, dont une table de bornes sur $K(n,R)$ pour $n \leq 33$ et $R \leq 10$, et une table de bornes sur $t[n,k]$ pour $k \leq n \leq 64$. Nous ne résistons pas au plaisir de livrer ici un court extrait de sa recension de cinq pages parue en 1999 dans Mathematical Reviews sous la plume du Professeur H.F. Mattson, Jr., de l'Université de Syracuse aux Etats-Unis :

> Covering radius of codes lay dormant for years after first appearing, unnamed, in Gorenstein, Peterson, and Zierler's 1960 paper [D. Gorenstein, W. W. Peterson and N. Zierler, Information and Control 3 (1960), 291–294; MR 22 9350]. (...) A second survey paper, by Cohen et al. [Appl. Algebra Engrg. Comm. Comput. 8 (1997), no. 3, 173–239; MR 98d:94047], had 280 items. The book under review, with far more complete coverage of the topic, has 714 entries in its bibliography.
>
> (...)

> The book has a full account of all aspects of covering radius. After introductory sections on finite fields and codes, one almost never finds a theorem stated without proof. The proofs are leisurely and complete. The book could thus be useful for beginners and experts alike.
>
> (...)
>
> This excellent book is smoothly written, with leisurely proofs and good motivation. There are a few new results in it, but the authors were too modest to mark them as new for the reader. I do have one complaint: the authors' grating neologisms "upperbound" and "upperestimate" (used as verbs) should be "bound above". As nouns they should be two words.
>
> The authors have obviously paid careful attention to their writing; there is a uniform style, fluid and clear, with no jarring changes from one chapter to the next. (...) It would be hard to imagine a better, more thorough, up-to-date, and authoritative treatment of covering codes than the one we find in this book.

En ce qui concerne les résultats de complexité liés à ces problèmes de recouvrement, nous renvoyons le lecteur à la Section 4.1, où nous énonçons plusieurs résultats de NP- et de Π_2-complétude.

1.1.3 Codes parfaits

Dans cette sous-section, nous nous plaçons dans le cas des codes ayant pour alphabet le corps fini $F_q = \{0, 1, \ldots, q-1\}$ (où q est une puissance de nombre premier). Un code est dit *parfait* si $d = 2R + 1$: les sphères de rayon $e = R$ remplissent tout l'espace et ont deux à deux une intersection vide. Les deux inégalités (1.1) sont alors vérifiées avec égalité.

Nous aurons besoin ici de la notion d'équivalence entre codes. Deux codes q-aires C_1 et C_2, de paramètres (n, K), sont dits *équivalents* s'il existe n permutations $\tau_1, \tau_2, \ldots, \tau_n$ sur F_q et une permutation σ des n coordonnées telles que, si $c_1 c_2 \ldots c_n \in C_1$, alors $\sigma\Big(\tau_1(c_1)\tau_2(c_2)\ldots \tau_n(c_n)\Big) \in C_2$. Dans le cas binaire, cela revient à l'existence d'un vecteur $\mathbf{a} \in F^n$ et d'une permutation σ des n coordonnées tels que $C_2 = \{\sigma(\mathbf{c}) + \mathbf{a} \ : \mathbf{c} \in C_1\}$.

Il est bien connu, depuis les années '70, que les seuls codes parfaits non triviaux sur F_q sont, à une équivalence près :

1) les codes binaires à répétition de longueur impaire (longueur $n = 2p+1$, dimension $k = 1$, distance minimale $d = 2p+1$, et rayon de recouvrement $R = p$, pour tout entier $p \geq 1$) ;

2) les codes ayant les mêmes paramètres que les codes q-aires de Hamming (longueur $n = (q^m-1)/(q-1)$, taille $K = q^{n-m}$, distance minimale $d = 3$, et rayon de recouvrement $R = 1$, pour tout entier $m \geq 2$) ;

3) le code de Golay binaire (longueur $n = 23$, dimension $k = 12$, distance minimale $d = 7$, et rayon de recouvrement $R = 3$) ;

4) le code de Golay ternaire (longueur $n = 11$, dimension $k = 6$, distance minimale $d = 5$, et rayon de recouvrement $R = 2$)

(voir par exemple [23, Sec. 11.1 et 11.2]).

On constate donc que seul le deuxième cas est susceptible de fournir des codes parfaits qui ne soient pas équivalents à des codes linéaires. Ce cas en fournit effectivement, et, même dans le cas binaire, on n'a pas encore décrit tous les codes parfaits ayant les mêmes paramètres que les codes de Hamming. La première famille de codes parfaits binaires non linéaires a été construite en 1962 (Vasiliev [74]). Depuis, plusieurs autres classes de ces codes ont vu le jour (voir par exemple [23, Sec. 11.3 et 11.4] pour un panorama de ces constructions, et des références).

●▷ Nous avons contribué à cet édifice [63], [76] dans le cas binaire : en utilisant des codes concaténés généralisés (Zinoviev [75]), nous avons obtenu une première famille de constructions de codes parfaits ayant les paramètres indiqués ci-dessus en 2), pour $q = 2$, pour lesquelles nous avons une borne inférieure sur le nombre de codes *non équivalents* ainsi construits [63]. En utilisant les mêmes idées, nous obtenons dans [76] de nouvelles constructions de codes parfaits ayant ces paramètres, où nous pouvons donner une borne inférieure sur le nombre de codes *différents* ainsi construits. Ces bornes inférieures ne sont pas les meilleures que l'on connaisse, mais ces constructions et ces bornes peuvent également s'appliquer pour construire d'autres codes, pas seulement parfaits.

•▷ Toujours dans le domaine des codes parfaits, mais attaqués sous un autre angle, nous nous sommes intéressés au problème suivant [2] : considérons des codes binaires, parfaits, étendus, c'est-à-dire des codes ayant les paramètres suivants : longueur $n = 2^t$ ($t \geq 2$ entier), cardinal 2^{n-1-t}, distance minimale $d = 4$, sur F. On sait que n codes parfaits étendus C_1, C_2, ..., C_n, peuvent constituer une partition de $E^n \subset F^n$, l'ensemble des vecteurs de poids pair, et que n codes parfaits étendus C_{n+1}, C_{n+2}, ..., C_{2n}, peuvent constituer une partition de $O^n = F^n \setminus E^n$, l'ensemble des vecteurs de poids impair. Etant donnée une autre partition, D_1, D_2, ..., D_n, de E^n, et D_{n+1}, D_{n+2}, ..., D_{2n}, de O^n, on définit la *matrice d'intersection* des partitions C et D, $\mathbf{IM}(C, D)$, par :

$$\mathbf{IM}(C, D) = [|C_i \cap D_j|]_{i=1,...,2n, j=1,...,2n},$$

et on cherche à construire des matrices d'intersection qui soient différentes, ou qui soient non équivalentes, ainsi qu'à évaluer leur nombre.

En utilisant des carrés latins, nous montrons que le nombre de matrices différentes est compris entre 2^{cn^2} et $2^{c'n^3}$, où n est suffisamment grand et c et c' sont des constantes positives (le nombre de matrices non équivalentes est du même ordre de grandeur).

1.2 Codes arithmétiques

Après les codes en blocs, nous développons ici un peu plus largement les notions de base relatives aux codes arithmétiques, que nous pensons moins connus, moins étudiés, et moins familiers, même pour les codeurs.

1.2.1 Poids et distances

Les codes arithmétiques permettent la correction et la détection d'erreurs dans les processeurs arithmétiques réalisant des opérations arithmétiques telles que : addition, soustraction, complémentation, décalage, ...

Soit r ($r \geq 2$) la base choisie pour représenter des entiers positifs : si I est un entier positif, la *représentation en base* r de I s'écrit $I = \sum_i a_i r^i$, où $0 \leq a_i < r$ pour tout i. Cette

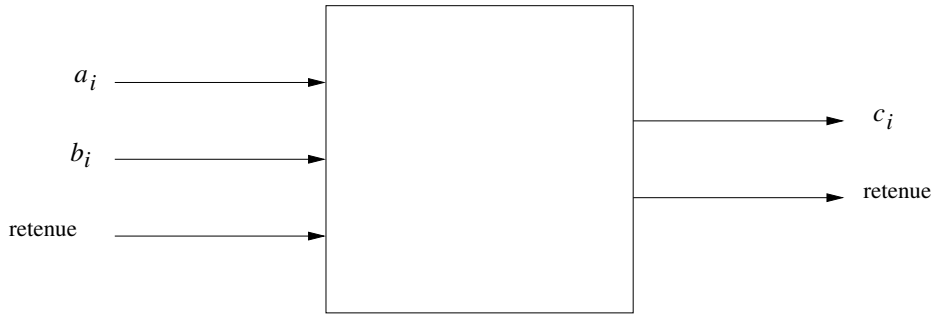

FIG. 4 – *L'unité i.*

représentation est unique. Lorsque $a_i = 0$ pour tout $i \geq n$, et $a_{n-1} \neq 0$, on peut écrire I sous forme d'un n-uple : $I = (a_{n-1}a_{n-2}\ldots a_1a_0)$ ou $I = a_{n-1}a_{n-2}\ldots a_1a_0$ (poids fort à gauche).

Le circuit effectuant l'addition de deux entiers positifs ainsi représentés est une suite d'unités élémentaires formant la somme c_i (modulo r) de leurs entrées (a_i, b_i, et une retenue) et une retenue (voir Figure 4). Une erreur de l'unité i provoque une somme (modulo r) c_i fausse, ou une retenue fausse, c'est-à-dire une erreur $\pm e_i r^i$ ($|e_i| < r$), ou $\pm e_{i+1} r^{i+1}$ ($|e_{i+1}| < r$). Si globalement on est en présence d'une erreur E (différence entre le résultat obtenu et le résultat exact), on pense alors très naturellement, pour définir le poids de l'erreur E, à considérer le nombre minimal de termes $\pm e_i r^i$ qui ont pour somme E. Formalisons cette approche du poids arithmétique : on appelle *représentation modifiée en base r* d'un entier I (positif, négatif, ou nul) toute représentation de la forme

$$I = \sum_i a_i r^i, \text{ où } |a_i| < r \text{ pour tout } i. \tag{1.4}$$

Cette représentation n'est en général pas unique. Toute représentation (1.4) comportant un nombre minimal de coefficients a_i non nuls est dite *minimale*. Une représentation minimale n'est généralement pas unique elle non plus.

Définitions. Le *poids arithmétique* d'un entier I, noté $W(I)$, est le nombre de termes non nuls d'une représentation modifiée minimale de I. La *distance arithmétique* entre deux entiers I_1 et I_2, notée $D(I_1, I_2)$, est le poids arithmétique de leur différence.

Chaque base r définit donc un poids arithmétique sur Z et une distance arithmétique sur

$Z \times Z$. Maintenant, comment calculer le poids arithmétique d'un entier? Ce n'est pas aussi élémentaire que dans le cas du poids de Hamming.

Il existe des algorithmes directs pour ce faire (Chiang et Reed [19]). Cependant, comme nous en aurons besoin par la suite (voir Section 5), nous allons décrire une nouvelle représentation modifiée, qui, étant minimale et simple à former, permet aussi un calcul simple du poids arithmétique. On appelle *représentation modifiée non adjacente en base* r (RMNA) d'un entier I toute représentation $I = \sum_{0 \le i \le n} a_i r^i$, où $|a_i| < r$ pour $i = 0, 1, \ldots, n$ et où de plus, pour $i = 0, 1, \ldots, n-1$, la condition suivante est vérifiée :

$$\Big(a_i a_{i+1} = 0\Big) \text{ ou } \Big(a_i a_{i+1} > 0 \text{ et } |a_i + a_{i+1}| < r\Big) \text{ ou } \Big(a_i a_{i+1} < 0 \text{ et } |a_i| < |a_{i+1}|\Big).$$

Le qualificatif « non adjacente » provient du cas binaire où la condition ci-dessus se réduit à $a_i a_{i+1} = 0$, ce qui signifie qu'il n'y a pas de termes non nuls adjacents.

La RMNA possède les propriétés suivantes (Clark et Liang [20]) : pour tout entier I, elle existe, elle est unique, elle est minimale, et elle est facile à établir, par exemple à partir de la représentation en base r de I ; il suffit en effet de calculer $(r+1)I$, de lui retrancher I en gardant les termes négatifs, et de diviser par r (en enlevant le dernier 0) pour obtenir la RMNA de I.

Nous allons maintenant changer légèrement de cadre, en considérant l'addition modulaire de deux éléments I_1 et I_2 de l'anneau $Z_m = \{0, 1, \ldots, m-1\}$ $(m > 0)$:

$$I_1 \oplus I_2 = \begin{cases} I_1 + I_2, & \text{si} \quad I_1 + I_2 < m, \\ I_1 + I_2 - m, & \text{si} \quad I_1 + I_2 \ge m. \end{cases}$$

Si le résultat exact $I_1 \oplus I_2$ vaut J et le résultat trouvé vaut K, on définit l'*erreur d'anneau* $F \in Z_m$ en posant $K = J \oplus F$. Par rapport à l'erreur E définie par $K = J + E$, on a $E = F$ ou $E = F - m$ selon que $E > 0$ ou $E < 0$, d'où les définitions suivantes (Rao et Garcia [71]) :

Définitions. On appelle *poids modulaire de Rao-Garcia* d'un entier $I \in Z_m$, et on note $W_{RG}(I)$, le plus petit des deux entiers $W(I)$ et $W(m-I)$. On appelle *distance modulaire de Rao-Garcia* entre deux entiers I_1 et I_2 de Z_m, et on note $D_{RG}(I_1, I_2)$, le poids modulaire de Rao-Garcia de leur différence modulaire.

Chaque couple (r, m) définit donc un poids modulaire sur Z_m (que l'on peut calculer par comparaison de deux poids arithmétiques) et une distance modulaire sur $Z_m \times Z_m$. Le problème est que ce poids et cette distance ne vérifient pas toujours l'inégalité triangulaire (par exemple, dès que $W(m) \geq 5$). Ils la vérifient toutefois dans les cas le plus souvent utilisés dans la pratique: $m = r^n$ ou $m = r^n \pm 1$. Jusqu'à la fin de cette section sur les codes arithmétiques, nous utiliserons les termes « poids » et « distance » même quand l'inégalité triangulaire n'est pas satisfaite, et « métrique » pour insister sur le fait qu'elle l'est. Ernvall [38], [39], [40] donne les conditions nécessaires et suffisantes sur r et m pour que D_{RG} soit une métrique.

Dans les cas où D_{RG} est une métrique, on peut se poser des problèmes concernant les codes parfaits (voir plus bas, page 23). Que l'inégalité triangulaire soit vérifiée est en effet crucial pour que deux sphères de rayon t dont les centres sont distants de $2t + 1$ soient disjointes !

Une autre définition du poids modulaire coexiste avec celle que nous venons de donner. Elle est plus récente (Clark et Liang [21]), satisfait l'inégalité triangulaire, mais se montre plus difficile à calculer.

Définitions. Le *poids modulaire de Clark-Liang* d'un entier I est défini par $W_{CL}(I) = \min\{W(J) : J \in Z, J = I \bmod m\}$. La *distance modulaire de Clark-Liang* entre deux entiers I_1 et I_2, notée $D_{CL}(I_1, I_2)$, est le poids modulaire de Clark-Liang de leur différence.

Chaque couple (r, m) définit donc un poids modulaire sur Z_m (ou sur Z) et une distance modulaire (qui est une métrique) sur $Z_m \times Z_m$ (ou sur $Z \times Z$).

A notre connaissance, la complexité du calcul de ce poids modulaire n'a été abordée nulle part (voir Section 4.2).

●▷ Le premier problème auquel nous nous sommes attaqué est de déterminer les cas où distance modulaire de Rao-Garcia et distance modulaire de Clark-Liang coïncident. On peut s'appuyer sur les résultats d'Ernvall susmentionnés donnant les cas où D_{RG} est une métrique, puisque D_{RG} ne peut être égale à D_{CL} si elle n'est pas une métrique. On a donc seulement à étudier les cas suivants: $W(m) = 1$, $W(m) = 2$, $W(m) = 3$ et la RMNA de m a l'une

parmi 22 formes possibles, ou enfin $W(m) = 4$ et la RMNA de m a l'une parmi 10 formes possibles ; dans le cas binaire, cela se réduit à $W(m) = 1$, $W(m) = 2$, ou $W(m) = 3$ et la RMNA de m a l'une des deux formes suivantes : $2^n + 2^{n-2} \pm 2^i$ $(i \leq n-4)$ ou $2^n - 2^j \pm 2^i$ $(n-5 \leq j \leq n-2,\ i \leq j-2)$.

