SOME DISTANCE PROBLEMS
IN CODING THEORY

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus, prof.
Dr. F.N. Kooij, voor een commissie
aangewezen door het college van dekens
in het openbaar te verdedigen op

vrijdag 16 januari 1987 te 16.00 uur

door

CORNELIS LEONARDUS MARIA VAN PUL
geboren te Rotterdam.
Dit proefschrift is goedgekeurd
door de promotorren:
Prof. dr. J.R. van Liemt
en
Prof. dr. ir. H.C.A. van Tilburg
ABSTRACT

This thesis is concerned with four topics from coding theory. The first one of these, treated in Chapter 1, is that of coding in an imperfect computer memory with stuck-at-defects and random errors. This coding problem finds its origin in a paper by Kuznetsov and Tyul'kov (1974). After a short historical overview in Section 1.1, a description of the problem and some related problems is given in Section 1.2. The Sections 1.3 up to 1.6 deal with lower (i.e., constructions) and upper bounds for the various functions defined in Section 1.1. The function \( A(n,d) \), i.e., the largest size of any binary code of length \( n \) and minimum distance \( d \), plays an important role in these sections.

In Chapter 2 we treat two constructions for constant weight codes. These constructions result in improved lower bounds on the function \( A(n,d,w) \), i.e., the largest size of any binary constant weight code of length \( n \), minimum distance \( d \) and constant weight \( w \). This function plays an important role in determining upper bounds on the function \( A(n,d) \) (e.g., Linear Programming Bound and Johnson bound).

In Chapter 3 we give the complete solution of a problem formulated by Ahlswede, Ccanal and Kayhan in 1984. They define a constant distance code pair \( (A,B) \) as a pair of binary codes of length \( n \) such that for some \( \delta \in \mathbb{R} \),

\[
\forall a \in A \exists b \in B \quad |a(b)-b(a)| = \delta
\]

They prove that for such a code pair \(|A| \cdot |B| \geq 2^{2 + \lceil \log_2 \delta \rceil} \). With the help of coding theory Hall and van Lint gave a nice proof of this inequality and moreover characterized all code pairs for which equality holds.

Since for these code pairs \( \delta \in \mathbb{R} \) or \( \mathbb{Z} \), the question remained: "What happens when \( \delta \) is fixed?". Chapter 3 gives an answer to this question.

In Chapter 4 we discuss a problem which arose in connection with comma-free codes. Let \( N_q(n) \) denote the maximal number of codewords in any \( q \)-ary comma-free code of length \( n \). Eastman (1965) proved that
\[ w_n(q) = \frac{1}{n} \sum_{d|n} \mu(d)q^{n/d} - R_n(q) \text{ if } n \text{ is odd.} \]

For even wordlength \( n \) the situation is much more complicated. In 1964
Cohn and Feng proved that

\[ w_{2k}(q) < R_{2k}(q) \text{ if } q > t(k) + k, \]

where \( t(k) \) is the maximal cardinality of any \((0,1,*)\) tournament code of
length \( k \). Chapter 4 deals with the problem of determining lower and
upper bounds on \( t(k) \), \( k \in \mathbb{N} \).

In order to make this thesis self-contained, we start with a short
introduction to coding theory in Chapter C.
CONTENTS

Chapter 0  INTRODUCTION  1

Chapter 1  COMPUTER MEMORIES WITH "STUCK-AT" DEFECTS AND RANDOM ERRORS
1.1 Introduction  6
1.2 Coding for an imperfect computer memory  8
1.3 Upper bounds on $n_f(m,t)$  16
1.4 Constructions for (separable) $t$-defect-compatible matrices
   with $t = 3$  20
1.5 Generalized partitioned linear block codes  32
   References  42

Chapter 2  TWO CONSTRUCTIONS FOR CONSTANT WEIGHT CODES
2.1 Introduction  44
2.2 A generalization of the Johnson bound for constant weight codes  45
2.3 Lower bounds for $A(n,n^{1/3})$  50
   Appendix  57
   References  69

Chapter 3  CONSTANT DISTANCE CODE PAIRS
3.1 Introduction  70
3.2 The exact value of $M(n,d)$  72
3.3 Optimal constant distance code pairs  77
   References  81

Chapter 4  TOURNAMENT CODES
4.1 Introduction  82
4.2 A lower bound  86
4.3 An upper bound  91
4.4 The exact value of $t(k)$ for $k = 2,3,\ldots,9$
   References  98
   Remarks  100
   Curriculum Vitae  100
CHAPTER 3

INTRODUCTION

The purpose of this introduction is to make the reader familiar with some of the notions of coding theory; for a course in coding theory we refer the reader to [1] and [2]. We restricted ourselves to the binary case.

Let $\mathbb{F}_2^n$ be the $n$-dimensional vector space over $\mathbb{F}_2$. A block code $C$ of length $n$ over $\mathbb{F}_2$ is a subset of $\mathbb{F}_2^n$. The elements of $C$ are called codewords. The set of elements of $\mathbb{F}_2^n$ is called the alphabet of the code $C$.

A $k$-dimensional linear subspace of $\mathbb{F}_2^n$ is called a binary linear block code, or binary $[n,k]$-code.

The Hamming-weight $wt(x)$ of a vector $x \in \mathbb{F}_2^n$ is the number of non-zero coordinates of $x$. The Hamming-distance $d(x,y)$ of two vectors $x$ and $y$ in $\mathbb{F}_2^n$ is defined by $d(x,y) = wt(x \oplus y)$. In words: $d(x,y)$ is the number of coordinate places in which $x$ and $y$ differ. The minimum distance $d$ of a code $C$ is defined by

$$d = \min \{ d(x,y) \mid x, y \in C, x \neq y \}.$$ 

A block code of length $n$ and minimum distance $d$ is called an $(n,d)$-code. A $(n,d)$-code with $N$ codewords, we call an $(n,N,d)$-code. An $(n,M,d)$-code of which all codewords have the same Hamming-weight, we say, is called a constant weight code, or an $(n,M,d,w)$-code. A linear $[n,k]$-code with minimum distance $d$ is called an $[n,k,d]$-code (the minimum distance in a linear code equals the minimum weight among all non-zero codewords).

In the vector space $\mathbb{F}_2^n$, we define an inner product $( , )$ in the usual way.
\[ \forall x \in \mathbb{F}_2 \quad \forall y \in \mathbb{F}_2^n \quad (x, y) := x_1 y_1 \otimes x_2 y_2 \otimes \cdots \otimes x_n y_n, \]

where \( \otimes \) denotes the usual addition in \( \mathbb{F}_2 \). If \( \mathcal{C} \) is an \((n, k, d)\)-code, then the dual code \( \mathcal{C}^\perp \) of \( \mathcal{C} \) is defined by

\[ \mathcal{C}^\perp := \{ \mathbf{z} \in \mathbb{F}_2^n \mid \forall y \in \mathcal{C} \quad \langle \mathbf{z}, y \rangle = 0 \}. \]

The code \( \mathcal{C}^\perp \) is an \((n, n-k)\)-code.

A generator matrix \( G \) of an \((n, k)\)-code \( \mathcal{C} \) is a \( k \times n \) matrix, the rows of which form a basis of \( \mathcal{C} \). A parity-check matrix \( H \) of a linear code \( \mathcal{C} \) is a generator matrix of the code \( \mathcal{C}^\perp \). Both \( G \) and \( H \) define the code \( \mathcal{C} \). The matrices \( G \) and \( H \) satisfy \( GH^\top = 0 \) (evaluated in \( \mathbb{F}_2 \)).

Block codes are used for reliable transmission of information over noisy channels. Examples of noisy channels are: telephone wires, telegraph wires, computer memories, etc. A simple model of such a channel is the binary symmetric channel, i.e., a channel over which we can send two different symbols 0 and 1 and for which there is a probability \( p \) that a transmitted 0 (resp. 1) is interpreted by the receiver as a 1 (resp. 0). The following figure illustrates the information-transmission scheme:

![Diagram](image)

Fig. 1.

We use the following notation:

- \( u \in \{0, 1, \ldots, M-1\} \): \( u \) the input message set,
- \( z \in \mathbb{F}_2^n \) channel input word,
- \( x \in \mathbb{F}_2^n \) a channel output word,
- \( v \in \{0, 1, \ldots, M-1\} \) the
output message and $\mathbf{v} \in \mathbb{F}_2^n$ an error vector describing the noise on the binary symmetric channel.

The channel input word $\mathbf{x}$ and channel output word $\mathbf{y}$ are related as follows

$$\mathbf{y} = \mathbf{x} \oplus \mathbf{e},$$

where $\oplus$ is the usual addition in $\mathbb{F}_2$ which operates on the vectors componentwise.

In order to protect the information, sent over the BSC channel, one can use the codewords of a binary $(n,M,d)$-code $C$ as channel input words. A one-to-one mapping $\mathbf{v} \mapsto \mathbf{y} = \mathbf{y} = \mathbf{v} \oplus \mathbf{e},$ is used to map any message $\mathbf{u} \in C$ onto a codeword $\mathbf{y} = \mathbf{v} \oplus \mathbf{e} \in C$. The function $\Phi$ is called an encoding function for $C$. A particular decoding function $\Gamma$, $\Gamma : \mathbb{F}_2^n \rightarrow C$, for $C$ can be defined by

$$\mathbf{v} = \Gamma(\mathbf{y}) := \Phi^{-1}(\mathbf{v}),$$

where $\mathbf{y}$ is the (not necessarily unique) codeword of $C$ which lies closest to $\mathbf{y} = \mathbf{y} \oplus \mathbf{e}$. If $\text{wt}(\mathbf{e}) \geq \frac{d}{2}$, then one easily sees that $\mathbf{x}'$ is equal to $\mathbf{x}$ and hence $\mathbf{v}$ is equal to $\mathbf{u}$. We say that $C$ is a $\left[\frac{d}{2}\right]$-error-correcting code.

The decoding principle described above is known as maximum likelihood decoding. It requires the determination of the (not necessarily unique) codeword $\mathbf{y}'$ of $C$, which lies closest to the received channel output word $\mathbf{y}$. This is a laborious task if the cardinality of $C$ is big and $C$ has no structure whatsoever. The linear structure of a code can be utilized to make the decoding somewhat easier.

Let $C$ be a binary linear code with parity check matrix $H$. For every $\mathbf{x} \in \mathbb{F}_2^n$ we call $\mathbf{h}^T \mathbf{x}$ the syndrome of $\mathbf{x}$. From the above we have that the codewords of $C$ are characterized by syndrome $0$. The syndrome is an important tool in decoding received vectors $\mathbf{y}$. Since $C$ is a sub-group of $\mathbb{F}_2^n$, we can
partition $\mathcal{P}_2$ into cosets of $C$. Two vectors $\tilde{v}$ and $\tilde{w}$ are in the same coset iff they have the same syndrome $(\tilde{v}^T \cdot v^T) = x \cdot y \in \mathcal{P}_2$. Therefore, if a vector $\tilde{v}$ is received, where $\tilde{v} = g \cdot x$, $x \in \mathcal{P}_2$, then $\tilde{v}$ and $g$ have the same syndrome. It follows, that for maximum likelihood decoding of $y$ one must choose a vector $\tilde{g}$ of minimal weight in the coset with syndrome $\tilde{v}$ and then decode $\tilde{v}$ as $g = S^{-1}(\tilde{v} \cdot \tilde{g})$. The vector $\tilde{g}$ is called the coset leader. Again, if $u(x) = S^{-1}(\tilde{v} \cdot \tilde{g})$ then $\tilde{g}$ is equal to $g$ and hence we will decode $\tilde{v}$ correctly.

Since time is money, we must in general keep the time needed for the transmission of information as short as possible. Let $C$ be a binary $(n,R,d)$-code. Then the rate $R$ of $C$, defined by

$$R = \frac{1}{n} \log M,$$

is a measure for the efficiency of the code $C$. Since, for a message $u \in \mathcal{U}$, with $|u| = M$, we need on the average $R \cdot M$ bits to distinguish $u$ from all other messages in $\mathcal{U}$, the number $n(1-R)$ gives an indication of the loss of time in transmission when the code $C$ is used for error protection. It will be clear that the higher the rate of $C$ the lower the error-correcting capability of $C$. So knowledge of the following two functions is of utmost importance.

$A(n,d)$ : maximum number of codewords in any binary code (linear or non-linear) of length $n$ and minimum distance $d$, and

$B(n,d)$ : maximum number of codewords in any linear binary code of length $n$ and distance $d$. 
REFERENCES


CHAPTER 1

COMPUTER MEMORIES WITH "STUCK-AT" DEFECTS AND RANDOM ERRORS

1.1 INTRODUCTION

In this chapter we consider the problem of reliable storage of information in an imperfect binary computer memory. We consider a memory that is composed of a very large number of binary memory cells which are partitioned into memory units of n cells, in the block-length of the error-correcting code to be used. We are concerned with two types of imperfections that affect individual memory cells. The first type is a defective memory cell that is unable to store information; its current value cannot be changed. Such cells are called stuck-at cells. We distinguish between stuck-at-0 and stuck-at-1 cells. When a 1 is written into a stuck-at-0 cell an error results. The second type of imperfection is a noisy cell which is occasionally in error. The distinction between these two types of imperfections is that stuck-at defects are permanent, while errors caused by noise are transient.

By examining a memory unit it is possible to determine the locations and natures of the stuck-at cells. The side information that describes the state of the defects can be incorporated in the decoding of the encoding of block codes. Depending on how this stuck-at information is exactly used, this gives rise to a number of different coding (reliable storage) problems. We mention the two most interesting ones.

In the first case, the locations of the stuck-at-cells are assumed to be known only at the decoder. These cells then act as erasures. Thus, it makes sense to apply known techniques for decoding block codes with random errors and erasures to this case. We will not go into this problem. The interested reader is referred to [2]. We consider the complementary problem of incorporating stuck-at information in the encoding process.
This last problem was originated by Rumenov and Tsybakov in [11]. They consider coding for binary memory units that have a number t of stuck-at cells, where 1 ≤ t ≤ p < 1, p fixed. The assumption is that the locations and natures of the defects are known at the decoder but not at the encoder, by allowing the size of the memory unit to become large, they prove the existence of codes that are capable of storing information without error, for any rate 0 ≤ R < 1 - p. Moreover, they prove that such codes can be found within the class of additive codes (see [11]). In Section 1.2 we give an outline of this paper. At the end of this section we introduce the related problem of exhaustive test pattern generation. In both problems so-called t-defect-compatible matrices play a very important role. Also the equivalence of the notion of t-defect-compatibility and that of t-independence of sets is mentioned. This fact seems to be almost unknown.

In Section 1.3 we prove an upper bound for the largest possible length of a t-defect-compatible matrix with m rows. This bound gives a slight improvement on the one given in [9]. Section 1.4 deals with constructions for additive codes, capable of correcting all word defects of multiplicity t or less and hence by nature, also constructions for exhaustive pattern testing schemes. The constructions described there, in fact generate separable t-defect-compatible matrices.

In [19] Tsybakov introduces the problem of coding for binary memory units with both defects and random errors. Once again the locations and natures of the defects are assumed to be known at the encoder but not at the decoder. He introduces the concept of "matched adjacent codes" to solve this problem. In [17] Meggido calls these codes partitioned linear block codes. We will stick to that name. In Section 1.5 we use their ideas and one of our construction methods of Section 1.4 to construct codes that have a better performance than those given in [7]. With these codes the encoding process will take more time, the decoding process on the other hand not.

The problem of determining the capacity of imperfect computer memories when complete or partial defect information is available at
the encoder or at the decoder is not studied here. For this problem, we refer the interested reader to [4].

1.2 CODING FOR AN IMPERFECT COMPUTER MEMORY

1.2.1 An electronic model

The following figure illustrates the information-transmission (storage) scheme we are concerned with.

\[\text{Encoder} \rightarrow \text{Channel (memory unit)} \rightarrow \text{Decoder}\]

\[\text{Supplemental info. source} \rightarrow \text{noise}\]

Fig. 1.

We use the following notation:

\(u \in \{0, 1, \ldots, M-1\}\) - the input message set, \(x \in \mathbb{F}_2^n\) a channel input word, \(y \in \mathbb{F}_2^n\) a channel output word, \(v \in \{0, 1, \ldots, M-1\}\) the output message, \(e \in \mathbb{F}_2^n\) an error vector describing the noise on the channel and \(d = (0, 1, \delta)^n\) a word defect describing the states of the memory cells to be used.

The word defect \(d = (d_1, d_2, \ldots, d_n) \in \{0, 1, \delta\}^n\) has to be interpreted as follows:
\[
\begin{cases}
0, \text{ then the } i^{th} \text{ cell of the memory unit is stuck-at-0}, \\
1, \text{ then the } i^{th} \text{ cell of the memory unit is stuck-at-1}, \\
\delta, \text{ then the } i^{th} \text{ cell of the memory unit is defect-free}. 
\end{cases}
\]

The number \( t \) of coordinates of \( \alpha \) equal to 0 or 1 is called the multiplicity of the word defect \( \alpha \). By \( D_{\leq t}^2 \) we denote the set of word defects \( \alpha \in \{0,1,\delta\}^n \) with multiplicity \( t \) or less. Let the "or" operator \( \circ \colon D_{\leq t}^2 \times \{0,1,\delta\} \to D_{\leq t}^2 \) be defined by

\[
x \circ y = \begin{cases} 
    x & \text{if } y = \delta, \\
    d & \text{if } d \neq \delta.
\end{cases}
\]

The relation between the channel input word \( x \) and channel output word \( y \) can then be described by

\[
y = (x \circ y) \circ x.
\]  \hspace{1cm} (1)

The errors, described by the error vector \( e \), occur when reading the memory, so they affect the memory contents of defect-free cells as well as that of stuck-at cells.

**Example:** Let \( n = 6, x = (0,0,0,0,0,0), \alpha = (0,\delta,0,\delta,1,0) \) and \( e = (0,1,0,0,1,1) \). Then \( y = (x \circ \alpha) \circ e = (0,1,\delta,1,0,1) \).

§ 1.2.2 The class of additive codes

During the rest of this section we assume that there is no noise on the channel (memory), so \( \circ = \odot \) in (1). Furthermore, we assume that the stuck-at cells are randomly distributed over the memory. In [11] Kuznetsov and Tsybakov define a block code of length \( n \) for this memory as a
partition of $\mathbb{F}_2^n$ into $2^k$ subsets $A_u$, $u = 0, 1, \ldots, 2^k - 1$. They use the defect information, known at the encoder, to assign to a message $u$ a channel input word $x \in A_u$ in such a way that the stuck-at fault of the memory unit to be used do not alter $x$. The decoder, receiving the unaltered $x$, recognizes that $x$ belongs to the subset $A_u$ and so recovers the message $u$ correctly. The rate $R$ of the code is given by $R = (1 - \log_2 2^k) / n$. To find suitable partitions of $\mathbb{F}_2^n$, Fumihide Tanaka and Peres make use of so-called separable $t$-defect-compatible matrices, leading to the introduction of the class of additive codes. We need some definitions.

A word $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ is said to be compatible with the word defect $d = (d_1, d_2, \ldots, d_n) \in \{0, 1\}^n$ if $x_i \equiv d_i \pmod{2}$, for all $i \in \{1, 2, \ldots, n\}$ with $d_i = 0$ or 1. A binary $m \times n$ matrix $C$ is called a $t$-defect-compatible matrix, if for any word defect $d \in \mathbb{F}_2^n$, there is a row of $C$ which is compatible with $d$.

We are now ready to define the class of additive codes.

Let $C$ be a $2^k \times n$ binary matrix in which the first $k$ elements of each row form the binary representation of the number $i$ of that row ($i = 0, 1, \ldots, 2^k - 1$). A matrix with this property is said to be a separable matrix. Let, for any $u \in \{0, 1, \ldots, 2^k - 1\}$ and any $d \in \{0, 1\}^n$, $C(u, d)$ be a specified row of $C$. This specification will be made clear later on. For any $u \in \mathbb{F}_2$, the vector $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ is given by

$$x_i = C(u, i - 1)$$

The encoding function $\Phi: \mathbb{F}_2 \times \{0, 1, \ldots, 2^k - 1\} \rightarrow \mathbb{F}_2^n$ is defined by

$$\Phi(u, d) = u \oplus C(u, d).$$

The code (partition of $\mathbb{F}_2^n$) defined by $C$ is clearly given by
\[ X^n_2 = \bigcup_{u=0}^{n-1} \{ y \in \mathbb{F}_2^n : y \text{ is a row of } C \}. \]

The rate \( R \) of this code is equal to \( R = (n-r)/n = 1-r/n \). Different separable matrices \( C \) define a class of codes, which we call the class of additive codes.

The decoding function \( \Psi \) is \( \Psi: X_2^n \rightarrow \mathbb{F}_2^n \), for the additive code is defined by

\[ \Psi(x)_i = \sum_{j=1}^{n-1} (x_{i+j} \cdot c_{i+j}) \cdot 2^{i-1}, \]

where \( c = (c_1, c_2, \ldots, c_{n-1}) \) is that row of \( C \) with

\[ c_i = y_i \quad \text{for} \quad i = 1, 2, \ldots, r. \]

From the above it will be clear that a necessary and sufficient condition for the additive code, defined by the separable matrix \( C \), to correct all word defects of multiplicity \( t \) or less is that \( C \) is a separable \( t \)-defect-compatible matrix. For any \( u \in \mathbb{F}^n_2 \) and \( \delta \in \mathbb{F}^n_2 \) the row \( c(u, \delta) \) from \( C \) must then be the (not necessarily unique) row of \( C \) which is compatible with the word defect \( \delta' \) defined by

\[ \delta' = \begin{cases} \\delta & \text{if } d_1 = \delta, \\u_1 \oplus \delta & \text{if } d_1 = 0 \text{ or } 1, \end{cases} \]

where \( u_1 \) is the \( 1 \)st component of the vector \( u \) defined above.

**Example 2.** Let \( n = 4, r = 1, t = 1, C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \), \( u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \) and \( \delta = \begin{bmatrix} 0 \\ 5 \\ 6 \\ 0 \end{bmatrix} \).

