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

Contents

PREFACE

This document tries to be the survey of some fifteen years of research: the defence of my Thesis took place in 1985 at Ecole Nationale Supérieure des Télécommunications (ENST), under Gérard Cohen's supervision, and I obtained a permanent position at Centre National de la Recherche Scientifique (CNRS), at ENST, in 1987. I belong to the team "Mathematics of Computer Science and Networks", inside the Department "Computer Science and Networks".

All my research is **Discrete Mathematics** and **Combinatorics**; the main theme, **coding**, is seen from a multiple yet always combinatorial viewpoint: my vision of codes is of combinatorial objects floating in different discrete spaces, and their links with, for instance, the theory of complexity and its structured classes of problems, are quite natural to me.

I chose a thematic rather than chronological presentation, wishing to show the links between the different fields covered by my research, and to sometimes stress one result which I find more interesting, significant, or easier to explain, than others.

From a chronological standpoint, I will only say that my set of themes has progressively moved, according to encounters, circumstances, frequent and varied collaborations, likings and serendipity, from covering radius and arithmetic codes to perfect block codes and identifying codes, often with complexity issues in the background, active incursions into cryptography being rare. This trajectory is admittedly not rectilinear, however it is consistent and within the fields of research at ENST, information processing and communication — indeed, I often publish with some of my ENST colleagues.

The first section is devoted to codes (see Figure 1):
– block codes and their fundamental parameters, minimum distance d and covering radius R, joining in the relation $d = 2R + 1$ for perfect codes;
– arithmetic codes, using different representations of integers, raising metric problems, and still open to new perfect codes;

– identifying codes, spotting vertices in graphs, in particular the square, triangular, king, and hexagonal grids.

The links between codes and complexity, codes and cryptology, are described after the second section, devoted to a short account on the theory of complexity, and the third section, describing three public-key cryptosystems.

The fourth section shows some links between codes and complexity (see Figure 2):
– NP- or Π_2-completeness of problems dealing with block codes, study of problems for which a preprocessing is possible, algorithms for the construction of "good" codes;
– complexity and arithmetic codes: hardness of computing the Clark-Liang modular weight;
– NP-completeness of the existence of identifying codes of bounded size, algorithms for the construction of "small" identifying codes.

The fifth section underscores some links between codes and cryptology (see Figure 3), in particular the relations between nonadjacent modified representations, modular weight, and fast modular exponentiation for the RSA cryptosystem.

A short conclusion mentions some possibilities in the future.

I also added a slightly shortened version, written in a common pidgin, for some of my foreign colleagues.

Two appendices contain the complete list of my publications as well as some articles which seem significant to me (the latter only in the "hard copy" of this document — available on request).

My research was often done in collaboration, leading to articles or books. These exchanges are enriching experiences and it is my pleasure here to warmly thank all my co-authors (in order of appearance):
Gérard Cohen (ENST, France),
Neil Sloane (Bell Labs, USA),
Gerhard van Wee (Eindhoven University, the Netherlands),
Jean-Pierre Barthélemy (ENST Bretagne, France),
Patrick Solé (CNRS, France),

Vera Pless (University of Illinois, USA),
Grigori Kabatianski (Institute for Problems of Information Transmission [IPPI], Russia),
Irène Charon (ENST, France),
Olivier Hudry (ENST, France),
Simon Litsyn (Tel Aviv University, Israel),
Skip Mattson (Syracuse University, USA),
Iiro Honkala (Turku University, Finland),
Victor Zinoviev (IPPI, Russia),
Gilles Zémor (ENST, France),
David Naccache (Gemplus, France),
Sylvain Gravier (CNRS, France),
Michel Mollard (CNRS, France),
Charles Payan (CNRS, France),
Sergey Avgustinovich (Sobolev Institute of Mathematics [SIM], Russia),
and Faina Solov'eva (SIM, Russia).

The credit for a great deal of what follows is theirs.

Hence, the "we" I will use throughout this document, now will be a we of modesty, now will designate a set of authors.

COMBINATORICS

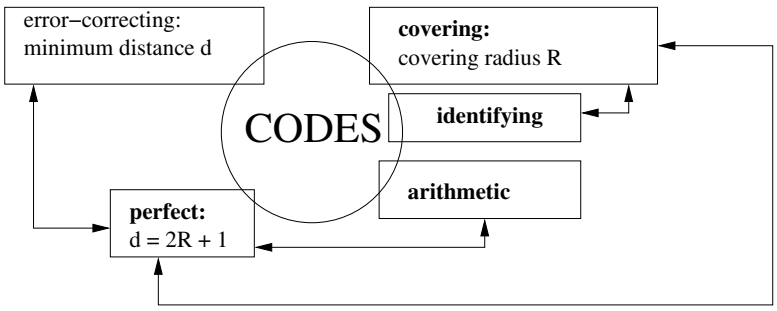

FIG. 1 – *Coding; in bold, the themes we studied.*

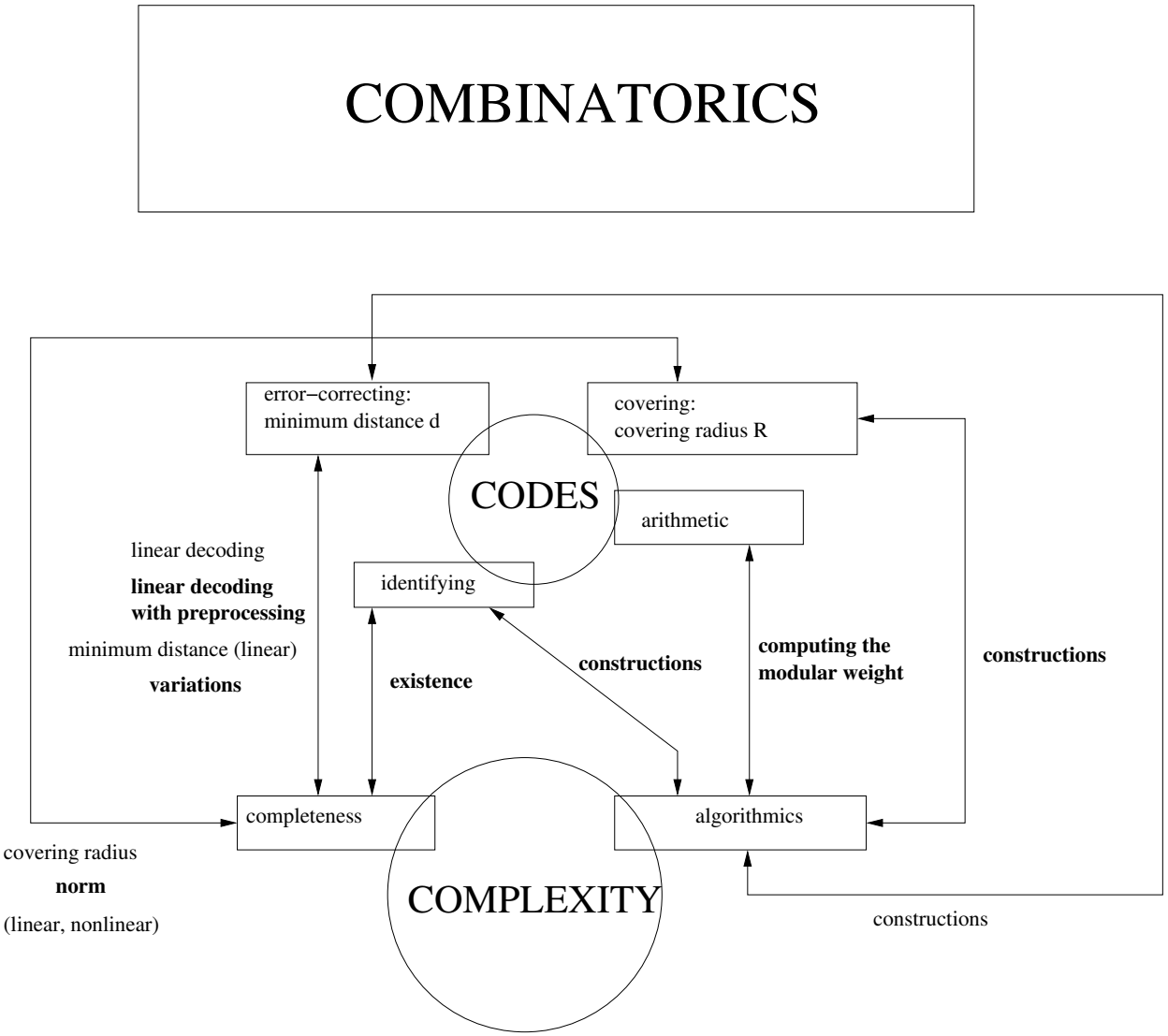

FIG. 2 – *Some links between coding and complexity; in bold, the links we studied.*

COMBINATORICS

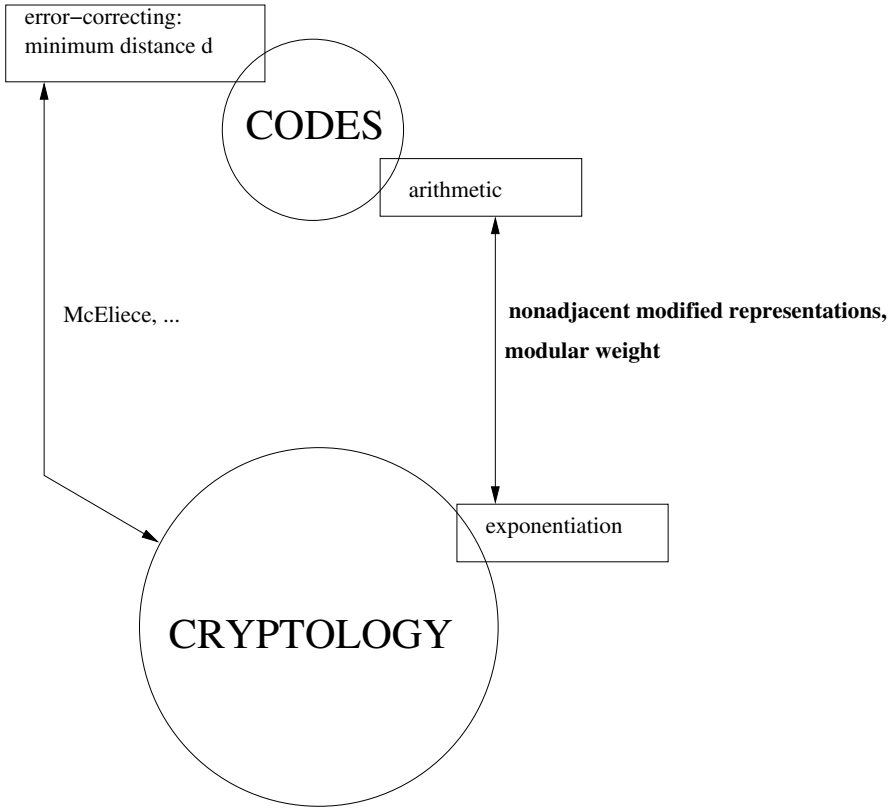

FIG. 3 – *Some links between coding and cryptology; in bold, the links we studied.*

1 Basic Facts in Coding

Our problematics can be stated in a very general way: we consider a discrete space (a vector space over a finite field, a ring of integers, a graph), and a metric (Hamming, Rao-Garcia, Clark-Liang, shortest path), and we study certain properties of certain subsets (called *codes*) in this space, properties relative to the metric.

We divided this section into three subsections. The first one is devoted to block codes in Hamming spaces, seen from two different viewpoints, the minimum distance (*error-correcting codes*) and the covering radius (*covering codes*). The second one deals with *arithmetic codes*, which, mostly used as error-correcting codes, present features deserving a separate study. The third section discusses *identifying codes*, which have been my main topic in the last two years. These codes can be seen as a particular case of covering codes, but we consider them in some graphs other than the Hamming n-cubes, and this is why they have a subsection of their own.

1.1 Block Codes

Most of the time we will use $F = F_2 = \{0, 1\}$, and, for the sake of simplicity, the definitions, notations, and basic notions are presented in the binary case. Their generalization to the finite field F_q, where q is a prime power, is straightforward. In particular, all operations below are modulo 2.