En fonction de r et m, nous avons obtenu une caractérisation quasi-totale [53], [54]. Seul a résisté un sous-cas de l'une des 22 formes possibles du cas où $W(m) = 3$. Pour $r \leq 13$, la caractérisation est complète ; en particulier :

•▷ Dans le cas $r = 2$, les deux distances modulaires D_{RG} et D_{CL} coïncident si et seulement si $W(m) \leq 2$.

Nous noterons $D_{CL} < D_{RG}$ lorsque les deux distances ne coïncident pas.

1.2.2 Codes arithmétiques

Les codes arithmétiques, conçus pour la correction d'erreurs portant sur des entiers, vont représenter ces entiers, sur lesquels on veut opérer (addition, addition modulaire, ...), de manière *redondante* ; par exemple, si on veut effectuer des additions modulaires, on peut coder les entiers $0, 1, 2, \ldots, B-1$ en les multipliant par un entier A : le code C va donc être constitué des entiers $0, A, 2A, \ldots, (B-1)A$. En posant $m = AB$, on pourra vérifier et corriger l'addition modulo m de deux mots de code de la manière suivante : si AI_1 et AI_2 sont deux mots de code, leur somme modulaire $AI_1 \oplus AI_2$ vaut $A(I_1+I_2)$ si $I_1+I_2 < B$ et $A(I_1+I_2-B)$ si $I_1 + I_2 \geq B$; c'est un multiple de A compris entre 0 et $A(B-1)$, donc un mot de code. La correction d'erreurs va consister à rechercher le mot de code qui est le plus proche (au sens de la métrique choisie) du résultat effectivement obtenu $I = AI_1 \oplus AI_2 \oplus F$; la détection d'erreurs peut se faire en calculant le reste de la division de I par A : ce reste, qui ne dépend que de F puisqu'il est égal au reste de la division de F par A, est appelé le *syndrome* de I. Comme pour les codes en blocs, le syndrome dépend de l'erreur (la maladie) et non du mot de code (le patient), ce qui explique cette terminologie médicale. Si le syndrome est nul (pas d'erreur, ou une erreur indétectable car multiple de A), on conclut que le résultat est correct ; s'il n'est pas nul, on détecte une erreur.

Remarque. Une telle détection doit être effectuée après un grand nombre d'additions : vérifier chaque addition en faisant une division ne serait ni économique ni fiable.

L'anneau Z_m représente l'ensemble des résultats possibles (éventuellement faux), $C \subseteq Z_m$ l'ensemble des résultats justes ; les entiers $0, 1, \ldots, B-1$ constituent l'*information*, et B est le *rang* de l'information. Le nombre de symboles nécessaires pour représenter un mot de code, dans la base r choisie, est la *longueur* du code. La *redondance* est donnée par $\log_r(A)$ et A est appelé le *générateur* du code.

Les codes de type $\{0, A, 2A, \ldots, (B-1)A\}$ sont appelés AN-*codes*. Mais d'une manière plus générale, de même que les codes en blocs linéaires sont des sous-espaces vectoriels alors que les codes non linéaires sont de simples sous-ensembles, on pourra considérer qu'un code arithmétique C est un sous-ensemble, sans structure particulière, de l'ensemble d'entiers Z_m. On peut ensuite se poser, pour les codes arithmétiques, tous les problèmes classiques de la théorie du codage, bien qu'ils paraissent ici beaucoup plus complexes à résoudre.

•▷ Le deuxième problème auquel nous nous sommes attaqué est l'étude de l'existence de codes arithmétiques parfaits (cf. Section 1.1.3), pour l'une ou l'autre des deux distances modulaires.

On se fixe r et m. Rappelons qu'un code e-correcteur (de distance minimale $d = 2e+1$) $C \subseteq Z_m$ est parfait si et seulement si

$$|C| \cdot V(e) = m, \tag{1.5}$$

où $V(e)$ est le volume de la sphère de rayon e (ce volume est indépendant du centre de la sphère et vaut $|\{y \in Z_m \ : W_{CL}(y) \leq e\}|$ ou $|\{y \in Z_m \ : W_{RG}(y) \leq e\}|$ selon le cas). Remarquons que pour un AN-code $C \subseteq Z_m$, $d = D_{RG}(C) = \min\{D_{RG}(x,y) \ : x \in C, y \in C, x \neq y\} = \min\{W_{RG}(x) \ : x \in C, x \neq 0\} = \min\{W(x) \ : x \in C, x \neq 0\}$, et qu'un AN-code parfait e-correcteur a pour générateur $V(e)$: $C = \{0, V(e), 2V(e), \ldots, (|C|-1)V(e)\}$ (cette dernière propriété est également vraie pour D_{CL}).

Comme on va le voir, la réponse à ce problème est loin d'être complète — en particulier loin d'être aussi complète que dans le cas des codes en blocs.

Considérons d'abord la distance modulaire de Rao-Garcia dans le cas binaire. On a déjà vu (page 22) qu'il s'agit d'une métrique si et seulement si la RMNA de m a l'une des formes suivantes : *f1*. 2^n ; *f2*. $2^n \pm 2^j$ $(j \leq n-2)$; *f3*. $2^n + 2^{n-2} \pm 2^i$ $(i \leq n-4)$; *f4*. $2^n - 2^j \pm 2^i$ $(n-5 \leq j \leq n-2,\ i \leq j-2)$. Astola [1] a établi les faits suivants, pour des AN-codes 1-correcteurs :

Dans le cas *f1*, aucun code parfait n'existe. Dans le cas *f2*, une condition nécessaire pour l'existence de codes parfaits est que $j = 0$ $(m = 2^n \pm 1)$, et on retrouve une classe de codes parfaits bien connus, les codes de Brown-Peterson (voir par exemple Rao [70]). Les cas *f3* et *f4* fournissent également de nombreux codes parfaits. Leur caractérisation n'a toutefois pas été établie.

Plaçons-nous maintenant dans le cas ternaire. Il existe une famille infinie de codes parfaits, donnés par $m = 3^n - 1$, $n = 2e+1$, $C = \{0, m/2\}$ (Gordon [44]). Ces codes, sortes de codes à répétition, sont e-correcteurs, et ce sont à ce jour les seuls codes parfaits connus (non triviaux) corrigeant plus d'une erreur.

●▷ Nous avons trouvé deux nouveaux codes parfaits ternaires corrigeant une erreur [58] :

Les AN-codes de générateur 37 et de moduli $m_1 = 3^9 - 2 \cdot 3^7 + 3^2$ et $m_2 = 3^9 + 1$ sont parfaits 1-correcteurs.

Je donne ici la démonstration de ce résultat, qui offre un aperçu des techniques d'arithmétique utilisées dans ce domaine. Soient $C_1 = \{0, 37, 74, \ldots, 15281\}$ (C_1 a 414 mots) et $C_2 = \{0, 37, 74, \ldots, 19647\}$ (C_2 a 532 mots) les deux codes considérés. Les moduli m_1 et m_2 sont tels que D_{RG} soit une métrique, et il est immédiat de vérifier que, dans les deux cas, le volume de la sphère de rayon un vaut 37 : pour m_1 par exemple, les entiers de poids modulaire zéro ou un sont : $0, 1, 15317, 2, 15316, 3, 15315, 6, 15312$, etc. Il faut enfin montrer que C_1 et C_2 sont 1-correcteurs. Soit x le plus petit entier strictement positif tel que $37x$ ait un poids arithmétique strictement inférieur à trois (c'est-à-dire égal à deux, car 37 est premier avec 3). Posons $37x = a \cdot 3^i + \varepsilon b \cdot 3^j$ $(0 \leq j < i, a = 1 \text{ ou } 2, b = 1 \text{ ou } 2, \varepsilon = \pm 1)$. D'après le théorème de Gauss, 3^j divise x, d'où $37 \cdot \dfrac{x}{3^j} = a \cdot 3^{i-j} + \varepsilon b$, et donc $W(37 \cdot \dfrac{x}{3^j}) = 2$. Alors, pour ne pas contredire la définition de x, $j = 0$: $a \cdot 3^i = -\varepsilon b \bmod 37$. L'étude des premières puissances

de 3 modulo 37 ($3, 9, 27, 7, 21, 26, 4, 12$ et 36) et de leur double ($6, 18, 17, \ldots$) montre que x est donné par $37x = 3^9 + 1 > \max\{m_1 - 37, m_2 - 37\}$. Donc tous les mots de code ont un poids arithmétique au moins égal à trois, ce qui clôt la démonstration.

En conclusion de cette étude très partielle des codes parfaits dans le cas de la distance modulaire de Rao-Garcia, remarquons qu'aucun code parfait n'est connu lorsque $r > 3$. Tous les codes parfaits que nous avons mentionnés sont des AN-codes (naturellement, des codes parfaits non AN peuvent en être déduits par addition ; mais nous n'avons pas d'exemple où des codes parfaits non AN existeraient sans que des AN-codes existent également). On est très loin d'une description totale des codes parfaits pour D_{RG}.

On en est encore plus éloigné pour la distance modulaire de Clark-Liang. Certains résultats obtenus pour D_{RG} fournissent des codes parfaits pour D_{CL} ... quand les deux distances coïncident. C'est le cas pour les codes de Brown-Peterson ($m = 2^n \pm 1$) ou pour les codes ternaires à répétition ($m = 3^n - 1$).

•▷ On peut se demander, lorsque $D_{CL} < D_{RG}$ et un code parfait $C \subseteq Z_m$ existe pour la métrique de Rao-Garcia, si ce même code C peut être parfait vis-à-vis de la distance de Clark-Liang. Nous avons démontré que la réponse est négative dans le cas des codes 1-correcteurs [55], et conjecturons qu'elle est négative dans tous les cas.

Pour le moment, on ne connaît pas de codes parfaits pour D_{CL} lorsque $D_{CL} < D_{RG}$.

Pour souligner toute la difficulté de cette entreprise (mais, c'est bien connu, il n'est pas nécessaire d'espérer pour entreprendre, ni de réussir pour persévérer), examinons le volume de la sphère, qui, en vertu de l'égalité (1.5), fournit une condition *nécessaire* à l'existence de codes parfaits : un code $C \subseteq Z_m$ ne peut être parfait e-correcteur que si $V(e)$ divise m.

Or, même dans le cas de la métrique de Rao-Garcia, *on ne sait pas calculer le volume de la sphère dans tous les cas* et quand on sait le faire (voir principalement Ernvall [41], [42]), on obtient des expressions extrêmement compliquées, que nous ne donnerons pas ici, de peur que le lecteur jusqu'ici bienveillant ne rejette au loin ce document en bâillant lugubrement. A partir de ces résultats partiels, en étudiant la divisibilité de m par $V(e)$, on peut établir quelques résultats d'inexistence de codes parfaits pour la métrique de Rao-Garcia, ou au

contraire trouver des paramètres de codes candidats à être parfaits. On obtient des résultats du genre (Gordon [44]) :

Pour $m = r^n \pm 1$, pour $e = 2$ et $V(e) < 2^{41}$, ou $e = 3$ et $V(e) < 2^{50}$, les seuls AN-codes parfaits e-correcteurs sont les codes ternaires $\{0, (3^5 - 1)/2\}$ et $\{0, (3^7 - 1)/2\}$, qui sont 2- et 3-correcteurs, respectivement (ce sont les codes à répétition que nous avons déjà mentionnés).

•▷ Nous avons établi les résultats suivants [58] :

1) Dans le cas $r = 2$, pour $m < 2^{33} + 2^{31} - 1$ et $e \geq 2$, aucun AN-code parfait n'existe.
2) Dans le cas $r = 3$, pour $m < 2 \cdot 3^{27} - 3^{26} - 2$ et $e \geq 2$, les seuls AN-codes parfaits sont les codes ternaires à répétition $\{0, (3^{2e+1} - 1)/2\}$.

On obtient de nombreux cas de divisibilité de m par $V(e)$ où on peut facilement éliminer les AN-codes (en établissant que le générateur $V(e)$ ou un de ses multiples est de poids $2e$ ou moins) mais où des codes non AN restent candidats.
•▷ On n'arrive alors qu'à éliminer les cas de petite taille, et nous n'avons trouvé aucun code parfait de ce type (voir [58], [59], et tous nos travaux précédents y mentionnés).

Pour conclure, on pourrait dire qu'en matière de codes arithmétiques parfaits, notre ignorance comporte peu de lacunes et qu'il faudrait, pour y remédier et espérer faire des avancées majeures, pouvoir élaborer des outils algébriques ou arithmétiques d'une puissance comparable à celle du théorème de Lloyd dans le cadre de l'espace de Hamming sur F_q.

•▷ Enfin, troisième problème concernant les codes arithmétiques, nous avons mené une étude asymptotique sur les codes arithmétiques binaires [49], dans les cas (où les deux distances modulaires coïncident) $m = 2^n$ et $m = 2^n \pm 1$. Les arguments usuels pour les codes correcteurs et couvrants mènent à des bornes de type Hamming et Varshamov-Gilbert : soient $M_a(n, d)$ la taille maximale d'un code arithmétique de distance minimale d, $R_a = R_a(n, d) = \frac{1}{n} \log_2 M_a(n, d)$ le *rendement* d'un tel code, $\delta = d/n$ sa *distance normalisée*, et H_2 la fonction entropie binaire.

Nous utilisons la notation $f(n) \lesssim g(n)$ lorsque n tend vers l'infini pour signifier que $f(n) \leq g(n)\big(1 + \varepsilon(n)\big)$ où $|\varepsilon(n)|$ tend vers 0 lorsque n tend vers l'infini.

Alors les inégalités asymptotiques suivantes sont vérifiées (Kabatianski [48]) :

$$R_a \lesssim (1-\delta/2)\Big(1 - H_2\Big(\frac{\delta/2}{1-\delta/2}\Big)\Big) \quad \text{— borne de Hamming arithmétique,}$$

$$R_a \gtrsim (1-\delta)\Big(1 - H_2\Big(\frac{\delta}{1-\delta}\Big)\Big) \quad \text{— borne de Varshamov-Gilbert arithmétique,}$$

lorsque n tend vers l'infini.

Remarquons que la borne de Varshamov-Gilbert garantit l'existence de codes de rendement asymptotiquement non nul pour $\delta < 1/3$ (voir Figure 5). Par ailleurs, la borne de Hamming montre que le rendement tend vers zéro lorsque $\delta \geq 2/3$. Ceci peut être immédiatement amélioré ($1/2$ au lieu de $2/3$) en observant qu'un poids arithmétique peut valoir au plus $(n+1)/2$.

Une autre remarque simple est que le poids arithmétique d'un entier x est toujours inférieur ou égal au poids de Hamming de sa représentation binaire. On peut donc toujours appliquer une borne supérieure connue pour les codes en blocs dans l'espace de Hamming, en particulier la borne de McEliece-Rodemich-Rumsey-Welch. Nous avons cependant obtenu une borne supérieure qui bat cette application de la borne de McEliece-Rodemich-Rumsey-Welch aux codes arithmétiques.