**Encoding:** To encode we determine \( y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \), \( \delta' = \begin{bmatrix} 5 \\ 6 \\ 1 \\ 6 \end{bmatrix} \), and \( c(u, \delta) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \). We store

\[ x = \Psi(u, \delta) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \]
Decoding: To decode retrieve from the computer memory the vector \( \mathbf{y} = [0, 0, 1, 1, 0, 0, 1, 1, 0] \) and from \( \mathbf{C} \) the row \( \mathbf{s} \) with index \( 1 \cdot 1 = 1 \); so \( \mathbf{s} = (1, 1, 1, 1) \). Now compute the value \( v = \mathbf{y} \cdot \mathbf{s} = (0 \ast 1) + (0 \ast 1, 0) + (1 \ast 1) + (1 \ast 1) = 0. \)

**Example 3.**

Let \( n = 3, r = 2, \ t = 2, \mathbf{C} \) as above, \( \mathbf{u} = 1 \) and \( \mathbf{s} = (1, 1, 1, 1) \).

Forming: \( \mathbf{y} = [0, 0, 1], \mathbf{d} = [1, 0, 0] \) and \( \mathbf{s} \cdot \mathbf{u} = (1, 0, 0) \).

So store the vector \( \mathbf{y} \cdot \mathbf{s} = (0, 0, 1) \ast (1, 1, 0) = (1, 1, 1). \)

Decoding: Retrieve \( \mathbf{y} \cdot \mathbf{s} = [1, 1, 1] \) and the row \( \mathbf{r} \) of \( \mathbf{C} \) with index \( 1 \cdot 1 \cdot 1 = 1 \); so \( \mathbf{r} = (1, 1, 0) \). Compute \( v = \mathbf{y} \cdot \mathbf{r} = 1 \cdot 1 = 0. \)

We define the function \( R(n, t) \) by

\[
R(n, t) = \text{the maximal value of } \mathbf{r} \text{ for which there exists a code with rate } \mathbf{r} \text{ that is capable of correcting all } \mathbf{t} \text{-defects of multiplicity } t \text{ or less.}
\]

In [11] Nemetsov and Pavlov prove the following surprising result.

**Theorem 1.** For any \( n, t \in \mathbb{N}, \ 1 \leq t \leq n, \)

\[
1 - \frac{\log \text{ in } 2^{n \choose t}}{n} \leq R(n, t) \leq \frac{t}{n}.
\]

(2)

The upper bound in Theorem 1 is obvious. The lower bound is a consequence of the existence of separable \( t \)-defect-compatible matrices of size \( 2^{n} \times n, \) with

\[
\log \text{ in } 2^{n \choose t}. \]

(3)
The existence of such matrices is proved by using a probabilistic "counting" argument. Since, for any fixed \( p \in [0,1] \),

\[
1 = \frac{1}{n} \log \left( \frac{n}{1 - p} \right) + \log \frac{1}{1 - p + o(n^{-1})}, \quad \text{for } \frac{1}{n} \to 0 \text{ and } n \to \infty,
\]

we have the following consequence of this theorem.

**Corollary 2.** Let \( p \) be any fixed number in \([0,1]\) and let \( \epsilon > 0 \). Then, for \( n \) sufficiently large (depending on \( p \) and \( \epsilon \)), there exists an additive code of length \( n \) that is capable of correcting all word defects of multiplicity \( np \) or less, for a rate \( R \), \( 1 - p - \epsilon > R > 1 - p \).

\[\square\]

\section{1.1.3 Some related problems}

From § 1.2.2 it will be clear that separable t-defect-compatible matrices play an important role in the reliable storage of information in an imperfect computer memory with stuck-at defects. Therefore, we define

\[ r(n,t) := \text{the minimal value of } k \text{ for which there exists a } 2^t \times n \text{ separable t-defect-compatible matrix, } n \in \mathbb{N}, \quad 1 \leq t \leq n. \]

The functions \( s(n,t) \) and \( r(n,t) \) are related by

\[ n s(n,t) \geq r(n,t). \]

In the conventional approach to logic circuit testing, a set of test vectors, to be applied at the circuit inputs, is derived from an analysis made on the circuit under test. Typical faults one wishes to determine are stuck-at-0 and stuck-at-1 faults at the gate level. Such a test-generation procedure requires a substantial amount of computer time due
to the necessary analysis and simulation to be carried out. Due to the growth of the number of logic circuits on a VLSI-chip, the conventional way of logic test generation becomes more and more impractical. Not only the computer time grows excessively, also the simple stuck-at fault model becomes more inadequate. A partial solution to this problem is to use exhaustive pattern testing schemes for testing several logic circuits simultaneously.

In this approach a VLSI-chip is considered to have \( n \) binary inputs. Each input may influence many outputs, but due to certain partitioning techniques each output is assumed to depend on at most \( t \) inputs \((1 \leq t \leq n)\). To test the chip, any set of \( t \) or less inputs feeding an output is provided with all possible input patterns. By checking the correctness of the outputs, any single hard fault or combination of hard faults, which results in a permanent alteration of the truth table, associated with an output function, is noticed. So we are left with the problem of generating a minimal set of test vectors of length \( n \), to provide simultaneously all input patterns to each of a collection of input subsets of size \( t \) or less. From the above, it may be clear that the rows of an \( n \times n \) \( t \)-defect-compatible matrix form such a set. Therefore, we define

\[ m(n,t) \] - the minimal value of \( n \) for which there exists an \( n \times n \) \( t \)-defect-compatible matrix.

The relation between the functions \( m(n,t) \) and \( m(n,1) \) is given by

\[ m(n,t) = n - \log m(n,1). \]

For a more detailed description of the problem of logic circuit testing, the reader is referred to [5,13].

Most authors who work on those two fields of research do not seem to be aware of the fact that the notion of \( t \)-defect-compatibility is equivalent to that of \( t \)-independence of sets. Consider the \( j \)th column of
an $m \times n$ t-defect-compatible matrix as the characteristic vector of a
subset $A_i$ of the set $A = \{1, 2, \ldots, m\}$. Let $\mathcal{F}$ denote the collection of subsets $A_i$ of $A$, $i = 1, 2, \ldots, n$, i.e., $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$. The t-defect-compatibility property can then be formulated as

For any t-tuple of subsets $A_{i_1}, A_{i_2}, \ldots, A_{i_t}$ from $\mathcal{F}$, all $2^n$
intersections

$$
\bigcap_{i=1}^{t} A_{i_i}
$$

are non-empty, where each $A_{i_i}$ can be either $A_{i_i}$ or $A \setminus A_{i_i}$.

In [9] Kleitman and Spencer call such a collection a $t$-independent
collection of subsets of an $n$-element set. In [9] a lower bound on the
size of such a collection is proved that coincides with the upper bound
on $\rho(n, t)$ given by (3). In Section 1.3 we mention some of their results
translated in the terminology of t-defect-compatible matrices.

For later use we give two more definitions. For any $r, n, t \in \mathbb{N}$ we
define

$$
\rho(n, t) := \text{the maximal value of } n \text{ for which there exists a } t^n \times n
$$

separable t-defect-compatible matrix, and

$$
\rho(t, n) := \text{the maximal value of } n \text{ for which there exists an } n \times n
$$
t-defect-compatible matrix.

The relations between $\rho(n, t)$ and $\rho(t, n)$ respectively $\min(n, t)$ and $\rho(n, t)$
are given by

$$
\rho(n_0, t) \leq n_0 \iff n(n_0, t) \leq t_0
$$

and

$$
\rho(n_0, t) \leq n_0 \iff n(n_0, t) \leq t_0
$$
We conclude this section with a table of known values of \( z(n,t), n(n,t) \) and \( v(n,t) \), for \( t = 0,1,m - 1 \) and \( n \).

<table>
<thead>
<tr>
<th>( t )</th>
<th>( z(n,t) )</th>
<th>( n(n,t) )</th>
<th>( v(n,t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>t</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1 - 1/( n )</td>
</tr>
<tr>
<td>( m - 1 )</td>
<td>( n - 1 )</td>
<td>( 2^{m-1} )</td>
<td>1/( n )</td>
</tr>
<tr>
<td>( m )</td>
<td>( n )</td>
<td>( 2^n )</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1.

1.3 UPPER BOUNDS ON \( a(t,n) \)

In [9] Kleitman and Spencer consider the problem of determining the largest size of a \( t \)-independent family of subsets of an \( n \)-element set. From the previous section we know that this is equivalent with determining the largest value of \( n \), for which there exists an \( n \times n \)-\( t \)-defect-compatible matrix. We have denoted this maximal value by \( n(t,n) \). In [9] Kleitman and Spencer solve this problem for \( t = 2 \) (see Theorem 3) and give asymptotic upper and lower bounds for \( n(t,n) \), where \( t \) is fixed and \( n \) tends to infinity. Although, from a coding point of view, determination of such bounds is of almost no interest, we found this problem interesting enough to work on. In this section we prove a slight improvement on the upper bound given in [9].

We first give the solution for \( t = 3 \) in Theorem 3. Because of our interest in separable \( t \)-defect compatible matrices, the value of \( a(t,2) \) is also mentioned.

**Theorem 3.** [9] For all \( n \in \mathbb{N}, \; n \geq 4 \) and \( t \geq 2 \) we have

\[
\frac{\binom{n-1}{t}}{\binom{n}{t}} \leq a(t,n) \leq \frac{\binom{n}{t-1}}{\binom{n}{t}}.
\]
The values of $af(n,2)$ and $m(n,2)$ are attained by the following construction.

**Construction.**

Let $a \in \mathbb{N}$, $a \geq 4$. We define $C$ to be the $n \times \left(\frac{m-1}{2}\right)$ matrix with columns all binary vectors of length $a$ and Hamming weight $\left\lfloor \frac{m}{2}\right\rfloor$, of which the first coordinate is equal to zero (see Fig. 2, below).

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

Fig. 2. A $6 \times 10$, 3-defect-compatible matrix.

From the above figure, it is easy to see that this construction indeed yields a 3-defect-compatible matrix. If $a = 2^k$ the matrix is separable.

As a consequence of Theorem 3 we find the following values for $m(n,2)$, $r(n,2)$ and $R(n,2)$ (see also § 1.2.3). Let, for any $n \in \mathbb{N}$, $n_0 \in \mathbb{N}$ be defined by

$$
\begin{bmatrix}
\binom{n_0}{2} - 1 \\
\binom{n_0 - 1}{2} \\
\end{bmatrix} \leq n \leq \begin{bmatrix}
\binom{n_0}{2} - 1 \\
\binom{n_0 - 1}{2} \\
\end{bmatrix}
$$

and take $n_0 = \left\lceil \log n \right\rceil$. Then

$$
\sigma(n,2) = n_0, \quad r(n,2) = n_0, \quad \text{and} \quad n - \frac{n_0 - 1}{n} \leq m(n,2) \leq 1 - \frac{n_0 - 1}{n}.
$$
We now turn our attention to the case $t = 1$. In [9] Kleitman and Spencer prove the following lower and upper bound on $\text{nf}(m,t)$.

**Theorem 5.** For all $t \in \mathbb{N}$, $t \geq 3$, $t$ fixed, we have

\[
\text{nf}(m,t) \geq \left\lfloor \frac{\log(1 - 2^{-t})}{t} \right\rfloor \text{ or } m = w,
\]

and

\[
\text{nf}(m,t) \leq \left\lfloor \frac{\log(1 - 2^{-t})}{t} \right\rfloor \text{ or } m = w,
\]

where $\log(x) = \log(1 - 2^{-t})$, and $\log(x) = \log(1 - 2^{-t})$, and $\log(x) = \log((1 - 2^{-t})/(1 - 2^{-t}))$, with $K(p) = -\log p - (1-p) \log(1-p)$, the well-known binary entropy function.

Let, as defined in Chapter 1, $A(n,d)$ denote the largest value of $N$ for which there exists a binary $(n, m, d)$ code. The following theorem uses the function $A(n,d)$ to derive an upper bound on $\text{nf}(m,t)$, $t \geq 4$.

**Theorem 5.** For any $m, t \in \mathbb{N}$, $t \leq 6$, we have

\[
\text{nf}(m,t) \leq \max_{0 \leq k \leq m} \min \left\{ \text{nf} \left( \frac{m}{2}, t-3 \right) + 2, A(n,d) \right\}.
\]

**Proof.** Let $C$ be an $m \times m$ $(m,t)$-defect compatible matrix. Let $A$ be the binary code with as codewords all the columns of $C$ and $\overline{C}$. From the definition of a $(m,t)$-defect compatible matrix, it follows that $C$ and $\overline{C}$ have no columns in common. Let $d$ be the minimum distance of $A$. Then,

\[
\text{nf}(m,t) = |A| \leq A(n,d).
\]

Since $A$ has minimum distance $d$, there are two columns of $C$, w.l.o.g., the first two columns $b_1$ and $b_2$, with $d(b_1, b_2) = 2$ or $n - d$. Assume $d(b_1, b_2) = 2$ (the case $d(b_1, b_2) = n - d$ goes analogously). Consider the matrices $C_1$ respectively $C_2$ consisting of those rows of $C$ for which
the first entry is equal to 0 and the second entry is equal to 1, respectively the first entry equal to 1 and the second entry equal to 0. From the $t$-defect-compatibility of $C$, it follows immediately that the matrices $C_1$ and $C_2$, which are formed by deleting the first two columns of $C$, respectively $C_2$, are $(t-2)$-defect-compatible matrices. Let $n_i$ be the number of rows of $C_i$, $i=1,2$. Then $n_1 + n_2 = 6$. Since both matrices have $nf(m,t) - 2$ columns, we have

$$nf(m,t) - 2 \geq \min \left( nf(n_1,t-2), nf(n_2,t-2) \right) \geq nf(n, t-2).$$

(5)

The last inequality follows from the fact that for fixed $m, nf(n,t)$ is an increasing function of $n$. Together, (5) and (6) give the desired inequality (4).

As a consequence of Theorem 5, we have

**Corollary 6.** Let $u_0^t = u_0$, $u_0^t = u_1$ and let, for all $t \geq 0$, $u_1^t$ be defined by

$$u_1^t = \max_{0 \leq s \leq t} \min \left( \frac{t}{2} \frac{t}{2} u_0^t, n(t - \log(1 - \delta)) \right).$$

Then, for any $t \in T$, $t \geq 5$,

$$nf(n,t) \leq \frac{1}{16} 16 \log \log t + O(1), \quad t \text{ fixed and } n \to \infty.$$ 

**Proof.** Use induction on $t$ and the well-known KNUTH upper bound [14] as an estimate for $H(m, t)$ in (4).

Corollary 6 gives a slight improvement on the upper bound on $nf(n,t)$ of Theorem 4, when $t$ is greater than or equal to 4. In Table 2 we list the values of $u_0^t$, $u_1^t$, and $u_0^t$ for $t = 3, 4, 5, 6, 8$ and 10.
<table>
<thead>
<tr>
<th>t</th>
<th>$t_e$</th>
<th>$u_e$</th>
<th>$u'_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.0642</td>
<td>0.3112</td>
<td>0.3112</td>
</tr>
<tr>
<td>4</td>
<td>0.0322</td>
<td>0.1467</td>
<td>0.1467</td>
</tr>
<tr>
<td>5</td>
<td>0.00916</td>
<td>0.0727</td>
<td>0.0643</td>
</tr>
<tr>
<td>6</td>
<td>0.00878</td>
<td>0.0316</td>
<td>0.0328</td>
</tr>
<tr>
<td>8</td>
<td>$7.068 \cdot 10^{-4}$</td>
<td>$6.821 \cdot 10^{-3}$</td>
<td>$7.662 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>10</td>
<td>$1.409 \cdot 10^{-4}$</td>
<td>$2.069 \cdot 10^{-3}$</td>
<td>$1.885 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 2.

1.4 Constructions for (Separable) $t$-Defect-Compatible Matrices with $t \geq 3$.

Many authors have considered the problem of constructing (separable) $t$-defect-compatible matrices $[1,3,4,11,10]$. In this section we describe two construction methods that, to our knowledge, yield the best results. The first one is due to Bauschbach [3]. This construction uses a small $t$-defect-compatible matrix to generate a larger one. So $t$ stays fixed, while the length $n$ grows. The second construction allows $t$ to grow proportionally with $n$ and is therefore used to determine a 'constructive' asymptotic lower bound on $n(n,p)$ for $p$ fixed, $0 < p < 1$ and $n \to \infty$. We also use this construction to derive some lower bounds on $n(n,t)$, for $t \leq 10$ and $t \geq 10$.

§ 1.4.1 A construction for $t$-defect-compatible matrices of length $n$, with $t < n$.

In this paragraph we describe the construction method, for $t$-defect-compatible matrices, found by Bauschbach in [3]. The adjustments necessary to make the resulting matrix separable are given. The construction uses a small $t$-defect-compatible matrix to generate a larger one. These small
matrices can, for instance, be constructed by the method of § 1.4.2.

**Construction.**
Let $A$ be a $n_0 \times n_0$ t-defect-compatible matrix, where $n_0$ is a prime power $n_0 = p^k$. Let $C$ be an $n_0$-ary linear MDS code with dimension $k$, $2 \leq k \leq \frac{4n_0}{3}$, and length $m = (k-1) \cdot \left(\frac{p^2}{2} - 1\right)$. Since $m > n_0$, these codes are easy to construct (see [14]). Let $B$ be the $m \times n_0$ matrix with as columns the codewords of $C$. Let $\phi : \mathcal{V}_0 \rightarrow \{\text{columns of } A\}$ be a bijection. Construct the $m \times n_0$ binary matrix $C$ by replacing each entry $b$ of $B$ by $\phi(b).

**Theorem 1.** The matrix $C$, constructed above, is a t-defect-compatible matrix.

**Proof.** To prove the t-defect-compatibility of $C$, let $C'$ be any $m_0 \times t$ submatrix of $C$ and let $d'$ be any binary vector of length $t$. We have to show that $d'$ is contained in the row set of $C'$. To prove this we go back to the code $B$. Let $b_j = (b_{1,j}, b_{2,j}, \ldots, b_{m,j})$ be that codeword of $B$ that corresponds to the $j$th column of $C$. Since every coordinate $b_{i,j}$, $1 \leq i \leq m$, is replaced by a column of the t-defect-compatible matrix $A$, we are done if we can show that there is an $i$, $1 \leq i \leq m$, such that

$$
\{b_{1,j}^2 \mid d_{i,j} = 0\} \cap \{b_{2,j}^2 \mid d_{i,j} = 1\} = \emptyset.
$$

(7)

From the t-defect-compatibility of $A$ we then have that $d'$ is contained in the row set of the submatrix $C'' = (b_{1,j}^2, b_{2,j}^2, \ldots, b_{m,j}^2)$ of $C'$.

So suppose that (7) does not hold for any $i \in \{1, 2, \ldots, m\};$ so $d' \neq 0, 1$. We calculate the sum

$$
\sum_{i \mid d_{i,j} = 0} \sum_{j \mid d_{i,j} = 1} d_{i,j}^2 b_{i,j}^2
$$

We will calculate...
in two different ways. Firstly, since (7) does not hold for any coordinate \( i \in \{1, 2, \ldots, n\} \), each coordinate contributes at most \( n_0(d') \cdot n_1(d') = 1 \) to the sum, where \( n_\lambda(d') := |\{j | d'_{ij} = \lambda\}|, \lambda = 0, 1 \). Hence

\[
\sum_{i | d'_i = 0} \sum_{j | d'_j = 1} d(i, j) \cdot n_0(d') \cdot n_1(d') = 1.
\]

Secondly, since the minimal distance of \( \mathcal{B} \) is equal to \( m - k + 1 \) (\( \mathcal{B} \) is MDS), we also have

\[
\sum_{i | d'_i = 0} \sum_{j | d'_j = 1} d(i, j) \cdot n_0(d') \cdot n_1(d') = (m - k + 1).
\]

Thus, we may conclude that

\[
n_0(d') \cdot n_1(d') = (m - k + 1) \leq n_0(d') \cdot n_1(d') - 1
\]

or equivalently

\[
m \leq (k + 1) \cdot n_0(d') \cdot n_1(d') \leq (k + 1)(n_0 + n_1) - 1.
\]

A contradiction with \( m \geq (k + 1)(n_0 + n_1) - 1 \).

\[\Box\]

The bounds on \( n(m, 1) \) and \( n(m, t) \), that result from this construction are so unattractive that we do not give them here. We confine ourselves to an example for \( t = 3 \) and refer the interested reader to [3].

**Example 4.** Let \( A \) be the \( 8 \times 8 \) 3-defect-compatible matrix with as codewords the codewords of the \( (4,1,2) \) binary code. Let \( n_1, n_2 \) denote the size of the 3-defect-compatible matrix after \( i \) successive applications of the above construction with maximal \( k_s \), so \( n_0 = 8 \) and \( n_0 = 4 \). Then we find the following values for \( n_1 \) and \( n_2 \), \( i = 1, 2, 3 \).
We see that the number $n_1$ grows excessively with respect to the number $n_2$, but nevertheless, it does not result in a lower bound on $n(2,3)$ of the form $n(2,3) \geq 2^{2n}$, a fixed.

Although, we feel that Behrend's construction is of little importance (it is too small compared to $n$) for the construction of additive codes, we adjusted the construction somewhat in order to make it yield (weak) separable $t$-defect-compatible matrices. A binary $n \times m$ matrix is called weakly separable if there exists a $n \times \lceil \log n \rceil$ submatrix of $A$ that has a different row. Matrices like this can also be used to define an additive code.