The Hamming space, F_2^n $(= F^n)$, is the set of binary vectors of length n, and the *Hamming distance*, d, between two vectors $\mathbf{x} = x_1x_2\ldots x_n \in F^n$ and $\mathbf{y} = y_1y_2\ldots y_n \in F^n$ is $d(\mathbf{x},\mathbf{y}) = |\{i \in \{1,2,\ldots,n\} \ : x_i \neq y_i\}|$. The (Hamming) *weight*, $w(\mathbf{x})$, of $\mathbf{x}$ is its (Hamming) distance to the all-zero vector. The distance between $\mathbf{x}$ and a nonempty subset $Y \subseteq F^n$ is $d(\mathbf{x}, Y) = \min\{d(\mathbf{x},\mathbf{y}) \ : \mathbf{y} \in Y\}$. The *sphere* of centre $\mathbf{x}$ and radius t is

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

Its volume $V(t)$ does not depend on its centre. Vector $\mathbf{x}$ is said to be *t-covered* (or *covered* if there is no ambiguity) by $\mathbf{y} \in F^n$ if $d(\mathbf{x},\mathbf{y}) \leq t$, and by a nonempty subset $Y \subseteq F^n$ if it is

covered by at least one element of Y.

A *binary code* C of *length* n and size K ($K \geq 2$) is a set of K binary vectors of length n. Its elements are called *codewords*.

If C is a vector subspace of dimension k in F^n, it is called *linear*. It can be defined by a *generator matrix*, $\mathbf{G}$, with dimensions $k \times n$, the rows of which form a basis of C. The *dual code* of C, $C^\perp$, is the set of vectors which are *orthogonal* with all vectors in C:

$$C^\perp = \{\mathbf{x} = x_1 x_2 \dots x_n \in F^n : \forall \mathbf{c} = c_1 c_2 \dots c_n \in C, < \mathbf{x}, \mathbf{c} >= \sum_{1 \leq i \leq n} x_i c_i = 0\}.$$

Code $C^\perp$ is also a vector subspace in F^n, of dimension $n - k$, and any generator matrix $\mathbf{H}$, of dimensions $(n-k) \times n$, characterizes C:

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

where $\mathbf{0}$ is the all-zero vector of length $n - k$ and T the symbol of transposition. Matrix $\mathbf{H}$ is the *parity-check matrix* of C. The *syndrome* of $\mathbf{y} \in F^n$ is $\mathbf{y}\mathbf{H}^T \in F^{n-k}$. So a vector is a codeword if and only if its syndrome is zero.

The main parameters of a code C are its *minimum distance*, $d(C)$ or d, and its *covering radius*, $R(C)$ or R. We denote C by $(n, K, d)R$, and $[n, k, d]R$ if it is linear. Also used are (n, K), $[n, k]$, (n, K, d), $[n, k, d]$, $(n, K)R$, or $[n, k]R$.

Definition. The minimum distance of C is:

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

If $e = \lfloor \frac{d-1}{2} \rfloor$, the spheres of radius e centred at the codewords have pairwise empty intersections, and e is the largest integer with this property. When C is linear,

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

Still in the linear case, the minimum distance of C can be characterized by the parity-check matrix: it is the smallest positive integer d such that $\mathbf{0}^T$ (of length $n - k$) is the sum of d columns of a parity-check matrix (of dimensions $(n-k) \times n$) of C.

Definition. The covering radius of C is:

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

In other words, the codewords R-cover F^n, and R is the smallest integer with this property.

In the linear case, the covering radius of C can be characterized by the parity-check matrix: it is the smallest integer R such that any transpose vector of length $n-k$ is the sum of at most R columns of a parity-check matrix (of dimensions $(n-k) \times n$) of C.

A code $(n, K, d)R$ satisfies the following inequalities:

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

called the "sphere-packing bound" or "Hamming bound", and the "sphere-covering bound", respectively.

1.1.1 Error-Correcting Codes

Error-correcting codes are designed in order to correct errors occurring during transmission over a noisy channel. Consider the *binary symmetric memoryless* channel: zeros and ones are transmitted, and with a probability $p < 1/2$ a '1' is wrongly transformed into a '0', or a '0' into a '1'. A block of k information symbols $\mathbf{u} = u_1 u_2 \dots u_k$ is coded by a codeword $\mathbf{c} = c_1 c_2 \dots c_n \in C$, with $n \geq k$.

Consider a linear code C with parameters $[n, k, d]$, generator matrix $\mathbf{G}$ and parity-check matrix $\mathbf{H}$. The error-detection and error-correction capacity of C is directly linked to its minimum distance: after the coding of $\mathbf{u}$ by the vector of length n, $\mathbf{c} = \mathbf{uG} \in C$, and the transmission of $\mathbf{c}$, the receiver receives $\mathbf{z} = \mathbf{c} + \mathbf{e}$, where $\mathbf{e} \in F^n$ is the error vector. Since $e = \lfloor \frac{d-1}{2} \rfloor$, if $w(\mathbf{e}) \leq e$, then $\mathbf{c}$ is the unique codeword which is closest to $\mathbf{z}$. The parameter e is the *error-correcting capacity* of C, and C is an *e-error-correcting code*; we'll say equally that C can correct e errors or an error of weight e.

Decoding (retrieving $\mathbf{c}$) can be done using $\mathbf{H}$: compute the syndrome of $\mathbf{z}$, $\mathbf{y} = \mathbf{zH}^T$, next $\mathbf{c}^* = \mathbf{z} + \mathbf{x}$, where $\mathbf{x}$ is a minimum-weight solution of $\mathbf{xH}^T = \mathbf{y}$. Indeed, $\mathbf{c}^*\mathbf{H}^T = \mathbf{zH}^T + \mathbf{xH}^T = \mathbf{y} + \mathbf{y} = \mathbf{0}$: $\mathbf{c}^*$ is the codeword closest to $\mathbf{z}$.

We face two crucial problems in coding:

1) Find "large" and "short" codes, linear or not, with a "large" minimum distance.

Either we fix n and d and search for a code (n, K, d) (or $[n, k, d]$) with the largest possible size K (or dimension k), or we fix n and k and search for a code $[n, k, d]$ with the largest possible minimum distance d, or we fix d and k and search for the smallest possible length n for a code $[n, k, d]$.

2) Find fast decoding algorithms.

As we shall see in Section 4.1, these are hard problems (and this hardness can be used in cryptography, cf. Section 3.3). However, vast classes of codes with fast decoding algorithms exist (e.g., BCH or Goppa codes — see page 37), but this aspect is not part of our research.

1.1.2 Covering Codes

We are interested in the following problem: find "small" and "long" codes, with a "small" covering radius; usually, we consider $K(n, R)$, the smallest possible size K for a code $(n, K)R$, or $t[n, k]$, the smallest possible covering radius R for a code $[n, k]R$. With given codimension $m = n - k$, the entries in a table of $t[n, k]$ are on a diagonal parallel to the main diagonal. As n increases, we move down the diagonal and, typically, $t[n, n-m]$ remains constant for several consecutive values of n, then drops. These points of change signal a value of the *length function* ℓ: if $t_0 = t[n, n-m] < t[n-1, n-1-m]$, then $\ell(m, t_0) = n$: $\ell(m, R)$ is the smallest length n for which there is a binary linear code $[n, n-m]R$.

•▷ Among the works published on this topic since our Thesis, [30], [31], [62], [28], [23] (see also Section 4.1 for results from [15] and [46]), we would like to mention the following results:

In the case of codes with small length or size, we can use linear inequalities on the weight distribution, induction using the notion of "balanced" codes (having as many ones as zeros on each column), 2-surjectivity (existence of the pairs '00', '01', '10', and '11' on any two columns), and partitionings of codes allowing the construction of longer codes; for instance,

one can show that $K(2p+3,p)=7$ for any $p \geq 1$. In particular,

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

has length $2p+3$, 7 codewords, and covering radius p.

Using embedded R-error-correcting codes, with minimum distance $2R+1$, gave lower bounds on $K(n,R)$. If $A(n,d)$ denotes the largest possible size of a code with length n and minimum distance d (with the convention $A(n,d)=1$ if $d>n$), we obtain

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

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

provided the denominators are positive. Neither (1.2) nor (1.3) is always better than the other.

Using "piecewise constant" codes and a Steiner system (Ss), we built a code with length 11, covering radius 1 and size 192, showing that $K(11,1) \leq 192$. A code C is piecewise constant if: when partitioning its length n into $n = n_1 + n_2 + \ldots + n_t$ and its elements $\mathbf{c}$ into $\mathbf{c} = \mathbf{c}_1|\mathbf{c}_2|\ldots|\mathbf{c}_t$, where $|$ stands for concatenation and $\mathbf{c}_i$ has length n_i, if C contains a vector $\mathbf{c}$ such that $w(\mathbf{c}_1)=w_1, w(\mathbf{c}_2)=w_2, \ldots, w(\mathbf{c}_t)=w_t$, then it contains all

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

such vectors. A Ss $S(t,k,v)$ is a particular design: it is a set of blocks (k-subsets) of a v-set S such that any t-subset of S is contained in exactly one block. There is a Ss $S(4,5,11)$, containing 66 blocks. These 66 blocks and their complements (i.e., 66 vectors of weight 5

and 66 vectors of weight 6) cover all vectors of weights 4 to 7. Let $11 = 6 + 5$ and $(w_1, w_2) = (0,1), (0,2)$, or $(2,0)$. This is a piecewise constant code of size 30, covering all vectors of weight 3 or less, and its complement covers all vectors of weight 8 or more. This yields a code $(11, 192)1$, still the best today — on the other hand, $K(11,1) \geq 180$ (Blass and Litsyn [9]).

The notion of *normality* was created by Graham and Sloane [45] for linear codes. We generalized it to nonlinear as well as to nonbinary codes. We describe it here in the binary case. Let C be a code $(n, K)R$. For i between 1 and n, denote by $C_0^{(i)}$ (respectively, $C_1^{(i)}$) the set of codewords whose i-th component is '0' (respectively, '1'). The integer

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

is the *norm* of C with respect to i, and $N_{\min} = \min\{N^{(i)} : i = 1, 2, \ldots, n\}$ is the *minimum norm* of C (we use the convention $d(\mathbf{x}, \emptyset) = \infty$). C is called *normal* if its minimum norm is at most $2R + 1$.

Normal codes can be used for efficient constructions; in particular, the existence of a normal code $(n, K)R$ allows the construction of codes $(n + 2p, K)R + p$ for any integer p and joins up the normality of codes and the conjecture $K(n + 2, R + 1) \leq K(n, R)$ (for $R < n$) as well as its linear variation $t[n + 2, k] \leq t[n, k] + 1$ (for $n \geq k \geq 1$).

We proved the first inequality, for fixed R, when n is large enough, and studied more in detail the cases $R = 1$ and $R = 2$: we showed that $K(n + 2, 2) \leq K(n, 1)$ for all $n \geq 2$, except maybe $n = 9$ and $n = 16$ (now, it is known that it is true for all $n \geq 2$), and that $K(n + 2, 3) \leq K(n, 2)$ for all n belonging to $\{1\} \cup \{3, \ldots, 7\} \cup \{20, \ldots, 28\} \cup \{43, 44\} \cup \{91, \ldots, 127\} \cup \{187, \ldots, 361\}$ and $n \geq 379$.

We proved that a *linear* code is normal if one of the following conditions holds: its length is ≤ 12; its dimension is ≤ 2; its minimum distance is ≤ 3; its covering radius is ≤ 2 (the current records are 15, 5, 4, and 3, respectively).

No binary linear abnormal code is known.

In the nonbinary case, the straightforward generalization of normality (for each i, we define q subcodes $C_a^{(i)}$ according to the value a of the i-th component, and C is normal if its

minimum norm is at most $qR + q - 1$) is less efficient; the same is true for a generalization (*subnormality*) where we consider *any* partition of the code into q subcodes. For instance, in the binary case, all perfect codes are normal and no subnormal code is known, whereas in the q-ary case, no perfect code is subnormal.

We disproved the conjecture $t[n,k] \leq t[n+1,k+1]+1$, for $n \geq k \geq 1$. As a consequence, in a table for $\ell(m,R)$, there exist arbitrarily long sequences of values which can be expressed with only two values of $t[n,k]$.

On this vast topic, Gérard Cohen, Iiro Honkala, Simon Litsyn, and myself have written a monography, "Covering Codes" [23], published in 1997. It has xxii+542 pages, 20 chapters (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 references and 24 tables, among which a table for $K(n,R)$, $n \leq 33$ and $R \leq 10$, and a table for $t[n,k]$, $k \leq n \leq 64$. Here is an excerpt of its 5-page review in Mathematical Reviews (1999), by Professor H.F. Mattson, Jr., Syracuse University, USA:

> 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.