•▷ Notre borne s'écrit asymptotiquement :

$$R_a \lesssim (1-\rho)\Big(1 - H_2\Big(\frac{\rho}{1-\rho}\Big)\Big),$$

$$\text{où } \rho = \frac{2}{3} - \sqrt{\frac{4}{9} - \frac{2}{3}\delta},\ \delta = d/n,\ \text{et } n \text{ tend vers l'infini,}$$

voir Figure 5. La technique employée est la suivante : on exploite le lemme de Bassalygo-Elias majorant la meilleure densité d'un code à l'aide de la meilleure densité dans un sous-espace, en l'occurrence l'ensemble des entiers d'un poids arithmétique donné w. Comme on ne dispose pas d'un équivalent de la borne de Johnson, on se place en fait dans un sous-espace plus grand, celui des vecteurs ternaires ayant w pour poids de Hamming. On utilise ensuite la borne de Johnson pour les codes ternaires, et le fait que la distance arithmétique entre deux

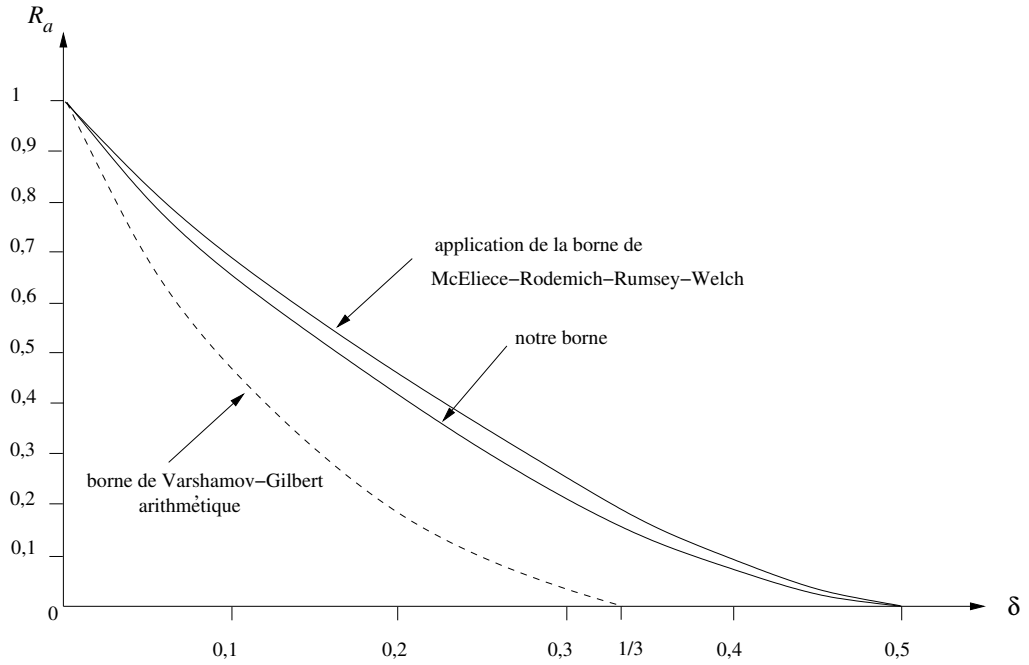

FIG. 5 – *Comportement asymptotique du rendement, R_a : borne inférieure, bornes supérieures.*

entiers est inférieure ou égale à la distance de Hamming entre les vecteurs ternaires de leur représentation modifiée non adjacente en base 2.

Dans le cas $r > 2$ cependant, on n'arrive pas à surpasser la simple application aux codes arithmétiques de bornes supérieures issues des codes en blocs.

1.3 Codes identifiants

Les codes identifiants, nouvellement apparus (Karpovsky, Chakrabarty, et Levitin [51], 1998), peuvent être vus comme une prolongation du thème des codes couvrants ; étant donnés un entier t et un graphe non orienté, connexe, fini ou infini, $G = (S, A)$, muni de la distance d du plus court chemin, on définit $B_t(u)$, la sphère de centre $u \in S$ et de rayon t, de la même manière que dans l'espace de Hamming, qui, en termes de graphes, n'est autre que le n-cube :

$$B_t(u) = \{v \in S : d(u, v) \leq t\}.$$

De même, on dit que le sommet u t-couvre (ou couvre s'il n'y a pas d'ambiguïté) tous les éléments de $B_t(u)$. On traite le plus souvent des graphes où le volume des sphères de rayon t est indépendant de leur centre, et dans ce cas on note $V(t)$ ce volume.

Toujours par analogie avec l'espace de Hamming, on appellera codes certains sous-ensembles particuliers de S, et mots de code leurs éléments ; un code t-couvrant ou couvrant $C \subseteq S$ est tel que les ensembles $B_t(s) \cap C$, $s \in S$, soient tous non vides. On dit que C est *t-identifiant* ou *identifiant* si de plus ces ensembles $B_t(s) \cap C$ sont tous distincts. L'ensemble des mots de code couvrant un sommet $s \in S$ est appelé l'*ensemble identifiant* de s.

On cherche alors à déterminer la plus petite densité, $D(G,t)$, d'un code t-identifiant dans G. Les graphes considérés peuvent être le n-cube, ou les réseaux rectangulaires ou carrés, triangulaires, ou hexagonaux, qui peuvent être appliqués, par exemple, à des réseaux de processeurs où l'on cherche à identifier un processeur défectueux s_0 (= un sommet quelconque du graphe) à l'aide de $|C|$ bits d'information fournis par $|C|$ processeurs sélectionnés (= mots de code) : les processeurs-mots de code signalent à un contrôleur, par exemple par un '1', s'il y a un processeur défectueux dans leur sphère de rayon t. Le contrôleur peut alors identifier le sommet s_0.

En empruntant des techniques simples aux codes couvrants dans l'espace de Hamming, on peut établir quelques bornes inférieures assez générales — toujours en supposant que les sphères de rayon t ont toutes le même volume, $V(t)$; si C est identifiant, il est aussi couvrant, et donc la deuxième des inégalités (1.1), la "sphere-covering bound", s'applique :

$$\frac{|C|}{|S|} \geq \frac{1}{V(t)}.$$

La propriété d'identification permet d'améliorer cette première borne : soit L_1 l'ensemble des sommets de S qui sont identifiés par un singleton de C ; alors $|S| - |L_1|$ sommets ont des ensembles identifiants de taille au moins deux. En utilisant le fait que $|C| \geq |L_1|$, on a : $|C| \cdot V(t) \geq 2(|S| - |L_1|) + |L_1| = 2|S| - |L_1| \geq 2|S| - |C|$. Donc :

$$\frac{|C|}{|S|} \geq \frac{2}{V(t)+1}, \tag{1.6}$$

et

$$D(G,t) \geq \frac{2}{V(t)+1}. \tag{1.7}$$

Si (1.6) est une égalité, on dira que C est un code *parfait*. Par exemple, dans un graphe G consistant en un cycle à six sommets, prendre trois sommets non adjacents deux à deux donne un code parfait.

●▷ Nous avons montré [26] qu'il n'existe aucun code parfait non trivial pour $t > 1$.

On peut améliorer (1.6) ou (1.7) de deux manières : des méthodes générales, valables pour tout t — nous en détaillerons un exemple en Section 1.3.2 —, ou des méthodes *ad hoc*, pour t petit (en fait, $t = 1$) et pour un graphe donné. Plus spécifiques et techniques, elles nous paraissent d'un intérêt moindre pour le lecteur de ce document, et nous nous contenterons d'en exposer les résultats. Le lecteur voulant en savoir plus est renvoyé à nos articles.

Les bornes supérieures sur $D(G,t)$ sont obtenues par construction ; là encore, ces constructions peuvent être générales — voir un exemple d'une telle construction en Section 1.3.3 —, ou particulières, pour de petites valeurs de t (voir Figure 7). Elles sont, dans ce dernier cas, obtenues soit « à la main », comme celle de la Figure 7, soit à l'aide d'heuristiques d'optimisation combinatoire (voir page 52).

Avant de passer à l'étude de quatre graphes particuliers, mentionnons déjà que le problème de décision correspondant à la recherche d'un code t-identifiant, de taille majorée, dans un graphe quelconque, est NP-complet pour tout t, et le reste si l'on se restreint aux graphes bipartis (voir Section 4.3 pour plus de développpements concernant ce résultat).

Remarquons que les quatre graphes que nous allons maintenant étudier sont *infinis*. Les constructions de codes identifiants que nous donnerons seront *périodiques* et pourront être décrites de manière simple.

1.3.1 La grille carrée

La grille carrée infinie, deux-dimensionnelle, G_C, a pour ensemble de sommets $S = Z \times Z$ et pour ensemble d'arêtes

$$A_C = \{\{u,v\} : u - v \in \{(0,1),(1,0)\}\}$$

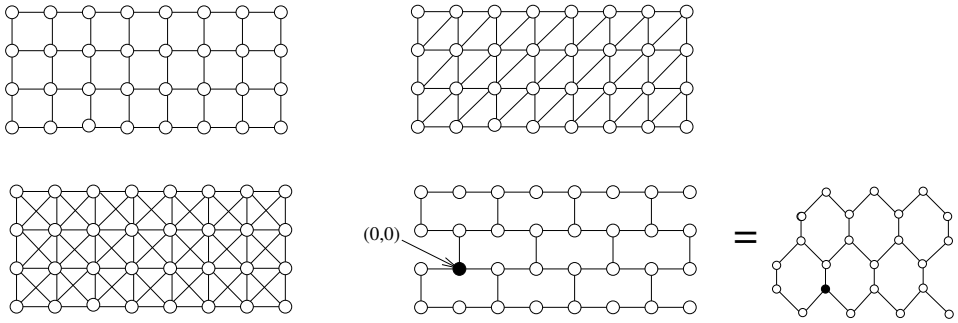

FIG. 6 – *Fragments de nos quatre graphes infinis, deux-dimensionnels.*

(voir Figure 6).

●▷ Nous avons établi les bornes suivantes [24], [22], [26], [47], [16], [12] :

$$\begin{gathered} 15/43 \leq D(G_C, 1) \leq 0,35\ ; \\ D(G_C, 2) \leq 5/29\ ; \\ D(G_C, t) \geq \frac{3}{8t+4}\ ; \\ D(G_C, t) \leq \frac{2}{5t},\ \text{pour } t \text{ pair}\ ; \\ D(G_C, t) \leq \frac{2t}{5t^2-2t+1},\ \text{pour } t \text{ impair}. \end{gathered} \tag{1.8}$$

La construction périodique de la Figure 7 montre que $D(G_C, 1) \leq 0,35$. Nous conjecturons que $D(G_C, 1) = 0,35$.

Une heuristique nous a fourni, pour t compris entre 3 et 6, des codes t-identifiants.

Notons enfin que la borne inférieure (1.8) est en $1/t$, alors que (1.7) donne une borne en $1/t^2$, puisque la sphère de rayon t est un polynôme du deuxième degré en t. Dans les trois autres graphes, nous établirons également des bornes en $1/t$, au lieu de $1/t^2$.

1.3.2 La grille triangulaire

La grille triangulaire infinie, deux-dimensionnelle, G_T, a pour ensemble de sommets $S = Z \times Z$ et pour ensemble d'arêtes

$$A_T = \{\{u, v\} : u - v \in \{(0,1), (1,0), (1,1)\}\}$$

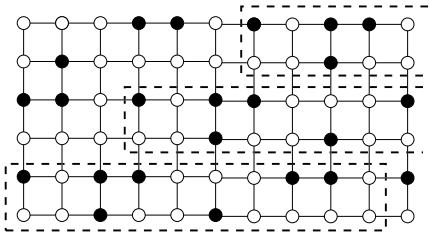

FIG. 7 – *Un code 1-identifiant, périodique, de densité* $0,35$*, sur la grille carrée infinie. Les mots de code sont en noir.*

(voir Figure 6).

Dans le cas $t = 1$, (1.7) est une égalité : il existe un code parfait, de densité $0,25$ (Karpovsky, Chakrabarty, et Levitin [51]).

●▷ Nous avons établi les bornes suivantes [26], [16], [12] :

$$D(G_T, t) \geq \frac{2}{6t+3} ;$$

$$D(G_T, t) \leq \frac{1}{2t+4}, \text{ pour } t = 0 \bmod 4 ;$$

$$D(G_T, t) \leq \frac{1}{2t+2}, \text{ pour } t = 1, 2 \text{ ou } 3 \bmod 4.$$

Une heuristique nous a fourni, pour t compris entre 2 et 6, des codes t-identifiants.

Pour donner un aperçu des idées utilisées pour obtenir des bornes inférieures, je donne ici l'esquisse de la démonstration de l'inégalité $D(G_T, t) \geq \frac{2}{6t+3}$.

Nous appelons *triangle* tout triplet (x, y, z) tel qu'il existe $i \in Z$ et $j \in Z$, avec $x = (i, j)$, $y = (i, j+1)$ et $z = (i+1, j+1)$. Soit $H_t(x, y, z) = \Delta_t(x, y) \cup \Delta_t(x, z) \cup \Delta_t(z, y)$, où $\Delta_t(x, y)$ désigne la différence symétrique des sphères de rayon t centrées sur x et y (voir Figure 8).

Il est aisé de constater d'une part que $|H_t(x, y, z)| = 6t + 3$, et d'autre part que, si C est un code t-identifiant, alors $|H_t(x, y, z) \cap C| \geq 2$. Des arguments de translation et de pavage du plan infini, rappelant le lemme de Bassalygo-Elias (cf. page 27), permettent alors d'en déduire, après quelques calculs, que la densité est au moins de $2/(6t + 3)$.

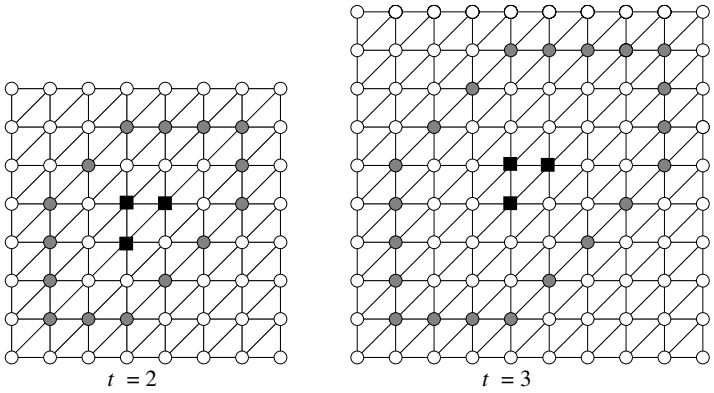

FIG. 8 – *En gris, les sommets appartenant à* $H_t(x, y, z)$.

1.3.3 La grille royale

La grille royale infinie, deux-dimensionnelle, G_R, a pour ensemble de sommets $S = Z \times Z$ et pour ensemble d'arêtes

$$A_R = \{\{u, v\} : u - v \in \{(0, 1), (1, 0), (1, 1), (1, -1)\}\}$$

(voir Figure 6). Elle est ainsi dénommée car sur un échiquier infini, la sphère de rayon t est l'ensemble des cases que le Roi peut atteindre en au plus t coups, en partant du centre de la sphère.

•▷ Nous avons pu établir la valeur *exacte* de $D(G_R, t)$ pour *toutes* les valeurs de t [27], [26], [16], [13] :

$$D(G_R, 1) = 2/9 \; ;$$

$$D(G_R, t) = \frac{1}{4t}, \text{ pour } t > 1.$$

La construction donnant la borne supérieure $1/4t$ pour tout t est facile à décrire :

$$C = \bigcup_{k \in Z} \{(2kt + \alpha, \alpha) : \alpha \in Z, \alpha \text{ pair}\}.$$

Il est moins facile de prouver que C est effectivement t-identifiant. Mais il est encore moins facile de prouver que $1/4t$ est aussi, pour $t > 1$, la borne inférieure sur $D(G_R, t)$!

1.3.4 La grille hexagonale

La grille hexagonale infinie, deux-dimensionnelle, G_H, a pour ensemble de sommets $S = Z \times Z$ et pour ensemble d'arêtes

$$A_H = \{\{u = (i,j), v\} : u - v \in \{(0,(-1)^{i+j+1}), (1,0)\}\}$$

(voir Figure 6).

●▷ Nous avons établi les bornes suivantes [33], [25], [26], [16], [12] :

$$16/39 \leq D(G_H, 1) \leq 3/7\,;$$

$$D(G_H, t) \geq \frac{2}{5t+3},\ \text{pour } t \text{ pair}\,;$$

$$D(G_H, t) \geq \frac{2}{5t+2},\ \text{pour } t \text{ impair}\,;$$

$$D(G_H, t) \leq \frac{8t-8}{9t^2-16t},\ \text{pour } t = 0 \bmod 4\,;$$

$$D(G_H, t) \leq \frac{8}{9t-25},\ \text{pour } t = 1 \bmod 4\,;$$

$$D(G_H, t) \leq \frac{8}{9t-34},\ \text{pour } t = 2 \bmod 4\,;$$

$$D(G_H, t) \leq \frac{8t-16}{(t-3)(9t-43)},\ \text{pour } t = 3 \bmod 4.$$

Une heuristique nous a fourni, pour t compris entre 2 et 8, des codes t-identifiants.

2 Éléments de complexité

Begin at the beginning, the King said gravely, and go on till you come to the end; then stop.