**Construction.**

Let $A$ be a $2^{n_0} \times n_0$ separable $t$-defect-compatible matrix, where $n_0$ is a prime power $2^{k_0}$. Let $B$ be a $n_0 \times m$ binary linear code with dimension $2^{k_0}n_0 - 1$ and a code length $n_0 - 1$. Let $E$ be the $n_0 \times n_0$ matrix with $1$s as codewords of $B$. Let the elements of $E$ be labelled by $a_1, a_2, \ldots, a_{n_0}$ and the columns of $A$ by $a_1, a_2, \ldots, a_{n_0}$ and let $s = \lceil \log m \rceil$. Assume $r_0 + s \leq n_0$ (this will always be the case). For any $v \in \{0,1,\ldots,m-1\}$ we define the vector $v \in \mathbb{F}_{n_0}^s$ by

$$v = (0,0,\ldots,0, v_0, v_0^2, v_0^4, v_0^8),$$

where $v = \sum_{i=1}^{s} v_i 2^{i-1}$.
And \( \Phi_{i_0} : \Phi_1(\mathcal{H}_{n}) \rightarrow \{\text{volumes of } A\} \) by

\[
\Phi_{i_0}(\mathcal{H}_{n}) = \sum_{k=0}^{n_i} \lambda_k \beta_{i,k}, \quad 1 \leq i \leq n, \quad n_i \geq 0,
\]

where \( \mathbf{1} \) is the all-one vector of length \( 2^n \). Construct the \( m \times 2^n \mathcal{H}_{n} \)-binary matrix \( \mathbf{B} \) by replacing each entry \( b \) in the \( v \)th row of \( \mathbf{B} \) by \( \Phi_{v}(\mathcal{H})( \mathcal{B} ) \), \( v \in \{0, 1, \ldots, m-1\} \).

**Theorem 3.** The matrix \( \mathbf{C} \), defined above, is a weak separable \( \mathcal{A} \)-defect-compatible matrix, if \( r_0 = r_0 \).

**Proof.** Since the entries in each row of \( \mathbf{B} \) are mapped on the columns of a \( \mathcal{A} \)-defect-compatible matrix, the \( \mathcal{A} \)-defect-compatibility of the matrix \( \mathbf{C} \) is a direct consequence of the proof of Theorem 7.

To prove the weak separability of \( \mathbf{C} \) we consider the codewords \( \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m \) of \( \mathbf{B} \). From the separability of \( \mathbf{A} \) and the definition of \( \Phi_{i,v} \), \( v \in \{0, 1, \ldots, m-1\} \), we immediately see that the \( r_0 \times r_0 \) (or \( r_0 \times 1 \)) submatrix of \( \mathbf{C} \) which corresponds with these codewords, consists of all different rows.

\( \mathbf{C} \) is a generalization of a construction method found by Bennett in [10].

From Corollary 2 we have that, for any \( p \in [0, 1] \) and \( n \) sufficiently large, there exists an additive code of length \( n \) that is capable of correcting all word defects of multiplicity \( n^p \) or less, for a rate \( n = 1 - p \). However, the question remains: "How to construct such a code?". In this section we describe a construction method for separable \( \mathcal{A} \)-defect-compatible matrices that gives a partial solution.
to this problem.

As a first reaction we and many others with us tried to solve this problem with the help of linear codes. Let $C$ be a binary $[n,k]$ code for which the dual code has minimum distance $t=1$. Then the $2^n \times n$ matrix $C$ with as rows the codewords of the code $C$ is easily seen to be a separable $t$-defect-compatible matrix. From the Gilbert-Varshamov bound one easily derives that this construction yields the following asymptotic upper bound on $R(n, np)$,

$$R(n, np) \leq n(1 - H(p) + o(1)),$$

for $p$ fixed, $0 < p < \frac{1}{2}$, and $n \to \infty$. \[ (8) \]

and so

$$R(n, np) \leq n(1 - H(p) + o(1)),$$

for $p$ fixed, $0 < p < \frac{1}{2}$, and $n \to \infty$.

However, not only the bound is poor, it is also cheating; no one as yet has found a construction of a family of binary linear codes that realizes the premises of the Gilbert-Varshamov bound. At present we only know that such families of good codes exist and can, for instance, be found within the class of Coppers codes [14, Ch. 12].

To our surprise, the following observation shows that it is rather simple to find such a construction for $t$-defect-compatible matrices; the resulting upper bound for $R(n, np)$ is even sharper than $(8)$. Let $C$ be the matrix with as rows all binary words of length $n$, which have weight $\left\lceil \frac{n}{2} \right\rceil$ or weight $n - \left\lfloor \frac{n}{2} \right\rfloor$. It is clear that $C$ is a $t$-defect-compatible matrix for any $t, 0 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor$. The asymptotic upper bound on $R(n, np)$, $0 < p < \frac{1}{2}$, that results from this construction reads

$$R(n, np) \leq n\left(1 - \frac{1}{2} + o(1) \right), \text{where } p \text{ is fixed, } 0 < p < \frac{1}{2} \text{ and } n \to \infty.$$ 

Although, this bound "improves" $(8)$, it is not really sharp. The resulting $t$-defect-compatible matrices, however, may be of interest for the generation of exhaustive test patterns, because of the simple
structure, the resulting test sets can be effectively implemented (see [17]).

Since we believe that \( \lceil \pi(n,k) \rceil \) is about as big as \( n(n,1) \), for all values of \( n \) and \( k \), \( 0 < k < n \), the simplicity of the above construction convinced us, that it must be possible to find a similar construction method for separable \( t \)-defect-compatible matrices. A generalization of a construction method for separable \( t \)-defect-compatible matrices found by postmanov in [10], does the trick.

**Construction.**

Let \( A \) be a binary \( [2^m,k,m] \) code with \( I(2^m) \) and minimum distance \( d \), \( [I(2^m-2),1:2^m/(2^m-1)] \). Let \( G \) be a generator matrix of \( A \) with the all-one vector as top-row. Let \( R \) be a parity check matrix of an \( [n,k,k(2^m-1)] \) binary even code which has the all-one vector as top-row. We define the \( 2^m \times n \) matrix \( C \) by

\[
C = \begin{bmatrix} \frac{t^m}{2^m} & \ldots & \frac{1}{2^m} \\ \ldots & \ldots & \ldots \\ \frac{1}{2^m} & \ldots & \frac{t^m}{2^m} \end{bmatrix}
\]

where \( ^tR \) is the complementary matrix of \( R \).

**Theorem 1.** The matrix \( C \) defined above is a \( t \)-defect-compatible matrix if \( t \leq 3 \). If \( C \) contains the generator matrix of \( 2^m \) as a submatrix, then \( C \) can be made separable.

**Proof.** To prove the \( t \)-defect-compatibility, it suffices to show that for any subset \( J \subseteq \{1,2,\ldots,n\} \) with \( |J| = t \) and any \( \mathbf{g} \in \mathbb{Z}_2^k \), there is an \( \mathbf{h} \in \{1,2,\ldots,2^m\} \) such that

\[
\sum_{i \in J} C_{i,j} = x \quad \text{or} \quad j \neq g,
\]

where \( H \) is the \( k \times t \) matrix that consists of those columns of \( H \) which have a column index belonging to \( J \) and \( e_{\mathbf{g}} \) is the \( t \)-th basis vector of \( \mathbb{Z}_2^k \).
Suppose there is a $J \subseteq \{1,2,\ldots,n\}$, $|J| = r$ and a $g \in \mathbb{F}_2^n$ such that (9) does not hold for any $k \in \{1,2,\ldots,2^r\}$, then

$$\sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( g \cdot h_j \right) \right)^2 = 2^r - 2.$$ 

So

$$\sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( g \cdot h_j \right) \right)^2 = 2^r - 2 \quad \text{if} \quad \text{wt}(g) = 0 \mod 2. \tag{10}$$

On the other hand, we have

$$\sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( g \cdot h_j \right) \right)^2 = \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( g \cdot h_j \right) \right)^2 \quad \text{if} \quad \text{wt}(g) \neq 0 \mod 2.$$

$$= \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( g \cdot h_j \right) \right)^2 = \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( \sum_{g \in \mathbb{F}_2^n, \quad g \in J} \left( g \cdot h_j \right) \right)^2 \quad \text{if} \quad \text{wt}(g) = 0 \mod 2. \tag{11}$$

The inequality is consequence of the fact, that for any $g \in \mathbb{F}_2^n \setminus \{0\}$ with $\text{wt}(g) = 0 \mod 2$, the word $g \cdot h_j$ is $(g,0) \in A\{0,1\}$. For, since $A$ has the all-one vector as top-row, $\text{wt}(g) = 0 \mod 2$ and $g \neq 0$, the first coordinate of $g \cdot h_j$ is equal to 0 and $g \cdot h_j \neq 0$. So, since the top-row of $G$ is also 1, we may conclude that $g \cdot h_j \in A\{0,1\}$. Together (10) and (11) give
\[ z^k - d \leq \frac{2^k - 2}{2^{k-1} - 1}, \]

or equivalently
\[ d \geq \frac{2^k - 2}{2^{k-1} - 1} \cdot \frac{2^k - 2}{2^{k-1} - 1}. \]

This is a contradiction with \( d \leq \frac{2^k - 2}{2^{k-1} - 1} \) if \( t \neq 2 \). So \( C \) is a \( t \)-defect-compatible matrix if \( t \neq 3 \).

The separability of \( C \), when \( G \) contains the generator matrix of \( BM(1, \ell) \) as a submatrix, is obvious.

\[ \Theta \]

Remark: For the case \( t = 3 \), the above construction can somewhat be simplified. Let \( A \) be a binary \( (m, n, [2^m - 1]) \)-code with the property that for all \( \bar{x} \in \mathbb{F}_2 \), \( \bar{x} A \) also \( \bar{x} A \). Let \( A_0 \), respectively \( A_1 \) denote the matrix with an columns the codewords of \( \bar{x} A \) of which the first coordinate is equal to 0 respectively equal to 1. Then the \( 2m \times n \) matrix \( C \) defined by

\[ C = \begin{pmatrix} A_0 \\ A_1 \end{pmatrix} \]

is a \( 3 \)-defect-compatible matrix. When \( m = 2^k \) and \( A \) contains \( BM(1, \ell) \) as a subcode, the matrix \( C \) can be made separable. This is in essence the construction for \( 3 \)-defect-compatible matrices Knesslkey gave in [10].

In order to make the above construction work we have to generate the matrices \( G \) and \( H \) which are mentioned there. The matrix \( C \) is the most important one. Suitable candidates for \( G \) are the generator matrices of the codes we describe in Theorem 10. For a proof of this theorem and construction of these codes we refer to [14].

**Theorem 10.** Let \( \ell = 2^{k-1} - 1 \) and let \( r \) be any number in the range \( 1 \leq r \leq \ell \). Then there exists two
Let \( A \) be one of the two \([2^r, r(\ell - 1 + 2) + 1, 2^{r - 1} - 2^{r - 1 - 1}]\) codes of Theorem 10. Then \( 1 \in A \), since \( H(1,r) \subseteq A \). Since \( 2^{r - 1} - 2^{r - 1 - 1} \geq 1 \), if \( \ell \geq 1 \), we have, according to Theorem 9, that the existence of an \([n, n-(\ell - 1 + 2)r - 1, 2^{\ell + 1}]\) binary code, \( 3 \leq \ell + 1 \), gives rise to the existence of a \( 2^{\ell + 1} \times n \) separable \( t \)-defect-compatible matrix.

In Table 3 we give some lower bounds on \( n(r,t) \), for moderate values of \( r \) and \( t \), which result from this construction. To generate the matrices \( \mathcal{C} \), we did not only use the codes from Theorem 10, but we also used codes that result from Winkelman's construction method, which we described in [16]. For the matrices \( \mathcal{H} \) we used a table search [14,20]. The letter \( \mathbf{h} \) in the upper left corner of an entry indicates that the corresponding lower bound on \( n(r,t) \) is attained by a linear code whose dual code has minimum distance \( t + 1 \) and dimension \( r \). The letter \( \mathbf{k} \) indicates that this lower bound is attained by the construction of Rusnotsov [10].
<table>
<thead>
<tr>
<th>i</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>h</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>h</td>
<td>5</td>
<td>h</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>h</td>
<td>7</td>
<td>h</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>h</td>
<td>9</td>
<td>h</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>2/3</td>
<td>26</td>
<td>h</td>
<td>12</td>
<td>h</td>
<td>13</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2/11</td>
<td>513</td>
<td>h</td>
<td>15</td>
<td>h</td>
<td>16</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>2/2</td>
<td>268</td>
<td>h</td>
<td>17</td>
<td>h</td>
<td>18</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2/13</td>
<td>513</td>
<td>h</td>
<td>19</td>
<td>h</td>
<td>20</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>2/21</td>
<td>257</td>
<td>h</td>
<td>21</td>
<td>h</td>
<td>22</td>
<td>-</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3. Lower bounds on \( m(x,t) \)
We conclude this section with the promised "constructive" lower bound for $X(n,p)$. Let $G$ be a \((2\ell + 1)(\ell - 1 + 2) + 1 \times 2\ell + 1\) generator matrix of one of the codes mentioned in Theorem 10 with the all-one vector as top-row. Let $H$ be any \((2\ell + 1) (\ell - 1 + 2) + 1 \times (\ell - 1 + 2) + 1\) regular binary matrix with the all-one vector as top-row. Then, from Theorem 9, the matrix $C$ defined by

\[
C = \begin{bmatrix}
\frac{\partial G}{\partial \mathbf{w}} \\
\frac{\partial G}{\partial \mathbf{h}}
\end{bmatrix},
\]

is a separable \((1 + 1)\)-defect-compatible matrix of size

\[
2 \ell^2 + 2 \times (\ell^2 + 1)(\ell - 1 + 2) + 1.
\]

Let $n = (2\ell + 1)(\ell - 1 + 2) + 1$, then the above construction shows

\[
x(n,1+1) \leq 2\ell + 2.
\]

Now take $1 = \ell - k$, $k$ fixed and let $\ell$ tend to infinity. Then, since

\[
\lim_{\ell \to \infty} \frac{k \cdot n + 1}{n} = \lim_{\ell \to \infty} \frac{k \cdot \ell + 1}{(2\ell + 1)(\ell - 1 + 2)} = \frac{1}{\pi(\ell + 2)} \quad \text{we find, for any} \quad k \in \mathbb{N} \cup \{0\}
\]

\[
x(n,1+1) \leq x\left\lfloor \frac{1}{\pi(k + 2)} - o(1) \right\rfloor, \quad k \text{ fixed and } n \to \infty.
\]

Since $x(n,t)$ is an increasing function of $n$ if $t$ is fixed, we constructively showed

\[
x(n,\eta) \leq x\left(n(2p + o(1)), p \text{ fixed}, 0 < p \leq 1 \right), \quad n \to \infty.
\]

Hence, for any $p$, $0 < p \leq 1$, the above construction can be used to generate
a family of additive codes of length \( n \), \( n \in \mathbb{N} \), that are capable of correcting all word defects of multiplicity up or less, for a rate \( R(n,p) \), for which

\[
R(n,p) = \frac{2^p}{n}.
\]

### 1.5 Generalized Partitioned Linear Block Codes

§ 1.5.1 Partitioned linear block codes

In [19], Tsybakov introduces the problem of coding for binary computer memory units with both defects and random errors. The locations and natures of the defects are assumed to be known at the encoder but not at the decoder. Recall from Section 1.1 that such a \( n \)-cell memory unit is defined by

\[
Y = [x \oplus d] \in \mathbb{Z}_2^n,
\]

where \( x \in \mathbb{F}_2^n \) is a channel input word, \( \gamma \in \mathbb{F}_2^n \) a channel output word, \( d \) a word defect \( d \in \mathbb{Z}_2^n \) and \( \gamma \) an error vector of weight \( \leq \) or less. To solve this problem,

Tsybakov defines the codewords of a binary \((n, k, d - 2^r + 1)\) code \( C \) as channel input words. The code \( C \) is partitioned into a number of subcodes \( C_0, C_1, \ldots, C_{2^r - 1} \), each of which forms a \( t \)-defect-compatible set. He uses the defect information, known at the encoder, to assign to each message \( x \in \{0, 1, \ldots, M - 1\} \) a channel input word \( x \in C_h \) which is compatible with \( x \). The decoder, receiving \( y = (x \oplus d) \oplus e \), sees that

\[
d(y, C_h) < d(y, C_j),
\]

for all \( j \neq h \), and so recovers the message \( x \) correctly. The rate \( R \) is defined by \( R = \log \frac{M}{N} \).

Since linear block codes are very suited for this coding strategy, Tsybakov introduces the concept of partitioned linear block codes.
(in [10] these codes are called matched adjacent). We give a formal definition.

An \((n, x, k_1, k_2)\) partitioned linear block code is a pair of linear codes \(C_0 \subset \mathbb{F}_2^n\), \(C_1 \subset \mathbb{F}_2^k\) of dimension \(k_0\) and \(k_1\), respectively such that \(C_0 \cap C_1 = \{0\}\). The direct sum \(C = C_0 \otimes C_1 = \{u_0 \oplus u_1 \mid u_0 \in C_0, u_1 \in C_1\}\) forms the set of channel input words. The partition of \(C\) into subcodes is described by

\[
C = \bigcup_{C_0 \cap C_1} (C_0 \otimes C_1)\,
\]

The rate \(R\) is equal to \(k_1/n\).

To define an encoding \(\Phi\) and a decoding \(\Psi\), \(\mathbb{F}_2^n \rightarrow C\) and \(\mathbb{F}_2^n \rightarrow \mathbb{F}_2^k\), we need some more definitions.

Let \(G_0\) and \(G_1\) be generator matrices for \(C_0\) and \(C_1\), respectively. Let \(S\) be a parity check matrix for \(C = C_0 \otimes C_1\), and let \(R_1\) be any \(k_1 \times n\) binary matrix such that \(C_1^T R_1^{-1} = 0\) and \(G_0 R_1^T = 0\).

We are ready to define the encoding and decoding functions \(\Phi\) and \(\Psi\), respectively.

Take the message set \(U\) equal to \(\mathbb{F}_2^n\) and let, for any \(u \in \mathbb{F}_2^n\) and any \(a \in \mathbb{F}_2^k\), \(\bar{u}: (u, a)\) be a specified vector of \(\mathbb{F}_2^n\) (see the proof of Theorem 11).

The encoding \(\Phi\), \(\Phi : \mathbb{F}_2^n \times \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n \oplus C\), is defined by

\[
\Phi(u, a) = u \oplus G_1 \cdot \bar{u}(u, a) \cdot G_0^	op
\]

The decoding \(\Psi\), \(\Psi : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^k\), is defined by

\[
\Psi(y) = (y \oplus \hat{a}) \cdot R_1^T
\]

where \(\hat{a} \in \mathbb{F}_2^k\) is chosen to minimize \(v(c)\) subject to \(A(\hat{a}) = u\).
the syndrome of \( y \) with respect to the code \( C \). The vector \( \hat{e} \) is an estimate for the error \( e \) in (1).

For any \([n,k_0,k_1]\) partitioned linear block code \((C_0,C_1)\) a pair of minimum distances \((d_0^*,d)\) is adjoined, where \( d_0^* \) is the minimum distance of the code \( C_0 \) and \( d_0^* \) is the minimum distance of the dual code \( C_0^* \) of \( C_0 \).

**Theorem 11.** Let \((C_0,C_1)\) be an \([n,k_0,k_1]\) partitioned linear block code with minimum distance pair \((d_0^*,d)\). Then \((C_0,C_1)\) is capable of correcting all word defects of multiplicity \( t \) or less and random errors of weight \( s \) or less. If \( t < d_0^* \) and \( 2s < d \).

**Proof.** For any \( y \in \mathbb{F}_2^{k_1} \) and \( g \in C_0^n \) we take \( g(y,g) \) equal to \( g \in \mathbb{F}_2^{k_1} \) such that \( g_0 \) is compatible with the word defect \( \hat{d} \in \mathbb{F}_2^t \) defined by

\[
\hat{d} = \begin{cases} 
0 & \text{if } d_1 = 0, \\
(0^t)_1 \odot d_1 & \text{if } d_1 = 1, 2, 3, \ldots , m.
\end{cases}
\]

Since any \( t \)-columns of \( g_0 \) are linearly independent this is possible.

With this choice of \( g(y,g) \) it is clear, from the definitions of \( \hat{d} \) and \( \bar{y} \), that \((C_0,C_1)\) is indeed a \( t \)-defect- \( s \)-error-correcting code.

**Example 5.** Let \((C_0,C_1)\) be the \([7,1,1]\) partitioned linear block code defined by

\[
C_0 = \{(1,1,1,1,1,0,0)\} \quad \text{and} \quad C_1 = \begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}.
\]
Then \((C_2, C_1)\) has a minimum distance pair \((2,3)\). Note that \(C\) is the \([7,4,3]\) single error-correcting encoding code. We can take \(n\) and \(d_1\) equal to
\[
\begin{bmatrix}
1 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}, \quad \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
Let \(u = (1,0,1)\) be the message to be stored in a computer memory unit with word defect \(d = (5,5,0,0,0,5,0)\) and error vector \(e = (0,0,1,0,0,0,0)\).

**Encoding:** To store the message \(u = (1,0,1)\) we first compute the word defect \(d'\) defined in the proof of Theorem 11 and the vector \(x(u, d)\).

We find \(d' = (5,5,1,0,0,5,0)\) and so \(x(u, d) = (1)\). Hence
\[
x = x(u, d) \otimes x(d, d) =
= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
\]

**Decoding:** To decode retrieve the vector \(y = (1,0,1)\) \(e = (0,1,0,0,0,1,0,1)\) from the memory unit and calculate the syndrome \(z = \gamma y^T\). We find \(z = (1,0,1)\). Since \(z\) is equal to the fourth column of \(E\), the decoder estimates \(e\) by \(\hat{e} = (0,0,0,1,0,0,0,3) = e\). Hence
\[
\gamma y = (y \otimes e) \otimes d = (1,0,1) - e.
\]

§ 1.5.2 Generalized partitioned linear block codes

In the coding of an \([n, k, d]\) partitioned linear block code \((C_2, C_1)\), the entire code \(C_2\) is used for making the defects of the memory unit. As we have seen in § 1.4.2, this is not always necessary. The class of generalized partitioned linear block codes makes advantage of this observation. We start with a definition.