For complexity results on covering problems, we refer the reader to Section 4.1, where several NP- and Π_2-completeness results are stated.

1.1.3 Perfect Codes

In this subsection, we consider codes over the finite field $F_q = \{0, 1, \ldots, q-1\}$ (where q is a prime power). A code is *perfect* if $d = 2R + 1$: the spheres of radius $e = R$ fill the whole space and have pairwise empty intersections. Inequalities (1.1) meet with equality.

Two q-ary codes C_1 and C_2, with parameters (n, K), are *equivalent* if there exist n permutations $\tau_1, \tau_2, \ldots, \tau_n$ over F_q and one permutation σ of the n coordinates such that, if $c_1 c_2 \ldots c_n \in C_1$, then $\sigma\Big(\tau_1(c_1)\tau_2(c_2)\ldots\ \tau_n(c_n)\Big) \in C_2$. In the binary case, this means the existence of a vector $\mathbf{a} \in F^n$ and a permutation σ of the n coordinates such that $C_2 = \{\sigma(\mathbf{c}) + \mathbf{a}\ : \mathbf{c} \in C_1\}$.

Up to equivalence, the only nontrivial perfect q-ary codes are:
1) the binary repetition codes with odd length (length $n = 2p+1$, dimension $k = 1$, minimum distance $d = 2p + 1$, covering radius $R = p$, for any integer $p \geq 1$);
2) codes having the same parameters as the q-ary Hamming codes (length $n = (q^m - 1)/(q-1)$, size $K = q^{n-m}$, minimum distance $d = 3$, covering radius $R = 1$, for any integer $m \geq 2$);
3) the binary Golay code (length $n = 23$, dimension $k = 12$, minimum distance $d = 7$,

covering radius $R = 3$);

4) the ternary Golay code (length $n = 11$, dimension $k = 6$, minimum distance $d = 5$, covering radius $R = 2$)

(see for instance [23, Sec. 11.1 and 11.2]).

Only Case 2) can yield perfect codes which are not equivalent to linear codes. The first construction of perfect binary nonlinear codes dates back to 1962 (Vasiliev [74]). See for instance [23, Sec. 11.3 and 11.4] for a survey of other constructions, and references.

•▷ We added new constructions [63], [76] in the binary case: using generalized concatenated codes (Zinoviev [75]), we obtained a first family of constructions, for which we gave a lower bound on the number of *nonequivalent* codes [63]. Using the same ideas, we construct in [76] new perfect codes and give a lower bound on the number of *different* codes. These bounds are not the best, but our constructions can be applied to construct codes other than perfect.

•▷ Still about perfect codes, we studied the following problem [2]: consider binary extended perfect codes, i.e., with the following parameters: length $n = 2^t$ ($t \geq 2$), size 2^{n-1-t}, minimum distance $d = 4$, over F. It is known that n extended perfect codes $C_1, C_2, \ldots, C_n$, can partition $E^n \subset F^n$, the set of even vectors, and that n extended perfect codes C_{n+1}, C_{n+2}, ..., C_{2n}, can partition $O^n = F^n \setminus E^n$, the set of odd vectors. Given a second partition, D_1, D_2, ..., D_n, of E^n, and D_{n+1}, D_{n+2}, ..., D_{2n}, of O^n, we define the *intersection matrix* of the partitions C and D, $\mathbf{IM}(C, D)$, by:

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

and we try to construct different or nonequivalent intersection matrices, and to estimate their number.

Using Latin squares, we show that the number of different matrices is between 2^{cn^2} and $2^{c'n^3}$, where n is large enough and c and c' are positive constants (the number of nonequivalent matrices is of the same order of magnitude).

1.2 Arithmetic Codes

We detail the basic notions of arithmetic codes, which are less familiar even to coding theorists.

1.2.1 Weights and Distances

Arithmetic codes are designed for error detection and correction in arithmetic processors performing arithmetic operations such as addition, subtraction, complementation, shifting.

Let r ($r \geq 2$) be the radix with which we represent a positive integer I: the *radix r representation* of I is $I = \sum_i a_i r^i$, where $0 \leq a_i < r$ for all i. This representation is unique. When $a_i = 0$ for all $i \geq n$, and $a_{n-1} \neq 0$, I can be written as a n-tuple: $I = (a_{n-1}a_{n-2} \ldots a_1 a_0)$ or $I = a_{n-1}a_{n-2} \ldots a_1 a_0$.

The addition of two positive integers is performed by a set of elementary units computing the sum c_i (modulo r) from the inputs (a_i, b_i, and a carry) and a carry (see Figure 4). One error in unit i leads to a false sum (modulo r) c_i, or to a false carry, i.e., an error $\pm e_i r^i$ ($|e_i| < r$), or $\pm e_{i+1} r^{i+1}$ ($|e_{i+1}| < r$). If globally we have an error E (difference between actual and exact results), it is natural to define the weight of E as the minimum number of terms $\pm e_i r^i$ which sum up to E. Formally: a *radix r modified representation* of I (positive, negative, or zero) is any representation

$$I = \sum_i a_i r^i, \text{ where } |a_i| < r \text{ for all } i. \tag{1.4}$$

This representation is not unique. Any representation (1.4) with a minimum number of nonzero coefficients a_i is called *minimum*. A minimum representation is not unique either.

Definitions. The *arithmetic weight* of I, $W(I)$, is the number of nonzero terms in a minimum modified representation of I. The *arithmetic distance* between I_1 and I_2, $D(I_1, I_2)$, is the arithmetic weight of their difference.

Each radix r defines arithmetic weight and distance. How to compute an arithmetic weight? This is not as elementary as in the case of Hamming weight.

There are direct algorithms (Chiang and Reed [19]). But, since we'll need it in Section 5,

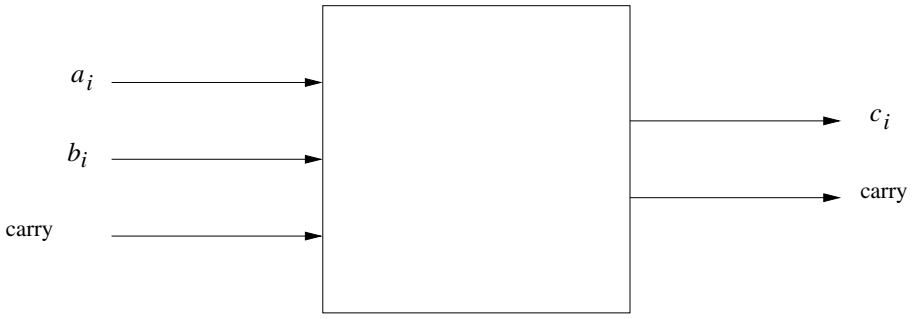

FIG. 4 – *Unit i.*

we now describe a new modified representation, which is minimum and easy to establish. The *radix r nonadjacent modified representation* (NAMR) of I is a representation $I = \sum_{0 \le i \le n} a_i r^i$, where $|a_i| < r$ for $i = 0, 1, \ldots, n$ and, for $i = 0, 1, \ldots, n-1$:

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

The word "nonadjacent" comes from the binary case, where the above condition is $a_i a_{i+1} = 0$.

The NAMR exists for all integers I, is unique, minimum, and easy to compute from the radix r representation (Clark and Liang [20]).

Let us now consider the modular addition of two elements I_1 and I_2 in the ring $Z_m = \{0, 1, \ldots, m-1\}$ $(m > 0)$:

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

If the exact result $I_1 \oplus I_2$ is J and the actual result is K, we define the *ring error* $F \in Z_m$ by $K = J \oplus F$. Compared to E defined by $K = J + E$, we have $E = F$ or $E = F - m$ depending on whether $E > 0$ or $E < 0$; hence the following definitions (Rao and Garcia [71]):

Definitions. The *Rao-Garcia modular weight* of $I \in Z_m$, $W_{RG}(I)$, is the smaller of $W(I)$ and $W(m - I)$. The *Rao-Garcia modular distance* between I_1 and $I_2 \in Z_m$, $D_{RG}(I_1, I_2)$, is the Rao-Garcia modular weight of their modular difference.

Each couple (r, m) defines a modular weight (which can be computed by comparing two arithmetic weights) and a modular distance. However, the triangle inequality is not satisfied

in all cases. It *is* satisfied in the most commonly used moduli: $m = r^n$ or $m = r^n \pm 1$. From now on, we use "weight" and "distance" even when the triangle inequality is not satisfied and "metric" when we want to insist that it is. Ernvall [38], [39], [40] gives necessary and sufficient conditions on r and m for D_{RG} to be a metric.

When D_{RG} is a metric, perfect codes can exist (see below, page 19): the triangle inequality is necessary to have two spheres of radius t, with centres at distance $2t + 1$, disjoint.

A second definition of modular weight exists, which is more recent (Clark and Liang [21]), satisfies the triangle inequality, but seems more difficult to compute.

Definitions. The *Clark-Liang modular weight* of I is $W_{CL}(I) = \min\{W(J) : J \in Z, J = I \bmod m\}$. The *Clark-Liang modular distance* between I_1 and I_2, $D_{CL}(I_1, I_2)$, is the Clark-Liang modular weight of their difference.

Each couple (r, m) defines modular weight and metric.

To our knowledge, the complexity of computing this modular weight is not mentioned anywhere (see Section 4.2).

•▷ The first problem we tackled is to determine when the Rao-Garcia and Clark-Liang modular distances coincide. We can use Ernvall's results on when D_{RG} is a metric, since D_{RG} cannot equal D_{CL} if it is not a metric. Hence, we have only to investigate the following cases: $W(m) = 1$, $W(m) = 2$, $W(m) = 3$ and the NAMR of m has one among 22 possible forms, or $W(m) = 4$ and the NAMR of m has one among 10 possible forms; in the binary case, this reduces to $W(m) = 1$, $W(m) = 2$, or $W(m) = 3$ and the NAMR of m is $2^n + 2^{n-2} \pm 2^i$ $(i \leq n - 4)$ or $2^n - 2^j \pm 2^i$ $(n - 5 \leq j \leq n - 2, i \leq j - 2)$.

Depending on r and m, we obtained a quasi-complete characterization [53], [54]. Only one subcase of one of the possible 22 forms (for $W(m) = 3$) was not totally solved. For $r \leq 13$, the characterization is complete; in particular:

•▷ When $r = 2$, the two modular distances D_{RG} and D_{CL} coincide if and only if $W(m) \leq 2$.

We note $D_{CL} < D_{RG}$ when they do not coincide.

1.2.2 Arithmetic Codes

Arithmetic codes, designed for error correction in operations on integers, represent these integers with *redundancy*; if we perform modular additions, we code the integers $0, 1, 2, \ldots, B-1$ by multiplying them by an integer A: code C contains the integers $0, A, 2A, \ldots, (B-1)A$. Letting $m = AB$, we can check and correct the addition modulo m of two codewords AI_1 and AI_2: their modular sum $AI_1 \oplus AI_2$ is $A(I_1 + I_2)$ if $I_1 + I_2 < B$ and $A(I_1 + I_2 - B)$ if $I_1 + I_2 \geq B$; it is a multiple of A between 0 and $A(B-1)$, i.e., a codeword. Error correction is to search for the codeword closest to the actual result $I = AI_1 \oplus AI_2 \oplus F$; error detection can be done by dividing I by A: the remainder, depending only on F, is the *syndrome* of I. If it is zero (no error, or an error multiple of A), we conclude that the result is correct; if not, an error is detected.

The ring Z_m represents the set of all possible, maybe wrong, results, $C \subseteq Z_m$ the set of correct results; the integers $0, 1, \ldots, B-1$ represent the *information*, and A is called the *generator* of the code.

The codes $\{0, A, 2A, \ldots, (B-1)A\}$ are called AN-*codes*. But more generally, just as linear block codes are vector subspaces and nonlinear codes are simple subsets, we can define an arithmetic code C as a subset of Z_m. All classical problems in coding arise for arithmetic codes, but seem more difficult to solve.