Lewis Carroll

Notre but principal est de donner ici une approche intuitive de la notion de *complétude* dans la *hiérarchie polynomiale*.

Nous ne traiterons que de *problèmes de décision*, constitués d'une entrée et d'une question à laquelle on ne peut répondre que OUI ou NON. Un algorithme A résout un problème π si, appliqué à une entrée I quelconque de π, il donne la réponse correcte à cette entrée. Une estimation de la *taille* d'une entrée I de π est donnée par tout encodage « raisonnable » de I (par exemple, un encodage raisonnable d'un entier m demande $\log m$ bits ; nous verrons toutefois plus loin (page 47), au sujet des codes linéaires, les résultats paradoxaux, au demeurant inévitables, que peut induire cette notion de taille). La fonction de *complexité en temps* d'un algorithme A résolvant π est, pour chaque taille d'entrée possible, le temps *maximal* nécessaire à A pour résoudre une entrée de cette taille. Un algorithme *polynomial* (en temps) est un algorithme dont la fonction de complexité (en temps) peut être bornée par un polynôme $p(n)$, où n est la taille de l'entrée considérée. La classe des problèmes que l'on peut résoudre à l'aide d'un algorithme polynomial est désignée par P.

Une *réduction polynomiale* d'un problème π_1 à un problème π_2 est une construction polynomiale qui transforme toute entrée de π_1 en une entrée équivalente de π_2 (par équivalente, nous entendons que les deux entrées ont la même réponse). Ainsi, une telle transformation permet de convertir un éventuel algorithme polynomial résolvant π_2 en un algorithme polynomial résolvant π_1.

Nous présentons maintenant la classe NP : un problème de décision appartient à NP s'il peut être résolu à l'aide d'un algorithme *non déterministe polynomial*, c'est-à-dire un algorithme constitué de deux étapes : une étape de *divination* et une étape polynomiale de *vérification*. La première étape procure une structure s. La seconde étape est déterministe et répond correctement OUI ou NON. Illustrons ces notions sur l'exemple célébrissime du problème du Voyageur de Commerce (VC), pour lequel l'entrée est constituée par un en-

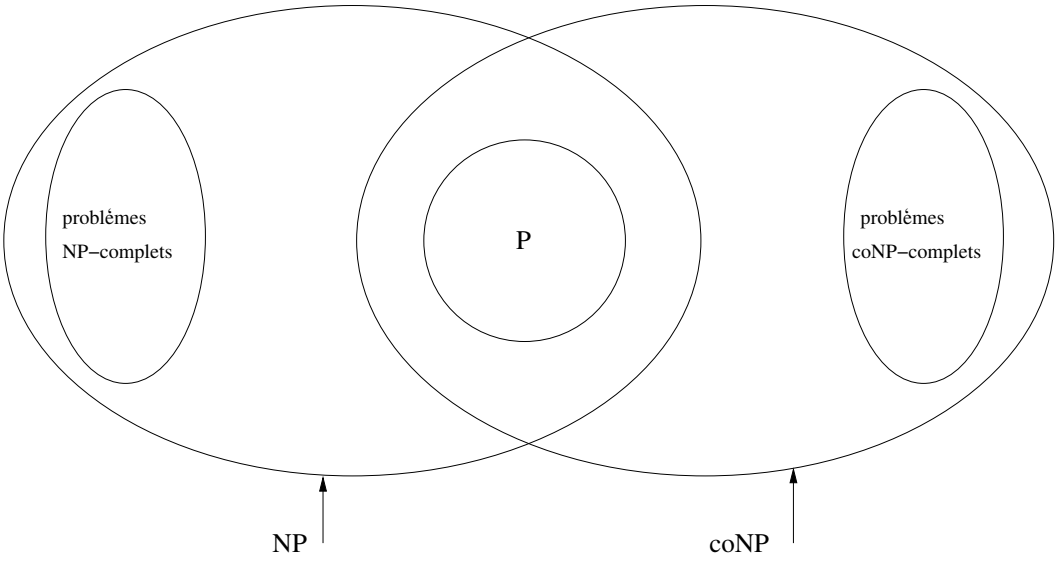

FIG. 9 – *Classes de complexité, si NP ≠ coNP.*

semble de villes, l'ensemble des distances (entières) inter-villes, et une borne supérieure B, et la question est de savoir s'il existe un cycle hamiltonien de longueur au plus B ; l'étape de divination fournit une suite s de villes et l'étape de vérification établit, en temps polynomial, si s est ou n'est pas un cycle hamiltonien de longueur au plus B.

Soit S un ensemble de problèmes ; on désigne par coS l'ensemble des problèmes qui sont complémentaires de ceux de S (leurs réponses sont inversées). On sait que P = coP ⊆ NP ∩ coNP, mais il ne semble pas que l'appartenance à NP implique l'appartenance à coNP (voir Figure 9). Par exemple, le problème complémentaire de VC consiste à déterminer si *tous* les cycles hamiltoniens ont une longueur au moins $B + 1$, et on ignore comment vérifier une réponse affirmative à cette question sans examiner au moins une très grande proportion de tous les cycles hamiltoniens possibles, chose que l'on ne sait pas faire en temps polynomial.

Parmi les problèmes appartenant à NP, certains ont la propriété suivante : tous les problèmes de NP peuvent leur être polynomialement réduits. Nous notons NP-C cette classe particulière de problèmes non déterministes polynomiaux, et qualifions de NP-*complets* ses membres. Si un problème de NP-C pouvait être résolu en temps polynomial, alors tout problème de NP le pourrait également, et P serait égal à NP. La question ouverte « P=NP ? »

est un des plus grands défis de la théorie de la complexité.

Les problèmes NP-complets peuvent donc être vus comme les plus difficiles à l'intérieur de la classe NP. Par exemple VC est NP-complet (Karp [50]), ainsi que 3-satisfiabilité (3-SAT) (Cook [35]), dont l'entrée est constituée par un ensemble de variables et un ensemble de clauses comportant exactement trois littérales distinctes (une littérale est soit une variable x_i soit une variable barrée $\overline{x}_j$), et la question consiste à savoir s'il existe une affectation des variables telle que chaque clause contienne au moins une littérale mise à Vrai (en d'autres termes, la formule booléenne E peut-elle être satisfaite, si $E = \mathcal{C}_1 \wedge \mathcal{C}_2 \wedge \ldots \wedge \mathcal{C}_m$, chaque clause $\mathcal{C}_i = x_{i_1} \vee x_{i_2} \vee x_{i_3}$ pour $i = 1, 2, \ldots, m$, et x_{i_1}, x_{i_2}, et x_{i_3} sont trois littérales distinctes ? L'expression E ci-dessus est dite *sous forme normale conjonctive*).

Certains problèmes pourraient être plus difficiles que les problèmes NP-complets et des classes de problèmes dont la complexité est (apparemment !) croissante peuvent être définies, formant ce que l'on appelle la *hiérarchie polynomiale*. La notion de complétude peut alors s'étendre à l'intérieur de ces classes : un problème π appartenant à un ensemble S de la hiérarchie polynomiale est dit S-*complet* si tout problème de S peut lui être polynomialement réduit.

En particulier, la hiérarchie polynomiale contient des classes notées Π_0, Π_1, ..., Π_k, ..., et Σ_0, Σ_1, ..., Σ_k, ..., possédant les propriétés suivantes : $\Pi_0 = \Sigma_0 = \mathrm{P}$, $\Sigma_1 = \mathrm{NP}$, $\Pi_1 = \mathrm{coNP}$, $\Pi_k = \mathrm{co}\Sigma_k$, $\Sigma_k \cup \Pi_k \subseteq \Sigma_{k+1} \cap \Pi_{k+1}$ (voir Figure 10).

On peut dire en gros qu'un problème appartient à Σ_k s'il peut être résolu à l'aide d'un algorithme non déterministe polynomial ayant accès à un *oracle* (c'est-à-dire un sous-programme) qui fournit, *en un pas de calcul*, une solution à un problème appartenant à Σ_{k-1}. Une autre caractérisation informelle de Σ_k est obtenue en représentant l'entrée d'un problème π par une chaîne z ; alors $\pi \in \Sigma_k$ si et seulement si $\pi = \{z : \exists y_1 \forall y_2 \ldots Q y_k R(z, y_1, y_2, \ldots, y_k)\}$, où les quantificateurs alternent, Q est égal à $\forall$ (respectivement, $\exists$) si k est pair (respectivement, impair), R est une relation reconnaissable en temps polynomial, et les longueurs des chaînes $y_1, y_2, \ldots, y_k$ sont polynomialement bornées par la longueur de la chaîne z. La même caractérisation vaut pour Π_k, avec l'alternance $\forall\exists\forall\ldots$ pour les quantificateurs. Le problème suivant appartient à Π_k ; il est même Π_k-complet (Meyer et Stockmeyer [67]) :

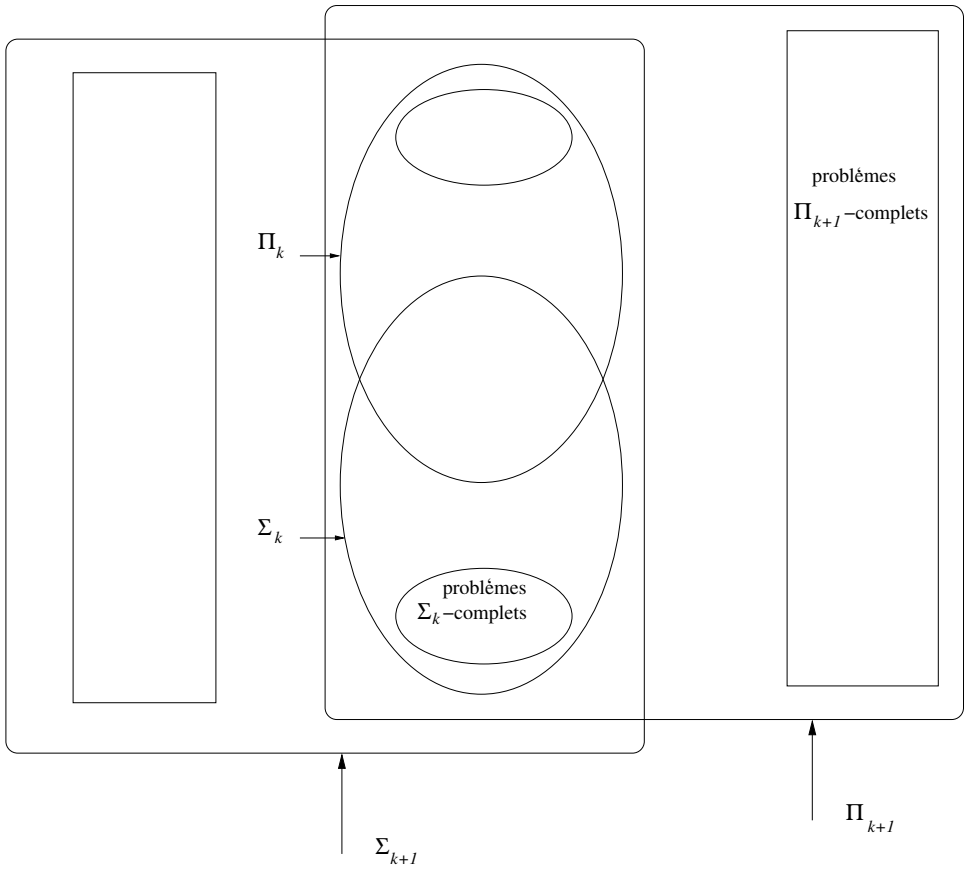

FIG. 10 – *Classes de complexité, si* $\Sigma_{k+1} \neq \Pi_{k+1}$, $k \geq 1$.

Nom : $\forall_1\exists_2\forall_3 \ldots Q_k$-3-satisfiabilité, où les quantificateurs alternent et Q est égal à $\forall$ (respectivement, $\exists$) si k est impair (respectivement, pair).

Entrée : k entiers $m_1, m_2, \ldots, m_k$, et une expression booléenne quantifiée $\forall u_{1,1} \ldots \forall u_{1,m_1} \exists u_{2,1} \ldots \exists u_{2,m_2} \forall u_{3,1} \ldots \forall u_{3,m_3} \ldots Q u_{k,1} \ldots Q u_{k,m_k} E$, où E est sous forme normale conjonctive, chaque clause contient exactement trois littérales distinctes, et les variables quantifiées sont toutes les variables de E.

Question : Est-il vrai que pour pour toute affectation des variables $u_{1,1}, \ldots, u_{1,m_1}$, il existe une affectation des variables $u_{2,1}, \ldots, u_{2,m_2}$, telle que pour pour toute affectation des variables $u_{3,1}, \ldots, u_{3,m_3}, \ldots,$ E soit satisfaite ?

Afin de montrer qu'un problème π est S-complet, on vérifie qu'il appartient à S, puis que chaque problème de S peut lui être polynomialement réduit. Pour cette seconde étape, il suffit de montrer qu'un problème π_0 qu'on sait être S-complet est polynomialement réductible à π, car tous les problèmes de S sont polynomialement réductibles à π_0 et le processus de réduction est transitif.

Les résultats de complétude sont des résultats *conditionnels* ; par exemple, établir la NP-complétude d'un problème π signifie qu'un algorithme polynomial résolvant π n'existe que si P=NP. De manière analogue, pour tout entier $k \geq 1$, la Σ_k-complétude de π implique $\pi \in \Sigma_k \setminus \Sigma_{k-1}$, à moins que $\Sigma_k = \Sigma_{k-1}$. On ignore si la hiérarchie polynomiale est finie ou infinie. La première alternative se produit si P=NP ; elle se produit également s'il existe un entier $k_0 \geq 1$ tel que $\Sigma_{k_0} = \Pi_{k_0}$, car on peut alors prouver que pour tout $k \geq k_0$ on aurait : $\Sigma_k = \Pi_k = \Pi_{k_0}$.

On pense généralement que P$\neq$NP, c'est-à-dire qu'il n'existe pas d'algorithme polynomial pour les problèmes NP-complets.

En présence d'un problème NP-complet (ou plus élevé dans la hiérarchie polynomiale), on peut appliquer des heuristiques telles que le recuit simulé, les algorithmes génétiques, la méthode tabou, ou le bruitage (voir Sections 4.1, page 48, et 4.3, page 52).

•▷ Sur la théorie de la complexité et ses outils formels (problèmes et langages, encodages raisonnables, taille d'un problème, classes de problèmes, complétude, réductions, machines

de Turing, déterministes et non déterministes, certificat succinct, la classe NP, problèmes fortement NP-complets, problèmes pseudo-polynomiaux, oracle, hiérarchie polynomiale, ...) — dont certains ont été décrits ci-dessus — ainsi que ses applications aux codes en blocs, à la cryptographie, et à la quantification vectorielle — voir aussi Sections 3 et 4 —, nous avons rédigé, Jean-Pierre Barthélemy, Gérard Cohen, et moi-même, un ouvrage intitulé « Complexité algorithmique et problèmes de communications » [6], paru en 1992. Il compte xxxviii+228 pages, six chapitres (1. Problèmes et langages 2. Machines, langages et problèmes, les classes P et NP 3. Problèmes et langages NP-difficiles 4. Complexité et codage 5. Complexité et cryptologie 6. Quantification vectorielle) et a fait en 1993 l'objet d'une courte recension par le Professeur Cristian Calude (Université d'Auckland, Nouvelle-Zélande) dans Mathematical Reviews :

> The book represents a clear, synthetical and deep presentation of the problem P =? NP. It contains six chapters (Problems and languages, Classes P and NP, NP-hard problems, Complexity and coding, Complexity and cryptology, Vector optimization). It is a serious, updated rival of the famous Garey-Johnson 1979 book. As in most cases in the history of mathematics, the challenging open problem P =? NP generates many other problems, often interesting in themselves.

Cet ouvrage a été traduit en anglais (parution en 1996 [7]).

3 Éléments de cryptographie

No puedo combinar unos caracteres dhcmrlchtdj *que la divina Biblioteca no haya previsto y que en alguna de sus lenguas secretas no encierren un terrible sentido.* *Jorge Luis Borges*

Nous allons ici nous contenter de décrire trois cryptosystèmes dits *à clé publique*, car ce sont ces trois systèmes qui seront concernés au premier chef dans la suite de ce document. Cette description est volontairement très succincte.