An \([n, k_0, k_1, k_2]\) generalized linear block code consists of a
triple of binary linear codes $C_0, C_1, C_2 \subseteq \mathbb{F}_2^n$ of dimension $k_0$, $k_1$, and $k_2$ respectively, such that $C_j \subseteq C_j \subseteq \{0, 1\}, j \in \{0, 1, 2\}$, and a binary code $I$ of length $k_0 + k_1$, which is separable on the first $k_0$ coordinate places. The direct sum $C = C_0 \oplus C_1 \oplus C_2$ forms the set of channel input words. Let $G_0, G_1$ and $G_2$ be generator matrices of the codes $C_0, C_1$, and $C_2$, respectively. Then $C$ is partitioned into

$$C = \bigcup_{\lambda \in \mathbb{F}_2^2} \{ \lambda \# \left( \begin{array}{c} G_0 \\ -G_1 \\ G_2 \end{array} \right) \}, \quad \lambda \in \mathbb{F}_2^2,$$

The rate $R$ is equal to $(k_1 + k_2)/n$.

Let $\delta_0$ be a parity check matrix of the direct sum $C = C_1 \oplus C_2 \oplus C_3$. Let $\delta_0$ be any $n \times n$ matrix such that $G_0 \delta_0^T = I_{k_0}$ and

$$\left( \begin{array}{c} 0 \\ 0 \\ \delta_0 \end{array} \right), \quad 0 = k_1 + k_2 + k_0.$$

Let $\delta_{1,2}$ be any $(k_1 + k_2) \times n$ matrix such that

$$\left( \begin{array}{c} \delta_1 \\ \delta_2 \end{array} \right), \quad \delta_{1,2} \delta_{1,2}^T = k_1 + k_2.$$

Then we can define the encoding

$$(u, \delta) \mapsto ku \in \{0, 1\}^{n} = \mathcal{C}$$

and the decoding

$$(y, \delta) \mapsto ku = \mathcal{C}$$

where $x(u, \delta)$ is a specified codeword of $I$ (see the proof of Theorem 12) and

$$y(u) = ku \in \{0, 1\}^{n} \delta_{1,2}^T,$$

where $z$ is chosen to minimize $w(x)$ subject to $z \delta = \bar{x} = \delta y$. And $\alpha$ is that codeword of $I$ that on the first $k_0$ coordinate places is equal to the vector $(y \delta) \delta_{1,2}^T$.

**Theorem 12.** Let $(C_0, C_1, C_2, I)$ be a $(n, k_0, k_1, k_2)$ generalized partitioned
linear block code, for which

( i) the direct sum $C = C_0 \oplus C_1 \oplus C_2$ has minimum distance $d = 2s + 1$,
( ii) the dual code of $C_0 \oplus C_1$ has minimum distance $d'_{\perp} = 2 \left\lfloor \frac{k_0 - 1}{2} \right\rfloor$,
( iii) $1 \in C_0$ and $C_0$ contains $1$ as top row,
( iv) the $(k_0 - 2)_{k_0 - 1}$ matrix, with as columns those codewords of $C$ that have a $1$ as first coordinate, is the generator

matrix of a binary $[2^{k_0 - 1}, k_0 + k_1, (2^{k_0 - 2} - 2)]_{k_0 - 1}$ code and
( v) for any $\mathbf{z} \in \mathcal{Z}$ also $1 \in \mathcal{Z}$.

Then $(C_0^L, C_1^L, C_2^L)$ is a $t$-defect, $s$-error-correcting code.

**Proof.** From the properties (i)-(v) and Theorem 9 of § 1.4.2 we have that $(\mathcal{Z}_{C_i^L}^L, 1 \in \mathcal{Z})$ forms a separable $t$-defect-compatible set. Hence for any $y \in \mathbb{F}_2^{k_1 + k_2}$ and any $\mathbf{z} \in \mathcal{Z}^L$ there is a $\mathbf{z} \in \mathcal{Z}$ such that $$(\mathcal{Z}_{C_i^L}^L) \quad \text{is compatible with the word defect } \mathbf{d},$$ defined by

$$d'_i = \left\{ \begin{array}{ll}
0 & \text{if } d_i = 0, \\
1 + \left\lfloor \frac{d_i}{2} \right\rfloor & \text{if } d_i = 1, i = 1, \ldots, n.
\end{array} \right.$$ 

Choose $\pi(n, d)$ to be equal to $\mathbf{z}$. With this choice for $\pi(n, d)$ and the definitions of $\mathcal{Z}$ and $\mathcal{V}$, the assertion of Theorem 12 is immediate.

With the help of primitive binary BCH codes of length $n = 31, 63, 127$ and 255 we constructed the following $(n, k_0, k_1, k_2)$ generalized partitioned $t$-defect, $s$-error-correcting linear block codes listed in Tables 4, 5, 6 and 7. The rate of such a code is equal to $(k_1 + k_2)/n$. The rate of the corresponding partitioned $t$-defect, $s$-error-correcting code of the same length $n$, given in [7], is equal to $k_2/n$ or $(k_3 + 1)/n$. So the gain in
state is at least \((k_n-1)/n\).

On the other hand, the encoding process of a generalized partitioned linear block code is more complicated than the encoding process of a partitioned linear block code. In both cases, the determination of the vector \(g[w, d]\) (see Theorems 11 and 12), amounts to solving an equation like

\[ g \circ d' = g' \]

where the matrix \(G'\) and the vector \(g'\) are directly determined by the vector \(g\), the word defect \(d\) and the code used. However, in the case of a partitioned linear block code any solution \(g\) will do, while in the case of a generalized partitioned linear block code one has to find a solution \(g\) of the above equation within the set \(I\). This will take more time.

The decoding process is in both cases the same.
Table 4. Generalized partitioned t-defect, e-error-correcting linear block codes of length $n = 31$.

<table>
<thead>
<tr>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$t$</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$t$</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>11</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>15</td>
<td>5</td>
<td>1</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>2</td>
<td>16</td>
<td>5</td>
<td>2</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>15</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>11</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>15</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>11</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>1</td>
<td>15</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 5. Generalized partitioned t-defect, e-error-correcting linear block codes of length $n = 31$.

<table>
<thead>
<tr>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$t$</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$t$</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>38</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>44</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>5</td>
<td>32</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>30</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>26</td>
<td>4</td>
<td>12</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>32</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>23</td>
<td>4</td>
<td>12</td>
<td>7</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>29</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>17</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>23</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>11</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>17</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>45</td>
<td>5</td>
<td>12</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>11</td>
<td>3</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>40</td>
<td>5</td>
<td>17</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>11</td>
<td>9</td>
<td>4</td>
<td>32</td>
<td>5</td>
<td>13</td>
<td>6</td>
<td>38</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>13</td>
<td>9</td>
<td>4</td>
<td>26</td>
<td>5</td>
<td>13</td>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>15</td>
<td>9</td>
<td>4</td>
<td>21</td>
<td>5</td>
<td>13</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>44</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td>17</td>
<td>5</td>
<td>6</td>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>30</td>
<td>4</td>
<td>2</td>
<td>11</td>
<td>5</td>
<td>10</td>
<td>6</td>
<td>1</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>$k_0$</td>
<td>$k_1$</td>
<td>$k_2$</td>
<td>$s$</td>
<td>$k_0$</td>
<td>$k_1$</td>
<td>$k_2$</td>
<td>$s$</td>
<td>$k_0$</td>
<td>$k_1$</td>
<td>$k_2$</td>
<td>$s$</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>110</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>63</td>
<td>4</td>
<td>7</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>105</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>7</td>
<td>56</td>
<td>4</td>
<td>9</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>98</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>49</td>
<td>4</td>
<td>10</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>91</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>42</td>
<td>4</td>
<td>11</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>84</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>35</td>
<td>4</td>
<td>13</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>77</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>28</td>
<td>4</td>
<td>15</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>70</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>21</td>
<td>4</td>
<td>15</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>63</td>
<td>3</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>105</td>
<td>5</td>
<td>7</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>56</td>
<td>3</td>
<td>9</td>
<td>10</td>
<td>5</td>
<td>96</td>
<td>5</td>
<td>2</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>49</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>91</td>
<td>5</td>
<td>3</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>42</td>
<td>3</td>
<td>11</td>
<td>10</td>
<td>5</td>
<td>84</td>
<td>5</td>
<td>4</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>35</td>
<td>3</td>
<td>12</td>
<td>10</td>
<td>5</td>
<td>77</td>
<td>5</td>
<td>5</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>28</td>
<td>3</td>
<td>13</td>
<td>10</td>
<td>5</td>
<td>70</td>
<td>6</td>
<td>6</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>21</td>
<td>3</td>
<td>14</td>
<td>10</td>
<td>5</td>
<td>63</td>
<td>7</td>
<td>7</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>14</td>
<td>3</td>
<td>15</td>
<td>10</td>
<td>5</td>
<td>56</td>
<td>6</td>
<td>9</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>16</td>
<td>10</td>
<td>5</td>
<td>49</td>
<td>6</td>
<td>10</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>17</td>
<td>10</td>
<td>5</td>
<td>42</td>
<td>6</td>
<td>11</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>18</td>
<td>10</td>
<td>5</td>
<td>35</td>
<td>6</td>
<td>13</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>19</td>
<td>10</td>
<td>5</td>
<td>28</td>
<td>5</td>
<td>15</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>21</td>
<td>5</td>
<td>15</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>21</td>
<td>10</td>
<td>5</td>
<td>14</td>
<td>5</td>
<td>6</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>22</td>
<td>10</td>
<td>5</td>
<td>91</td>
<td>6</td>
<td>4</td>
<td>14</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>23</td>
<td>10</td>
<td>5</td>
<td>84</td>
<td>6</td>
<td>3</td>
<td>14</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 6. Generalized partitioned t-defect, s-error-correcting linear block codes of length n = 127.
<table>
<thead>
<tr>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$r_0$</th>
<th>$r_1$</th>
<th>$t_0$</th>
<th>$t_1$</th>
<th>$k_2$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>238</td>
<td>3</td>
<td>1</td>
<td>63</td>
<td>7</td>
<td>14</td>
<td>222</td>
</tr>
<tr>
<td>63</td>
<td>230</td>
<td>3</td>
<td>2</td>
<td>63</td>
<td>7</td>
<td>14</td>
<td>214</td>
</tr>
<tr>
<td>63</td>
<td>222</td>
<td>3</td>
<td>3</td>
<td>63</td>
<td>7</td>
<td>14</td>
<td>206</td>
</tr>
<tr>
<td>63</td>
<td>214</td>
<td>3</td>
<td>4</td>
<td>63</td>
<td>7</td>
<td>14</td>
<td>198</td>
</tr>
<tr>
<td>63</td>
<td>206</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>190</td>
</tr>
<tr>
<td>63</td>
<td>198</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>182</td>
</tr>
<tr>
<td>63</td>
<td>190</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>174</td>
</tr>
<tr>
<td>63</td>
<td>182</td>
<td>3</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>166</td>
</tr>
<tr>
<td>63</td>
<td>174</td>
<td>3</td>
<td>9</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>158</td>
</tr>
<tr>
<td>63</td>
<td>166</td>
<td>3</td>
<td>10</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>150</td>
</tr>
<tr>
<td>63</td>
<td>158</td>
<td>3</td>
<td>11</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>142</td>
</tr>
<tr>
<td>63</td>
<td>150</td>
<td>3</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>134</td>
</tr>
<tr>
<td>63</td>
<td>142</td>
<td>3</td>
<td>13</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>126</td>
</tr>
<tr>
<td>63</td>
<td>134</td>
<td>3</td>
<td>14</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>118</td>
</tr>
<tr>
<td>63</td>
<td>126</td>
<td>3</td>
<td>15</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>110</td>
</tr>
<tr>
<td>63</td>
<td>118</td>
<td>3</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>102</td>
</tr>
<tr>
<td>63</td>
<td>110</td>
<td>3</td>
<td>17</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>94</td>
</tr>
<tr>
<td>63</td>
<td>102</td>
<td>3</td>
<td>18</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>86</td>
</tr>
<tr>
<td>63</td>
<td>94</td>
<td>3</td>
<td>19</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>78</td>
</tr>
<tr>
<td>63</td>
<td>86</td>
<td>3</td>
<td>20</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>70</td>
</tr>
<tr>
<td>63</td>
<td>78</td>
<td>3</td>
<td>21</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>62</td>
</tr>
<tr>
<td>63</td>
<td>70</td>
<td>3</td>
<td>22</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>54</td>
</tr>
<tr>
<td>63</td>
<td>62</td>
<td>3</td>
<td>23</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>46</td>
</tr>
<tr>
<td>63</td>
<td>54</td>
<td>3</td>
<td>24</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>38</td>
</tr>
<tr>
<td>63</td>
<td>46</td>
<td>3</td>
<td>25</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>30</td>
</tr>
<tr>
<td>63</td>
<td>38</td>
<td>3</td>
<td>26</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>22</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>3</td>
<td>27</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>63</td>
<td>22</td>
<td>3</td>
<td>28</td>
<td>10</td>
<td>7</td>
<td>14</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 7. Generalised partitioned t-defect, a-error-correcting linear block codes of length n = 255.
REFERENCES


CHAPTER 2.

TWO CONSTRUCTIONS FOR CONSTANT WEIGHT CODES

2.1 INTRODUCTION

In this chapter we discuss two construction methods for constant weight codes, which improve several of the best known lower bounds on $A(n,d,w)$ in [1,3,6], where $A(n,d,w)$ denotes the maximal cardinality of any binary constant weight code of length $n$, minimum distance $d$ and constant weight $w$. Although, our interest in the function $A(n,d,w)$ finds its origin in the fact that this function plays an important role in the determination of upper bounds on $A(n,d)$ (e.g., Johnson bound, linear programming bound), the function $A(n,d,w)$ is also interesting in its own right. Besides the obvious connection with t-designs and Hadamard matrices we would like to mention the application of constant weight codes as a set of protocol sequences for the multiple-access collision channels without feedback [5,7].

The first construction method, treated in Section 2.2, results from proving a generalization of the well-known Johnson upper bound on $A(n,d,w)$. Unlike the generalization, the resulting construction does improve several of the best known results on $A(n,d,w)$ in [1,3,6]. A table of improved results is given.

In Section 2.3 we treat a construction method for constant weight codes with minimum distance $4$. In order to make this construction work, one needs to partition the set $[n]$, $n \in \mathbb{N}$ with $w \in \mathbb{N}$ into as small as possible number of constant weight codes with minimum distance $4$, for $n \geq 6n - 1$ or $6(n+1)$ and $w \in \mathbb{N}$ this last problem is equivalent to that of determining a packing of Steiner triple systems of order $n$. Since this construction method results in many improvements of the lower bounds on
A(n,4,w,n \geq 24), given in [1,3,8], a revised table of the function A(n,4,w,n \geq 24), is included.

2.2 A GENERALIZATION OF THE JOHNSON BOUND FOR CONSTANT WEIGHT CODES

Let A(n,26,w) denote the maximum number of codewords in any binary code of length n, minimum distance 2\ell and constant weight w. The following upper bound on A(n,26,w) is well known [4].

**Theorem 1.** (Johnson bound)

\[
A(n,26,w) \leq \sum_{\ell=1}^{w} A(n-\ell,26,w-\ell) \leq \sum_{\ell=1}^{w} A(n-\ell,26,w-\ell).
\]

Applying Theorem 1 k times we obtain the following bound:

\[
A(n,26,w) \leq \left( \sum_{\ell=1}^{w} A(n-k,26,w-k) \right) \leq \left( \sum_{\ell=1}^{w} A(n-k,26,w-k) \right).
\]

At the International Workshop "Convolutional Codes Multi-User Communication" Zinoviev [8] presented the following generalization of the Johnson bound [1].

**Theorem 2.** For any integers k and \ell with 0 \leq \ell \leq n, the following inequality holds

\[
A(n,26,w) \leq \left( \binom{w}{\ell} A(n-k,26,w-\ell) \right).
\]

where w = \ell - k if \ell \leq k/2 and w = -k + \ell if \ell \geq k/2.

If we take \ell = k in Theorem 2 we get the Johnson bound (1).
We now give a further improvement of the Johnson bound stated in the next theorem.

**Theorem 3.** For any two integers \( k \) and \( \ell \) with \( 0 \leq \ell \leq k \leq n \), we have

1. \( A(n, 2\ell, w) \leq \binom{n}{k} \frac{A(n-k, 2(\ell - \ell), w - \ell)}{\binom{2\ell}{\ell}} \) if \( \ell \leq n/2 \)

and

2. \( A(n, 2\ell, w) \leq \binom{n}{k} \frac{A(n-k, 2(\ell + \ell), w - \ell)}{\binom{2\ell}{\ell}} \) if \( \ell > n/2 \).

**Remark.** Note that the denominators in Theorem 3 are greater than the corresponding denominators in Theorem 2.

**Proof.** Let \( C \) be an \((n, 2\ell, w)\) binary constant weight code with \(|C| = A(n, 2\ell, w)\) and let \( \ell = \min \left( \frac{n}{2}, w \right) \). For every binary vector \( b \) of length \( n \) and weight \( k \) (notation \( b \in V_n^k \)) we define the code \( C^b \) by

\[
C^b = \{ \alpha \in C : \text{wt}(\alpha \oplus b) \leq \ell \},
\]

where \( \alpha \oplus b \) is \((c_1, c_2, \ldots, c_n)\). To make things clear we give an example. Let \( b \in V_n^2 \) be given by

\[
b = (1, 1, \ldots, 1, 0, 0, \ldots, 0, 0, 0, \ldots, 0),
\]

\[
\ell = (1, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0).
\]

Then

\[
C^b \subseteq C = \{ (0, 0, \ldots, 0, 0, 0, \ldots, 0, 1, 1, \ldots, 1, 0, 0, \ldots, 0) \}.
\]

From the above it will be clear that, for any two codewords \( c_1, c_2 \in C \) and \( c_{\ell} \in C^b \) of \( C^b \) we have
\[ d(c_1 \cdot c_2, c_1 \cdot c_2, c_1 \cdot c_2, c_1 \cdot c_2) = d(c_1, c_2)^2 \cdot d(c_1 \cdot b, c_2 \cdot b)^2 \]

\[ \geq 2^{\ell - \text{wt}(c_1 \cdot b)} - \text{wt}(c_1 \cdot b). \]

So if, for every \( c \in C_{n-k} \), an arbitrary set of \( \ell - \text{wt}(c \cdot b) \) coordinates that are one are changed into zeros, we get a code \( C'_{n-k} \) with constant weight \( w - \ell \) and minimum distance at least \( 2(\ell - \ell) \).

From the definition of \( C'_{n-k} \), one easily sees that all codewords of \( C'_{n-k} \) have zeros where \( b \) has ones. So certainly all codewords of \( C'_{n-k} \) have zeros where \( b \) has ones. This means that we can puncture the code \( (C'_{n-k} \setminus k \text{ times}) \) (delete the \( k \) coordinates where \( b \) has ones) to get a constant weight code \( C''_{n-k} \) of length \( n-k \), minimum distance at least \( 2(\ell - \ell) \) and constant weight \( w - \ell \). Thus

\[ |C''_{n-k}| = |C'_{n-k}| \leq \lambda(n-k, 2(\ell - \ell), w - \ell), \text{ for all } b \in V_k. \]

We now show that there is a \( b \in V_k \) such that

\[ |C_{n-k}| \geq \lambda(n, 25, w). \]

To do this, we calculate the number \( N \) of pairs \( (c, b) \) of the set \( \{ (c, b) | c \in C_{n-k}, \text{ wt}(c \cdot b) = \ell \} \) in two different ways.

Since, for any \( c \in C \), there are \( \ell \sum_{j=0}^{\ell} \binom{n-w}{j} \) vectors in \( V_k \) that have not more then \( \ell \) ones in common with \( c \), we have

\[ N = \sum_{z \in C} \ell \sum_{z \in b} \binom{n-w}{\ell-j} \lambda(n, 25, w) \ell \sum_{j=0}^{\ell} \binom{n-w}{j} \binom{\ell-j}{j}. \]

On the other hand, this number also equals
\[ N = \sum_{\mathbf{x} \in \mathbb{F}_2^n} |C_{\mathbf{x}}|, \]

hence we have

\[ \sum_{\mathbf{x} \in \mathbb{F}_2^n} |C_{\mathbf{x}}| = \mathcal{A}(n,2d,w) \sum_{D \subseteq [n]} \frac{\binom{n}{|D|} (n-w)}{(n-|D|) (n+1)} \sum_{\mathbf{x} \in \mathbb{F}_2^n} \frac{\binom{n-w}{w}}{\binom{n-w-|D|}{w}}. \]

Hence there is a \( \mathbf{y} \in \mathbb{F}_2^n \) with

\[ |C_{\mathbf{y}}| < A(n,2d,w). \]

Together with \( |C_{\mathbf{y}}| \leq A(n-k,2d,w-\ell) \) this last inequality proves the first part of the theorem.

To prove the second inequality of the theorem, we apply the first one to the complementary code \( \overline{C} \) of \( C \), with \( \ell' = k - \ell \) and \( k \). This gives

\[ a(n,2d,w) = \mathcal{A}(n,2d,w) \sum_{D \subseteq [n]} \frac{\binom{n}{|D|} (n-w)}{(n-|D|) (n+1)} \mathcal{A}(n-k,2(\ell'-k+\ell),n-w-k+\ell), \]

\[ = \sum_{j=0}^{n-k} \binom{n-k}{j} \mathcal{A}(n-k-j,2(\ell'-k+\ell),n-w+1-j). \]

As in [9] the proof of the above theorem has an immediate consequence, which is stated in the next theorem.

**Theorem 4.** Let there exists an \( (n,2d,w) \) constant weight code with \( N \) codewords and let \( k, \ell \) be arbitrary integers with \( 0 \leq k \leq n \). Then there exists an \( (n-k,2\ell,n-w-\ell) \) constant weight code with \( N' \) codewords, where
$$u = \delta - \ell$$ and $$n' = \sum_{i=0}^{\ell} \binom{\delta}{i} \binom{n-i}{\ell-i}$$ if $$\ell \leq x/2$$.

$$v = \ell - k + \ell$$ and $$n' = \sum_{i=\ell}^{k} \binom{\ell}{i} \binom{n-i}{\ell-i}$$ if $$\ell > x/2$$.