●▷ The second problem we tackled is the existence of perfect arithmetic codes (cf. Section 1.1.3). Given r and m, recall that an e-error-correcting code (with minimum distance $d = 2e + 1$) $C \subseteq Z_m$ is perfect if and only if

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

where $V(e)$ is the volume of the sphere of radius e (which does not depend on the centre and is equal to $|\{y \in Z_m \ : W_{CL}(y) \leq e\}|$ or $|\{y \in Z_m \ : W_{RG}(y) \leq e\}|$). Observe that for an AN-code $C \subseteq Z_m$, $d = D_{RG}(C) = \min\{D_{RG}(x, y) \ : x \in C, y \in C, x \neq y\} = \min\{W_{RG}(x) \ : x \in C, x \neq 0\} = \min\{W(x) \ : x \in C, x \neq 0\}$, and that a perfect e-error-correcting AN-code has generator $V(e)$: $C = \{0, V(e), 2V(e), \ldots, (|C|-1)V(e)\}$ (this holds also for D_{CL}).

The answer to this problem is far from being as complete as for block codes.

We first consider the Rao-Garcia modular distance in the binary case. We saw (page 18) that it is a metric if and only if the NAMR of m is: *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] established the following facts for single-error-correcting AN-codes:

In Case *f1*, no perfect code exists. In Case *f2*, a necessary condition for the existence of perfect codes is that $j = 0$ ($m = 2^n \pm 1$), and this is a well-known class of perfect codes, the Brown-Peterson codes (see, e.g., Rao [70]). Cases *f3* and *f4* also yield many perfect codes.

In the ternary case, there is an infinite family of perfect codes, given by $m = 3^n - 1$, $n = 2e+1$, $C = \{0, m/2\}$ (Gordon [44]). These sorts of repetition codes are e-error-correcting and are the only known nontrivial perfect codes correcting more than one error.

●▷ We found two new perfect ternary single-error-correcting codes [58]:

The AN-codes with generator 37 and moduli $m_1 = 3^9 - 2 \cdot 3^7 + 3^2$ and $m_2 = 3^9 + 1$ are perfect single-error-correcting.

I give the proof, which provides an insight into the arithmetic techniques used in this field. Let $C_1 = \{0, 37, 74, \ldots, 15281\}$ (C_1 has 414 words) and $C_2 = \{0, 37, 74, \ldots, 19647\}$ (C_2 has 532 words). Moduli m_1 and m_2 are such that D_{RG} is a metric, and in both cases, the volume of the sphere of radius one is 37: for m_1 for instance, the integers of modular weight zero or one are: $0, 1, 15317, 2, 15316, 3, 15315, 6, 15312$, etc. We have to show that C_1 and C_2 are 1-error-correcting. Let x be the smallest positive integer such that $37x$ has arithmetic weight < 3 (i.e., $= 2$). Let $37x = a \cdot 3^i + \varepsilon b \cdot 3^j$ ($0 \leq j < i, a = 1$ or $2, b = 1$ or $2, \varepsilon = \pm 1$). By Gauss' theorem, 3^j divides x, whence $37 \cdot \dfrac{x}{3^j} = a \cdot 3^{i-j} + \varepsilon b$, and $W(37 \cdot \dfrac{x}{3^j}) = 2$. Then $j = 0$: $a \cdot 3^i = -\varepsilon b \bmod 37$. Listing the first powers of 3 modulo 37 ($3, 9, 27, 7, 21, 26, 4, 12$ and 36) and their doubles ($6, 18, 17, \ldots$) shows that x is given by $37x = 3^9 + 1 > \max\{m_1 - 37, m_2 - 37\}$. So all codewords have arithmetic weight at least three, which ends the proof.

In conclusion of this very partial study in the case of the Rao-Garcia modular metric, observe that no perfect code is known for $r > 3$, and that the Brown-Peterson codes (for $m = 2^n \pm 1$), or the ternary repetition codes (for $m = 3^n - 1$), are also perfect codes for the Clark-Liang

modular metric, since in these cases the two modular metrics coincide.

•▷ One can ask, when $D_{CL} < D_{RG}$ and a perfect code $C \subseteq Z_m$ exists for the Rao-Garcia metric, if the same code C can be perfect with respect to the Clark-Liang metric. We proved that the answer is no for 1-error-correcting codes [55] and conjecture that it is no in all cases.

No perfect code is known for D_{CL} when $D_{CL} < D_{RG}$.

To see the hardness of the problem, consider the volume of the sphere, which, by (1.5), gives a *necessary* condition for the existence of perfect codes: a code $C \subseteq Z_m$ can be perfect e-error-correcting only if $V(e)$ divides m.

Now, even in the case of the Rao-Garcia metric, *we do not know the volume of the sphere in all cases*. From partial results (see Ernvall [41], [42]), inexistence results or parameters for candidates can be established, for instance (Gordon [44]):

For $m = r^n \pm 1$, for $e = 2$ and $V(e) < 2^{41}$, or $e = 3$ and $V(e) < 2^{50}$, the only perfect e-error-correcting AN-codes are the ternary codes $\{0, (3^5 - 1)/2\}$ and $\{0, (3^7 - 1)/2\}$, which are 2- and 3-error-correcting, respectively (these are the aforementioned repetition codes).

•▷ We obtained the following results [58]:

1) When $r = 2$, for $m < 2^{33} + 2^{31} - 1$ and $e \geq 2$, no perfect AN-code exists.
2) When $r = 3$, for $m < 2 \cdot 3^{27} - 3^{26} - 2$ and $e \geq 2$, the only perfect AN-codes are the ternary repetition codes $\{0, (3^{2e+1} - 1)/2\}$.

Many cases of divisibility of m by $V(e)$ can be ruled out for AN-codes (by showing that the generator $V(e)$ or one of its multiples is of weight $2e$ or less) but codes other than AN remain candidates.

•▷ Then we managed only to rule out the small cases, and we found no such perfect codes (see [58], [59], and previous works mentioned there).

In conclusion, little is known, and we would need stronger algebraic or arithmetic tools, such as the Lloyd theorem for the q-ary Hamming space.

•▷ Finally, third problem concerning arithmetic codes, we studied the asymptotic behaviour of arithmetic binary codes [49], when $m = 2^n$ and $m = 2^n \pm 1$. The usual techniques for

error-correcting and covering codes lead to bounds of Hamming and Varshamov-Gilbert type: let $M_a(n,d)$ be the maximum size of an arithmetic code with minimum distance d, $R_a = R_a(n,d) = \frac{1}{n}\log_2 M_a(n,d)$ the *rate* of such a code, $\delta = d/n$ its *normalized distance*, and H_2 the binary entropy.

When using $f(n) \lesssim g(n)$ when n goes to infinity, we mean that $f(n) \leq g(n)\big(1+\varepsilon(n)\big)$, where $|\varepsilon(n)|$ tends to 0 when n tends to infinity.

Then the following asymptotic inequalities hold (Kabatianski [48]):

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

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

when n goes to infinity.

Observe that the Varshamov-Gilbert bound guarantees the existence of codes with non-zero asymptotic rate for $\delta < 1/3$ (see Figure 5). On the other hand, the Hamming bound shows that the rate tends to zero when $\delta \geq 2/3$. This can be immediately improved ($1/2$ instead of $2/3$), since an arithmetic weight is at most $(n+1)/2$.

Another simple remark is that the arithmetic weight of an integer x is at most the Hamming weight of its binary representation. Hence, one can apply any upper bound on block codes in Hamming space, in particular the McEliece-Rodemich-Rumsey-Welch bound. We managed however to find an upper bound which is better than the application of the McEliece-Rodemich-Rumsey-Welch bound to arithmetic codes.

●▷ Our asymptotic bound reads:

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

$$\text{where } \rho = \frac{2}{3} - \sqrt{\frac{4}{9} - \frac{2}{3}\delta},\ \delta = d/n, \text{ and } n \text{ goes to infinity,}$$

see Figure 5. The technique is the following: we use the Bassalygo-Elias lemma, bounding above the best density of a code by means of the best density in a subspace, here the set of

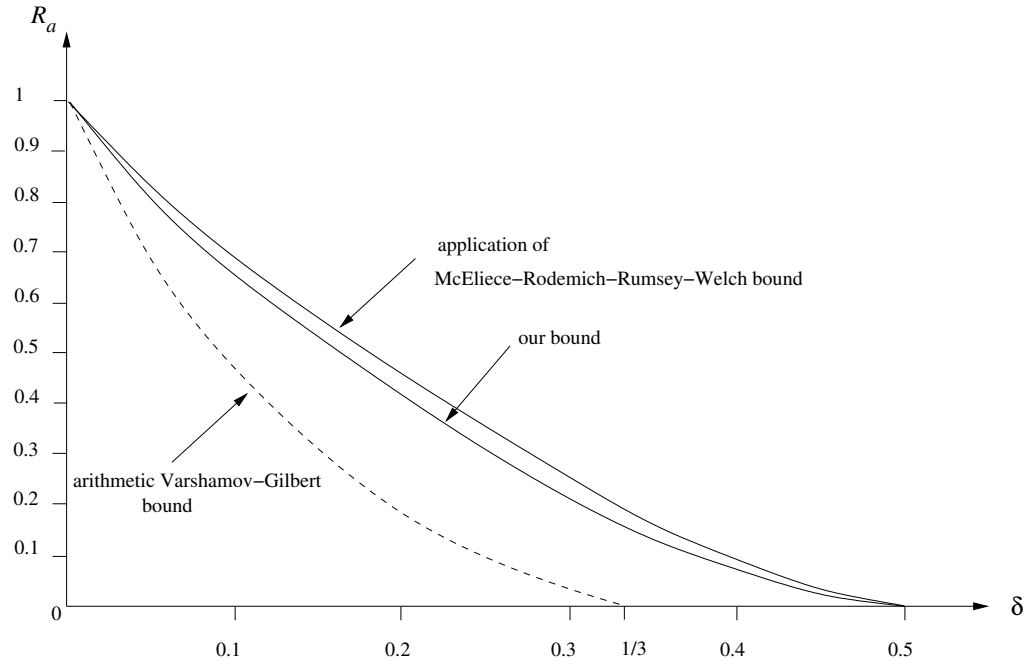

FIG. 5 – *Asymptotic behaviour of the rate, R_a: lower bound, upper bounds.*

integers with given arithmetic weight w. Since we do not have an analogue of the Johnson bound, we actually consider a larger subspace, the set of ternary vectors with Hamming weight w. Next we use the Johnson bound for ternary codes, together with the fact that the arithmetic distance between two integers is at most the Hamming distance between the ternary vectors of their nonadjacent modified representations in radix 2.

When $r > 2$, we could not improve on the simple application to arithmetic codes of upper bounds from block codes.

1.3 Identifying Codes

Identifying codes are new (Karpovsky, Chakrabarty, and Levitin [51], 1998) and can be seen as an extension of the theme of covering codes; given an integer t and an undirected, connected, finite or infinite, graph $G = (V, E)$, with the shortest path distance d, we define $B_t(u)$, the sphere of centre $u \in V$ and radius t, as in the Hamming space, which, in terms of

graphs, is the n-cube:

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

Similarly, a vertex u t-covers (or covers if there is no ambiguity) all vertices in $B_t(u)$. We usually deal with graphs for which the volume of the spheres does not depend on the centre, and note $V(t)$ this volume.

Codes are subsets of V and their elements are codewords; a t-covering or covering code $C \subseteq V$ is such that the sets $B_t(v) \cap C$, $v \in V$, are all nonempty. Then C is *t-identifying* or *identifying* if moreover these sets $B_t(v) \cap C$ are distinct. The set of codewords covering $v \in V$ is the *identifying set* of v.

We search for the smallest density, $D(G,t)$, of a t-identifying code in G. The graph can be the n-cube, or the square, triangular, or hexagonal grids, with possible applications to processor networks where we want to spot a malfunctioning element.

General lower bounds can be established — assuming that the volume of the spheres of radius t is $V(t)$: if C is identifying, it is also covering, hence the second inequality in (1.1) holds:

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

Using the identifying condition, we can improve on this bound: let L_1 be the set of vertices of V identified by a singleton in C; then $|V| - |L_1|$ vertices have identifying sets of size at least two. Since $|C| \geq |L_1|$, we have $|C| \cdot V(t) \geq 2(|V - |L_1|) + |L_1| = 2|V| - |L_1| \geq 2|V| - |C|$:

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

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

If (1.6) holds with equality, C is *perfect*. For instance, in a graph G consisting of a cycle with six vertices, three pairwise nonadjacent vertices form a perfect code.

●▷ We proved [26] that there is no perfect nontrivial code for $t > 1$.

One can improve on (1.6) or (1.7) in two ways: general methods, valid for all t — we detail an example in Section 1.3.2 —, or *ad hoc* methods, for small t (actually, $t = 1$).