La cryptographie à clé publique est née en 1976 (Diffie et Hellman [37]). Elle ne nécessite pas d'échange de clé sur un canal sûr, mais fait apparaître le besoin de disposer de *fonctions à sens unique avec trappe*, c'est-à-dire de fonctions faciles à calculer et difficiles à inverser, *sauf si l'on dispose d'une information supplémentaire*, appelée la trappe du système. De telles fonctions ne sont apparues qu'en 1978, et nous allons maintenant en présenter trois, le RSA, le sac à dos et le McEliece.

3.1 Le cryptosystème RSA

Le nom de ce cryptosystème vient des noms de ses concepteurs, Rivest, Shamir, et Adleman [72]. Un utilisateur B désirant recevoir des messages chiffrés choisit deux grands nombres premiers p et q et forme leur produit $n = pq$. L'ensemble des messages en clair $\mathcal{M}$ et des messages chiffrés $\mathcal{C}$ est l'ensemble des entiers de 0 à $n-1$. Connaissant p et q, il est élémentaire de trouver deux entiers positifs e et d tels que pour tout entier $M \in \mathcal{M}$,

$$M^{de} = M^{ed} = M \bmod n.$$

Il suffit en effet de choisir e entre 2 et n et premier avec $(p-1)(q-1)$, puis de calculer d (à l'aide de l'algorithme d'Euclide) de manière à vérifier $ed = 1 \bmod (p-1)(q-1)$. Inverser une exponentiation modulaire, à gauche comme à droite, revient donc à calculer une autre exponentiation modulaire, les exposants étant associés par une relation de type $cc' = 1 \bmod (p-1)(q-1)$.

Sont ensuite rendus publics e et n, alors que sont gardés secrets d, p, et q, qui constituent la trappe du système.

Toute personne A ayant connaissance de e et n peut envoyer à B un message $M \in \mathcal{M}$, dont la confidentialité sera préservée, de la manière suivante : A calcule le message chiffré $C \in \mathcal{C}$, $C = M^e \bmod n$. Le destinataire, muni de la clé secrète d, calcule $C^d = (M^e)^d = M^{ed} = M \bmod n$.

Le cryptanalyste interceptant le message chiffré C est placé devant le problème suivant : il sait que $C = M^e \bmod n$ où C, e, et n lui sont connus mais M inconnu ; il sait également que $M = C^d \bmod n$ où C et n lui sont connus mais M et d inconnus. Les deux problèmes, retrouver M à partir de $M^e \bmod n$ (extraction de racine e-ième modulo n) sans factoriser n, ou retrouver d sans factoriser n, n'ont encore connu aucune approche satisfaisante depuis la création du RSA, et la sécurité de ce cryptosystème repose apparemment sur la difficulté, reconnue, de la factorisation.

La mise en œuvre pratique du RSA requiert une procédure rapide pour le calcul d'une exponentiation modulaire, puisque c'est elle que l'on utilise dans les deux opérations, chiffrement et déchiffrement. Voir Section 5 pour une méthode permettant d'accélérer les exponentiations modulaires.

3.2 Le cryptosystème du sac à dos

Il existe plusieurs cryptosystèmes utilisant le problème de décision suivant :

Nom : SAC A DOS.

Entrée : n entiers strictement positifs $a_1, a_2, \ldots, a_n$, un entier strictement positif S.

Question : Existe-t-il n nombres x_i ($x_i = 0$ ou 1) tels que $\sum_{1 \leq i \leq n} a_i x_i = S$?

Ce problème est NP-complet (Karp [50]). Sa difficulté présumée a été utilisée dans plusieurs cryptosystèmes ; nous décrivons ici la première tentative, due à Merkle et Hellman [66]. Un utilisateur B désirant recevoir des messages chiffrés choisit deux grands nombres m et w, premiers entre eux (de cette manière, il existe un entier w' tel que $ww' = 1 \bmod m$). Il choisit également une suite $\mathbf{a}$ (de grande longueur) d'entiers $a_1, a_2, \ldots, a_n$ vérifiant la propriété suivante : pour tout i entre 2 et n, $a_i > \sum_{1 \leq j \leq i-1} a_j$ (une telle suite d'entiers est dite *super-croissante*). Enfin il choisit une permutation σ sur $\{1, 2, \ldots, n\}$. Il peut alors « brouiller » le

sac à dos super-croissant en le transformant en un sac à dos $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n)$ défini par $a'_i = wa_{\sigma(i)} \bmod m$ pour $i = 1, 2, \ldots, n$.

Le sac à dos $\mathbf{a}'$ est ensuite rendu public, alors que sont gardés secrets le sac à dos $\mathbf{a}$, la permutation σ, et les entiers w et m, qui constituent la trappe du système. L'ensemble des messages en clair $\mathcal{M}$ est l'ensemble des vecteurs binaires de longueur n alors que l'ensemble des messages chiffrés $\mathcal{C}$ est l'ensemble des entiers de 0 à $\sum_{1\leq i\leq n} a'_i$ (ou, de manière équivalente, l'ensemble des vecteurs binaires de longueur $\lfloor \log_2(\sum_{1\leq i\leq n} a'_i) \rfloor + 1$).

Toute personne A ayant connaissance de $\mathbf{a}'$ peut envoyer à B un message $\mathbf{M} = (M_1, M_2, \ldots, M_n) \in \mathcal{M}$, dont la confidentialité sera préservée, de la manière suivante : A calcule le message chiffré $C \in \mathcal{C}$, $C = \sum_{1\leq i\leq n} M_i a'_i$. Le destinataire, muni de la clé secrète $\mathbf{a}$, σ, w, et m, calcule $Cw' = \sum_{1\leq i\leq n} M_i a'_i w' = \sum_{1\leq i\leq n} M_i a_{\sigma(i)} \bmod m$. Si m a été choisi de manière à être plus grand que $\sum_{1\leq i\leq n} a_i$, alors $Cw' = \sum_{1\leq i\leq n} M_i a_{\sigma(i)}$, et il est élémentaire pour B de retrouver le vecteur $\mathbf{M}$, en utilisant la propriété de super-croissance de la suite initiale.

Le cryptanalyste interceptant le message chiffré C doit résoudre une entrée, apparemment quelconque, d'un problème NP-complet.

3.3 Le cryptosystème de McEliece

Ce système a été imaginé par McEliece en 1978 [64]. Il utilise le problème de décision suivant, et illustre de manière immédiate les liens pouvant exister entre codage et cryptographie :

Nom : Décodage Linéaire (DL).
Entrée : Une matrice binaire $\mathbf{H}$, un vecteur binaire $\mathbf{y}$, un entier w.
Question : Existe-t-il un vecteur binaire $\mathbf{x}$, de poids au plus w, tel que $\mathbf{xH}^T = \mathbf{y}$?

Ce problème est NP-complet (Berlekamp, McEliece, et van Tilborg [8] — cf. Section 4.1). Dans ce qui suit, tous les vecteurs et matrices sont binaires. Soient $n = 2^m$ et t un entier positif. Un utilisateur B désirant recevoir des messages chiffrés construit une matrice génératrice $\mathbf{G}$, de dimensions $k \times n$, d'un code de Goppa binaire, C_{Goppa}, de paramètres $[n, k \geq n - mt, d \geq 2t + 1]$. Il choisit ensuite une première matrice « brouilleuse » $\mathbf{S}$, qui est une matrice non singulière de dimensions $k \times k$: faire le produit $\mathbf{SG}$ revient à effectuer des

combinaisons linéaires de lignes de $\mathbf{G}$. Il construit une deuxième matrice brouilleuse $\mathbf{P}$, qui est une matrice de permutation de dimensions $n \times n$ (c'est-à-dire une matrice obtenue à partir de la matrice identité $n \times n$ par permutation de ses lignes) : faire le produit $\mathbf{G}' = (\mathbf{SG})\mathbf{P}$ revient à permuter des colonnes de $\mathbf{SG}$.

Sont ensuite rendus publics la matrice $\mathbf{G}'$ et l'entier t, alors que sont gardées secrètes les matrices $\mathbf{G}$, $\mathbf{S}$, et $\mathbf{P}$, qui constituent la trappe du système. L'ensemble des messages en clair $\mathcal{M}$ est l'ensemble des vecteurs binaires de longueur k, alors que l'ensemble des messages chiffrés $\mathcal{C}$ est l'ensemble des vecteurs binaires de longueur n.

Toute personne A ayant connaissance de $\mathbf{G}'$ et t peut envoyer à B un message $\mathbf{M} \in \mathcal{M}$, dont la confidentialité sera préservée, de la manière suivante : A calcule le message chiffré $\mathbf{C} \in \mathcal{C}$, $\mathbf{C} = \mathbf{MG}' + \mathbf{E}$, où $\mathbf{E}$ est un vecteur aléatoire de longueur n et de poids t (notons que le vecteur $\mathbf{E}$, qui ne fait pas partie de la trappe, doit être gardé secret par l'expéditeur). Le destinataire, muni de la clé secrète $\mathbf{G}$, $\mathbf{S}$, et $\mathbf{P}$, calcule $\mathbf{CP}^{-1} = \mathbf{MG}'\mathbf{P}^{-1} + \mathbf{EP}^{-1} = (\mathbf{MS})\mathbf{G} + \mathbf{EP}^{-1}$, où, grâce au fait que $\mathbf{P}$ est une matrice de permutation, $\mathbf{EP}^{-1}$ a le même poids que $\mathbf{E}$. Un algorithme rapide (linéaire en n) de décodage de C_{Goppa} peut alors être appliqué au vecteur $\mathbf{CP}^{-1}$ pour obtenir le vecteur $\mathbf{MS}$, le poids de $\mathbf{EP}^{-1}$ se situant en-deçà de la capacité de correction. Il ne reste plus qu'à multiplier à droite par $\mathbf{S}^{-1}$ pour retrouver le message $\mathbf{M}$.

Le cryptanalyste interceptant le message chiffré $\mathbf{C}$ doit résoudre une entrée, apparemment quelconque, d'un problème NP-complet.

Notons que si le vecteur $\mathbf{E}$ est pris de poids inférieur à t, on peut utiliser le système de McEliece en combinant correction d'erreurs et confidentialité : lors de la transmission de $\mathbf{C}$, un vecteur d'erreur $\mathbf{E}'$ dû au canal peut venir s'ajouter à $\mathbf{E}$; tant que le poids de $\mathbf{E} + \mathbf{E}'$ ne dépasse pas t, on retrouve le vecteur $\mathbf{MS}$ lors du décodage.

4 Liens entre codage et complexité

4.1 Liens entre complexité et codes en blocs

La complexité de plusieurs problèmes de décision associés à des problèmes importants pour les codes correcteurs ou pour les codes couvrants a pu être déterminée. Etudions-en deux, concernant les codes correcteurs :

Nom : Décodage Linéaire (DL).
Entrée : Une matrice binaire $\mathbf{H}$, un vecteur binaire $\mathbf{y}$, un entier w.
Question : Existe-t-il un vecteur binaire $\mathbf{x}$, de poids au plus w, tel que $\mathbf{xH}^T = \mathbf{y}$?

On a vu (page 10) que ce problème de décision est associé au décodage d'un code binaire linéaire C, donné par sa matrice de parité $\mathbf{H}$. On peut objecter qu'une matrice de parité n'est pas quelconque : elle a plus de colonnes que de lignes, elle est de rang maximal, ... Ces restrictions ne simplifient en fait pas le problème général.

Nom : Poids Minimal dans un code binaire linéaire (PM).
Entrée : Une matrice binaire $\mathbf{H}$, un entier w.
Question : Existe-t-il un vecteur binaire non nul $\mathbf{x}$, de poids au plus w, tel que $\mathbf{xH}^T = \mathbf{0}$?

Il est facile (cf. page 9) de constater que ce problème de décision revient à donner une borne supérieure sur la distance minimale d'un code linéaire.

Le premier problème, DL, est NP-complet (Berlekamp, McEliece, et van Tilborg [8]). La question de savoir si son énoncé est le mieux adapté pour traduire le problème du décodage linéaire a été posée par Bruck et Naor [11]. En effet, on peut considérer que le code (ou sa matrice de parité), une fois choisi, n'est plus modifié, et que seuls changent les vecteurs successivement reçus par le décodeur : il est possible d'appliquer un *prétraitement* à la matrice $\mathbf{H}$, qui pourrait permettre ensuite de traiter de manière efficace (en temps polynomial) les vecteurs au moment de leur réception. On peut donc voir l'entrée du problème DL comme étant formée de deux parties, la matrice $\mathbf{H}$ constituant une partie « fixe », et le syndrome du vecteur reçu une partie « mobile ». Le problème avec prétraitement (DLAP) s'énonce ainsi en retirant la matrice $\mathbf{H}$ de l'entrée. On obtient alors un résultat de complexité moins

fort que la NP-complétude, mais pouvant tout de même entraîner l'effondrement rapide de la hiérarchie polynomiale : l'existence d'un algorithme polynomial pour DLAP impliquerait que $\Pi_2 = \Sigma_2$ ($= \Pi_k = \Sigma_k$ pour tout $k \geq 2$) (Bruck et Naor [11]).

•▷ Nous avons donné une nouvelle démonstration, plus directe, et s'appliquant aussi immédiatement à tout autre alphabet que l'alphabet binaire, de ce résultat [56], [57].

•▷ Dans la même veine, considérons le problème SAC A DOS énoncé en Section 3.2. Son utilisation en cryptographie suppose que les n entiers a_1, ..., a_n, liés à la clé, ne sont pas modifiés pendant un certain temps, durant lequel seuls varient les messages, c'est-à-dire l'entier S. On peut alors voir l'entrée du problème SAC A DOS comme étant formée de deux parties, les entiers a_i formant une partie « fixe », et l'entier S une partie « mobile ». Le problème avec prétraitement s'obtient donc en retirant les a_i de l'entrée. Nous avons montré [57] que l'existence d'un algorithme polynomial pour ce problème impliquerait que, comme pour DLAP, $\Pi_2 = \Sigma_2$.

Remarquons que ces deux problèmes (DL et SAC A DOS) sont à la base de deux des trois cryptosystèmes à clé publique décrits en Section 3, mais que les deux résultats ci-dessus, concernant la complexité de leur résolution avec prétraitement, ne constituent pas une garantie supplémentaire de sécurité : dans leurs applications cryptographiques, on est en réalité en présence d'entrées particulières, polynomiales, mais qui ont été brouillées.

Le deuxième problème, PM, est NP-complet (Vardy [73], 1997, dix-neuf ans après sa conjecture dans [8]). Avant que ce résultat ne soit enfin obtenu, la NP-complétude de nombreuses variations autour de ce thème avait été établie : Poids exact [8], Poids moyen (Diaconis et Graham [36]), Poids incongru, Poids maximal, et Poids encadré (Ntafos et Hakimi [68]), où la question est de savoir s'il existe un mot de code de poids égal à w, égal à $\lfloor n/2 \rfloor$ (n étant la longueur du code), au plus w et non multiple d'un entier donné, plus grand que w, et compris entre deux entiers donnés, respectivement.

•▷ Quant à nous, nous avions montré [60], entre autres, que déterminer s'il existe un mot de code de poids au plus w dont les $\lfloor w\frac{p}{p+1} \rfloor$ premières composantes valent '1', est NP-complet pour p fixé, $p \geq 3$.

Ce résultat est presque optimal au sens suivant : si l'on remplace $wp/(p+1)$ par $w-\lambda$, où λ est une constante, alors le problème possède un algorithme polynomial. En effet, il suffit de compléter, de toutes les façons possibles, avec au plus λ '1', le vecteur de longueur n dont les $w-\lambda$ premières composantes valent '1', et de tester l'appartenance au code. Le nombre de tests requis vaut $\sum_{i=0}^{\lambda} \binom{n-(w-\lambda)}{i}$, qui est de l'ordre de n^{λ}, i.e., polynomial en n. Avec λ fraction de w au lieu de λ constante, c'est un problème NP-complet.

Dans le cas ternaire, notons qu'il a par exemple été établi (Barg [5]) que déterminer s'il existe un mot de code de poids égal à la longueur du code, est NP-complet.

Passons aux codes couvrants.