**Proof.** One of the codes $$C_{n'}$$ for $$\ell \in V_{n'}$$ defined in the proof of Theorem 3 does the job.

The next table contains the improved (according to the table in [6]) lower bounds on $$A(n,6,v)$$ that follow from Theorem 4, using the constant weight codes formed by the codewords of weight 8 respectively 12 from the [24,12,8] Golay code. For completeness we also mention the lower bounds that can be found in [5] (second column) and the improvements given by V. B. Slepnev in [8] (third column). Our improved results are stated in the fourth column. The values of $$k$$ and $$\ell$$, needed to obtain these results, are given in the last column.

<table>
<thead>
<tr>
<th>$$A(n,25,w)$$</th>
<th>Upper and lower bounds from [6]</th>
<th>Lower bound from [8]</th>
<th>Lower bound from Th. 4</th>
<th>Values of $$k$$ and $$\ell$$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(22,6,7)</td>
<td>475-1120</td>
<td>-</td>
<td>498</td>
<td>2 1</td>
</tr>
<tr>
<td>A(21,6,7)</td>
<td>465-626</td>
<td>-</td>
<td>570</td>
<td>3 1</td>
</tr>
<tr>
<td>A(20,6,7)</td>
<td>510-651</td>
<td>320</td>
<td>450</td>
<td>4 1</td>
</tr>
<tr>
<td>A(19,6,7)</td>
<td>520-520</td>
<td>260</td>
<td>338</td>
<td>5 1</td>
</tr>
<tr>
<td>A(18,6,7)</td>
<td>160-249</td>
<td>198</td>
<td>243</td>
<td>6 1</td>
</tr>
<tr>
<td>A(17,6,7)</td>
<td>140-140</td>
<td>141</td>
<td>166</td>
<td>7 1</td>
</tr>
<tr>
<td>A(16,6,7)</td>
<td>90-156</td>
<td>95</td>
<td>108</td>
<td>8 1</td>
</tr>
<tr>
<td>A(15,6,7)</td>
<td>1574-5064</td>
<td>-</td>
<td>1360</td>
<td>2 1</td>
</tr>
<tr>
<td>A(21,6,10)</td>
<td>1286-2702</td>
<td>-</td>
<td>1388</td>
<td>3 2</td>
</tr>
<tr>
<td>A(20,6,9)</td>
<td>738-1362</td>
<td>-</td>
<td>760</td>
<td>4 3</td>
</tr>
<tr>
<td>A(19,6,8)</td>
<td>332-734</td>
<td>360</td>
<td>408</td>
<td>5 4</td>
</tr>
<tr>
<td>A(18,6,8)</td>
<td>232-420</td>
<td>-</td>
<td>229</td>
<td>5 4</td>
</tr>
</tbody>
</table>

**Table 1.** Improved lower bounds on $$A(n,6,w)$$.

To find the lower bound $$A(18,6,8) \geq 229$$ (*) one uses Theorem 4, starting with the $$(23,1288,6,12)$$ constant weight code, taking $$k=5$$ and $$\ell=4$$. 
2.3 LOWER BOUNDS FOR $A(n,4,w)$

§ 2.3.1 Introduction

In this paper we describe a construction method for constant weight codes with minimum distance 4 that improves several of the known results in [1], [3] and [6]. The method is based on the following observation.

**Lemma 5.** Let $n_1$ and $n_2$ be two positive integers and let $n = n_1 + n_2$. Then

$$A(n,4,w) \geq \max \left\{ \begin{array}{ll}
\frac{w-5}{2} \\
\sum_{i=0}^{w-3} T(i+21, n_1, w-i, n_2, w-i+1, n_1, w-i+1)
\end{array} \right\}
,$$

where $T(w_1, n_1, w_2, n_2, 4)$ — the maximal number of codewords in any binary code of length $n_1 + n_2$, minimum distance 4, with exactly $w_1$ 1's in the first $n_1$ coordinate places and exactly $w_2$ 1's in the last $n_2$ coordinate places (such a code will be denoted by an $(n_1, n_2, 4; w_1, w_2)$ code).

**Proof.** Let $C(w_1, n_1, w_2, n_2)$ denote a binary $(n_1, n_2, 4; w_1, w_2)$ code. The lemma is proved if we can show that the codes $C(0)$ and $C(1)$ defined by

$$C(0) := \bigcup_{i=0}^{w-3} C(i+21, n_1, w-i, n_2), \quad C(1) := \bigcup_{i=0}^{w-3} C(i+21, n_1, w-i, n_2),$$

both have minimum distance 4. We will prove this for $i = 0$.

Let $y = (y_1, y_2, \ldots, y_n) \in C(0)$, with $y_1, y_2 \in \{0,1\}$ and $y_3, y_4 \in \{0,1\}^2$ be two distinct codewords of $C(0)$. Suppose that $w(y_1) = 21$ and $w(y_2) = 21$, so $w(y_3) = w(y_2) - 21$ and $w(y_4) = w(y_2) - 21$. Then if $i = 0$ we have

$$(y_1, y_2, y_3) \in C(2i, n_1, w-21, n_2)$$

and so $d(y_1, y_2) = 4$, while if $i = 1$ we have

$$(y_1, y_2, y_3, y_4) \in C(4i+2, n_1, w-21, n_2, 2)$$

and so $d(y_1, y_2) = 4$. Therefore, both codes have minimum distance 4 and...
\[ \delta(w, v) = d(w_1, y_1) + d(y_2, v_2) + d(v_1, v_1) + d(y_2, v_2) + d(v_1, v_1) = 2w - 2v. \]

From this it is clear that our construction method involves the construction of \((n_1, n_2) v_1, v_2\) binary codes, which we treat in §2.3.2.

§ 2.3.2 The construction of \((n_1, n_2) v_1, v_2\) codes

Let \(V^0_n\) denote the set of all binary vectors of length \(n\) and weight \(w\). \((C^i(w, n))_{i=0}^{n-1}\) denotes a partition of \(V^0_n\) into \(k\) mutually disjoint constant weight codes, each with minimum distance \(d\) and constant weight \(w\). Assume that the constant weight codes are numbered in such a way that \(|C^0(w, n)| \geq |C^1(w, n)| \geq \ldots \geq |C^{n-1}(w, n)|\) holds. The construction of a \((n_1, n_2) v_1, v_2\) code is as follows:

Let \((C^i(w_1, n_1))_{i=0}^{n_1-1}\) and \((C^j(w_2, n_2))_{j=0}^{n_2-1}\) be partitions of \(V^0_{w_1}\) and \(V^0_{w_2}\) as we have defined above. The code \(C(w_1, n_1, v_1, n_2, n_2)\) is then defined by

\[ C(w_1, n_1, v_1, n_2, n_2) := \bigcup_{i=0}^{n_1-1} C^i_w(w_1, n_1) \odot C^j(v_2, n_2). \]  

(3)

where \(A \odot B = \{ (a, b) \mid a \in A, b \in B \} \).

**Lemma 6.** The code \(C(w_1, n_1, v_1, n_2, n_2)\) defined in (3) is a binary \((n_1, n_2) v_1, v_2\) code. The number of codewords is given by

\[ |C(w_1, n_1, v_1, n_2, n_2)| = \sum_{i=0}^{n_1-1} |C^i(w_1, n_1)| \cdot |C^j(v_2, n_2)|. \]  

(4)
**Proof.** We only prove that the minimum distance is 4. The rest then follows immediately. Let \( y = (y_1, y_2) \in C^1(w_1, n_1) \times C^1(w_2, n_2) \) and \( z = (z_1, z_2) \in C^1(w_3, n_3) \times C^1(w_4, n_4) \) be two distinct codewords of \( C(w_i, n_i, w_j, n_j) \). Then there are two cases:

1. \( i = j \), then \( y_1 \neq z_1 \) or \( y_2 \neq z_2 \). Hence \( d(y, z) = \min \{d_1(y_1, z_1), d_2(y_2, z_2)\} \geq 4 \).
2. \( i \neq j \), then \( y_1 \neq z_1 \) and \( y_2 \neq z_2 \). Hence \( d(y, z) = \min \{d_1(y_1, z_1), d_2(y_2, z_2)\} \geq 2 \times 2 = 4 \).

**Remark.** In (3) we used the codes \( C^1(w_i, n_i) \) and \( C^1(w_i, n_i) \) to form the direct sum \( C^1(w_i, n_i) \times C^1(w_i, n_i) \), \( i = 0, 1, \ldots, \min \left\{ \frac{n}{2}, k \right\} - 1 \). Other combinations are possible. However, from (3) and the assumption about the ordering of the codes in a partition, it follows that no other combination gives a larger code \( C(w_i, n_i, w_j, n_j) \).

We are left with the problem of finding suitable partitions of \( V_n^0 \) for \( 0 < v \leq n \). One way of solving this problem is to look at the construction method for constant weight codes that Graham and Sloane described in [1]; this method partitions \( V_n^0 \) into mutually disjoint constant weight codes with minimum distance 4, which given a partition \( \{C^1(w_i, n_i)\}_{i=0}^{\min \{n/2, k\}} \) of \( V_n^0 \), \( 0 < v < n \). Using these partitions in (3), we find \( (n_1, n_2, w_1, w_2) \) codes \( C(w_1, n_1, w_2, n_2) \) of \( C^1(w_1, n_1, w_2, n_2) \). For every \( (n_1, n_2, w_1, w_2) \), such \( n_1 \geq n_2 \), \( 0 < w_1 \leq n_1 \), and \( 0 < w_2 \leq n_2 \).

Taking \( n_1 = n_2 = n \) and \( 0 < v < n \) we find codes \( C(0) \) and \( C(1) \) with \( |C(0)| = |C(1)| = \left(\begin{array}{c} n \\binom{2}{v} \end{array}\right)/\binom{2}{v}, \) from which we conclude that \( \lambda(2n, 4, v) = \left(\begin{array}{c} 2n \\binom{2}{v} \end{array}\right)/\binom{2}{v} \). The lower bound was also found by Graham and Sloane [1]. From the above it will be clear that we can expect to find better results if, for instance, we are able to find partitions of \( V_n^0 \) into fewer than \( n \) mutually disjoint constant weight codes. The determination of such partitions is postponed to the appendix.

**Example.** Let \( n = 16 \) and \( v = 7 \). From [1] we have \( \lambda(16, 4, 7) \geq 15 \). Taking \( n_1 = n_2 = 8 \) and using the partitions of \( V_n^0 \) as determined.
in the appendix, we find codes \( C(i, 0, 7, 21, 8), i = 0, 1, 2, 3 \), with
\[
|C(0, 0, 7, 21, 8)| = \sum_{i=0}^{3} |C^{i}(0, 0)| \cdot |C^{i}(7, 21)| = 1,
\]
\[
|C(1, 0, 5, 21, 8)| = \sum_{i=0}^{6} |C^{i}(2, 5)| \cdot |C^{i}(5, 21)| = 7.43 \times 224,
\]
\[
|C(4, 0, 1, 21, 8)| = \sum_{i=0}^{6} |C^{i}(4, 0)| \cdot |C^{i}(4, 21)| = 2.14, 8 + 2.12, 8 + 10.3 + 8.0 + 1.0 - 500
\]
and
\[
|C(6, 0, 1, 21, 8)| = \sum_{i=0}^{6} |C^{i}(6, 0)| \cdot |C^{i}(6, 1)| = 7.41 \times 28.
\]
Hence, for the code \( C(0) \) defined in (2) we find
\[
|C(0)| = \sum_{i=0}^{0} |C^{i}(21, 0, 7, 21, 8)| = 813,
\]
giving us the improved lower bound \( A(16, 4, 7) \geq 813.\)

**Example 2.** Let \( n = 10 \) and \( w = 5 \). From (6) we have \( A(19, 4, 5) \geq 612 \). Take \( n_1 = 9 \)
and \( n_2 = 10 \). With the help of the appendix, we find codes
\( C(21, 1, 9, 4, 21, 10), i = 0, 1, 2, 3 \), with
\[
|C(1, 9, 4, 10)| = \sum_{i=0}^{8} |C^{i}(1, 9)| \cdot |C^{i}(4, 10)|
\]
\[
= 2.1.27 + 1.26 + 7.1.26 + 2.1.1.2 = 206,
\]
\[
|C(1, 9, 2, 10)| = \sum_{i=0}^{6} |C^{i}(3, 9)| \cdot |C^{i}(2, 10)| = 7.12.5 = 420,
\]
\[
|C(5, 0, 2, 10)| = \sum_{i=0}^{6} |C^{i}(5, 9)| \cdot |C^{i}(10, 10)| = 16.1 = 16.
\]
Thus, for the code $C(1)$ defined by (2) we find

$$|C(1)| = \sum_{i=0}^{2} |C(2i+1,9,4-2i,10)| = 642.$$ 

Hence, we have $A(19,4,5) \leq 642$.

However, since any constant weight code $A$ of length 9, minimum distance 4 and constant weight 5, can be transformed into $(9,10,4;5,0)$-code by adding a codeword of $A$ a tail of 10 zeros, it is easy to find a $(9,10,4;5,0)$-code $C'(5,9,0,10)$ with $|C'(5,9,0,10)| = A(9,4,5) = 18$. Replacing $C(5,9,0,10)$ by $C'(5,9,0,10)$ in the above construction, gives $A(19,4,5) \leq 644$.

We conclude this paragraph with a revised table of lower and upper bounds for the function $A(n,4,w)$ in the range $n \leq 24$. The entries in this table with an asterisk in the upper left corner are the improved lower bounds found by the above described method, using the partitions given in the appendix. On checking these entries the reader must be aware of the fact that we have used the trick explained in Example 2 several times. The entries without an asterisk are from [1],[3] and [6].
<table>
<thead>
<tr>
<th>4</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0</td>
<td>14</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>12</td>
<td>18</td>
<td>18</td>
<td>12</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>13</td>
<td>30</td>
<td>36</td>
<td>30</td>
<td>12</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>17</td>
<td>35</td>
<td>66</td>
<td>66</td>
<td>35</td>
<td>17</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>20</td>
<td>51</td>
<td>73</td>
<td>64</td>
<td>132</td>
<td>75</td>
<td>64</td>
<td>51</td>
<td>20</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>76</td>
<td>65</td>
<td>118</td>
<td>158</td>
<td>150</td>
<td>118</td>
<td>65</td>
<td>26</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>28</td>
<td>91</td>
<td>169</td>
<td>275</td>
<td>271</td>
<td>316</td>
<td>275</td>
<td>165</td>
<td>91</td>
<td>28</td>
<td>7</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>33</td>
<td>104</td>
<td>222</td>
<td>373</td>
<td>302</td>
<td>370</td>
<td>222</td>
<td>105</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>37</td>
<td>140</td>
<td>308</td>
<td>902</td>
<td>813</td>
<td>1164</td>
<td>913</td>
<td>592</td>
<td>305</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>44</td>
<td>154</td>
<td>426</td>
<td>854</td>
<td>1320</td>
<td>1608</td>
<td>1605</td>
<td>1320</td>
<td>854</td>
<td>424</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>48</td>
<td>198</td>
<td>504</td>
<td>1260</td>
<td>1436</td>
<td>2760</td>
<td>3150</td>
<td>2760</td>
<td>1936</td>
<td>1260</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>9</td>
<td>57</td>
<td>220</td>
<td>664</td>
<td>1496</td>
<td>3024</td>
<td>4330</td>
<td>5490</td>
<td>5490</td>
<td>4330</td>
<td>3024</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>60</td>
<td>285</td>
<td>831</td>
<td>2120</td>
<td>4103</td>
<td>7112</td>
<td>9197</td>
<td>9197</td>
<td>7112</td>
<td>2120</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td>70</td>
<td>315</td>
<td>1071</td>
<td>2056</td>
<td>5708</td>
<td>10045</td>
<td>13142</td>
<td>18587</td>
<td>18587</td>
<td>13142</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>11</td>
<td>73</td>
<td>385</td>
<td>1386</td>
<td>3927</td>
<td>7880</td>
<td>15124</td>
<td>23450</td>
<td>33442</td>
<td>33442</td>
<td>15124</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>11</td>
<td>83</td>
<td>416</td>
<td>1771</td>
<td>5313</td>
<td>11266</td>
<td>22530</td>
<td>36006</td>
<td>52987</td>
<td>52987</td>
<td>22530</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>12</td>
<td>88</td>
<td>498</td>
<td>2011</td>
<td>704</td>
<td>15267</td>
<td>37795</td>
<td>56267</td>
<td>85816</td>
<td>105499</td>
<td>124052</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2. Bounds on A(n, 4, w)**
§ 2.3.3 An optimal \((18,21,8,6)\)-code

We conclude this chapter with an \((18,21,8,6)\)-code. From [6] we have that \(A(16,8,6) \leq 2\); so the code is optimal. The \((18,21,8,6)\)-code is given by the matrix \(A\) below, the rows of which are the codewords.

\[
A = \begin{bmatrix}
A_1 & A_2 & A_3 \\
A_4 & A_5 & A_6 \\
A_7 & A_8 & A_9 \\
\end{bmatrix}
\]

where \(A_1, A_2, A_3\) are circulant matrices with top row \(A_1\) given by

\[
A_1 = (100000), \quad A_2 = (110000) \text{ and } A_3 = (001100)
\]

It is easy to prove that the rows of \(A\) indeed form an \((18,21,8,6)\)-code.
In this appendix a partition of $V_n^w$ is understood to be a partition of $V_n^w$ into constant weight codes of length $n$, minimum distance $d$ and constant weight $w$ as defined in § 2.3.2. The partitions of $V_n^w$ which we give here, are used in our construction method of Section 2.3 to find the results stated in Table 2.

**Definition:** Let $\{C^t(w,n)\}_{t=0}^{1} \ldots , x-1$ be a partition of $V_n^w$. Then the number vector $E_{n,n}$ of $\{C^t(w,n)\}_{t=0}^{1} \ldots , x-1$ is defined by

$$E_{n,n} = \left(\sum_{t=0}^{1} |C^t(w,n)|, \sum_{t=1}^{x-1} |C^t(w,n)|, \ldots , |C^{x-2}(w,n)|\right)$$

For the determination of the number of codewords in the codes $C(0)$ and $C(1)$ of (2) and hence for the determination of a lower bound on $\binom{n}{d,n}$, these number vectors are all we have to know. The next lemma gives the number vectors of partitions of $V_n^w$ for $w \leq 2$. The proof is simple and is left to the reader.

**Lemma.** For every $w \leq n$ there are partitions of $V_n^w$ with number vectors satisfying:

1. $E_{0,n} = n$, $E_{1,n} = (n)$, $E_{2,n} = \binom{n}{2}$,
2. $E_{0,n} = \sum_{t=0}^{1} |C^t(w,n)|$, if $n$ is odd,

$$E_{0,n} = \left\{\begin{array}{ll}
\binom{n-1}{2}, & \text{if } n \text{ is odd,} \\
\binom{n}{2}, & \text{if } n \text{ is even.}
\end{array}\right.$$
In order to limit the amount of writing we frequently use the following notation. Let \( C \) be a binary code of length \( n \) with coordinate set \( X = \{0,1,\ldots,n-1\} \) and let \( p \) be a permutation of \( X \). Then we denote by \( p(C) \in \{0,1\}^X \) the vector
\[
p(C) = (c_0, c_1, c_2, \ldots, c_{n-1})
\]
and by \( p(C) \) the code
\[
p(C) = \{p(c) \mid c \in C\}.
\]

<table>
<thead>
<tr>
<th>( n )</th>
<th>( S )</th>
<th>( k )</th>
<th>Number vector ( E_{k,n} )</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>6</td>
<td>( {4,4,4,4,2,2} )</td>
<td>A.1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>6</td>
<td>( {7,7,6,6,5,4} )</td>
<td>A.1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>7</td>
<td>( {8,8,8,8,8,8,8} )</td>
<td>A.2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>6</td>
<td>( {9,14,11,12,10,8} )</td>
<td>A.1</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>7</td>
<td>( {12,12,12,12,12,12} )</td>
<td>A.2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>8</td>
<td>( {15,16,15,16,15,16,14} )</td>
<td>A.3</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>10</td>
<td>( {15,13,13,13,13,13,13,13,13,13} )</td>
<td>A.4</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>10</td>
<td>( {12,12,12,12,12,12,12,12,12,12} )</td>
<td>A.5</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>10</td>
<td>( {10,14,12,12,12,12,12,12,12,12} )</td>
<td>A.6</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>11</td>
<td>( {17,17,17,17,17,17,17,17,17,17,17} )</td>
<td>A.7</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>11</td>
<td>( {17,17,17,17,17,17,17,17,17,17,17} )</td>
<td>A.7</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>11</td>
<td>( {16,60,55,55,47,44,41,32,31,30,5} )</td>
<td>A.6</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>11</td>
<td>( {20,23,22,22,22,22,22,22,22,22,22} )</td>
<td>A.7</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>11</td>
<td>( {21,51,51,51,51,40,40,40,40,40,40} )</td>
<td>A.7</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>12</td>
<td>( {73,70,70,70,70,64,64,64,64,64,64} )</td>
<td>A.7</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>11</td>
<td>( {92,90,90,90,90,80,80,80,80,78,78} )</td>
<td>A.7</td>
</tr>
</tbody>
</table>

Table 3. Number vectors
A.1 Partitions of $V_4^7$, $V_3^7$ and $V_4^8$

The six arrays below form a partition of $V_4^7$ into six mutually disjoint constant weight codes with minimum distance 24. The number vector of the partition is $E_{3,7} = (7,7,6,6,5,4)$.

![Arrays](image)

If we consider the vectors that have a zero in the $k^{th}$ coordinate place we find, after deleting this coordinate, a partition of $V_4^6$ with number vector $E_{3,6} = (4,4,4,4,2,2)$.