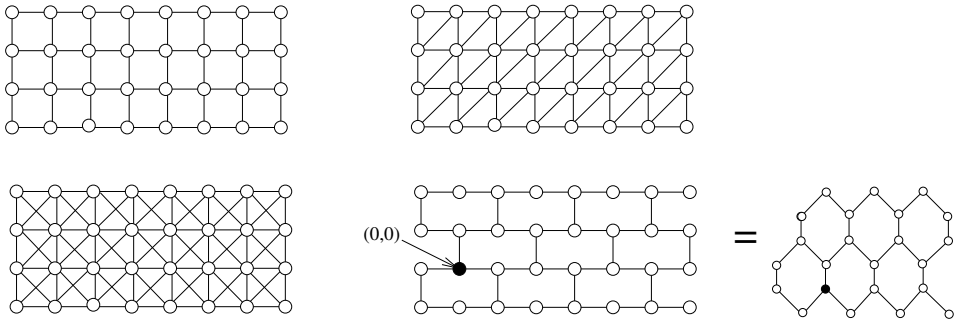

FIG. 6 – *Fragments of our four infinite two-dimensional graphs.*

Upper bounds on $D(G,t)$ are by construction, either general — see an example in Section 1.3.3 —, or specific, for small values of t (see Figure 7). In this case, they are obtained either "by hand", like in Figure 7, or by combinatorial optimization heuristics (see page 44).

Before studying four particular graphs, let us mention that the decision problem corresponding to the search for a t-identifying code, of bounded size, in a graph, is NP-complete for all t (see Section 4.3 for more development about this result).

Observe that the following four graphs are *infinite*. The constructions of identifying codes will be *periodic* and described in a simple way.

1.3.1 The Square Grid

The infinite two-dimensional square grid, G_S, has vertex set $V = Z \times Z$ and edge set

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

(see Figure 6).

•▷ We proved the following bounds [24], [22], [26], [47], [16], [12]:

$$15/43 \leq D(G_S,1) \leq 0.35\,;$$

$$D(G_S,2) \leq 5/29\,;$$

$$D(G_S,t) \geq \frac{3}{8t+4}\,; \tag{1.8}$$

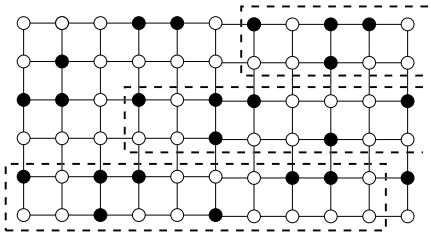

FIG. 7 – *A* 1*-identifying periodic code, with density* 0.35*, in the infinite square grid. Codewords are in black.*

$$D(G_S, t) \leq \frac{2}{5t}, \text{ for } t \text{ even;}$$
$$D(G_S, t) \leq \frac{2t}{5t^2 - 2t + 1}, \text{ for } t \text{ odd.}$$

The periodic construction of Figure 7 shows that $D(G_S, 1) \leq 0.35$. We conjecture that $D(G_S, 1) = 0.35$.

For t between 3 and 6, t-identifying codes were constructed using heuristics.

Observe that the lower bound (1.8) is in $1/t$, whereas (1.7) is in $1/t^2$, since the sphere of radius t is in t^2. In the other three graphs, we also have bounds in $1/t$, instead of $1/t^2$.

1.3.2 The Triangular Grid

The infinite two-dimensional triangular grid, G_T, has vertex set $V = Z \times Z$ and edge set

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

(see Figure 6). When $t = 1$, (1.7) holds with equality: there is a perfect code, with density 0.25 (Karpovsky, Chakrabarty, and Levitin [51]).

●▷ We proved the following bounds [26], [16], [12]:

$$D(G_T, t) \geq \frac{2}{6t + 3};$$
$$D(G_T, t) \leq \frac{1}{2t + 4}, \text{ for } t = 0 \bmod 4;$$
$$D(G_T, t) \leq \frac{1}{2t + 2}, \text{ for } t = 1, 2 \text{ or } 3 \bmod 4.$$

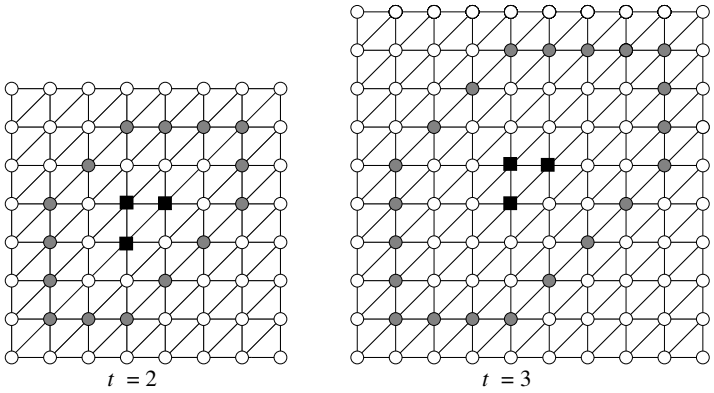

FIG. 8 – *In grey, the vertices belonging to* $H_t(x, y, z)$.

For t between 2 and 6, t-identifying codes were constructed using heuristics.

To give an insight into the ideas used for general lower bounds, I give the sketch of the proof of $D(G_T, t) \geq \dfrac{2}{6t+3}$.

We call *triangle* any triple (x, y, z) such that there exist $i \in Z$ and $j \in Z$, with $x = (i, j)$, $y = (i, j+1)$ and $z = (i+1, j+1)$. Let $H_t(x, y, z) = \Delta_t(x, y) \cup \Delta_t(x, z) \cup \Delta_t(z, y)$, where $\Delta_t(x, y)$ is the symmetric difference of the spheres of radius t centred at x and y (see Figure 8).

It is easy to see that $|H_t(x, y, z)| = 6t+3$, and that, if C is t-identifying, then $|H_t(x, y, z) \cap C| \geq 2$. Arguments using translations and tilings of the infinite plane, similar to the Bassalygo-Elias lemma (cf. page 22), imply that the density is at least $2/(6t+3)$.

1.3.3 The King Grid

The infinite two-dimensional king grid, G_K, has vertex set $V = Z \times Z$ and edge set

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

(see Figure 6). Its name comes from the fact that, on an infinite chessboard, the sphere of radius t is the set of squares that a King can reach in at most t moves, starting from the centre.

•▷ We obtained the *exact* value of $D(G_K, t)$ for *all* values of t [27], [26], [16], [13]:

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

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

The construction yielding the upper bound $1/4t$ for all t is easy to describe:

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

It is less easy to prove that C is indeed t-identifying. But it is still less easy to prove that $1/4t$ is also, for $t > 1$, the lower bound on $D(G_K, t)$.

1.3.4 The Hexagonal Grid

The infinite two-dimensional hexagonal grid, G_H, has vertex set $V = Z \times Z$ and edge set

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

(see Figure 6).

•▷ We proved the following bounds [33], [25], [26], [16], [12]:

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

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

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

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

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

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

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

For t between 2 and 8, t-identifying codes were constructed using heuristics.

2 Basic Facts in Complexity

Our goal is to give an intuitive approach of *completeness* in the *polynomial hierarchy*.

We deal only with *decision problems*, consisting of an instance and a question whose answer is YES or NO. An algorithm A solves a problem π if, applied to any instance I of π, it gives the correct answer. An estimation of the *size* of an instance I of π is given by any "reasonable" encoding of I (for instance, a reasonable encoding of an integer m requires $\log m$ bits; we'll see however (page 41), about linear codes, the paradoxical results induced by this notion of size). The *time complexity function* of an algorithm A solving π is, for each possible instance size, the *maximum* time required by A to solve an instance of that size. A *polynomial(-time)* algorithm is one whose time complexity function can be bounded by a polynomial $p(n)$, where n is the size of the instance we consider. The class of polynomial-time solvable problems is denoted by P.

A *polynomial reduction* from a problem π_1 to a problem π_2 is a polynomial construction mapping any instance of π_1 into an equivalent instance of π_2 (the answer is the same for both instances). Thus, such a transformation provides the means for converting any polynomial algorithm solving π_2 into a corresponding polynomial algorithm solving π_1.

Next, we introduce the class NP: a decision problem belongs to NP if it can be solved by a *nondeterministic polynomial(-time) algorithm*, i.e., an algorithm constiting in two stages: a *guessing stage* and a polynomial-time *checking stage*. The first stage provides a structure s. The second stage is deterministic and correctly answers YES or NO. For instance, consider the well-known Travelling Salesman (TS) problem, for which the instance is a set of cities, the set of integer distances between the cities, and an upper bound B, and the question is whether there exists a Hamiltonian cycle of length at most B; the guessing stage provides a sequence s of cities and the checking stage checks in polynomial time if s is a Hamiltonian cycle of length at most B.

For a set S of problems, let coS be the set of problems that are complementary to those of S (their answers are reversed). We have $\mathrm{P} = \mathrm{coP} \subseteq \mathrm{NP} \cap \mathrm{coNP}$, but membership of NP does not seem to imply membership of coNP (see Figure 9). For instance, the complement

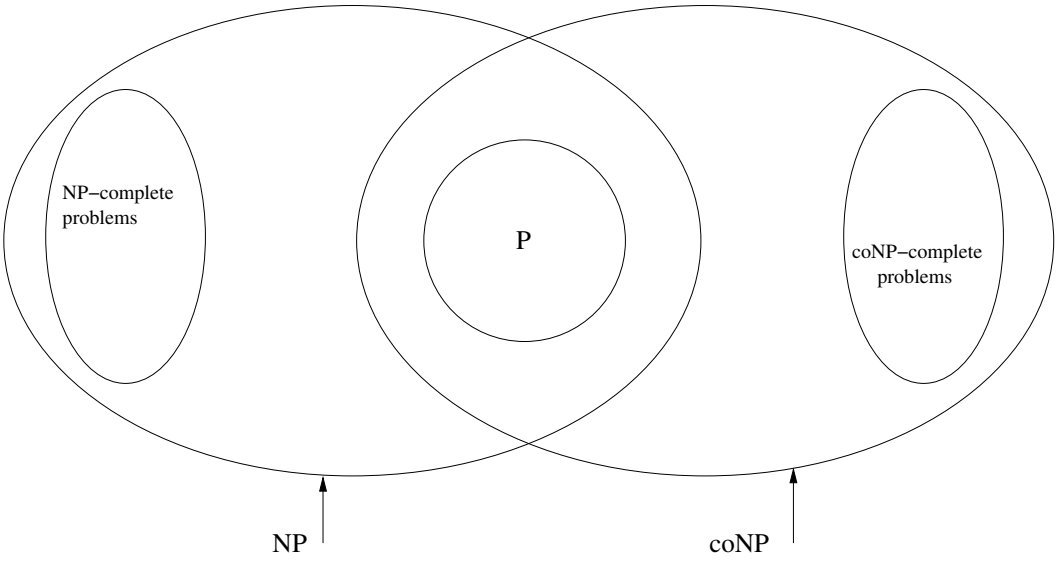

FIG. 9 – *Complexity classes, if NP ≠ coNP.*

of TS is to determine whether *all* Hamiltonian cycles have length at least $B + 1$, and there is no known way to verify a YES answer short of examining a very large proportion of all possible Hamiltonian cycles, which is not known to be achievable in polynomial time.

Among problems in NP, some have the property that all other problems in NP can be polynomially reduced to them. We denote by NP-C this class and call NP-*complete* its members. If one problem in NP-C could be solved in polynomial time, then so could every problem in NP, and P would be equal to NP. The question "P=NP?" is still open. The NP-complete problems can be seen as the most difficult in NP. For instance, TS is NP-complete (Karp [50]), and so is 3-satisfiability (3-SAT) (Cook [35]), for which the instance is a set of variables and a set of clauses containing exactly three different literals (a literal is either a variable x_i or a negated variable $\overline{x}_j$), and the question is whether there exists a truth assignment to the variables such that each clause has at least one true literal (in other words, can the Boolean formula E be satisfied, if $E = \mathcal{C}_1 \wedge \mathcal{C}_2 \wedge \ldots \wedge \mathcal{C}_m$, each clause $\mathcal{C}_i = x_{i_1} \vee x_{i_2} \vee x_{i_3}$ for $i = 1, 2, \ldots, m$, and x_{i_1}, x_{i_2}, and x_{i_3} are three distinct literals? Such an expression for E is called its *conjunctive normal form*).