Nom : Rayon de Recouvrement d'un code binaire Linéaire (RRL).
Entrée : Une matrice binaire $\mathbf{H}$ (de dimensions $m \times n$), un entier w.
Question : Pour tout vecteur binaire $\mathbf{y}$ (de longueur m), existe-t-il un vecteur binaire $\mathbf{x}$ (de longueur n), de poids au plus w, tel que $\mathbf{x}\mathbf{H}^T = \mathbf{y}$?

Nom : Rayon de Recouvrement d'un code binaire (RR).
Entrée : Un code binaire C (de longueur n), un entier w.
Question : Pour tout vecteur binaire $\mathbf{y}$ (de longueur n), existe-t-il un mot de code $\mathbf{c}$ tel que $d(\mathbf{c},\mathbf{y}) \leq w$?

On a vu (page 9) comment le premier problème, RRL, répond au problème de borner supérieurement le rayon de recouvrement d'un code binaire linéaire ; il est Π_2-complet (McLoughlin [65]). Le même problème pour des codes non linéaires, RR, est coNP-complet (Frances et Litman [43]), c'est-à-dire plus bas dans la hiérarchie polynomiale, alors qu'il a pour sous-problème RRL. Ce résultat à première vue paradoxal s'explique par la représentation plus compacte d'un code linéaire : la taille d'un problème portant sur des codes linéaires de paramètres $[n,k]$, donnés par leur matrice génératrice ou de manière équivalente par leur matrice de parité, est en $n \cdot k = n \cdot \log_2 |C|$, alors que, s'agissant de codes non linéaires de paramètres (n,K), donnés de manière explicite mais non économique par la liste des mots de code, la taille du problème est en $n \cdot K = n \cdot |C|$.

•▷ Nous avons montré que le même résultat est valide pour le calcul de la norme minimale (définie en page 14) d'un code [46] : borner supérieurement la norme minimale d'un code binaire linéaire est Π_2-complet, le même problème pour des codes non linéaires est coNP-complet.

Ces résultats de complexité n'empêchent toutefois pas de rechercher des codes ayant de bons paramètres, au moins pour de petites longueurs, et l'utilisation d'heuristiques itératives telles que le bruitage ou le recuit simulé, pour ne citer qu'elles, ont permis la construction de nouveaux codes.

La méthode dite du bruitage (voir, entre autres, Charon et Hudry [14]), qui sera décrite plus en détail en Section 4.3, en liaison avec la construction de codes identifiants, a, par exemple, été appliquée avec succès à la construction de codes correcteurs sur F_4 (Bogdanova [10]), permettant ainsi l'amélioration de bornes inférieures sur la quantité $A(n, d)$, définie en page 12, pour des codes quaternaires de longueur allant jusqu'à 12.

•▷ Nous l'avons appliquée avec moins de réussite aux codes couvrants [15], dans l'espoir d'améliorer des bornes supérieures sur la quantité $K(n, 1)$ (cf. page 11), pour n compris entre 9 et 12. Nous avons seulement retrouvé les meilleures bornes supérieures connues. (Note ajoutée le 27 avril 2001 : en ce qui concerne la longueur 9, cela n'est pas étonnant. En effet, à l'époque, nous savions seulement que $57 \leq K(9, 1) \leq 62$; or il vient d'être montré que $K(9, 1) = 62$ [69].)

4.2 Liens entre complexité et codes arithmétiques

Comme nous l'avons dit plus haut (page 21) en présentant le poids modulaire de Clark-Liang, il nous semble que le problème de la complexité de son calcul n'a jamais été abordé. Dans les cas où les deux distances modulaires, de Rao-Garcia et de Clark-Liang, coïncident, le calcul peut se faire par comparaison de deux poids arithmétiques. Dans le cas général cependant, combien de poids arithmétiques doit-on calculer, et quelle est la taille des nombres qu'il faudra considérer ?

Rappelons que le problème est le suivant : étant donnés une base r, un modulo m, et un

entier I compris entre 0 et $m-1$, quel est le minimum, $W_{CL}(I)$, de l'ensemble $\{W(J) : J = I + km, k \in Z\}$, où $W(J)$ désigne le poids arithmétique de J, c'est-à-dire le nombre minimal de termes non nuls dans un représentation modifiée en base r de J ; nous rappelons également que la RMNA de J donne son poids arithmétique.

•▷ Nous avons effleuré ce sujet dans [61] : on sait que la distance modulaire D_{CL} est graphique (van Lint [52]) ; c'est la distance du plus court chemin dans le graphe de Cayley $G = (S, A)$ ayant pour ensemble de sommets $S = Z_m$ et pour générateurs $\{xr^i \bmod m : |x| < r, i = 0, 1, \ldots\}$. Les générateurs fournissent les entiers de poids modulaire 1, et permettent de déterminer les voisins de tout sommet de S. L'algorithme de parcours de graphe en largeur, à partir de 0, construit successivement les ensembles de sommets de poids modulaire 2, 3, ..., $W_{CL}(I)$. Sa complexité est majorée par

$$\sum_{s \in S} \deg(s) = 2|A| < m^2.$$

Une autre approche, statistique, pourrait être utilisée pour estimer le poids modulaire de Clark-Liang ; voir page 56.

4.3 Liens entre complexité et codes identifiants

Considérons, à t fixé, le problème de décision suivant :

Nom : Code t-identifiant.

Entrée : Un graphe connexe biparti $G = (S, A)$, un entier $k \leq |S|$.

Question : Existe-t-il un code t-identifiant $C \subseteq S$ de cardinal au plus k?

•▷ Nous avons démontré [26], [18] que ce problème est NP-complet. J'en donne ici la démonstration, assez simple, dans le cas $t = 1$. L'appartenance à NP étant facile à vérifier, il reste à faire une réduction polynomiale d'un problème NP-complet, en l'occurrence 3-SAT (cf. page 37) :

Nom : 3-satisfiabilité (3-SAT).

Entrée : Un ensemble ε de clauses sur un ensemble X de variables, chaque clause contenant

exactement trois littérales distinctes.

Question : Existe-t-il une une affectation des variables telle que chaque clause contienne au moins une littérale mise à Vrai ?

A partir de $\varepsilon = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$ et $X = \{x_1, x_2, \ldots, x_n\}$, nous construisons un graphe G biparti et un entier k tels que ε puisse être satisfaite si et seulement si G admet un code 1-identifiant de taille au plus k.

Pour chaque variable $x_i \in X$, on construit $G_{x_i} = (S_{x_i}, A_{x_i})$, où

$$S_{x_i} = \{a_i, b_i, x_i, \overline{x}_i, c_i, d_i\},$$

$$A_{x_i} = \{\{a_i, b_i\}, \{b_i, x_i\}, \{b_i, \overline{x}_i\}, \{x_i, c_i\}, \{\overline{x}_i, c_i\}, \{c_i, d_i\}\}.$$

Pour chaque clause $\mathcal{C}_j = \{u_{j,1}, u_{j,2}, u_{j,3}\}$, on construit le graphe $G_{\mathcal{C}_j} = (S_{\mathcal{C}_j}, A_{\mathcal{C}_j})$, qui contient deux sommets, α_j et β_j, et une arête, $\{\alpha_j, \beta_j\}$, et nous y ajoutons l'ensemble de trois arêtes $A'_{\mathcal{C}_j} = \{\{\alpha_j, u_{j,1}\}, \{\alpha_j, u_{j,2}\}, \{\alpha_j, u_{j,3}\}\}$.

Le graphe G a pour ensemble de sommets l'union des ensembles S_{x_i} et $S_{\mathcal{C}_j}$, et pour ensemble d'arêtes l'union des ensembles A_{x_i}, $A_{\mathcal{C}_j}$, et $A'_{\mathcal{C}_j}$ ($1 \leq i \leq n$, $1 \leq j \leq m$) ; il est biparti. Enfin, $k = 3n + m$.

Si ε peut être satisfaite, alors on peut construire un code 1-identifiant C, de taille égale à k, de la manière suivante : pour tout i entre 1 et n, b_i, c_i, et celle des deux littérales, x_i ou $\overline{x}_i$, qui est vraie, appartiennent à C ; pour tout j entre 1 et m, $\alpha_j \in C$.

Réciproquement, supposons que C soit un code 1-identifiant. Alors $|C \cap S_{\mathcal{C}_j}| = 1$ ou 2, et, de toutes façons, α_j est nécessairement couvert par un mot de code qui ne couvre pas β_j. Ensuite, il est facile de constater que $|C \cap S_{x_i}| \geq 3$, et que, si $|C \cap S_{x_i}| = 3$, alors exactement une des deux littérales, x_i ou $\overline{x}_i$, appartient à C. Par conséquent, $|C| \geq m + 3n = k$, donc $|C| = k$, donc $|C \cap S_{x_i}| = 3$, et mettre x_i à Vrai si $x_i \in C$, à Faux si $\overline{x}_i \in C$, est une affectation cohérente des variables de X. Comme α_j doit être distingué de β_j, il est couvert par un mot de code correspondant à une littérale de la clause $\mathcal{C}_j$, et ce pour tout j, ce qui montre que dans chaque clause il y a au moins une littérale vraie, et achève notre démonstration.

Une conséquence immédiate de ce résultat est que, si t n'est pas fixé mais fait partie de

l'entrée du problème, ce dernier est NP-complet (car il reste dans NP).

Nous avons dit, en page 28, que le problème des codes identifiants est récent. Cependant, une notion assez proche, celle des *ensembles localisateurs-dominateurs*, existe depuis plus longtemps (voir par exemple Colbourn, Slater, et Stewart [34]) ; la définition en est la suivante : un ensemble de sommets (nous préférons dire, en conformité avec tout ce qui précède, un code) est (t-)localisateur-dominateur si tous les sommets du graphe *qui ne sont pas mots de code* ont des ensembles identifiants non vides et distincts. La nuance (de taille) avec les codes identifiants est que les mots de code ont ici un statut spécial.

Il se trouve que nous n'avons appris l'existence de ce concept qu'après avoir commencé à développer nos recherches sur les codes identifiants dans une certaine direction, que les recherches menées par Slater et d'autres auteurs (d'ailleurs uniquement pour $t = 1$) étaient parties dans une autre direction que la nôtre, et que des méthodes communes n'ont donc guère émergé. Cependant, dans [34] il est établi que le problème de décision suivant est NP-complet :

Nom : Code 1-localisateur-dominateur.
Entrée : Un graphe connexe $G = (S, A)$, un entier $k \leq |S|$.
Question : Existe-t-il un code 1-localisateur-dominateur $C \subseteq S$ de cardinal au plus k ?

Cela nous a inspirés pour établir le résultat de NP-complétude concernant l'existence de codes t-identifiants de cardinal majoré.

•▷ Nous avons étendu [18] ce résultat de NP-complétude pour les codes 1-localisateurs-dominateurs à tout entier t, et ce pour les graphes bipartis.

Enfin, nous avons généralisé ces deux notions (code identifiant et code localisateur-dominateur) au cas des graphes orientés, pour montrer que là aussi, les problèmes de décision correspondants sont NP-complets, pour tout t fixé ou non fixé, et ce même lorsque l'on se restreint aux graphes bipartis [17].

D'un côté, nous avons donc établi des résultats de complexité théorique ; de l'autre, nous avons cherché à construire des codes identifiants aussi petits que possible, dans les quatre graphes décrits en Section 1.3.

•▷ Dans ce but, nous avons développé [16] des algorithmes de construction que nous allons maintenant décrire brièvement.

L'objectif étant de construire des codes de faible densité dans des graphes *infinis*, nous nous sommes bornés à rechercher des codes périodiques. Après avoir montré que, pour considérer *tous* les codes périodiques de $Z \times Z$, il suffit en fait de considérer des motifs rectangulaires à l'intérieur desquels on place les mots de code, nous avons procédé de la manière suivante :

on se fixe les entiers t, w, h, α ($0 \leq \alpha < w$), et c ($c \leq w \times h$), et on recherche un sous-ensemble C_R, de cardinal c, d'un rectangle R de largeur w et de hauteur h, tel que, en translatant R par les vecteurs $(w, 0)$ et (α, h), on obtienne un code C qui soit t-identifiant. Dans l'exemple de la Figure 7, les valeurs prises par w, h, α, et c sont 10, 2, 3, et 7, respectivement. On appelle *solution* tout sous-ensemble C_R de R de taille c, et on définit une *fonction-objectif* f pour chaque solution. Cette fonction, qui tient le compte des sommets n'étant t-couverts par aucun mot de code, et des paires de sommets qui sont t-couverts par les mêmes mots de code, doit valoir zéro pour que C_R induise un code t-identifiant.

On peut ensuite appliquer à ce modèle des méthodes itératives de descente, descente pure ou descente avec bruitage par exemple (cf. page 48). Décrivons ici le bruitage : on fixe arbitrairement c mots de code à l'intérieur de R. On considère à tour de rôle chacun des mots de code, et on calcule le déplacement qui minimise la valeur de la fonction-objectif f. A chaque fois, on déplace le mot de code soit sur l'emplacement permettant de minimiser f, soit au hasard, avec une probabilité de se trouver dans le second cas que l'on va faire diminuer progressivement, d'une valeur initiale (valant typiquement $0,2$ ou $0,3$) jusqu'à la mettre à 0. L'algorithme s'arrête lorsque $f = 0$ ou lorsqu'un certain nombre de déplacements (typiquement, 300 fois le nombre d'éléments de R) a été effectué.

En balayant les petites valeurs des paramètres w, h, α, c, on cherche à construire des codes dont la densité $\frac{c}{wh}$ améliore les meilleures valeurs connues. Voir [16] pour tous les résultats.

5 Liens entre codage et cryptographie

De nombreux liens unissent codage et cryptographie, dont le but commun est de protéger la transmission d'information, soit contre les erreurs de transmission, soit contre des attaques menaçant la confidentialité ou l'intégrité des données.

Un exemple très élégant reliant ces deux aspects de la sécurité est fourni par le système de McEliece (cf. Section 3.3), qui utilise des codes correcteurs, et peut sous certaines conditions se comporter en code correcteur en même temps qu'en cryptosystème (voir page 44). Sa sécurité repose sur la difficulté du décodage linéaire.

Il nous a été donné d'étudier une autre relation entre codage et cryptographie, existant entre les différentes représentations d'entiers utilisées dans les codes arithmétiques (voir Section 1.2.1) et le système RSA, qui a besoin de procédures rapides pour calculer des exponentiations modulaires (voir Section 3.1).

Rappelons que l'on désire calculer $M^e \bmod n$ ou $M^d \bmod n$, où n est le produit de deux grands nombres premiers p et q, et où e et d sont deux entiers fixés, reliés par l'égalité modulaire $ed = 1 \bmod (p-1)(q-1)$.

Si $d = d_{\ell-1}d_{\ell-2}\ldots d_1 d_0$ est la représentation en base 2 de d (avec $d_{\ell-1} = 1$), la méthode bien connue des « multiplications et mises au carré successives » permet d'effectuer l'exponentiation M^d en faisant ℓ mises au carré et w multiplications, où w est le nombre de composantes d_i non nulles : on pose $R = 1$ et à chaque étape i, $1 \leq i \leq \ell$, on fait $R \leftarrow R^2$, et si $d_{\ell-i} = 1$, on fait de plus $R \leftarrow MR$. Le R final vaut M^d.

●▷ Les deux idées de base que nous allons maintenant exploiter sont les suivantes [32], [29] : 1) En utilisant la représentation modifiée non adjacente (RMNA — cf. page 20) de d, on peut espérer avoir moins de composantes non nulles et ainsi économiser des multiplications. 2) $M^{d+k(p-1)(q-1)} = M^d \bmod n$, pour tout entier k. En explorant un ensemble d'entiers k, on peut espérer trouver un exposant $d + k(p-1)(q-1)$ ayant une représentation comptant « peu » de composantes non nulles et ainsi économiser des multiplications.

Ecrivons donc la RMNA binaire de d :

$$d = \sum_{i=0}^{\ell'-1} d'_i 2^i = d'_{\ell'-1} d'_{\ell'-2} \dots d'_1 d'_0,$$

où les d'_i valent 0, 1, ou -1, $d'_{\ell'-1} = 1$, $d'_i d'_{i+1} = 0$, et $\ell' \leq \ell + 1$.