Since the distance between two distinct vectors in the same code $C^k(3,7) (k=0, \ldots, 5)$ is exactly four, we may adjoin to each $C^k(3,7)$ ($k=0, 1, \ldots, 5$) the complements of its codewords. Adding the overall parity check then yields a partition of $V_4^8$ with number vector $E_{4,8} = (14,14,12,12,10,8)$. 
A.3 Partitions of $V_3^\alpha$, $V_3^\beta$ and $V_3^{12}$

The problem of finding a partition of $V_3^\alpha$, for $n = 6n + 1$ or $n = 6n + 3$ into a number (as small as possible) of mutually disjoint constant weight codes, is the same as trying to find a packing with Steiner triple systems of order $n$. (i.e., a partition of the set of triples of $n$ elements into $n-2$ disjoint Steiner triple systems). In [2] Poonnam gives the solution of the above problem for eleven values of $n$, including $n=13$. This solution is given on the following page and is used to determine a partition of $V_3^{12}$ with number vector $K^*_{3,12} = (20,20,20,20,20,20,20,20,20,20)$. One can also find several references to the above problem in [2]. The existence of a packing of order 9 was found by Kirkman (see [22]) and rediscovered several times (also by 08). This packing is given below. We use our terminology.

Let $C$ be the constant weight code shown in Fig. 1 and let $p_4$ be the permutation $(0,1,2,3,4,5,6)(7)(8)(9)$, where $p_4(i) = i + 1$ (mod 9). Then we define the codes $C^\alpha_{4}(1,3,5,7)$, $i = 0,1,2,3,5,6$, by

$$
C^\alpha_{1}(2,3) = p_4(0), \quad i = 0,1,2,3,5,6.
$$

These codes form a partition of $V_3^\alpha$ with number vector $K_{3,9} = (12,12,12,12,12,12)$. This codewords with a zero in the last coordinate form, after deleting this coordinate, a partition of $V_3^\beta$ with number vector $K_{3,6} = (8,8,8,8,8,8)$. 

```
\begin{tabular}{cccccc}
  0 & 1 & 2 & 3 & 4 & 5 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 & 1 & 1 \\
\end{tabular}
```

Fig. 1.
Let $D$ be the code shown in Fig. 2 and let $\pi_2$ be the permutation 
$(0,1,2,3,4,5,6,7,8,9,10)(11)(12)$. Then the codes $C^i(3,11), \ i = 0, \ldots, 10,$ defined 
below form a partition of $\mathbb{Z}^{11}_3$.

$$C^i(3,11) = \pi_2^i(D), \quad i = 0, \ldots, 10.$$ 

Shortening these codes gives a partition of $\mathbb{Z}^{12}_3$ with number vector


![Fig. 2.]

A.3 A partition of $\mathbb{Z}^{12}_4$

Let $C$ be the $(9,18,4,4)$ code shown in Fig. 3 and let $\pi$ be the 
permutation $(0,1,2,3,4,5,6)(7)(8)$. Then the partition $(C^i(4,9)), \ i = 0, \ldots, 9,$
with number vector $E_{9, 9} = (16, 16, 16, 16, 16, 16, 16, 16, 16)$ is defined by

\[ C^L(4, 9) = p^1(C), \quad i = 0, 1, \ldots, 6, \]
\[ C^H(4, 9) = p^2((1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0) \mid 1-0, 1-0, \ldots, 1-0). \]

\[ C = \begin{array}{cccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array} \]

Fig. 3.

### A.4 A partition of $v_{10}^0$

We now give a partition of $v_{10}^0$ with number vector $E_{9, 10} = (13, 13, 13, 13, 13, 13, 13, 13, 13, 13)$. Let $p$ be the permutation $(0, 1, 2, 3, 4, 5, 6, 7, 8) \{9\}$ and let $P_1, P_2, P_3$ and $P_4$ be the constant weight codes shown in Fig. 9. Then the partition of $v_{10}^0$ is given by

\[ C^L(3, 10) = P_3^1(C), \quad i = 0, 1, 2 \text{ and } i = 0, 1, 2, \text{ and } C^H(3, 10) = P_3. \]
A.5 A partition of $\mathcal{P}_4$

Let $\mathcal{C}$ be the $(0, 37, 4, 4)$ constant weight code shown in Fig. 5 and let $p$ be the permutation $(0, 1, 2, 3, 4, 5, 6)(7)(8)(9)$. We define the codes $\mathcal{C}^1(4, 10)$, $\mathcal{C}^2(4, 10)$, $\mathcal{C}^3(4, 10)$, $\mathcal{C}^4(4, 10)$, $\mathcal{C}^5(4, 10)$, and $\mathcal{C}^6(4, 10)$ by $\mathcal{C}_1$ and $\mathcal{C}_2$ are as defined in Fig. 5:

$\mathcal{C}^0(4, 10) := \mathcal{C}$, $\mathcal{C}^1(4, 10) := p^2(\mathcal{C})$, $\mathcal{C}^2(4, 10) := p^4(\mathcal{C})$,
$\mathcal{C}^3(4, 10) := \mathcal{C}^0(\mathcal{C}) \setminus \{p^2(\mathcal{C}_2)\}$, $\mathcal{C}^4(4, 10) := \mathcal{C} \setminus \{p(\mathcal{C}_1), p(\mathcal{C}_2)\}$,
$\mathcal{C}^5(4, 10) := \mathcal{C}^3(\mathcal{C}) \setminus \{p^3(\mathcal{C}_1), p^3(\mathcal{C}_2)\}$ and
$\mathcal{C}^6(4, 10) := \mathcal{C}^5(\mathcal{C}) \setminus \{p^5(\mathcal{C}_1), p^5(\mathcal{C}_2)\}$.
Together with \( C^7(4, 10), C^8(4, 10) \) and \( C^9(4, 10) \), defined by the arrays shown in Fig. 5a, they define a partition of \( F_{10}^4 \) with number vector \( \mathbf{b}_{4, 10} = (27, 27, 27, 24, 24, 24, 24, 12, 12, 41). \)
A.6 A partition of $V_{5}^{10}$ and $V_{5}^{11}$

From [6] we have, that the rows of $A$ (see the figure below) and the sums of pairs of rows of $A$ form a $(11,66,4,6)$ constant weight code.

\[
\begin{array}{ccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

Taking the complements of all these codewords and permuting the coordinates \(\{0,1,2,\ldots,10\}\) with the permutation \((0,10,11,7,2,8,4,3,6,5,9)\) one finds the \((11,66,5,4)\) code $D$ shown in Fig. 6 (the coordinates are renumbered).

Let $D'$ be the subcode $D' := D \setminus \{e_1, e_2, \ldots, e_{28}\}$, where the $e_i$, \(i = 1, 2, \ldots, 28\), are defined in Fig. 6. Let $p$ be the permutation \((0,1,2,3,4,5,6,7,8,9,10)\), then we define the following partition of $V_{5}^{11}$:

- \(C^0(5,11) := D = D' \cup \{e_1, e_2, \ldots, e_{28}\}\),
- \(C^1(5,11) := p^3(D') \cup \{p^3(e_i)\} \quad | i = 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 22, 23, 24, 25, 27\),
- \(C^2(5,11) := p^2(D') \cup \{p^2(e_i)\} \quad | i = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 21, 22, 24, 26\),
- \(C^3(5,11) := p^4(D') \cup \{p^4(e_i)\} \quad | i = 1, 3, 5, 7, 9, 12, 13, 16, 17, 19, 22, 23\),
- \(C^4(5,11) := p^4(D') \cup \{p^4(e_i)\} \quad | i = 2, 11, 12, 14, 16, 19, 22, 23\) \cup \{\{0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1\}\}
- \(C^5(5,11) := p^2(D') \cup \{p^2(e_i)\} \quad | i = 5, 8, 16, 20\) \cup \{\{0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0\}\}
$C^0(5,11) = \ell^0(5,11) = (1,1,0,0,0,0,1 | 1,0,0,1)$.

$C^0(5,11), C^0(5,11), C^0(5,11)$ and $C^{10}(5,11)$ are defined by the array shown in Fig. 6a.

The number vector is $E_{x,11} = (66,62,55,50,47,41,32,30,3,5)$.
A.7 Partitions found with help of the method described in Section 2.3

Since the partitions in Table 3, with A.7 as reference, are all found in the same way, we only give one, namely that of $V'_{3}$, the others are left to the reader.

Let $\mathcal{C}^{1}(v,5)$, $\mathcal{C}^{1}(v,6)$, and $\mathcal{C}^{1}(v,7)$ be the partitions of $V'_{v}$ ($v = 0, 1, 2, 3$) respectively $V'_{v}$ ($v = 0, 1, 2, 3$) which we found in A.1 or just preceding. Then we define the following partition of $V'_{3}$.

$$C^{1}(3,11) = \mathcal{C}^{1}(3,5) \# C^{1}(0,6)$$

$$U \left( \bigcup_{j=0}^{4} \mathcal{C}^{1}(2,5) \# C^{1}(j+1) \text{ mod } C^{1}(2,6) \right), \quad i = 0, 1, 2, 3, 4,$$

$$C^{1}(3,11) = \mathcal{C}^{1}(0,5) \# C^{1}(3,6)$$

$$U \left( \bigcup_{j=0}^{4} \mathcal{C}^{1}(2,5) \# C^{1}(j+1) \text{ mod } C^{1}(1,6) \right), \quad i = 0, 1, \ldots, 5.$$

From § 2.2.2 it follows that every $C^{1}(3,11)$, $i = 0, 1, \ldots, 10$, is a constant weight code of length 11, minimum distance 2 and constant weight 3. From this we also have

$$|C^{1}(3,11)| = 17, \quad i = 0, 1, 2, 3, 4, \quad |C^{1}(2,11)| = 14, \quad i = 5, 6, 7, 8$$

and

$$|C^{1}(2,11)| = 12, \quad i = 9, 10.$$

One further easily sees that all words of weight 3 are different and on $\mathcal{C}^{1}(3,11)$, $i = 0, 1, \ldots, 10$, is a partition of $V'_{3}$ with number vector

$$B_{3,11} = (17, 17, 17, 17, 17, 14, 14, 14, 14, 14, 12, 12).$$
REFERENCES


CHAPTER 3

CONSTANT DISTANCE COIL PAIRS

1.1 INTRODUCTION

In this chapter we are concerned with a problem formulated by Alías, Gómez and Fang [1] in 1981. They defined a constant distance code pair \((A, B)\) as a pair of binary codes of length \(n\) such that, for some \(\delta \in \mathbb{N}, 0 < \delta < n,

\[ \forall a \in A \quad \forall b \in B \quad |d(a, b) - \delta|. \]

If \((A, B, \delta, S) \subseteq \mathbb{F}_2^m\), a code pair for which the above property holds, we write \(M(A, B) = \delta\). They were interested in the following function defined below:

\[ M(n, \delta) = \max \{|A| - |B| \mid A \subseteq \mathbb{F}_2^n, B \subseteq \mathbb{F}_2^n, \delta(A, B) = \delta, \}. \]

In [1] Alías, Gómez and Fang proved the following upper bound on \(M(n, \delta)\).

**THEOREM 1.**

\[ M(n, \delta) \leq 2^\left(\frac{\delta}{2}\right), \text{ for all } n, \delta \in \mathbb{N} \text{ with } 0 < \delta < n. \]

They gave the following examples, where equality holds in Theorem 1.
\[ A_i = \{(0,0), (0,1), \ldots, (i-1,0)\} \subset \{0\}^i, \text{ for } i = 0, 1, \]
\[ B_i = \{(0,1), (1,0), \ldots, (i-1,1)\} \subset \{0\}^i, \]
\[ C_i = \{(0,0), (1,1), \ldots, (i-1,2)\} \subset \{0\}^i, \]

where \( c = 0 \) if \( n \) is even and \( c = 1 \) if \( n \) is odd. One immediately sees that
\[ A_i, B_i \subset \mathbb{Z}_2, \] \( c(A_i, B_i) = \frac{3}{2} + i \) and \( \vert A_i \vert - \vert B_i \vert = \frac{3}{2} + \frac{i}{2}, \) \( i = 0, 1. \)

In [2] Hall and van Lint proved Theorem 1 using the observation that for an equidistant code pair \((A, B)\), for any \( a \in A \) and \( b \in B \), the codes \( a^+ A \) and \( b^+ B \) are orthogonal even weight codes. Moreover, they proved that essentially the only code pairs for which equality holds in Theorem 1 are the ones given in the example above. To be more precise we need a definition.

Two code pairs \((A, B)\) and \((A', B')\), \( A, B, A', B' \subset \mathbb{Z}_2^0 \) are called equivalent if there exists a permutation \( \sigma \) of the positions of codewords and an \( \rho \in \mathbb{Z}_2^0 \) such that
\[ \rho \in \sigma(A) \cup \rho \in \sigma(B) = (A', B'). \]

where \( \sigma(A) = \{(a_1, \ldots, a_n) \mid (\sigma(a_1, \ldots, a_n)) \in A\} \). In [2] Hall and van Lint proved that any code pair for which equality holds in Theorem 1 is equivalent to one of the code pairs given in the example above. Since for these examples \( \delta = \frac{3}{2} \) or \( \delta = \frac{5}{2} \), the question remained: "What is the exact value of \( M(n, \delta) \), for \( \delta = \frac{3}{2} \) or \( \delta = \frac{5}{2} \)?".

In this chapter we will give an answer to this question. In Section 3.2 we determine the exact value of \( M(n, \delta) \) for all \( n \) and \( \delta \) with \( 0 \leq \delta \leq n \).

In Section 3.3 we additionally characterize all constant distance code pairs \((A, B)\) of length \( n \) and constant distance \( \delta \) with
\[ \vert A \vert - \vert B \vert = M(n, \delta). \]
3.2 The Exact Value of $M(n,\delta)$

From now on $A$ and $\delta$ will always denote two binary codes of length $n$ such that $d(A,B) = \delta$ and $|A| \cdot |B| = M(n,\delta)$. Such a code pair is called optimal. The following lemma shows that without loss of generality we can assume $\delta = \frac{n}{2}$.

**Lemma 2.** For any $n, \delta \in N$, we have

$$M(n,\delta) = M(n,n-\delta).$$

**Proof.** Let $(A,B)$ be a constant distance code pair with constant distance $\delta$. Then obviously $d(A,B) = n - \delta$. The result follows.

From now on we assume $0 \leq \delta \leq \frac{n}{2}$. The following examples give a lower bound on $M(n,\delta)$.

**Example 1.**

$$A^L_{n,\delta} = \{(0,0), (1,1)\} \subset \{0\}^{n-2\delta}$$

$$B^A_{n,\delta} = \{(0,1), (1,0)\} \subset \{0\}^{n-2\delta}, \text{ for } l = 0,1,2,\ldots,\delta,$$

where $A^L_{n,\delta} = \{c \in \{0\}^n \mid \text{wt}(c) = \delta\}.$

We have $A^L_{n,\delta}, B^A_{n,\delta} \in 2^U, d(A^L_{n,\delta}, B^A_{n,\delta}) = \delta$ and $|A^L_{n,\delta}| \cdot |B^A_{n,\delta}| = 2^{2\delta(\frac{n-2\delta}{2})}.

Hence

$$M(n,\delta) \geq |A^L_{n,\delta}| \cdot |B^A_{n,\delta}| = 2^{2\delta(\frac{n-2\delta}{2})}.$$ \hspace{1cm} (1)

The following theorem states that this bound is tight.
THEOREM 3. For all \( n \in \mathbb{N} \), \( \delta \in \mathbb{N} \) with \( 0 \leq \delta \leq \frac{n}{2} \), we have

\[
M(n, \delta) = \max \left( 2^{\frac{n(n-2)}{2}}, \frac{n!}{(n-\delta)!} \right).
\]

REMARK. One easily checks that

\[
\max \left( 2^{\frac{n(n-2)}{2}}, \frac{n!}{(n-\delta)!} \right) = \frac{n!}{(n-\delta)!}, \quad \text{if } n(n-1) \leq 4\delta(n-\delta).
\]

For this reason the inequality \( n(n-1) \leq 4\delta(n-\delta) \) will play an important role in the proof of Theorem 3.

Before we can prove Theorem 3, we need to do some preliminary work. Looking at the code pairs given in Example 1, it seems more or less natural to consider pairs of positions of codewords. That is why we define, for every \( i,j \) with \( 1 \leq i \leq n \) and \( 1 \leq j \leq n \),

\[
\Delta_{ij} = \text{the number of pairs } (a, b), \ a \in A, \ b \in B \text{ such that } a_i - a_j + b_i + b_j \equiv 1 \mod 2.
\]

The condition \( a_i - a_j + b_i + b_j \equiv 1 \mod 2 \) says that the positions \( i \) and \( j \) contribute exactly 1 to the Hamming distance between \( a \) and \( b \). That is,

\[
d(a, b) = \delta_i \delta_j (a_i, b_i, a_j, b_j) = 1. \quad \text{Since } \delta(A, B) = \delta, \text{we have}
\]

\[
\sum_{0 \leq i < j \leq n} \Delta_{ij} = \delta(n-\delta) \binom{|A| \cdot |B|}{\delta(n-\delta)} M(n, \delta) \cdot M(n, \delta).
\]

It follows that there is an \( a_{ij} \) with \( a_{ij} \geq 2\delta(n-\delta) M(n, \delta) / n(n-1) \).

Using a permutation of the positions of codewords we can take care that

\[
a_{ij} \geq 2\delta(n-\delta) M(n, \delta) / n(n-1).
\]

We now try to find an upper bound on \( a_{ij} \). For this reason, we partition the codes \( A \) and \( B \) as follows:
\[ A = \{(0,0) \mid A_{00} \cup E(1,1) \} \cup A_{11} \cup E(1,0) \} \cup A_{10} \]

and
\[ B = \{(0,0) \mid S_{00} \cup E(1,1) \} \cup S_{11} \cup E(1,0) \} \cup S_{10} \],

where \( A_{00}, B_{00} \subseteq \mathbb{F}_2^n \), \( n \in \{0,1\} \). Notice that some of the sets \( A_{\mu} \), \( S_{\mu} \) may be empty. With this terminology we can write \( n_{12} = \sigma_{12} \) as
\[ \sigma_{12} = (|A_{00}| + |A_{10}|) (|S_{00}| + |S_{10}|) + (|A_{11}| + |A_{10}|) (|S_{01}| + |S_{11}|) . \]

The following two lemmas are useful in finding an upper bound on \( n_{12} \).

**LEMMA 4.** For every \( \epsilon, \mu \in \{0,1\} \) the following holds;

- If \( A_{\mu} \cap A_{\mu} = \emptyset \), then \( S_{\mu} = S_{\mu} \mu = \emptyset \) and \( A_{\mu} - A_{\mu} \).

where \( \emptyset = 1 \mod 2 \) and \( \emptyset = 1 \mod 2 \), by symmetry, the roles of the \( A_{\mu} \)'s and \( S_{\mu} \)'s are interchangeable.

**PROOF.** Without loss of generality we may take \( \epsilon \) and \( \mu \) equal to 0. Since \( A_{00} \cap A_{10} = \emptyset \), there is an \( \alpha \in \mathbb{F}_2^m \) such that \( (0,0) \{\alpha\} \) and \( (1,1) \{\alpha\} \) both belong to \( A \). But, for any \( \beta \in \mathbb{F}_2^m \) we then have
\[ d((0,0) \{\alpha\} , (0,0) \{\beta\}) = d((1,1) \{\alpha\} , (0,0) \{\beta\}) \leq 2 \]
and
\[ d((0,0) \{\alpha\} , (1,1) \{\beta\}) = d((1,1) \{\alpha\} , (1,1) \{\beta\}) \leq 2 . \]

So \( \delta_{10} = \delta_{10} = 0 \).

From \( \delta_{10} = \delta_{10} = 0 \) and \( |A| : |E| = (n, \emptyset) \), it now easily follows that \( (0,0) \{\alpha\} (1,1) \{\alpha\} \subseteq A \). So \( A_{00} = A_{11} \).

The following lemma is obvious.

**LEMMA 5.** The code pairs \( (A_{00} \cup A_{10} , S_{00} \cup S_{10}) \) and \( (A_{11} \cup A_{10} , S_{01} \cup S_{11}) \)
are constant distance code pairs of length \( n - 2 \) and constant distance \( \delta - 1 \).

We are now ready to give an upper bound on \( d(\delta) \) and so indirectly an upper bound on \( H(n, \delta) \). We have to consider three cases, the first one of which is special.

**Case I:** \( A_{\epsilon} \cap A_{\delta} = \emptyset \) and \( B_{\epsilon} \cap B_{\delta} \neq \emptyset \), for some \( \epsilon, \delta \in \{0, 1\} \).

With Lemma 4 we then have

\[
H(n, \delta) = |A| + |B| = d_{12} = (|A_{\epsilon}| + |B_{\delta}|) + (|B_{\epsilon}| + |B_{\delta}|)
\]

\[= |A_{\epsilon}| + |B_{\delta}| - |A_{\delta}| + |B_{\delta}|\]

And so with Lemma 5

\[H(n, \delta) = d_{12} \leq H(n - 2, \delta + 1)\]

**Case II:** \( A_{\epsilon} \cap A_{\delta} = \emptyset \) and \( B_{\epsilon} \cap B_{\delta} = \emptyset \), for some \( \epsilon, \delta \in \{0, 1\} \). Exchanging the \( A_{\epsilon} \)'s and \( B_{\epsilon} \)'s gives an equivalent situation.

With Lemma 4 and 5 we then have

\[
d_{12} = (|A_{\epsilon}| + |A_{\delta}|) + (|B_{\epsilon}| + |B_{\delta}|) = 2|A_{\epsilon}| + |B_{\epsilon} \cup B_{\delta}| \leq 2 H(n - 2, \delta + 1).
\]

**Case III:** \( A_{\epsilon} \cap A_{\delta} = \emptyset \) and \( B_{\epsilon} \cap B_{\delta} = \emptyset \), for all \( \epsilon, \delta \in \{0, 1\} \).