Some problems might be harder than the NP-complete problems and classes of problems

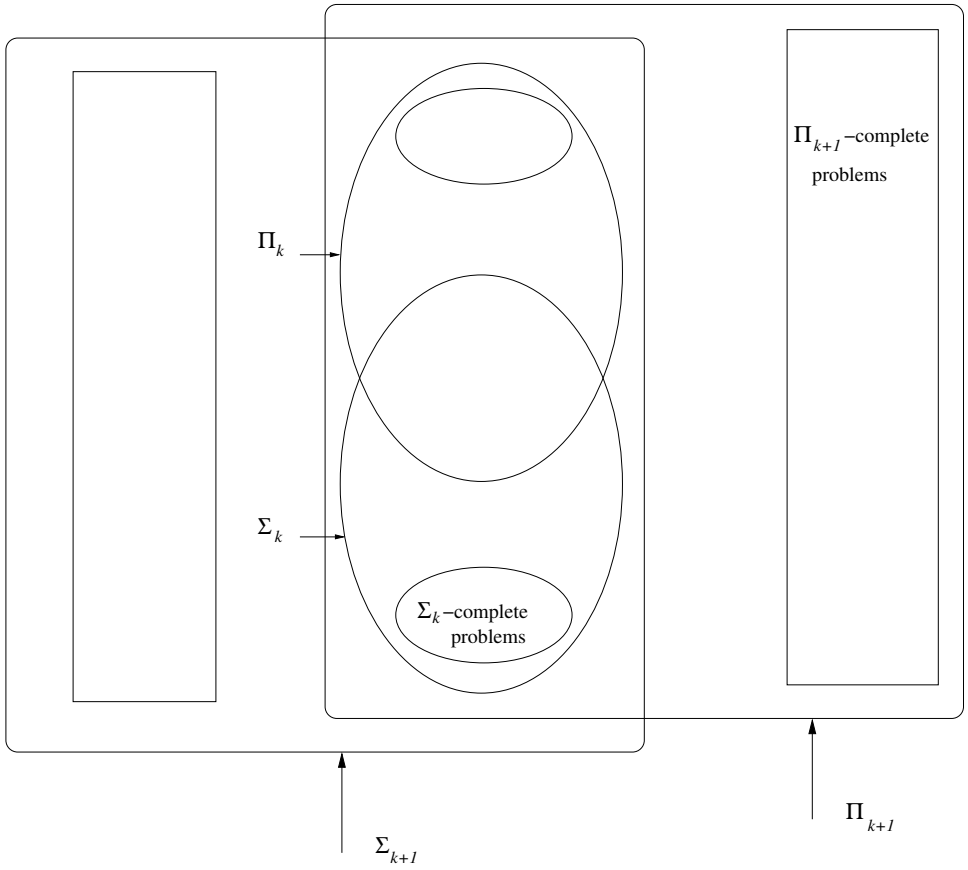

FIG. 10 – *Complexity classes, if* $\Sigma_{k+1} \neq \Pi_{k+1}$, $k \geq 1$.

of increasing *apparent* difficulty can be defined, which form the *polynomial hierarchy*. The notion of completeness can be extended inside these classes: a problem π belonging to a class S of the polynomial hierarchy is S-*complete* if every problem in S can be polynomially reduced to π.

In particular, the polynomial hierarchy contains classes denoted by Π_0, Π_1, ..., Π_k, ..., and Σ_0, Σ_1, ..., Σ_k, ..., with the following properties: $\Pi_0 = \Sigma_0 = \mathrm{P}$, $\Sigma_1 = \mathrm{NP}$, $\Pi_1 = \mathrm{coNP}$, $\Pi_k = \mathrm{co}\Sigma_k$, $\Sigma_k \cup \Pi_k \subseteq \Sigma_{k+1} \cap \Pi_{k+1}$ (see Figure 10).

Roughly speaking, a problem is in Σ_k if it can be solved by a nondeterministic polynomial algorithm with access to an *oracle* (a subroutine) that provides, *in one step of computation*, solutions to a problem in Σ_{k-1}. Another rather informal characterization of Σ_k is to represent the instance of a problem π by a string z; now $\pi \in \Sigma_k$ if and only if $\pi = \{z : \exists y_1 \forall y_2 \ldots$

$Qy_kR(z, y_1, y_2, \ldots, y_k)\}$, where the quantifiers alternate, Q stands for $\forall$ (respectively, $\exists$) if k is even (respectively, odd), R is a polynomial-time recognizable relation, and the lengths of the strings y_1, y_2, ..., y_k are polynomially bounded by the length of the string z. The same characterization holds for Π_k, with the alternating quantifiers $\forall\exists\forall\ldots$. The following problem is Π_k-complete (Meyer and Stockmeyer [67]):

Name: $\forall_1\exists_2\forall_3\ldots Q_k$-3-satisfiability, where the quantifiers alternate and Q stands for $\forall$ (respectively, $\exists$) if k is odd (respectively, even).

Instance: k integers $m_1, m_2, \ldots, m_k$, a quantified Boolean expression $\forall u_{1,1}\ldots\forall u_{1,m_1}\ \exists u_{2,1}\ldots\exists u_{2,m_2}\forall u_{3,1}\ldots\forall u_{3,m_3}\ldots Qu_{k,1}\ldots Qu_{k,m_k}E$, where E is in conjunctive normal form, each clause contains exactly three distinct literals, and the quantified variables are all the variables of E.

Question: Is it true that for every truth assignment to $u_{1,1}, \ldots, u_{1,m_1}$, there exists a truth assignment to $u_{2,1}, \ldots, u_{2,m_2}$, such that for every truth assignment to $u_{3,1}, \ldots, u_{3,m_3}, \ldots$, E is satisfied?

To prove that a problem π is S-complete, we have to check that it belongs to S, and that every problem in S can be polynomially reduced to π. For the second step, it is sufficient to prove that some known S-complete problem π_0 is polynomially reducible to π, since all problems in S are polynomially reducible to π_0 and the reduction process is transitive.

Completeness results are *conditional*; for example, the NP-completeness of a problem π means that a polynomial algorithm solving π exists if and only if P=NP. Analogously, for $k \geq 1$, the Σ_k-completeness of π implies that $\pi \in \Sigma_k \setminus \Sigma_{k-1}$, unless $\Sigma_k = \Sigma_{k-1}$. It is not known whether the polynomial hierarchy is finite or infinite. The first alternative occurs if P=NP; it also occurs if for some $k_0 \geq 1$, $\Sigma_{k_0} = \Pi_{k_0}$, since it can be shown that this would imply that for all $k \geq k_0$, $\Sigma_k = \Pi_k = \Pi_{k_0}$.

It is widely believed that P$\neq$NP, i.e., that no polynomial algorithm exists for NP-complete problems.

When faced to NP-complete problems (or even higher in the hierarchy), one can use heuristics such as simulated annealing, genetic algorithms, taboo search, or noising (see

Sections 4.1, page 41, and 4.3, page 44).

•▷ On the theory of complexity and its formal tools (problems and languages, reasonable encodings, size of a problem, classes of problems, completeness, reductions, deterministic and nondeterministic Turing machines, short certificate, class NP, strongly NP-complete problems, pseudo-polynomial problems, oracle, polynomial hierarchy, ...), as well as its applications to block codes, cryptography, and vector quantization — see also Sections 3 and 4 — Jean-Pierre Barthélemy, Gérard Cohen, and myself have written a book, "Algorithmic Complexity and Communication Problems" [6] (in French), published in 1992. It has xxxviii+228 pages, six chapters (1. Problems and languages 2. Machines, languages and problems, classes P and NP 3. NP-hard problems and languages 4. Complexity and coding 5. Complexity and cryptology 6. Vector quantization) and was shortly reviewed in 1993 by Professor Cristian Calude (Auckland University, New-Zealand) in 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.

Our book was translated into English (published in 1996 [7]).

3 Three Cryptosystems

Here we shall describe shortly three *public-key* cryptosystems which will appear again in the next sections. Public-key cryptography was born in 1976 (Diffie and Hellman [37]). It does not require the exchange of a key on a secure channel, but needs *trapdoor one-way functions*, i.e., functions which are easy to compute and hard to invert, *unless one has additional information*, called the trapdoor of the system. Such functions were designed only in 1978, and we shall now describe three of them, the RSA, the knapsack and the McEliece.

3.1 The RSA Cryptosystem

The name comes from its designers, Rivest, Shamir, and Adleman [72]. A user B wishing to receive private messages chooses two large primes p and q and computes $n = pq$. The set of plaintext messages $\mathcal{M}$ and of ciphertext messages $\mathcal{C}$ is the set of integers from 0 to $n-1$. Knowing p and q, it is elementary to find two positive integers e and d such that for any integer $M \in \mathcal{M}$, $M^{de} = M^{ed} = M \bmod n$.

Indeed, it suffices to choose e between 2 and n, coprime with $(p-1)(q-1)$, and to compute d (using Euclid's algorithm) such that $ed = 1 \bmod (p-1)(q-1)$. Hence, inverting a modular exponentiation, to the left or to the right, is computing another modular exponentiation, with the exponents satisfying $xx' = 1 \bmod (p-1)(q-1)$.

Are public e and n, are secret d, p, and q, the trapdoor. Any A knowing e and n can send to B a message $M \in \mathcal{M}$, whose privacy is protected: A computes the ciphertext message $C \in \mathcal{C}$, $C = M^e \bmod n$. The receiver, with the secret key d, computes $C^d = (M^e)^d = M^{ed} = M \bmod n$.

The cryptanalyst intercepting C faces the following problem: he knows that $C = M^e \bmod n$ where C, e, and n are known but M unknown; he knows that $M = C^d \bmod n$ where C and n are known but M and d unknown. The two problems, retrieving M from $M^e \bmod n$ (extraction of the e-th root modulo n) without factoring n, or retrieving d without factoring n, have had no satisfactory solution since the introduction of RSA, and it is believed that the security of RSA is based on the hardness of factoring large integers.

RSA requires fast modular exponentiation, since this operation both enciphers and deciphers. See Section 5 for methods improving the speed of modular exponentiations.

3.2 The Knapsack Cryptosystem

Several cryptosystems use the following NP-complete decision problem (Karp [50]):

Name: KNAPSACK.

Instance: $n+1$ strictly positive integers $a_1, a_2, \ldots, a_n, S$.

Question: Are there n numbers x_i ($x_i = 0$ or 1) such that $\sum_{1 \le i \le n} a_i x_i = S$?

We describe the first one, by Merkle and Hellman [66]. A user B wishing to receive private messages chooses two large numbers m and w, coprime (so that there exists an integer w' such that $ww' = 1 \bmod m$). He chooses a *superincreasing* sequence $\mathbf{a}$ (of large length) of integers $a_1, a_2, \ldots, a_n$: for all i between 2 and n, $a_i > \sum_{1 \le j \le i-1} a_j$. Finally, he chooses a permutation σ over $\{1, 2, \ldots, n\}$. He "scrambles" the superincreasing knapsack, transforming it into a knapsack $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n)$ defined by $a'_i = w a_{\sigma(i)} \bmod m$ for $i = 1, 2, \ldots, n$.

The knapsack $\mathbf{a}'$ is public, the knapsack $\mathbf{a}$, the permutation σ, and the integers w and m, are the secret trapdoor. The set of plaintext messages $\mathcal{M}$ is the set of binary vectors of length n, the set of ciphertext messages $\mathcal{C}$ is the set of integers from 0 to $\sum_{1 \le i \le n} a'_i$ (or the set of binary vectors of length $\lfloor \log_2(\sum_{1 \le i \le n} a'_i) \rfloor + 1$). Any A knowing $\mathbf{a}'$ can send to B a message $\mathbf{M} = (M_1, M_2, \ldots, M_n) \in \mathcal{M}$, whose privacy is protected: A computes the ciphertext message $C \in \mathcal{C}$, $C = \sum_{1 \le i \le n} M_i a'_i$. The receiver, with the secret key $\mathbf{a}$, σ, w, and m, computes $Cw' = \sum_{1 \le i \le n} M_i a'_i w' = \sum_{1 \le i \le n} M_i a_{\sigma(i)} \bmod m$. If m is larger than $\sum_{1 \le i \le n} a_i$, then $Cw' = \sum_{1 \le i \le n} M_i a_{\sigma(i)}$, and B can retrieve $\mathbf{M}$, using the superincreasing property.