Cette représentation est minimale et fournit le poids arithmétique de d. Or on peut montrer que le poids arithmétique moyen d'un entier dont la RMNA est de longueur ℓ vaut $\ell/3$, alors que le poids de Hamming moyen d'un vecteur binaire de longueur ℓ vaut $\ell/2$.

L'inconvénient de la RMNA est qu'elle comporte des '-1'. On peut pourtant facilement contourner cet obstacle en regroupant les '1' et les '-1', pour avoir à effectuer une seule division modulaire : écrivons $d = d^+ - d^-$ avec $d^+ = \sum_{i \in A} 2^i$, $d^- = \sum_{i \in B} 2^i$; alors $M^d = M^{d^+}/M^{d^-}$.

Etudions maintenant le gain moyen obtenu par le remplacement de l'exposant d par $\tilde{d} = d + k(p-1)(q-1)$; dans ce qui suit, nous considérerons qu'une multiplication coûte α fois plus qu'une mise au carré, et, par souci de simplicité, nous étudierons seulement le cas où d et $\tilde{d}$ sont représentés par leur représentation binaire, et non par leur RMNA.

On souhaite minimiser, parmi un ensemble d'exposants possibles $\tilde{d}$, la quantité $\ell(\tilde{d}) + \alpha w(\tilde{d})$, où $\ell(\tilde{d})$ est la longueur, et $w(\tilde{d})$ le nombre de composantes non nulles, de la représentation de $\tilde{d}$. Cette quantité exprime en effet l'équivalent du nombre de mises au carré à effectuer pour calculer $M^{\tilde{d}}$.

On pose $\ell = \ell(d)$, $\tilde{\ell} = \ell(\tilde{d})$, $\ell(k) = t\ell$, $\phi = (p-1)(q-1)$, et on fait les approximations suivantes (rappelons que l'entier n est de l'ordre de 500 bits actuellement et que sa taille est susceptible d'être augmentée, et que la clé secrète d doit être *grosso modo* du même ordre de grandeur) :

$$\tilde{\ell} = \ell(d + k\phi) \approx \ell(k\phi) = \ell(k) + \ell(\phi) \text{ et } \ell \approx \ell(\phi).$$

On en déduit que $\tilde{\ell} \approx (1+t)\ell$. On fait maintenant l'hypothèse que, lorsque k décrit l'ensemble des entiers de longueur $\ell(k) = t\ell$, l'ensemble des $2^{t\ell}$ vecteurs de longueur $\tilde{\ell} = (1+t)\ell$ représentant les $\tilde{d}$ se comporte comme un ensemble de vecteurs choisis aléatoirement et indépendamment parmi les $2^{\tilde{\ell}}$ vecteurs binaires de longueur $\tilde{\ell}$. Dans ce cas, l'espérance du

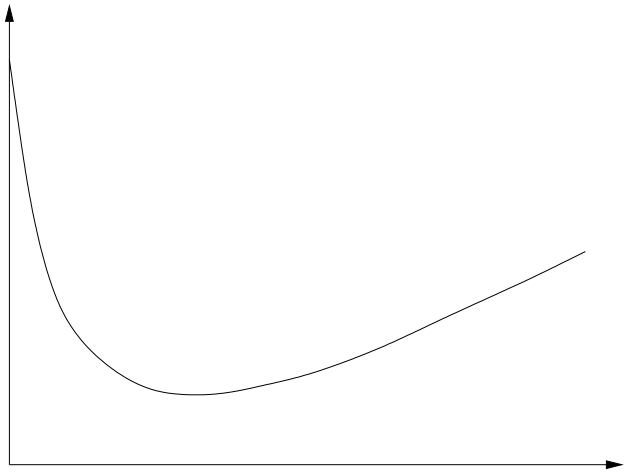

FIG. 11 – *Evolution du coût d'une exponentiation modulaire en fonction de t.*

nombre de vecteurs de poids u dans l'ensemble vaut :

$$E_u = 2^{t\ell} \times \frac{\binom{\tilde{\ell}}{u}}{2^{\tilde{\ell}}}$$

et dépasse 1 tant que

$$\binom{\tilde{\ell}}{u} \geq 2^{\ell}. \tag{5.9}$$

Soit $\tilde{u} = \min_{E_u \geq 1} u$, et posons $\tilde{u} = y\tilde{\ell}$. En utilisant la fonction entropie binaire H_2, on peut réécrire approximativement (5.9) : $\tilde{\ell} H_2(y) = \ell$, soit encore $H_2(y) = 1/(1+t)$. Le nombre moyen de mises au carré, $\ell(\tilde{d}) + \alpha w(\tilde{d})$, s'écrit donc maintenant

$$\tilde{\ell} + \alpha\tilde{u} = \tilde{\ell}(1 + \alpha y) = \ell(1+t)\Big(1 + \alpha H_2^{-1}\left(\frac{1}{t+1}\right)\Big).$$

Son évolution qualitative en fonction de t est représentée en Figure 11.

Une étude plus précise que ce que nous venons de brièvement présenter a été menée en [32], [29] : calcul du minimum de la courbe de coût et de la valeur de t pour laquelle il est atteint, avec différentes hypothèses sur le rapport α entre le coût d'une multiplication et celui d'une mise au carré, dans les deux cas de la représentation binaire et de la représentation modifiée. Par exemple, sous l'hypothèse d'une multiplication coûtant deux fois plus qu'une mise au carré, dans le cas de la représentation binaire exposé ci-dessus, on a un gain moyen d'un peu plus de 9% (pour une longueur d'exposant augmentant d'un peu moins de 11%).

Enfin des simulations ont été effectuées, qui corroborent ces résultats et valident le modèle théorique choisi en justifiant les approximations et hypothèses faites.

L'étude évoquée ci-dessus pour le cas des représentations modifiées pourrait être menée de la même manière, cette fois en cherchant à minimiser $W(\tilde{d})$ au lieu de $\ell(\tilde{d}) + \alpha W(\tilde{d})$, et l'on voit que cela revient à estimer le poids modulaire de Clark-Liang de d modulo ϕ, puisque l'on recherche $\tilde{d} = d + k\phi$, de poids arithmétique minimal (cf. Sections 1.2.1 et 4.2).

Cette possible approche statistique, visant à estimer le poids modulaire attendu, ne dispense pas d'une recherche théorique sur la complexité de ce problème, qu'il serait souhaitable de savoir mesurer mieux qu'en Section 4.2.

6 Perspectives

Je choisis le chemin qui monte. Pourquoi? C'est sans logique, sans certitude.

Nikos Kazantzaki

Le thème des codes identifiants est un champ encore relativement neuf, et suffisamment vaste pour se prêter à beaucoup d'investigations de type combinatoire ou géométrique. J'envisage donc d'y consacrer la principale partie de mes recherches à venir.

On peut par exemple imaginer de nouvelles techniques pour établir des bornes inférieures sur le cardinal d'un code identifiant (résultats de non-existence) ; s'intéresser à des classes de graphes telles que les chaînes ou les arbres [étude en cours], ou, pour les grilles infinies étudiées en Section 1.3, passer de la dimension deux (le plan) à la dimension trois (l'espace) ; établir la complexité du problème dans le cas de certains graphes [étude en cours pour le n-cube] ; appréhender, dans le n-cube, le comportement des sphères lorsqu'on augmente leur rayon : au bout d'un moment, leur « pouvoir d'identification » diminue — à la limite, si on prend des sphères de rayon n, on ne peut plus identifier aucun sommet.

Enfin, dans le cadre des grilles infinies susmentionnées, une généralisation possible semble particulièrement riche : soit $S = Z \times Z$ l'ensemble des sommets. Jusqu'ici, nous avons considéré des arêtes et des sphères, lesquelles sphères sont des *motifs* qui vont être utilisés pour couvrir S de manière à identifier tous les sommets : par exemple, dans le cas de la grille royale, une sphère de rayon t est un carré dont le côté compte $2t + 1$ sommets. Maintenant, abandonnons cette approche et envisageons directement un motif, par exemple un carré dont le côté compte un nombre *pair* de sommets ; ce n'est plus une sphère, mais on peut néanmoins chercher à couvrir S avec ce motif, de manière que deux sommets distincts ne soient pas couverts de la même façon. Toutes sortes de motifs peuvent alors être utilisés...

ANNEXE :

LISTE COMPLÈTE DES PUBLICATIONS, PAR ORDRE CHRONOLOGIQUE

• A. Lobstein : Rayon de recouvrement de codes binaires non-linéaires, *Traitement du Signal*, vol. 1–2–1, pp. 105–114, 1984.

• A. Lobstein : Contributions au codage combinatoire : ordres additifs, rayon de recouvrement, Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure des Télécommunications, Paris, 165 pages, 1985.

• A. Lobstein, G. Cohen, N.J.A. Sloane : Recouvrements d'espaces de Hamming binaires, *Comptes-Rendus de l'Académie des Sciences*, Sér. I, vol. 301, pp. 135–138, 1985.

• A. Lobstein : When are modular weights identical?, EUT Report 86-WSK-05, University of Technology of Eindhoven, the Netherlands, 54 pages, 1986. [53]

• G. Cohen, A. Lobstein, N.J.A. Sloane : Further results on the covering radius of codes, *IEEE Trans. on Inform. Theory*, vol. 32, pp. 680–694, 1986. [30]

• G. Cohen, A. Lobstein, N.J.A. Sloane : On a conjecture concerning coverings of Hamming space, *Lecture Notes in Computer Science*, No. 228, pp. 79–89, New York : Springer-Verlag, 1986. [31]

• A. Lobstein, G. Cohen : Sur la complexité d'un problème de codage, *RAIRO Informatique Théorique et Applications*, vol. 21–1, pp. 25–32, 1987. [60]

• A. Lobstein : On modular weights in arithmetic codes, *Lecture Notes in Computer Science*, No. 311, pp. 56–67, New York : Springer-Verlag, 1988. [54]

• A. Lobstein : Comments on "A note on perfect arithmetic codes", *IEEE Trans. on Inform. Theory*, vol. 34, pp. 589–590, 1988. [55]

• A. Lobstein : On the nonexistence of a perfect binary arithmetic code with modulus 1791, Rapport Interne 88D013, Ecole Nationale Supérieure des Télécommunications, Paris, 12 pages, 1988.

• A. Lobstein : On the nonexistence of a perfect binary arithmetic code with modulus 4097, Rapport Interne 88D014, Ecole Nationale Supérieure des Télécommunications, Paris, 23 pages, 1988.

• A. Lobstein : A new proof for the complexity of linear decoding with preprocessing, Rapport Interne 89D006, Ecole Nationale Supérieure des Télécommunications, Paris, 8 pages, 1989. [56]

• A. Lobstein, G.J.M. van Wee : On normal and subnormal q-ary codes, *IEEE Trans. on Inform. Theory*, vol. 35, pp. 1291–1295, 1989. Correction to "On normal and subnormal q-ary codes", *IEEE Trans. on Inform. Theory*, vol. 36, p. 1498, 1990. [62]

• J.P. Barthélemy, G. Cohen, A. Lobstein : Eléments d'algorithmique moderne, Polycopié 90INF002, Ecole Nationale Supérieure des Télécommunications, Paris, 245 pages, 1990.

• A. Lobstein : On perfect binary arithmetic codes which can correct two errors or more, *Ars Combinatoria*, vol. 29, pp. 24–27, 1990.

• A. Lobstein : The hardness of solving Subset Sum with preprocessing, *IEEE Trans. on Inform. Theory*, vol. 36, pp. 943–946, 1990. [57]

• A. Lobstein : Quelques problèmes de métriques dans les codes arithmétiques, Rapport Interne 91D003, Ecole Nationale Supérieure des Télécommunications, Paris, 32 pages, 1991. [58]

• A. Lobstein, P. Solé : Arithmetic codes – Survey, recent and new results, *Lecture Notes in Computer Science*, No. 539, pp. 246–258, New York : Springer-Verlag, 1991. [61]

• A. Lobstein : Results on the nonexistence of some perfect arithmetic codes, Rapport Interne 91D014, Ecole Nationale Supérieure des Télécommunications, Paris, 21 pages, 1991.

• G. Cohen, S. Litsyn, A. Lobstein, G. Zémor (Eds.) : Algebraic Coding, First French-Soviet Workshop, Proceedings, *Lecture Notes in Computer Science*, No. 573, New York : Springer-Verlag, 158 pages, 1992.

• A. Lobstein : On perfect arithmetic codes, *Discrete Mathematics*, vol. 106/107, pp. 333–336, 1992. [59]

• J.P. Barthélemy, G. Cohen, A. Lobstein : Complexité algorithmique et problèmes de communications, Paris : Masson, xxxviii+228 pages, 1992. [6]

• G. Cohen, S. Litsyn, A. Lobstein, G. Zémor (Eds.) : Algebraic Coding, First French-Israeli Workshop, Proceedings, *Lecture Notes in Computer Science*, No. 781, New York : Springer-Verlag, 326 pages, 1994.

• A. Lobstein, V. Pless : The length function: a revised table, *Lecture Notes in Computer Science*, No. 781, pp. 51–55, New York : Springer-Verlag, 1994.

• G. Kabatianski, A. Lobstein : On Plotkin-Elias type bounds for binary arithmetic codes, *Lecture Notes in Computer Science*, No. 781, pp. 263–269, New York : Springer-Verlag, 1994. [49]

• I. Charon, O. Hudry, A. Lobstein : A new method for constructing codes, *Proc. IVth Internat. Coll. on Algebraic and Combinatorial Coding*, Novgorod, pp. 62–65, 1994. [15]

• G. Cohen, S. Litsyn, A. Lobstein, H.F. Mattson, Jr. : Covering radius 1985–1994, Rapport Interne 94D025, Ecole Nationale Supérieure des Télécommunications, Paris, 76 pages, 1994.

• J.P. Barthélemy, G. Cohen, A. Lobstein : Algorithmic Complexity and Communication Problems, London : University College of London, xx+256 pages, 1996. [7]

• G. Cohen, S. Litsyn, A. Lobstein, H.F. Mattson, Jr. : Covering radius 1985–1994, *Applicable Algebra in Engineering, Communication and Computing*, vol. 8–3, 67 pages, 1997. [28]

• G. Cohen, I. Honkala, S. Litsyn, A. Lobstein : Covering Codes, Amsterdam : Elsevier, xxii+ 542 pages, 1997. [23]

• A. Lobstein, V. Zinoviev : On new perfect binary nonlinear codes, *Applicable Algebra in Engineering, Communication and Computing*, vol. 8–5, pp. 415–420, 1997. [63]

• G. Cohen, A. Lobstein, G. Zémor : Comment accélérer une exponentiation modulaire, Rapport Interne 97D006, Ecole Nationale Supérieure des Télécommunications, Paris, 26 pages, 1997. [32]

• G. Cohen, A. Lobstein, D. Naccache, G. Zémor : How to improve an exponentiation black box, *Lecture Notes in Computer Science*, No. 1403, pp. 211–220, New York : Springer-Verlag, 1998. [29]

• I. Honkala, A. Lobstein : On the complexity of calculating the minimum norm of a code, *Proc. Workshop on Coding and Cryptography '99*, Paris, pp. 21–27, 1999. [46]

• G. Cohen, I. Honkala, A. Lobstein, G. Zémor : New bounds for codes identifying vertices in graphs, *Electronic Journal of Combinatorics*, vol. 6(1), R19, `http://www.combinatorics.org`, 1999. [24]

• G. Cohen, A. Lobstein, G. Zémor : Identification d'une station défaillante dans un contexte radio-mobile, *Aspects Algorithmiques des Télécommunications (AlgoTel '99)*, Actes, pp. 19–22, 1999. [33]

• G. Cohen, S. Gravier, I. Honkala, A. Lobstein, M. Mollard, Ch. Payan, G. Zémor : Improved identifying codes for the grid, *Electronic Journal of Combinatorics*, vol. 6(1), Comments to R19, `http://www.combinatorics.org`, 1999. [22]

• G. Cohen, I. Honkala, A. Lobstein, G. Zémor : Bounds for codes identifying vertices in the hexagonal grid, *SIAM Journal on Discrete Mathematics*, vol. 13, No. 4, pp. 492–504, 2000. [25]

• I. Charon, I. Honkala, O. Hudry, A. Lobstein : Identifying codes, Rapport interne Télécom Paris-2000D009, Paris, 67 pages, 2000.