Lemma 5 now gives

\[
d_{12} = (|A_{00}| + |A_{11}|) + (|B_{00}| + |B_{11}|) = (|A_{00}| + |B_{00}|) + (|A_{11}| + |B_{11}|) -
\]

\[= |A_{00} \cup A_{11}| + |B_{00} \cup B_{11}| \leq H(n - 2, \delta + 1).
\]

Together with (1) case II and case III give
\[ \text{Hence, we have proved the following inequality} \]
\[ \min(\delta, \lambda) \leq \max \left\{ \frac{n(n-1)}{2 \delta(n-1)}, \frac{n(n-2)(n-3)}{6(\delta - 1)} \right\}. \]

An induction argument now completes the proof of Theorem 1.

**Proof of Theorem 1.** We use induction on \( \delta \). First note that the theorem is obviously true for \( \delta = 0 \) and all \( n \in \mathbb{N} \).

Let \( \delta \geq 1 \) and suppose that for all \( n \in \mathbb{N} \) with \( n \leq 2(\delta - 1) \) the following equality holds:

\[ \text{Min}(n-2, \delta - 1) = \max \left\{ \frac{n(n-1)}{2 \delta(n-1)}, \frac{n(n-2)(n-3)}{6(\delta - 1)} \right\}. \]

From (3) it follows that, for any \( n \geq 2(\delta - 1) \),

\[ \min(\delta, \lambda) \leq \left\{ \begin{array}{l}
\frac{n(n-1)}{2 \delta(n-1)} \text{ if } n(n-1) \leq 4\delta(n-1), \\
\frac{n(n-1)}{2 \delta(n-1)} \text{ if } n(n-1) \geq 4\delta(n-1) \end{array} \right. \]

So we have to distinguish two cases.

First suppose \( n(n-1) \leq 4\delta(n-1) \), we then have

\[ \min(\delta, \lambda) \leq \frac{n(n-1)}{2 \delta(n-1)} \leq \frac{1}{4} \left\{ \frac{n(n-2)(n-3)}{6(\delta - 1)} \right\} \]

Secondly, suppose \( n(n-1) > 4\delta(n-1) \). Then

\[ n(n-1) = 4(n-1) + 1 > 4\delta(n-1) \quad \text{and} \quad n(n-1) > 4\delta(n-1) \quad \text{and} \quad n(n-1) > 4\delta(n-1) \quad \text{and} \quad n(n-1) > 4\delta(n-1). \]

From the remark directly below Theorem 1 we have
\[ \max \left\{ 2^{\frac{n-2}{2}} \begin{pmatrix} n-2 \cr \delta - 1 \end{pmatrix} \mid 0 \leq \delta \leq 1 \right\} \geq \begin{pmatrix} n-2 \cr \delta - 1 \end{pmatrix} \].

So

\[ n(n-1) \delta \cdot n(n-2) \delta - 1 = n(n-2) \begin{pmatrix} n-2 \cr \delta - 1 \end{pmatrix} \]

\[ = \begin{pmatrix} n \cr \delta \end{pmatrix} \cdot \max \left\{ 2^{\frac{n-2}{2}} \begin{pmatrix} n-2 \cr \delta - 1 \end{pmatrix} \mid 0 \leq \delta \leq 1 \right\} \].

Together with (3) both cases give

\[ n(n-1) \delta \cdot n(n-2) \delta - 1 = \max \left\{ 2^{\frac{n-2}{2}} \begin{pmatrix} n-2 \cr \delta - 1 \end{pmatrix} \mid 0 \leq \delta \leq 1 \right\} \].

3.3 OPTIMAL CONSTANT DISTANCE CODE PAIRS

In this section we shall prove that the code pairs of Example 1 are essentially the only optimal constant distance code pairs. The observation at the beginning of Section 3.2 shows us that we only need to consider the case \( 2 \delta \leq n \). So we assume \( 2 \delta \leq n \). In the following \((A,B)\) is an optimal constant distance code pair with constant distance \( \delta \).

The lemma below deals with a single case.

**Lemma 6.** If \(|A| \leq 2\) or \(|B| \leq 2\), then \((A,B)\) is equivalent to \((A^0, B^0)\) or \((A^1, B^1)\) defined in Section 3.2.

**Proof.** Without loss of generality we may assume that \(|A| \leq 2\). If \(|A| = 1\), then \(B = \{0\} \subseteq \mathcal{B}^0\) with \(|B| = 1\) and hence \((A,B)\) is equivalent to \((A^0, B^0)\).

So suppose \(|A| = 2\). Then \(A = \{a_1, a_2\}\) with \(d(a_1, a_2) = 2\), for some \( \lambda \in \mathbb{N} \) with \(2 \leq \lambda \). Since \((A,B)\) is easily seen to be equivalent to \((A^1, B^1)\) if \(d(a_1, a_2) = 2\), we only need to prove \(\lambda = 1\).

Counting the number of words \(x \in \mathcal{B}^0\) with \(d(a_1, x) = d(a_2, x) = \delta\), we find \(|B| = \begin{pmatrix} 2 \cr \delta \end{pmatrix} \). Hence with Theorem 3
\[
\max\left(2^k \binom{n-2i}{\frac{n}{2}-i} \mid 0 \leq i \leq \frac{n}{2}\right) = |A| = |B| = 2^{\frac{n}{2}} \frac{n}{2} - \frac{n}{2} \binom{n}{\frac{n}{2}}
\]

\[
\delta \geq \delta \binom{n-2i}{\frac{n}{2}-i} \max\left(2^k \binom{n-2i}{\frac{n}{2}-i} \mid 0 \leq i \leq \frac{n}{2}\right).
\]

So equality must occur everywhere, which implies \( \lambda = 1 \).

As a consequence of Lemma 5 we have:

**Corollary 7.** If \( \delta = 1 \), then \((A, B)\) is equivalent to either \((A^0, B^0)\)

or \((A^1, B^1)\) defined in Section 3.2.

**Proof.** If \( \delta = 1 \), then one "easily" seen to \( |A| \leq 2 \) or \( |B| \leq 2 \).

The following Lemma is very useful in proving Theorems 9.

**Lemma 8.** Using the notation of Section 3.2 we have, for every \( \varepsilon, \mu \in \{0, 1\} \)

and \( \rho \in x_2^{-2} \),

if \( A_{\rho \mu} \cup A_{\rho \mu} = x_2 \), then \( |A_{\rho \mu}| + |E_{\rho \mu}| = |B_{\rho \mu} \cup B_{\rho \mu}| \leq 1 \).

provided \( m = 1 \) and \( 0 \leq 2 \). The same holds if we interchange the \( A_{\rho \mu} \)'s and \( B_{\rho \mu} \)'s.

**Proof.** Without loss of generality we may take \( \varepsilon = \mu = 0 \) and \( \rho = 1 \). With

Leaves 4 we have \( B_{10} \cup B_{11} = 0 \). If \( B_{10} \cup B_{11} = 0 \) there is nothing to be proved.

So, let \( g \in B_{10} \cup B_{11} \). Then, for any \( a \in A_{00} \cup A_{11} \leq 0 \) we have

\( \delta = \delta + 2 \) or \( \lambda \). Since \( \min = 1 \) and \( \lambda \leq 2 \) implies \( \min \leq 2 \), the \( \min \leq 2 \) as proved above gives us \( \min \leq 1 \). So \( \delta = 1 \) is partitioned into \( A_{00} \)

and \( A_{11} \) (by Lemma 4, \( A_{00} \cap A_{11} = 0 \)), where \( A_{00} \) and \( A_{11} \) are given by
\[ A_{00} = \{ a' \in \mathbb{V}_0 \mid (a', b') = -1 \} \text{ and } A_{11} = \{ a' \in \mathbb{V}_0 \mid (a', b') = 0 \} \text{ if } b' \in \mathbb{V}_{11} \]
or
\[ A_{00} = \{ a' \in \mathbb{V}_0 \mid (a', b') > 0 \} \text{ and } A_{11} = \{ a' \in \mathbb{V}_0 \mid (a', b') = 1 \} \text{ if } b' \in \mathbb{V}_{00}. \]

Since any other \( b' \in \mathbb{V}^{n-2} \) with \( wt(b') = 1 \) involves a similar but different partition of \( \mathbb{V}_{n-1} \), we have \( |c_{00} \cup c_{11}| \leq 1 \).

We are now ready to give the characterization of all optimal constant distance code pairs.

**Theorem 4.** Any optimal constant distance code pair of length \( n \) and constant distance \( \delta \), \( 25 \leq n \), is equivalent to one of the code pairs

\[ (A_{\delta, n}, B_{\delta, n}), \quad \delta = 0, 1, \ldots, 5, \]

defined in Section 3.2.

**Proof.** We use induction on \( \delta \). With Corollary 7 we have that the theorem holds for \( \delta = 1 \). Suppose the theorem is true for \( \delta = \delta + 1 \) and let \((A, B)\) be an optimal constant distance code pair of length \( n \) and constant distance \( \delta \). Without loss of generality we may assume that (see Section 3.2)

\[ n_{12} = \frac{2\delta(n-\delta)}{m(n, \delta)}. \]

As in the proof of Theorem 3 we consider three cases.

**Case I:** \((A' \cap A_{\delta, n} = \emptyset \text{ and } \emptyset \cap B_{\delta, n} = \emptyset, \text{ for some } \emptyset \in \{0, 1\}.

Then \( n_{12} = \frac{1}{2}(\delta)\{A' \cap A_{\delta, n}\} \mid \{B' \cap B_{\delta, n}\}\mid \{A' \cap B_{\delta, n}\} = \{A \mid B_{\delta, n}\} = m(n-\delta+1) \).

And so with Lemma 4.8 and the induction hypothesis we have \((A, B)\) is equivalent to \((A' \cap A_{\delta, n})\) for some \( \emptyset \in \{1, 2, \ldots, 5\} \).

**Case II:** \((A' \cap A_{\delta, n} = \emptyset \text{ and } \emptyset \cap B_{\delta, n} = \emptyset, \text{ for some } \emptyset \in \{0, 1\}.

From Section 3.2 we then have \( n(n-1) \geq 4\delta(n-\delta), A_{\emptyset} = A_{\delta, n}, B_{\emptyset} = B_{\delta, n} \).
\[ a_{12} = 2|A_{12} - |E_{12} U B_{12}| = 2(n - 2,\bar{\delta} - 1). \] Since, \( n(n - 1) \geq 6(n - 6) \),

implies \((n - 2)\bar{\delta} - 3 > 4(\bar{\delta} - 1)(n - \bar{\delta} - 1)\). Lemma 5 and the induction hypothesis give, either

\[ A_{12} - A_{12} = \{x\}, \quad x \subseteq \mathbb{P}^n - 2 \text{ and } B_{12} U B_{12} = \mathbb{P}^n - \bar{\delta} - 1 \] (4)

or

\[ B_{12} - B_{12} = \{x\}, \quad x \subseteq \mathbb{P}^n - 2 \text{ and } A_{12} = \mathbb{P}^n - \bar{\delta} - 1. \] (5)

with Lemma 8, (4) gives \((n - 2)\bar{\delta} - 3(\bar{\delta} - 1)\), which

contradicts \(n(n - 1) \geq 6(n - 6)\). So (5) must hold. But then \(|B| = 1\)

(Lemma 9) and so with Lemma 5, \((A,B)\) equivalent to \((A_0,0, B_{n,0})\).

**Case III:** \(A_{12} \cap A_{12} = \emptyset\) and \(B_{12} \cap B_{12} = \emptyset\), for all \(e, u \in \{0,1\}\).

From Section 3.2 we have \(n(n - 1) \geq 6(n - 6)\) and

\[ a_{12} = |A_{00} U A_{12}| - |B_{00} U B_{12}| = |A_{01} U A_{12}| - |B_{01} U B_{12}| = 2(n - 2,\bar{\delta} - 1). \] (6)

and the induction hypothesis new give that \(a_{00} U A_{12} \subseteq B_{00} U B_{12}\) and \((A_{01} U A_{12}, B_{01} U B_{12})\) are equivalent to \((A_{n,\bar{\delta} - 1}, B_{n,\bar{\delta} - 1})\).

\((n - 2)\bar{\delta} - 3 > 4(\bar{\delta} - 1)(n - \bar{\delta} - 1)\). So without loss of generality we may

assume that \(A_{00} U A_{12} = \{0\}\) and \(B_{01} U B_{12} = \mathbb{P}^n - \bar{\delta} - 1\), but then with Lemma 8,

\(|A_{12} U A_{12}| = 1\), so that \(|A| = 2\). Hence, with Lemma 6, \((A,B)\) is equivalent

\((A_{n,\bar{\delta} - 1}, B_{n,\bar{\delta} - 1})\).
REFERENCES


CHAPTER 4

TORRENT CODES

4.1 INTRODUCTION

In this chapter we discuss a problem which arose in connection with comma-free codes. A $q$-ary code $C$ of length $n$ is said to be comma-free if, for every pair of words $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ of $C$ the words $(a_1, a_2, \ldots, a_{n-1}, b_n)$ and $(b_1, b_2, \ldots, b_{n-1}, a_n)$ are not in $C$. Comma-free codes were first introduced by Stickel, Grifith and Zeggel [5] as a possible genetic coding scheme for protein synthesis. The general mathematical setting of such codes was presented by Golube, Gordon and Welsh in [3]. They considered the problem of finding the maximal cardinality of such a code.

Let $W_q(n)$ denote the maximal number of codewords in any $q$-ary comma-free code of length $n$. From the definition of a comma-free code $C$ we have that no two codewords of $C$ are a cyclic permutation of each other and every codeword $(a_2, a_3, \ldots, a_n, a_1)$ of $C$ is non-periodic, i.e., there is no $1 \leq i < n$, such that

$$(a_2, a_3, \ldots, a_n, a_1) = (a_i, a_{i+1}, \ldots, a_n).$$

Hence

$$W_q(n) < \frac{q^n}{d(n)}$$

where

$$d(n) = \frac{1}{n} \sum_{d|n} \mu(d) q^{n/d}$$

is the number of non-periodic cyclic equivalence classes of sequences of length $n$ formed from an alphabet of $q$ letters. The summation in (2) is taken over all divisors $d$ of $n$ and $\mu(d)$ is the Möbius function. In [3] Golube, Gordon and Welsh proved that $W_q(n)$ attains the upper bound $d(n)$.
for arbitrary $q$ if $n = 1, 3, 5, 7, 9, 11, 15$ and conjectured that this is indeed the case for all odd $n$. This conjecture was first proved by Eastman [2], who gave a construction for maximal comma-free codes of odd wordlength. A simpler construction for these codes was found by Scholz [9].

For comma-free codes of even length, the situation becomes surprisingly complicated. It was proved by Golomb, Gordon and Welch [11] that $w_n(q) < b_n(q)$ if $q > 2^{n/2}$. In particular $w_4(q) = \left\lfloor \frac{4}{3} \right\rfloor < b_4(q) = \left\lfloor \frac{4}{3} \right\rfloor$.

$w_4(q) = b_4(q)$ if $q = 1, 2, 3$ and $w_4(q) < b_4(q)$ if $q \geq 5$. The case $n = 4$ and $q = 4$ was later solved in [6] by exhaustive computer search, which found $w_4(4) = 57 < b_4(4)$.

An improvement on the relation between $k$ and $n$ such that $w_n(q) < b_n(q)$ for even $n$ was given by Jigga [8]:

$$w_n(q) < b_n(q) \text{ if } q > 2^{n/2} + n/2.$$ 

A further improvement based on Jigga's proof was given by Golomb and Tang [4]:

$$w_n(q) < b_n(q) \text{ if } q > (n/2) \log n/2 + n/2, n \geq 9,$$

where $c = (\ln 2)/0.71$. In Section 4.3 we give a proof of this result for $c = 0.5$. Moreover the proof is much simpler than that of Golomb and Tang [4]. We first present Jigga's result. The modifications are due to Golomb and Tang.

We consider the simpler problem of finding the maximal cardinality of a $q$-ary comma-free code $D'$ of length $n = 2k$ ($k \in \mathbb{N}$) in which every word is a cyclic shift of a word of the form $\ldots 0, a, b, 0, \ldots$, where $a$ and $b$ are two different symbols of the alphabet separated by $n/2$ zeros. So $|D'| \leq \left\lfloor \frac{q}{2} \right\rfloor$. Clearly, if $|D'| \leq \left\lfloor \frac{q}{2} \right\rfloor$, then $w_n(q) < b_n(q)$.

A half-word in $D'$ is a $k$-tuple which is either the initial or final half of some word in $D'$, for each symbol $d$ of the alphabet and $r \in \mathbb{N}$, $1 \leq k$, let $y(d, x)$ denote the half-word with $d$ at the $r$-th position.
and 0 everywhere else. The half-word \( y(d,r) \) is called initial resp.
final if it equals the initial half resp. final half of some word in \( \mathcal{D}' \).

To each symbol \( d \) we assign a word \( \mathbf{w}^d = (x_1, x_2, \ldots, x_k) \in \{0, 1, *, \}^k \),
where \( x_r \) is defined in the following way

\[
x_r = \begin{cases} 
2 & \text{if } y(d,r) \text{ is both initial and final}, \\
1 & \text{if } y(d,r) \text{ is final only}, \\
0 & \text{if } y(d,r) \text{ is initial only}, \\
* & \text{if } y(d,r) \text{ is neither initial nor final}.
\end{cases}
\]

**Example.** Let \( q = 5, \ n = 2k = 4 \) and let \( \mathcal{D}' \) be given by

\[
\mathcal{D}' = \begin{bmatrix}
1 & 0 & 2 & 0 \\
1 & 0 & 3 & 0 \\
1 & 0 & 4 & 0 \\
1 & 0 & 5 & 0 \\
2 & 0 & 3 & 0 \\
2 & 0 & 4 & 0 \\
2 & 0 & 5 & 0 \\
0 & 3 & 0 & 4 \\
0 & 3 & 0 & 5 \\
0 & 4 & 0 & 5 \\
\end{bmatrix}
\]

Then

\[
\mathbf{w}^1 = (0, *) , \quad \mathbf{w}^2 = (2, *, 1) , \quad \mathbf{w}^3 = (1, 0) \\
\mathbf{w}^4 = (1, 2) , \quad \mathbf{w}^5 = (1, 1).
\]

Juggs showed that the words \( \mathbf{w}^d \) have the following two properties if

\[
| \mathcal{D}' | = \binom{q}{2}.
\]

1. If \( d \neq b \), then \( x^a \) and \( x^b \) cannot both be 2, for any \( 1 \leq r \leq k \).

2. If \( d \neq b \), there exists an \( r, 1 \leq r \leq k \), such that \( (x^a_r, x^b_r) = (0, 1) \)
or \( (x^a_r, x^b_r) = (0, 1) \).

(In particular distinct letters of the alphabet must have distinct words).
The first property implies that the number of distinct words $\mathcal{A}^n$ containing a $b$ is smaller than or equal to $\kappa$ while the second one implies that the number of words $\mathcal{A}^n$ containing no $b$ is smaller than or equal to $\kappa^n$. So, if $|\mathcal{A}^n| = \binom{\kappa}{2}$ then $\kappa \geq 2^n / \kappa'$, unless $\mathcal{A}(q) < \mathcal{B}(q)$ if $q > 2^n / \kappa'$, $n/2$.

The improvement of Jajosy results by Golomb and Tang is a consequence of the following observation.

**Theorem 1.** If $|\mathcal{A}^n| = \binom{\kappa}{2}$ then for every two different symbols $b, c$ of the alphabet and every two different coordinates $x$ and $y$ we do not have $x^a_y \neq x^a_y$ and $x^b_y \neq x^c_y$.

This observation led to the definition of a $(0,1,\ast)$ tournament code. A $(0,1,\ast)$ tournament code $C$ of length $k$ is a subset of $(0,1,\ast)^k$ such that for any two distinct codewords $a, b \in C$:

(i) $d(a, b) > 1$,

(ii) $d(a, b) \geq \left[ \binom{k}{2} \right]$, $d(a, b) = \left[ \binom{k}{2} \right] - \binom{2}{2}$, where $d(a, b) = \left[ \binom{k}{2} \right] - \binom{2}{2}$ is the distance between $a$ and $b$.

Let $t(k)$ denote the maximum number of codewords in any $(0,1,\ast)$ tournament code of length $k$. Then we have $\mathcal{A}(q) < \mathcal{B}(q)$ if $q > t(k) + \frac{1}{2}$.

In [4] Golomb and Tang prove $t(k) \leq \log_2 k$ (logarithm to base 2), with $c = \log_2(1/\epsilon)$. This upper bound and method for establishing it were suggested by E. J. G. & C. D. (in Section 4.3). We give a simple proof of this upper bound for $c=0.8$. To be complete we first give some constructions for $(0,1,\ast)$ tournament codes in Section 4.2. We conclude this chapter with the determination of the exact value of $t(k)$ for $k=1, 2, 3, \ldots, 9$ in Section 4.4.
4.3 A LOWER BOUND

We repeat the definition of a \([0,1,*]\) tournament code \(C\) of length \(k\).

**Definition 1.** A code \(C\) of length \(k\) over the alphabet \([0,1,*]\) is called a tournament code if, for any two distinct codewords \(a, b\) the following two conditions hold:

1. \(d(a, b) > 1\),
2. \(\forall i : 1 \leq i \leq k, \left[\left(\begin{array}{c} a_i \\ b_i \end{array}\right) \not= \left(\begin{array}{c} 0 \\\n1 \\end{array}\right), \left(\begin{array}{c} 0 \\\n1 \\end{array}\right)\right]\).

Since \(d(a, b) > 1\), there is at least one coordinate \(i, 1 \leq i \leq k\), such that \((a_i, b_i) = (0,1)\) or \((1,0)\), if \((a_i, b_i) = (0,1)\), it follows from condition (11) that \((a_i, b_i) \not= (1,0)\) for all \(1 \leq j \leq k\), and we shall say \(a^* b\).

This defines the tournament.

**Definition 2.** The maximal value of \(|C|\) over all tournament codes of length \(k\) is called \(t(k)\).

From now on \(C\) will always denote a \((0,1,*\)) tournament code. The matrix with as rows the codewords of \(C\) will be denoted by \(C\). If \(|C| = t(k)\), we call the code optimal.