The cryptanalyst intercepting C must solve an apparently arbitrary instance of KNAPSACK.

3.3 The McEliece Cryptosystem

This system, designed by McEliece in 1978 [64], is based on the following problem, and shows a direct link between coding and cryptography:

Name: Linear Decoding (LD).

Instance: A binary matrix $\mathbf{H}$, a binary vector $\mathbf{y}$, an integer w.

Question: Is there a binary vector $\mathbf{x}$, of weight at most w, such that $\mathbf{xH}^T = \mathbf{y}$?

This problem is NP-complete (Berlekamp, McEliece, and van Tilborg [8] — cf. Section 4.1). Let $n = 2^m$ and t be an integer. A user B wishing to receive private messages builds a generator matrix $\mathbf{G}$, of dimensions $k \times n$, of a binary Goppa code, C_{Goppa}, of parameters $[n, k \geq n - mt, d \geq 2t + 1]$. He chooses a "scrambling" matrix $\mathbf{S}$, nonsingular, of dimensions $k \times k$: computing $\mathbf{SG}$ produces combinations of rows of $\mathbf{G}$. He chooses a second scrambling matrix $\mathbf{P}$, a permutation matrix of dimensions $n \times n$: computing $\mathbf{G}' = (\mathbf{SG})\mathbf{P}$ permutes the columns of $\mathbf{SG}$.

Are public $\mathbf{G}'$ and t, are secret $\mathbf{G}$, $\mathbf{S}$, and $\mathbf{P}$, which are the trapdoor. The set of plaintext messages $\mathcal{M}$ is the set of binary vectors of length k, the set of ciphertext messages $\mathcal{C}$ is the set of binary vectors of length n. Any A knowing $\mathbf{G}'$ and t can send to B a message $\mathbf{M} \in \mathcal{M}$, whose privacy is protected: A computes $\mathbf{C} \in \mathcal{C}$, $\mathbf{C} = \mathbf{MG}' + \mathbf{E}$, where $\mathbf{E}$ is a random vector of length n and weight t (vector $\mathbf{E}$, which is not part of the trapdoor, is kept secret by the sender). The receiver, with the secret key $\mathbf{G}$, $\mathbf{S}$, and $\mathbf{P}$, computes $\mathbf{CP}^{-1} = \mathbf{MG}'\mathbf{P}^{-1} + \mathbf{EP}^{-1} = (\mathbf{MS})\mathbf{G} + \mathbf{EP}^{-1}$, where, because $\mathbf{P}$ is a permutation matrix, $\mathbf{EP}^{-1}$ has the same weight as $\mathbf{E}$. A fast decoding algorithm (linear in n) for C_{Goppa} is applied to vector $\mathbf{CP}^{-1}$ in order to obtain $\mathbf{MS}$, since the weight of $\mathbf{EP}^{-1}$ is within correction capacity. The receiver multiplies to the right by $\mathbf{S}^{-1}$ and gets $\mathbf{M}$.

The cryptanalyst intercepting $\mathbf{C}$ must solve an apparently arbitrary instance of LD.

Note that if vector $\mathbf{E}$ has weight less than t, the McEliece system can be used combining both error correction and privacy: during the transmission of $\mathbf{C}$, an error vector $\mathbf{E}'$ from the channel can be added to $\mathbf{E}$; as long as the weight of $\mathbf{E} + \mathbf{E}'$ is not more than t, retrieving $\mathbf{MS}$ is possible.

4 Links Between Coding and Complexity

4.1 Links Between Complexity and Block Codes

The complexity of several decision problems associated to crucial problems in coding theory has been determined. The following two concern error-correcting codes:

Name: Linear Decoding (LD).

Instance: A binary matrix $\mathbf{H}$, a binary vector $\mathbf{y}$, an integer w.

Question: Is there a binary vector $\mathbf{x}$, of weight at most w, such that $\mathbf{x}\mathbf{H}^T = \mathbf{y}$?

We saw (page 9) that LD is associated to the decoding of a binary linear code C, given by a parity-check matrix $\mathbf{H}$.

Name: Minimum Weight in a binary linear code (MW).

Instance: A binary matrix $\mathbf{H}$, an integer w.

Question: Is there a nonzero binary vector $\mathbf{x}$, of weight at most w, such that $\mathbf{x}\mathbf{H}^T = \mathbf{0}$?

Associated to this decision problem is an upper bound on the minimum distance of a binary linear code (cf. page 8).

LD is NP-complete (Berlekamp, McEliece, and van Tilborg [8]). It has been questioned whether its statement is the best one for modeling linear decoding (Bruck and Naor [11]): the code (or the matrix $\mathbf{H}$) is not modified once it has been chosen, only the vector $\mathbf{y}$ changes. It would therefore be possible to apply a *preprocessing* to $\mathbf{H}$, in order to later process efficiently (in polynomial time) the vectors when they are received. We can see the instance of LD in two parts: matrix $\mathbf{H}$ is a "fixed" part, and $\mathbf{y}$, the syndrome of the received vector, a "mobile" part. The problem with preprocessing (LDWP) is formulated by removing $\mathbf{H}$ from the instance. Then one obtains a complexity result weaker than NP-completeness, but still capable of implying the early collapse of the polynomial hierarchy: the existence of a polynomial algorithm solving LDWP would imply that $\Pi_2 = \Sigma_2$ ($= \Pi_k = \Sigma_k$ for all $k \geq 2$) (Bruck and Naor [11]).

•▷ We gave a new proof of it, direct and also valid for any alphabet other than binary [56], [57].

•▷ In the same vein, consider the KNAPSACK problem (Section 3.2). In its cryptographic use, the n integers $a_1, \ldots, a_n$, are not modified for some time, during which only the message (the integer S) changes. Again, we can see the instance in two parts: a fixed part (the integers a_i) and a mobile part (the integer S), and the problem with preprocessing is stated with the integers a_i removed from the instance. We showed [57] that the existence of a polynomial algorithm for this problem would imply that, as for LDWP, $\Pi_2 = \Sigma_2$.

Observe that two of the three public-key cryptosystems described in Section 3 are constructed on the problems LD and KNAPSACK. However, the above results, dealing with their complexity with preprocessing, are no additional safeguard: in cryptography, we actually face polynomial instances, even if they have been scrambled so as to look arbitrary.

The second coding problem, MW, is NP-complete (Vardy [73], 1997, some nineteen years after it was conjectured in [8]). Before this breakthrough, the NP-completeness of several variations had been shown: Exact Weight [8], Average Weight (Diaconis and Graham [36]), Incongruent Weight, Maximum Weight, and Weight-Range (Ntafos and Hakimi [68]), where the question is whether there exists a codeword with weight equal to w, equal to $\lfloor n/2 \rfloor$ (n = length of the code), at most w and not a multiple of a given integer, at least w, and lying between two given integers, respectively.

•▷ As for us, we had proved [60], among others, that knowing whether there exists a codeword with weight at most w whose first $\lfloor w\frac{p}{p+1} \rfloor$ components are '1', is NP-complete for fixed $p \geq 3$.

This result is "almost optimal": if we replace $wp/(p+1)$ by $w - \lambda$, λ constant, then the problem is polynomial. Indeed, complete, in all possible ways, with at most λ '1', the length n vector whose first $w - \lambda$ components are '1', and check membership of the code. The number of checkings is $\sum_{i=0}^{\lambda} \binom{n-(w-\lambda)}{i}$, which is in n^λ, i.e., polynomial in n. If λ is a fraction of w instead of a constant, we have an NP-complete problem.

In the ternary case, note for instance that the existence of a codeword with weight equal to the length is NP-complete (Barg [5]).

Now we consider covering codes.

Name: Covering Radius of a binary Linear code (CRL).

Instance: A binary matrix $\mathbf{H}$ (of dimensions $m \times n$), an integer w.

Question: For any binary vector $\mathbf{y}$ (of length m), is there a binary vector $\mathbf{x}$ (of length n), of weight at most w, such that $\mathbf{x}\mathbf{H}^T = \mathbf{y}$?

Name: Covering Radius of a binary code (CR).

Instance: A binary code C (of length n), an integer w.

Question: For any binary vector $\mathbf{y}$ (of length n), is there a codeword $\mathbf{c}$ such that $d(\mathbf{c}, \mathbf{y}) \leq w$?

We saw (page 9) how CRL corresponds to bounding above the covering radius of a binary linear code; CRL is Π_2-complete (McLoughlin [65]). The same problem for nonlinear codes, CR, is "only" coNP-complete (Frances and Litman [43]), whereas CRL is its subproblem. This paradoxical result can be explained by the more compact representation of a linear code: the size of a problem involving linear codes $[n, k]$, given by generator or parity-check matrices, is $n \cdot k = n \cdot \log_2 |C|$, whereas, for nonlinear codes (n, K), given explicitly but uneconomically by their elements, the size is $n \cdot K = n \cdot |C|$.

●▷ We proved that the same result is true for the minimum norm of a code (defined on page 12) [46]: bounding above the minimum norm of a binary linear code is Π_2-complete, the same for nonlinear codes is coNP-complete.

These results do not stop the search for codes with good parameters, at least for small lengths, and the use of iterative heuristics, such as noising or simulated annealing, gave new constructions. The noising method (see, e.g., Charon and Hudry [14]), described in Section 4.3 in connexion with the construction of identifying codes, has been successfully applied to the construction of error-correcting codes over F_4 (Bogdanova [10]), improving on lower bounds on $A(n, d)$ (defined on page 11) for quaternary codes with lengths up to 12.
●▷ We were less successful when we applied it to covering codes [15], hoping to improve some upper bounds on $K(n, 1)$ (cf. page 10), for n between 9 and 12: we only rediscovered the known upper bounds. (Note added on April 27, 2001: for length 9, this is not surprising. At that time, we knew only that $57 \leq K(9, 1) \leq 62$; now it has been proved that $K(9, 1) = 62$ [69].)

4.2 Links Between Complexity and Arithmetic Codes

As mentioned earlier (page 18) when presenting the Clark-Liang modular weight, its complexity has seemingly never been tackled. If the two modular distances, Rao-Garcia and Clark-Liang, coincide, then it is sufficient to compare two arithmetic weights. In the general case however, how many arithmetic weights, what sizes of integers need to be considered?

Recall that the problem is the following: given a radix r, a modulo m, an integer I between 0 and $m-1$, what is the minimum, $W_{CL}(I)$, in the set $\{W(J) : J = I+km, k \in Z\}$, where $W(J)$ is the arithmetic weight of J, i.e., the minimum number of nonzero terms in a radix r modified representation of J.

•▷ We skimmed over the topic in [61]: D_{CL} is graphical (van Lint [52]); it is the shortest path distance in the Cayley graph $G = (V, E)$ where $V = Z_m$ and the generators are $\{xr^i \bmod m : |x| < r, i = 0, 1, \ldots\}$. The generators give the integers with modular weight 1, thus we know the neighbours of any vertex in V. A breadth-first search, starting from 0, successively constructs the sets of vertices of modular weight 2, 3, ..., $W_{CL}(I)$. Its complexity is bounded above by

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

Another, statistical, approach could be used for estimating the Clark-Liang modular weight; see page 49.

4.3 Links Between Complexity and Identifying Codes

Consider, for fixed t, the following decision problem:

Name: t-Identifying Code.

Instance: A connected bipartite graph $G = (V, E)$, an integer $k \leq |V|$.

Question: Is there a t-identifying code $C \subseteq V$ of size at most k?

•▷ We proved [26], [18] that this problem is NP-complete. I give the proof for $t = 1$. Membership of NP is easy to check, and we polynomially reduce the NP-complete problem 3-SAT (cf. page 30):

Name: 3-satisfiability (3-SAT).

Instance: A set ε of clauses over a set X of variables, each clause containing exactly three distinct literals.

Question: Is there a truth assignment to the variables such that each clause contains at least one true literal?

From $\varepsilon = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, $X = \{x_1, x_2, \ldots, x_n\}$, we construct a bipartite graph G and an integer k such that ε can be satisfied if and only if there is a 1-identifying code of size at most k. For each variable $x_i \in X$, we construct $G_{x_i} = (V_{x_i}, E_{x_i})$, where

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

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

For each clause $\mathcal{C}_j = \{u_{j,1}, u_{j,2}, u_{j,3}\}$, we construct the graph $G_{\mathcal{C}_j} = (V_{\mathcal{C}_j}, E_{\mathcal{C}_j})$, which contains two vertices, α_j and β_j, and one edge, $\{\alpha_j, \beta_j\}$, to which we add the set of three edges $E'_{\mathcal{C}_j} = \{\{\alpha_j, u_{j,1}\}, \{\alpha_j, u_{j,2}\}, \{\alpha_j, u_{j,3}\}\}$.