• V. Zinoviev, A. Lobstein : On generalized concatenated constructions of perfect binary nonlinear codes, *Problemy Peredachi Informatsii*, vol. 36, No. 4, pp. 3–17, 2000 (en russe). Traduction anglaise : *Problems of Information Transmission*, vol. 36, No. 4, pp. 336–348, 2000. [76]

• G. Cohen, I. Honkala, A. Lobstein, G. Zémor : On identifying codes, *Proceedings of DIMACS Workshop on Codes and Association Schemes '99*, vol. 56, pp. 97–109, 2001. [26]

• G. Cohen, I. Honkala, A. Lobstein, G. Zémor : On codes identifying vertices in the two-dimensional square lattice with diagonals, *IEEE Transactions on Computers*, vol. 50, pp. 174–176, 2001. [27]

• S. Avgustinovich, A. Lobstein, F. Solov'eva : Intersection matrices for partitions by binary perfect codes, *IEEE Trans. on Inform. Theory*, vol. 47, pp. 1621–1624, 2001. [2]

• I. Charon, I. Honkala, O. Hudry, A. Lobstein : General bounds for identifying codes in some infinite regular graphs, *Electronic Journal of Combinatorics*, vol. 8(1), R39, `http://www.combinatorics.org`, 2001. [12]

• I. Charon, O. Hudry, A. Lobstein : Identifying codes with small radius in some infinite regular graphs, *Electronic Journal of Combinatorics*, vol. 9(1), R11, `http://www.combinatorics.org`, 2002. [16]

• I. Honkala, A. Lobstein : On the density of identifying codes in the square lattice, *Journal of Combinatorial Theory*, Ser. B, vol. 85, pp. 297–306, 2002. [47]

• I. Charon, O. Hudry, A. Lobstein : Identifying and locating-dominating codes: NP-completeness results for directed graphs, *IEEE Trans. on Inform. Theory*, vol. 48, pp. 2192–2200, 2002. [17]

• 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. 290/3, pp. 2109–2120, 2003. [18]

• I. Charon, I. Honkala, O. Hudry, A. Lobstein : The minimum density of an identifying code in the king lattice, *Discrete Mathematics*, vol. 276(1/3), pp. 95–109, 2004. [13]

Références

[1] J. Astola : A note on perfect arithmetic codes, *IEEE Trans. on Inform. Theory*, vol. 32, pp. 443–445, 1986.

[2] S. Avgustinovich, A. Lobstein, F. Solov'eva : Intersection matrices for partitions by binary perfect codes, *IEEE Trans. on Inform. Theory*, vol. 47, pp. 1621–1624, 2001.

[3] P. Balalaïka : Deafness caused by tomato injury. Observations on half a case, *Acta Pathol. Marignan*, vol. 1, pp. 1–7, 1515.

[4] K. Ball : On packing unequal squares, *Journal of Combinatorial Theory*, Ser. A, vol. 75, pp. 353–357, 1996.

[5] S. Barg : Some new NP-complete coding problems, *Problems of Information Transmission*, vol. 30–3, pp. 209–214, 1994.

[6] J.P. Barthélemy, G. Cohen, A. Lobstein : Complexité algorithmique et problèmes de communications, Paris : Masson, xxxviii+228 pages, 1992.

[7] J.P. Barthélemy, G. Cohen, A. Lobstein : Algorithmic Complexity and Communication Problems, London : University College of London, xx+256 pages, 1996.

[8] E.R. Berlekamp, R.J. McEliece, H.C.A. van Tilborg : On the inherent intractability of certain coding problems, *IEEE Trans. on Inform. Theory*, vol. 24, pp. 384–386, 1978.

[9] U. Blass, S. Litsyn : Several new lower bounds on the size of codes with covering radius one, *IEEE Trans. on Inform. Theory*, vol. 44, pp. 1998–2002, 1998.

[10] G. Bogdanova : Optimal codes over an alphabet of 4 elements, *Proc. Vth Internat. Coll. on Algebraic and Combinatorial Coding*, Sozopol, pp. 46–53, 1996.

[11] J. Bruck, M. Naor : The hardness of decoding linear codes with preprocessing, *IEEE Trans. on Inform. Theory*, vol. 36, pp. 381–385, 1990.

[12] I. Charon, I. Honkala, O. Hudry, A. Lobstein : General bounds for identifying codes in some infinite regular graphs, *Electronic Journal of Combinatorics*, vol. 8(1), R39, `http://www.combinatorics.org`, 2001.

[13] I. Charon, I. Honkala, O. Hudry, A. Lobstein : The minimum density of an identifying code in the king lattice, *Discrete Mathematics*, vol. 276(1/3), pp. 95–109, 2004.

[14] I. Charon, O. Hudry : The noising method: a new method for combinatorial optimization, *Operations Research Letters*, No. 14, pp. 133–137, 1993.

[15] I. Charon, O. Hudry, A. Lobstein : A new method for constructing codes, *Proc. IVth Internat. Coll. on Algebraic and Combinatorial Coding*, Novgorod, pp. 62–65, 1994.

[16] I. Charon, O. Hudry, A. Lobstein : Identifying codes with small radius in some infinite regular graphs, *Electronic Journal of Combinatorics*, vol. 9(1), R11, `http://www.combinatorics.org`, 2002.

[17] I. Charon, O. Hudry, A. Lobstein : Identifying and locating-dominating codes: NP-completeness results for directed graphs, *IEEE Trans. on Inform. Theory*, vol. 48, pp. 2192–2200, 2002.

[18] 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. 290/3, pp. 2109–2120, 2003.

[19] A.C.L. Chiang, I.S. Reed : Arithmetic norms and bounds of the arithmetic AN codes, *IEEE Trans. on Inform. Theory*, vol. 16, pp. 470–476, 1970.

[20] W.E. Clark, J.J. Liang : On arithmetic weight for a general radix representation of integers, *IEEE Trans. on Inform. Theory*, vol. 19, pp. 823–826, 1973.

[21] W.E. Clark, J.J. Liang : On modular weight and cyclic nonadjacent forms for arithmetic codes, *IEEE Trans. on Inform. Theory*, vol. 20, pp. 767–770, 1974.

[22] G. Cohen, S. Gravier, I. Honkala, A. Lobstein, M. Mollard, Ch. Payan, G. Zémor : Improved identifying codes for the grid, *Electronic Journal of Combinatorics*, vol. 6(1), Comments to R19, `http://www.combinatorics.org`, 1999.

[23] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein : Covering Codes, Amsterdam : Elsevier, xxii+542 pages, 1997.

[24] G. Cohen, I. Honkala, A. Lobstein, G. Zémor : New bounds for codes identifying vertices in graphs, *Electronic Journal of Combinatorics*, vol. 6(1), R19, `http://www.combinatorics.org`, 1999.

[25] G. Cohen, I. Honkala, A. Lobstein, G. Zémor : Bounds for codes identifying vertices in the hexagonal grid, *SIAM Journal on Discrete Mathematics*, vol. 13, No. 4, pp. 492–504, 2000.

[26] G. Cohen, I. Honkala, A. Lobstein, G. Zémor : On identifying codes, *Proceedings of DIMACS Workshop on Codes and Association Schemes '99*, vol. 56, pp. 97–109, 2001.

[27] G. Cohen, I. Honkala, A. Lobstein, G. Zémor : On codes identifying vertices in the two-dimensional square lattice with diagonals, *IEEE Transactions on Computers*, vol. 50, pp. 174–176, 2001.

[28] G. Cohen, S. Litsyn, A. Lobstein, H.F. Mattson, Jr. : Covering radius 1985–1994, *Applicable Algebra in Engineering, Communication and Computing*, vol. 8–3, 67 pages, 1997.

[29] G. Cohen, A. Lobstein, D. Naccache, G. Zémor : How to improve an exponentiation black box, *Lecture Notes in Computer Science*, No. 1403, pp. 211–220, New York : Springer-Verlag, 1998.

[30] G. Cohen, A. Lobstein, N.J.A. Sloane : Further results on the covering radius of codes, *IEEE Trans. on Inform. Theory*, vol. 32, pp. 680–694, 1986.

[31] G. Cohen, A. Lobstein, N.J.A. Sloane : On a conjecture concerning coverings of Hamming space, *Lecture Notes in Computer Science*, No. 228, pp. 79–89, New York : Springer-Verlag, 1986.

[32] G. Cohen, A. Lobstein, G. Zémor : Comment accélérer une exponentiation modulaire, Rapport Interne 97D006, Ecole Nationale Supérieure des Télécommunications, Paris, 26 pages, 1997.

[33] G. Cohen, A. Lobstein, G. Zémor : Identification d'une station défaillante dans un contexte radio-mobile, *Aspects Algorithmiques des Télécommunications (AlgoTel '99)*, Actes, pp. 19–22, 1999.

[34] C.J. Colbourn, P.J. Slater, L.K. Stewart : Locating dominating sets in series parallel networks, *Congressus Numerantium*, vol. 56, pp. 135–162, 1987.

[35] S.A. Cook : The complexity of theorem-proving procedures, *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, New York, pp. 151–158, 1971.

[36] P. Diaconis, R.L. Graham : The Radon transform on Z_2^k, *Pacific J. Math.*, vol. 118, pp. 323–345, 1985.

[37] W. Diffie, M.E. Hellman : New directions in cryptography, *IEEE Trans. on Inform. Theory*, vol. 22, pp. 644–654, 1976.

[38] S. Ernvall : When does the modular distance induce a metric in the binary case?, *IEEE Trans. on Inform. Theory*, vol. 28, pp. 665–668, 1982.

[39] S. Ernvall : When does the modular distance induce a metric?, *Annales Univ. Turku*, Ser. A, Math., No. 185, 64 pages, 1983.

[40] S. Ernvall : On the modular distance, *IEEE Trans. on Inform. Theory*, vol. 31, pp. 521–522, 1985.

[41] S. Ernvall : The Hamming bound for binary arithmetic AN codes, *Ars Combinatoria*, vol. 20–B, pp. 207–227, 1985.

[42] S. Ernvall : On the Hamming bound for nonbinary arithmetic AN codes, *Ars Combinatoria*, vol. 25–B, pp. 31–53, 1988.

[43] M. Frances, A. Litman : On covering problems of codes, *Theory of Computing Systems*, vol. 30–2, pp. 113–119, 1997.

[44] D.M. Gordon : Perfect multiple error-correcting arithmetic codes, *Mathematics of Computation*, vol. 49, pp. 621–633, 1987.

[45] R.L. Graham, N.J.A. Sloane : On the covering radius of codes, *IEEE Trans. on Inform. Theory*, vol. 31, pp. 385–401, 1985.

[46] I. Honkala, A. Lobstein : On the complexity of calculating the minimum norm of a code, *Proc. Workshop on Coding and Cryptography '99*, Paris, pp. 21–27, 1999.

[47] I. Honkala, A. Lobstein : On the density of identifying codes in the square lattice, *Journal of Combinatorial Theory*, Ser. B, vol. 85, pp. 297–306, 2002.

[48] G. Kabatianski : Bounds on the number of code words in binary arithmetic codes, *Problems of Information Transmission*, vol. 12–4, pp. 277–283, 1976.

[49] G. Kabatianski, A. Lobstein : On Plotkin-Elias type bounds for binary arithmetic codes, *Lecture Notes in Computer Science*, No. 781, pp. 263–269, New York : Springer-Verlag, 1994.

[50] R.M. Karp : Reductibility among combinatorial problems, in R.E. Miller & J.W. Thatcher (Eds.) Complexity of Computer Computations, New York : Plenum Press, pp. 85–103, 1972.

[51] M.G. Karpovsky, K. Chakrabarty, L.B. Levitin : On a new class of codes for identifying vertices in graphs, *IEEE Trans. on Inform. Th.*, vol. 44, pp. 599–611, 1998.

[52] J.H. van Lint : Introduction to Coding Theory, Chapitre 10, New York : Springer-Verlag, 1982.

[53] A. Lobstein : When are modular weights identical?, EUT Report 86-WSK-05, University of Technology of Eindhoven, the Netherlands, 54 pages, 1986.

[54] A. Lobstein : On modular weights in arithmetic codes, *Lecture Notes in Computer Science*, No. 311, pp. 56–67, New York : Springer-Verlag, 1988.

[55] A. Lobstein : Comments on "A note on perfect arithmetic codes", *IEEE Trans. on Inform. Theory*, vol. 34, pp. 589–590, 1988.

[56] A. Lobstein : A new proof for the complexity of linear decoding with preprocessing, Rapport Interne 89D006, Ecole Nationale Supérieure des Télécommunications, Paris, 8 pages, 1989.

[57] A. Lobstein : The hardness of solving Subset Sum with preprocessing, *IEEE Trans. on Inform. Theory*, vol. 36, pp. 943–946, 1990.

[58] A. Lobstein : Quelques problèmes de métriques dans les codes arithmétiques, Rapport Interne 91D003, Ecole Nationale Supérieure des Télécommunications, Paris, 32 pages, 1991.

[59] A. Lobstein : On perfect arithmetic codes, *Discrete Mathematics*, vol. 106/107, pp. 333–336, 1992.

[60] A. Lobstein, G. Cohen : Sur la complexité d'un problème de codage, *RAIRO Informatique Théorique et Applications*, vol. 21–1, pp. 25–32, 1987.

[61] A. Lobstein, P. Solé : Arithmetic codes – Survey, recent and new results, *Lecture Notes in Computer Science*, No. 539, pp. 246–258, New York : Springer-Verlag, 1991.

[62] A. Lobstein, G.J.M. van Wee : On normal and subnormal q-ary codes, *IEEE Trans. on Inform. Theory*, vol. 35, pp. 1291–1295, 1989. Correction to "On normal and subnormal q-ary codes", *IEEE Trans. on Inform. Theory*, vol. 36, p. 1498, 1990.

[63] A. Lobstein, V. Zinoviev : On new perfect binary nonlinear codes, *Applicable Algebra in Engineering, Communication and Computing*, vol. 8–5, pp. 415–420, 1997.

[64] R.J. McEliece : A public-key cryptosystem based on algebraic coding theory, *JPL DSN Progress Report* 42–44, pp. 114–116, 1978.

[65] A.M. McLoughlin : The complexity of computing the covering radius of a code, *IEEE Trans. on Inform. Theory*, vol. 30, pp. 800–804, 1984.

[66] R.C. Merkle, M.E. Hellman : Hiding information and signatures in trapdoor knapsacks, *IEEE Trans. on Inform. Theory*, vol. 24, pp. 525–530, 1978.

[67] A.R. Meyer, L.J. Stockmeyer : The equivalence problem for regular expressions with squaring requires exponential time, *Proc. 13th Ann. IEEE Symp. on Switching and Automata Theory*, Long Beach, pp. 125–129, 1972.

[68] S.C. Ntafos, S.L. Hakimi : On the complexity of some coding problems, *IEEE Trans. on Inform. Theory*, vol. 27, pp. 794–796, 1981.

[69] P.R.J. Östergård, U. Blass: On the size of optimal binary codes of length 9 and covering radius 1, *IEEE Trans. Inform. Th.*, vol. 47, pp. 2556–2557, 2001.

[70] T.R.N. Rao : Error Coding for Arithmetic Processors, Chapitre 4, New York : Academic Press, 1974.

[71] T.R.N. Rao, O.N. Garcia : Cyclic and multiresidue codes for arithmetic operations, *IEEE Trans. on Inform. Theory*, vol. 17, pp. 85–91, 1971.

[72] R.L. Rivest, A. Shamir, L. Adleman : A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, pp. 120–126, 1978.

[73] A. Vardy : The intractability of computing the minimum distance of a code, *IEEE Trans. on Inform. Theory*, vol. 43, pp. 1757–1766, 1997.

[74] J. Vasiliev : On nongroup close-packed codes, *Problemy Kibernetiki*, vol. 8, pp. 337–339, 1962 (en russe).

[75] V. Zinoviev : On generalized concatenated codes, *Colloquia Mathematica Societatis János Bolyai* 16, Topics in Information Theory, pp. 587–592, Keszthely, Hongrie, 1975.

[76] V. Zinoviev, A. Lobstein : On generalized concatenated constructions of perfect binary nonlinear codes, *Problemy Peredachi Informatsii*, vol. 36, No. 4, pp. 3–17, 2000 (en russe). Traduction anglaise : *Problems of Information Transmission*, vol. 36, No. 4, pp. 336–348, 2000.