**Lemma 3.** For every \(k \in \mathbb{N}\) there is an optimal code \(C\) of length \(k\) with \(\exists c \in C, \exists C\).

**Proof.** If \(C\) is optimal and \(c \not= C\), then clearly \(C\) must contain a word with distance 0 to \(c\), replace this word by \(c\) to obtain a new optimal code. Similarly for \(c\).

The following lemma is trivial.

**Lemma 4.** If \(C\) is optimal then \(\overline{C}\) is optimal.
Until recently Theorem 5 gave the best lower bound on \( t(h) \). The proof consists of a construction that produces a long tournament code from two shorter ones.

**Theorem 5.** \( t(k+\ell) \geq t(k) + t(\ell) - 1 \).

**Proof.** Let \( C \) be optimal of length \( k \) and let \( D \) be the top row of \( C \) and the bottom row. Similarly with \( D' \) for length \( \ell \). Consider the code with corresponding matrix,

```
  k   \ell
  e(k)  e(\ell)
  1 ... 1  0 ... 0
  c ... c
  0 ... 0
  d
  e(\ell)
```

Clearly, this is a tournament code of length \( k+\ell \) and cardinality \( t(k) + t(\ell) - 1 \).

**Corollary 5.** \( t(n) \geq 1 + n(t(n) - 1) \).

This shows that \( \lim k^{-1}t(k) \) exists (possibly \( \infty \)). For a while it was believed that this limit was \( 2 \) until Golumbic and Tart (1982) found that \( t(7) = 18 \). The following theorem due to Collins et al. (1984) shows that in fact the limit is \( \infty \).

**Theorem 7.** For \( n \in \mathbb{N} \) we have \( t(n) = \frac{1}{n(n-1)} n(n^2 + n - 1) - 2 \).
The following construction for a tournament code \( C \) of length \( n \) is due to Collins et al. (1984). The adjustments are from van Lam. [7]. We will now prove that the construction indeed gives a tournament code of size \( n(n^2-n+1) \); we refer to [7].

The code \( C \) we shall construct consists of \( \mathbb{Z}_n \) and all the cyclic shifts of the words of a set \( \{0, 1, 2, \ldots, n-1\} \). To define these words we number the coordinates with the integers mod \( n^2-n+1 \), starting with \(-1\) (i.e., for the first coordinate). The coordinates \( k \) will have their index written in the \((n+1)\)-ary system. So \((x,y)\) denotes the coordinate with index \((n+1)x+y\). Therefore \(0 \leq x \leq n-1, 0 \leq y \leq n^2-n-1\). The definition of the words \( c^k \) is as follows:

1) For each \( i \), take \( c^1_{i+1} = 1 \).
2) \( c^1 \) has
   \[ \begin{align*}
   0 & \text{ in coordinate place } (x,y) \text{ if } x \geq i \text{ and } x \equiv y \pmod{n-1}, \\
   1 & \text{ in coordinate place } (x,y) \text{ if } x < i \text{ and } x \equiv y \pmod{n-1}, \\
   * & \text{ otherwise.}
   \end{align*} \]

To be complete we mention that the first code in this class is the code of length \( n \) found by Golomb and Tang (1951). The second one of length \( n \) was independently found by Collins et al., independently by H. L. Le, Janos and Vertakol (1964). From their work we copied the following list of tournament codes of length \( n = 2, 3, \ldots, 13 \), the first digit of which are optimal (see Section 3.4). Obviously \( U(1) = 2 \). The optimal \( U(1,n) \) tournament code \( C_1 \) of length \( n \) is equal to \( C_1 = \{(1), (1)\} \).
Fig. 1. List of \( \{0,1,\ast\} \) tournament codes \( C_k \) of length \( k = 2,3,\ldots,10 \).
4.3 An Upper Bound

In this section we give a simple proof of Graham's upper bound on \( t(k) \) mentioned in Section 4.1. We first note that clearly \( t(k) \) is strictly increasing. The following observation essentially proves Graham's result.

Let \( C \) be a tournament code of length \( k \) with \( D, L \subseteq C \) (Lemma 3). By permuting rows and columns, the corresponding matrix \( C \) can be put in the following "standard form".

\[
\begin{bmatrix}
\ell & k-1 & \ell \\
A & B & \checkmark \\
\checkmark & E & \checkmark \\
F & G & \\
\end{bmatrix}
\]

Fig. 1.

Here, every column of \( A \) contains a 1 but no column of \( E \) contains a 1.
From definition 1, in particular condition (14), it follows that no column of \( A \) has a \( 0 \), while every column of \( B \) may have a \( 0 \). This shows that \( C \) is \( B \) has a standard form with \( \ell \leq \left\lfloor \frac{k-1}{2} \right\rfloor \).

**Theorem 9.** \( t(k) \leq t(k-1) + t(\ell) \), for some \( \ell \leq \left\lfloor \frac{k-1}{2} \right\rfloor \).

**Proof.** Let \( \mathcal{C} \) be an optimal code with \( C \) in "standard form" with \( \ell \leq \left\lfloor \frac{k-1}{2} \right\rfloor \) (see Fig. 1). By the definition of \( A \) the rows of the matrix \( A \) form a tournament code of length \( \ell \). So \( A \) has at most \( t(\ell) \) rows. Clearly the rows of \( \begin{pmatrix} \mathcal{C} \\ \mathcal{D} \end{pmatrix} \) form a tournament code of length \( k-1 \). The result follows.

**Corollary 9.** \( t(k) \leq \left\lceil \frac{k+1}{2} \right\rceil t\left(\left\lfloor \frac{k-1}{2} \right\rfloor\right) \).

**Proof.** Applying Theorem 8 \( \left\lceil \frac{k+1}{2} \right\rceil \) times and using the fact that \( t \) is a strictly increasing function, we have

\[
t(k) \leq t(k - \left\lfloor \frac{k-1}{2} \right\rfloor) + t\left(\left\lfloor \frac{k-1}{2} \right\rfloor\right) - t\left(\left\lfloor \frac{k-1}{2} \right\rfloor\right) = t\left(\left\lfloor \frac{k-1}{2} \right\rfloor\right) + \left\lceil \frac{k+1}{2} \right\rceil t\left(\left\lfloor \frac{k-1}{2} \right\rfloor\right) \leq t(k) \leq \left\lceil \frac{k+1}{2} \right\rceil t\left(\left\lfloor \frac{k-1}{2} \right\rfloor\right) .
\]

**Theorem 10.** \( t(k) < k^{0.5 \log k} \) for \( k > 7 \) (logarithm by base 2).

**Proof.** We use induction on \( k \). In Section 4.4 we will show that \( t(4) = 5 \), \( t(5) = 8 \), \( t(6) = 11 \), \( t(7) = 16 \), \( t(8) = 22 \) and \( t(9) = 31 \). With Theorems 8 and the above values of \( t(k) \), \( k = 4,5,\ldots,9 \), one easily checks that the assertion
is true for $8.5 < 16$ (see also Table 1 of Section 4.4).

Let $k \geq 7$, then from Corollary 0 and the induction hypothesis we have

$$t(k) \leq \left[\frac{k+3}{2}\right] t\left(\left[\frac{k-1}{2}\right]\right) +$$

$$\frac{k+3}{2} \cdot \left\lfloor \frac{k-1}{2} \right\rfloor \cdot 0.5 \log \left[\frac{k+1}{2}\right] +$$

$$\leq \left(\frac{k+3}{2}\right) \cdot \left(\frac{k-1}{2}\right) \cdot 0.5 \log \left(\frac{k}{2}\right) +$$

$$\frac{(k+3)^2}{k} \cdot 0.5 \log k \leq 0.5 \log k$$

There is a tremendous gap between the upper bound of this section and the lower bound of Section 4.2. The upper bound is probably not too good but improving it does not look easy. The following section gives an indication.

### 4.4 The Exact Value of $t(k)$ for $k = 2, 3, \ldots, 9$

In this section we will prove that the codes of length $k$, $2 \leq k \leq 9$, of Section 4.2 are optimal. It is clear that this is indeed the case for $k \leq 7$, while the case $k = 3$ follows directly from Theorem 0 and $t(1) = 2$. To prove $t(4) = 8$ and $t(5) = 8$, we use the following obvious lemma.

**Lemma 11.** For any tournament code $C$ of length $k$ we have,

$$\sum_{c \in C} \gamma^2(c) \leq \gamma^k,$$

where $\gamma_{+}(c) = |\{i : c_i = +1\}|$. 

PROOF. Since any two codewords from $C$ have distance greater than or equal to 1, any binary word of length $k$ can have distance 0 to at most one codeword of $C$. For each codeword $c \in C$ there are clearly $2^{k_s(c)}$ different binary words of length $k$ having distance 0 to $c$. The result follows.

We will use this lemma to show that $t(4) \leq 6$. From Theorem 6 we have $t(4) + t(3) + t(1) = 7$. Assume $t(4) = 7$ and let $C$ be an optimal code of length 4 with $s(C) = 6$. Since equality holds in Theorem 8 every column of $C$ can contain at most $t(1) = t(2) = 2^2 = 4$ non-$*$ elements. Hence $C$ contains at least $4 \times (7-3) = 16$ stars. From Lemma 11 and $s(C) = 6$ we then have

$$2 \cdot \sum_{c \in C} 2^{k_s(c)} \leq 2^4 \cdot \sum_{c \in C} n_s(c) \leq 16,$$

This is impossible. Hence $t(4) \leq 6$. The case $k=5$ is similar.

The case $k=6$ and $k=7$ are again a direct consequence of Theorem 6 and Section 4.3. On we are left with $k=8$ and $k=9$. For these cases we need some new machinery.

Let $C$ be a tournament code of length $k$ and let the coordinates be numbered from 1 up to $k$. Then we define

$$i < j : = \exists c \in C \left( c_i = 0 \text{ and } c_j = 1 \right).$$

Furthermore we define the vectors $a^r, b^r, c^r \in \{0,1\}^k$, $r = 1,2, \ldots, k$, by:

- $a^r$ has 1 in coordinate place $r$ if $i < k$,
  - * otherwise

and

- $b^r$ has 0 in coordinate place $r$ if $i < k$,
  - * otherwise.
Remark. The words \( a^r, b^r, r = 1,2, \ldots, k \) satisfy condition (ii) of (2).

**Lemma 12.** Let \( C \) be an optimal tournament code of length \( k \), for which \( c \in C \setminus \{g\} \) is minimal among all optimal codes of length \( k \) and let \( a^r \) and \( b^r \), \( r = 1,2, \ldots, k \), be defined as above. Then the set of words \( \{a^r \mid r = 1,2, \ldots, k\} \cup \{b^r \mid r = 1,2, \ldots, k\} \cup C \) satisfies condition (ii) of (2). Furthermore, for every \( r = 1,2, \ldots, k \) there is a unique word \( c \in C \) with distance 0 to \( a^r \). Similarly for \( b^r \).

**Proof.** The first assertion of Lemma 12 is a direct consequence of the definitions of \( a^r \) respectively \( b^r \), \( r = 1,2, \ldots, k \). So we only have to prove the second one.

Since \( C \) is optimal, there is at least one word of \( C \) that has distance 0 to \( a^r \). Assume there are two different codewords \( c \) and \( d \) in \( C \) that have distance 0 to \( a^r \). Since \( c \) and \( d \) have distance greater than or equal to 1, there is an \( i, 1 \leq i \leq k \), where \( c_i = 0 \) and \( d_i = 1 \) say. Since \( c \) and \( d \) both have distance 0 to \( a^r \), \( a^r_i \neq a^r_j \). From the definition of \( a^r \) we then have \( r \neq i \) and \( r \neq j \). So \( d_i = 1 \). Now define \( \mathbf{d}^r \in \{0,1\}^k \) by

\[
\mathbf{d}^r_i = \begin{cases} 1 & \text{if } i = r, \\ d_i & \text{if } i \neq r. \end{cases}
\]

Since, for all \( i \) with \( d_i^r = 0 \), also \( d_i = 0 \) and so \( a^r_i = a^r_j \), it follows \( r \neq i \).

So the words of \( \{a^r \cup C\} \) satisfy condition (ii) of (2).

But then \( C' = (\mathbf{d}^r) \cup C \setminus \{g\} \) is an optimal tournament of length \( k \) with

\[
\sum_{c \in C'} n_c(\mathbf{d}^r) = \sum_{c \in C} n_c(\mathbf{d}^r) + 1. \]

A contradiction.

The following lemma is "trivial." The words \( a^r, b^r, r = 1,2, \ldots, k \), are as defined above.
LEMMA 12. A codeword $c \in C, c \neq 0, 1$ that has distance greater than or equal to $1$ to all the words of $\{x^2 | r = 1, 2, \ldots, k\} \cup \{y^2 | x = 1, 2, \ldots, k\}$ has at least three coordinates equal to 0 and at least three coordinates equal to 1.

For any $\lambda \in \{0, 1, x\}$ and $\gamma \in \{0, 1, x\}^k$, let $n_\gamma(c)$ denote the number of coordinate planes $i$ with $c_i = \gamma_i$. The following lemma is obvious.

LEMMA 13. Let $C$ be an optimal code of length $k$. Then for all $c \in C$

$$(t(k) = k) \Rightarrow n_0(c) = n_1(c) = 1.$$  

PROOF. By permuting rows and columns one can achieve that $C$ looks like

$$
\begin{array}{c|c|c|}
A & B & D \\
\hline
x & y & x \\
\hline
C & E & D
\end{array}
$$

where each row of $A$ contains at least one $1$. It follows that $B$ is a matrix with all entries equal to 0 or $*$ and $F$ is a matrix with all entries equal to 0 or $*$. Hence the rows of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ respectively $\begin{bmatrix} A \end{bmatrix}$ form a tournament code of length $n-n_0(c)$ respectively $n-n_0(c)$. The result follows.

COROLLARY 15. $t(8) = 18$ and $t(9) = 21.$
**Proof.** From Section 4.2 we already have \( t(8) \geq 18. \)

Let \( C \) be an optimal code of length 8 for which the sum \( \sum_{g \in C} n_x(g) \) is minimal among all optimal codes of length 8. Let the words \( \alpha^x \) and \( \beta^x, x = 1, 2, \ldots, 8, \) be defined as above. From Lemma 14 and \( t(8) \geq 18 \) it follows that there is no codeword \( g \in C \) with both \( n_x(g) \geq 1 \) and \( n_3(g) \geq 3. \) Hence with Lemma 12 and 13, \( t(8) \geq \left| \{ x^3 \mid x = 1, 2, \ldots, 8 \} \right| + \left| \{ x^1 \mid x = 1, 2, \ldots, 8 \} \right| = 26 \times 6 + 2 = 18. \)

The proof of \( t(9) = 21 \) is similar to that of \( t(8) = 18. \)

---

We conclude this section with a small table of lower and upper bounds on \( t(k) \) for \( k = 10, 11, \ldots, 21. \) In the last column of Table 1 we indicate the theorems and lemmas we used to derive the upper bound.

The lower bounds are from Abebe, Janss, and Verhulst [1].

<table>
<thead>
<tr>
<th>( k )</th>
<th>Lower bound on ( t(k) )</th>
<th>Upper bound on ( t(k) )</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>23</td>
<td>27</td>
<td>Th 8</td>
</tr>
<tr>
<td>11</td>
<td>27</td>
<td>33</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>12</td>
<td>33</td>
<td>40</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>13</td>
<td>41</td>
<td>48</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>14</td>
<td>43</td>
<td>57</td>
<td>Th 8</td>
</tr>
<tr>
<td>15</td>
<td>46</td>
<td>73</td>
<td>Th 8</td>
</tr>
<tr>
<td>16</td>
<td>49</td>
<td>81</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>17</td>
<td>54</td>
<td>92</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>18</td>
<td>59</td>
<td>108</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>19</td>
<td>66</td>
<td>124</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>20</td>
<td>75</td>
<td>141</td>
<td>Th 8 + L 11</td>
</tr>
<tr>
<td>21</td>
<td>86</td>
<td>159</td>
<td>Th 8 + L 11</td>
</tr>
</tbody>
</table>

Table 1.


[7] Van LINT, J.H.: "(0, 1) Distance Problems in Combinatorics". Internal
Report Technological University Eindhoven.

SAMENVATTING

In dit proefschrift worden vier problemen uit de coderingstheorie behandeld, met als voornaamste doel het bepalen van bovengrenzen voor de maximale cardinaliteit van en het vinden van goede constructieve methoden voor de betreffende codes. Het begrip afstand speelt hierbij een belangrijke rol. Om de lezer snelst mogelijkdra te maken met de in dit proefschrift gebruikte terminologie, geven we in hoofdstuk 0 een korte inleiding in de coderingstheorie.

In hoofdstuk 1 houden wij ons voornamelijk bezig met het bepalen van blok codes voor de betrouwbaar overdracht van informatie in computer geheugen met defecten en zumbed errors. De hier behandelde constructieve methoden doen een stempel brengen op reeds bestaande constructies voor lineaire en niet-lineaire codes voor het binaire symmetrisch kanaal.

In hoofdstuk 2 behandelen we twee constructieve methoden voor constante gewicht codes. Voor de tweede constructie geeft scherpe verbeteringen op reeds bestaande ondergrenzen voor $A(n,k,w)$. Het bepalen van partities van $[n]\times [w] / \leq (g \times w)$ in zo weinig mogelijk constante gewicht codes met minimum afstand $d$ is in deze constructie van cruciaal belang.

In hoofdstuk 3 geven we de volledige classificatie van alle optimale code paren van lengte $n$ en constante afstand $d$, $n \in \mathbb{N}$, $0 < d < n$. Daarom bepalen we eerst de waarden van $\text{Min}(n,d)$,

$$\text{Min}(n,d) = \{A | |B| \leq A, B \subseteq \mathbb{F}_2^n, \Delta(A,B) = d\}.$$

In het laatste hoofdstuk houden we ons bezig met het bepalen van boven grenzen voor de maximale cardinaliteit $t(k)$ van $(n,1,k)$ toernooisch codes van lengte $n$, $k \in \mathbb{N}$. We geven een verandering van Graham's bovengrens voor $t(k)$, $k > 7$, en bepalen vervolgens de exacte waarden van $t(k)$ voor $k = 1, 2, ..., 9$. 
CURRICULUM VITAE


Vanaf 1974 studeerde hij verder aan de Technische Hogeschool in Eindhoven, waar hij in 1982 zijn doctoraal examen aflegde. Tijdens zijn studie was hij gedurende 2 jaar student-assistent.

Van september 1982 tot september 1986 was hij wetenschappelijk assistent bij de onderafdeling dat wiskunde en informatica van de bovengenoemde hogeschool. Thans is hij werkzaam bij Philips-LCTA, Eindhoven.
STELLINGEN

1. Zij $C$ een binair lineaire code met woord lengte $n$, dimensie $k$ en minimum afstand $d$. Dan geldt: $k < 12$, en dus $B(26,8) = 212$


2. Er bestaat een uniek decodeerbaar code paar $(C, P)$ voor het two-access binary adder channel met woord lengte $n$ waarvan de som rate $R_1 + R_2$ gelijk is aan
   $$R_1 + R_2 = 0.59745 \times 0.72032 = 1.11781$$


3. Zij $C$ een $(0,1,\ast)$ tournament code van lengte 10. Dan geldt $|C| = 23$. Daar volgt dit tabel 1 van hoofdstuk 4 dat
   $$s(10) = 23.$$

4. Zij $C$ een constant gewicht code van lengte 17, minimum afstand 8 en constant gewicht 8. Dan geldt: $|C| = 34$, en dus
   $$A(17,9,8) = 34.$$


5. Zij $m_k(n,\delta)$ het max $\{ |A| \cdot |B| \mid A, B \subseteq \{0, 1, \ldots, k-1\}, \delta(A, B) = \delta \}$.
   Vermoedelijk is de waarde van $m_k(n, \delta)$ gelijk aan
\[ N_k(n, \delta) = \begin{cases} \max \left\{ 10^{-1} \left( \frac{k-1}{k-2} \right)^{\frac{3}{2}} \left( \frac{1}{2} \right) \right\} & 0 \leq \delta \leq \max \left( \frac{2^{k-2}}{\delta}, \frac{1}{2} \right) \text{ als } k = 3, \\ \max \left\{ (\frac{5}{2})^{\frac{1}{2}} \left( \frac{1}{2} \right)^{\frac{3}{2}} \left( \frac{1}{2} \right) \right\} & 0 \leq \delta \leq \frac{5}{4} \text{ als } k = 4, \\ \end{cases} \]

Voor \( k = 4, \delta \) is dit bewezen door Ahlawade (to appear) en voor \( k = 10 \) door van Pul (unpublished).

6 De lineaire programmeringsgrens voor binaire codes is een generalisatie van de Plotkin grens.

7 Zij \( C \) een binair \( [n, k, d] \)-code met overdekkingstraal \( \delta \).
\[ \exists \| C_i \| = \{ x \in [0,1]^n | C_i \subseteq C \}, \quad i=1,2,\ldots,n \text{ en } \forall = 0,1. \]
De norm \( N \) van de code \( C \) wordt gedefinieerd door
\[ N := \min_{x \in \mathbb{F}_2^n} \max_{i=1}^n (d(x, C_i) + d(x, C_{\bar{i}})). \]
Dus geldt
\[ N \geq 2n - 1 + \left[ \frac{n}{2} \right]. \]

Als \( N \geq 2n - 1 \), dan noemen we de code \( C \) normaal. Uit bovenstaande ongelijkheid volgt dat elke enige binair lineaire code met minimum afstand \( \leq n \) normaal is. Vermoedelijk zijn alle binair lineaire codes normaal.


6 Standaardisatie van cryptosystemen leidt tot diversificatie.

9 Ondanks de resultaten van Tafman, Viardot en Zink, is het vermoedelijk waar dat de Gilbert-Varshamov grens voor binair krommen scherper is.


10 De ABC-regeling voor jonge onderzoekers is equivalent met de RKR-regeling voor beeldende kunstenaars.