Graph G has vertex set $V_{x_i} \cup V_{\mathcal{C}_j}$ and edge set $E_{x_i} \cup E_{\mathcal{C}_j} \cup E'_{\mathcal{C}_j}$ ($1 \leq i \leq n$, $1 \leq j \leq m$); it is bipartite. We set $k = 3n + m$.

If ε can be satisfied, then a 1-identifying code C, of size k, can be constructed as follows: for all i between 1 and n, b_i, c_i, and whichever of x_i and $\overline{x}_i$ is true, belong to C; for all j between 1 and m, $\alpha_j \in C$.

Conversely, assume that C is a 1-identifying code. Then $|C \cap V_{\mathcal{C}_j}| = 1$ or 2, and α_j is necessarily covered by a codeword which does not cover β_j. Next, $|C \cap V_{x_i}| \geq 3$, and, if $|C \cap V_{x_i}| = 3$, then exactly one of x_i or $\overline{x}_i$ belongs to C. So $|C| \geq m + 3n = k$, hence $|C| = k$, therefore $|C \cap V_{x_i}| = 3$, and setting x_i true if $x_i \in C$, false if $\overline{x}_i \in C$, is a valid truth assignment to the variables of X. Since α_j must be separated from β_j, it is covered by a codeword corresponding to a literal in clause $\mathcal{C}_j$, which shows that in each clause there is at least one true literal, and ends the proof.

An immediate consequence is that, if t is not fixed, the problem is NP-complete.

We mentioned on page 23 that identifying codes are recent. However, a close notion, that of *locating-dominating sets*, is older (see, e.g., Colbourn, Slater, and Stewart [34]): a subset of vertices (from now, we'll say: a *code*) is (t-)locating-dominating if all vertices *which are not codewords* have nonempty and distinct identifying sets.

We learnt the existence of this close concept after having started our own research on identifying codes, and it turned out that we had taken directions quite different from those developed by Slater and other authors (besides, they worked only on $t = 1$); therefore no common methods have emerged. However, in [34] it is proved that the following decision problem is NP-complete:

Name: 1-Locating-Dominating Code.
Instance: A connected graph $G = (V, E)$, an integer $k \leq |V|$.
Question: Is there a 1-locating-dominating code $C \subseteq V$ of size at most k?

This inspired us for proving the NP-completeness of the problem of existence of t-identifying codes with bounded size.

•▷ We extended [18] this NP-completeness result concerning 1-locating-dominating codes to all integers t, and for bipartite graphs.

We also generalized the two notions, of identifying code and of locating-dominating code, to directed graphs, and proved that the corresponding decision problems are NP-complete, too, for all t fixed or not, and for bipartite graphs [17].

On the one hand, we established complexity results; on the other hand, we built identifying codes with size as small as possible, in the four graphs described in Section 1.3.

•▷ We developed construction algorithms [16] which we now describe shortly.

The goal being to construct codes with low density in some *infinite* graphs, we searched only for periodic codes. Having showed that, in order to consider *all* periodic codes in $Z \times Z$, it suffices to consider rectangular tiles inside which we put codewords, we proceeded as follows: we fix integers t, w, h, α ($0 \leq \alpha < w$), and c ($c \leq w \times h$), and we search for a subset C_R, of size c, of a rectangle R with width w and height h, such that, by translating R by the vectors $(w, 0)$ and (α, h), we obtain a t-identifying code C. In the example of Figure 7, the

values of w, h, α, and c are 10, 2, 3, and 7, respectively. A *solution* is any subset C_R of R, with size c, and we define an *objective function* f for each solution. If this function, taking into account the vertices which are not t-covered by any codeword, and the pairs of vertices which are t-covered by the same codewords, is zero, then C_R induces a t-identifying code.

To this model, we can apply iterative descent methods, for instance with noising (cf. page 41): we arbitrarily fix c codewords inside R. We successively consider each of the codewords, and compute the move which minimizes f. Each time, we move the codeword either on the place minimizing f, or randomly, with a probability to be in the latter case progressively decreasing, from an initial value (typically, 0.2 or 0.3) down to zero. The algorithm stops when $f = 0$ or when a certain number of moves (typically, 300 times the number of vertices in R) has been made.

Scanning small values of w, h, α, c, we search for codes whose density $\frac{c}{wh}$ improves on the known upper bounds. See [16] for results.

5 Links Between Coding and Cryptography

Many links exist between coding and cryptography, whose common goal is to protect information transmission, from either transmission errors or attacks threatening the data privacy or integrity. An elegant example uniting these two aspects of security is given by the McEliece system (cf. Section 3.3), which uses error-correcting codes and can behave like a code and a cryptosystem at the same time (see page 37).

We were given the opportunity to investigate another relation between coding and cryptography, existing between the different representations of integers used in the framework of arithmetic codes (see Section 1.2.1) and the RSA system, which requires fast modular exponentiations (see Section 3.1).

We wish to compute M^e or M^d mod n, where $n = pq$ and e, d are two integers satisfying $ed = 1 \bmod (p-1)(q-1)$. If $d = d_{\ell-1}d_{\ell-2}\ldots d_1 d_0$ is the radix 2 representation of d (with $d_{\ell-1} = 1$), the well-known "square-and-multiply" method computes M^d with ℓ squarings and w multiplications, where w is the number of nonzero components d_i: set $R = 1$ and at each step i, $1 \leq i \leq \ell$, compute $R \leftarrow R^2$, and if $d_{\ell-i} = 1$, also compute $R \leftarrow MR$. The final R is M^d.

●▷ We exploited the following two basic ideas [32], [29]:
1) Using the nonadjacent modified representation (NAMR — cf. page 17) of d, we hope to have fewer nonzero components and save multiplications.
2) $M^{d+k(p-1)(q-1)} = M^d \bmod n$, for all k. We hope to find an exponent $d + k(p-1)(q-1)$ having a representation with "few" nonzero components, and save multiplications.

Consider the binary NAMR of d:

$$d = \sum_{i=0}^{\ell'-1} d'_i 2^i = d'_{\ell'-1}d'_{\ell'-2}\ldots d'_1 d'_0, \text{ where } d'_i = 0, 1, \text{ or } -1, d'_{\ell'-1} = 1, d'_i d'_{i+1} = 0, \ell' \leq \ell + 1.$$

This representation is minimum and gives the arithmetic weight of d. Now it can be shown that the average arithmetic weight of an integer whose NAMR has length ℓ is $\ell/3$, whereas the average Hamming weight of a length ℓ binary vector is $\ell/2$.

The drawback of the NAMR is that it contains '-1'. This can be circumvented by grouping together the '1' and the '-1', to have only one modular division to perform: let $d = d^+ - d^-$ with $d^+ = \sum_{i\in A} 2^i$, $d^- = \sum_{i\in B} 2^i$; then $M^d = M^{d^+}/M^{d^-}$.

Now we study the average improvement obtained by replacing d by $\tilde{d} = d + k(p-1)(q-1)$; we consider that the cost of a multiplication is α times the cost of a squaring, and for the sake of simplicity, we investigate only the case using the binary representation of d and $\tilde{d}$, not their NAMR. We wish to minimize, among a set of possible exponents $\tilde{d}$, the quantity $\ell(\tilde{d}) + \alpha w(\tilde{d})$, where $\ell(\tilde{d})$ is the length, and $w(\tilde{d})$ the number of nonzero components, of the representation of $\tilde{d}$, since this quantity expresses the equivalent number of squarings necessary to compute $M^{\tilde{d}}$.

We set $\ell = \ell(d)$, $\tilde{\ell} = \ell(\tilde{d})$, $\ell(k) = t\ell$, $\phi = (p-1)(q-1)$, and use the following approximations (the integer n is 500 bits long, and the secret key d must be *grosso modo* of the same order of magnitude):

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

As a consequence, $\tilde{\ell} \approx (1+t)\ell$. Now we assume that, when k ranges over the integers of length $\ell(k) = t\ell$, the set of the $2^{t\ell}$ vectors of length $\tilde{\ell} = (1+t)\ell$ representing the exponents $\tilde{d}$ behaves like a set of vectors chosen randomly and independently among the $2^{\tilde{\ell}}$ binary vectors of length $\tilde{\ell}$. Then, the expectation of the number of vectors with weight u in this set is:

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

and is greater than 1 as long as

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

Let $\tilde{u} = \min_{E_u \geq 1} u = y\tilde{\ell}$. Using the binary entropy H_2, an approximation of (5.9) is: $\tilde{\ell} H_2(y) = \ell$, or $H_2(y) = 1/(1+t)$. So the average number of squarings, $\ell(\tilde{d}) + \alpha w(\tilde{d})$, is

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

Its qualitative behaviour with t is given by Figure 11.

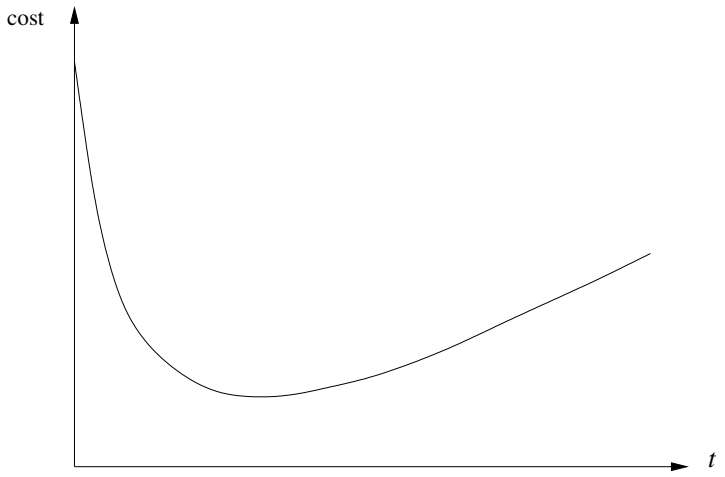

FIG. 11 – *Behaviour of the cost of a modular exponentiation.*

A more detailed report can be found in [32], [29]: minimum cost and value of t for which it is reached, with different assumptions on the ratio α between the cost of a multiplication and the cost of a squaring, in the case of binary representation as well as of modified representation. For instance, with the assumption that $\alpha = 2$, in the case of the binary representation we have just described, there is an average improvement of slightly more than 9%, for an exponent length increasing by slightly less than 11%.

Our simulations corroborated these results and validated our theoretical model, vindicating our approximations and assumptions.

The study of the case of modified representations could be done in the same way, this time trying to minimize $W(\tilde{d})$ instead of $\ell(\tilde{d}) + \alpha W(\tilde{d})$, and we see that this is exactly trying to estimate the Clark-Liang modular weight of d modulo ϕ, since we search for $\tilde{d} = d + k\phi$, with minimum arithmetic weight (cf. Sections 1.2.1 and 4.2).

This possible, statistical, approach, does not exempt from a more theoretical investigation on the complexity of this problem, in view of results better than those of Section 4.2.

6 Prospects

Identifying codes are relatively new and offer vast possibilities for combinatorial or geometric investigations. This is why I plan to dedicate most of my future research to this topic.

One can for instance think of new techniques for lower bounds on the cardinality of an identifying code (i.e., nonexistence results); study classes of graphs such as chains or trees [work in progress], or, for the infinite grids of Section 1.3, jump from dimension two to three; find the complexity of the problem for certain graphs [work in progress for the n-cube]; apprehend, in the n-cube, the behaviour of the spheres when we increase the radius: after a while, their "identifying power" decreases — when we take spheres of radius n, we cannot identify any vertex.

Finally, in the case of the aforementioned infinite grids, one possible generalization seems particularly rich: let $V = Z \times Z$ be the vertex set. Up to now, we considered edges and spheres, and spheres were *patterns* used for covering V and identifying the vertices: for instance, in the king grid, a sphere of radius t is a square with sides of size $2t + 1$. Now, we directly consider a pattern, e.g., a square with sides of *even* size. It is not a sphere, however we can try to use it and cover V, in such a way that two distinct vertices are differently covered. Now all sorts of patterns can be used...

APPENDIX:

COMPLETE LIST OF PUBLICATIONS, IN CHRONOLOGICAL ORDER

• 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.