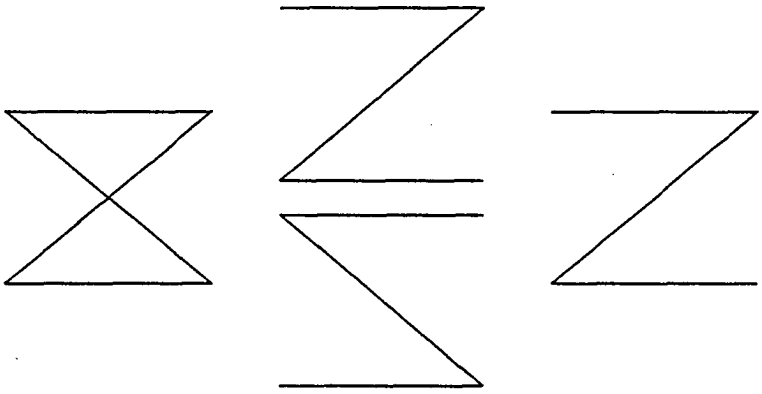
Bounds and Constructions for Binary Block Codes Correcting Asymmetric or Unidirectional Errors



J.H. Weber

Bounds and Constructions for Binary Block Codes Correcting Asymmetric or Unidirectional Errors

Bounds and Constructions for Binary Block Codes Correcting Asymmetric or Unidirectional Errors

Proefschrift

ter verkrijging van de graad van doctor aan de
Technische Universiteit Delft, op gezag van
de Rector Magnificus, prof. drs. P.A. Schenck,
in het openbaar te verdedigen ten overstaan van
een commissie aangewezen door het College van Dekanen
op donderdag 23 november 1989 te 16.00 uur

door

Jacobus Hendricus Weber

geboren te Schiedam

wiskundig ingenieur

Dit proefschrift is goedgekeurd door de promotoren
prof. dr. ir. D.E. Boekee
en
prof. dr. C. de Vroedt

Grenzen en constructies voor binaire blokcodes welke asymmetrische of unidirectionel fouten verbeteren

Aan *Karin*

Contents

Summary

This thesis deals with error correcting block codes for reliable transmission or storage of data in a communication system using a binary channel. Most classes of codes have been designed for use on symmetric channels, on which $0 \rightarrow 1$ cross-overs and $1 \rightarrow 0$ cross-overs occur with equal probability (*symmetric errors*). However, in certain applications, such as optical communications, the error probability from 1 to 0 is significantly higher than the error probability from 0 to 1. These applications can be modeled by an asymmetric channel, on which only $1 \rightarrow 0$ transitions can occur (*asymmetric errors*). Further, some recently developed memory systems behave like a unidirectional channel, on which, even though both $1 \rightarrow 0$ and $0 \rightarrow 1$ errors are possible, all errors are of the same type when sending a certain codeword (*unidirectional errors*).

Codes correcting symmetric errors have been studied extensively. Of course, these codes can also be used to correct asymmetric or unidirectional errors. However, it is likely that codes correcting asymmetric or unidirectional errors that need less redundancy than a comparable symmetric error correcting code can be constructed. The main object of this thesis is to provide upper and lower bounds on the maximum cardinality of a code of length n which corrects up to t asymmetric or unidirectional errors.

In Chapter 1 we sketch the overall communication system, give a brief introduction on binary block codes, and treat the symmetric, unidirectional, and asymmetric error types. Further, general conditions for error correction and/or detection capabilities of block codes are derived, where we focus on the correction of one single error type.

Upper bounds on the maximum cardinality of asymmetric or unidirectional error correcting codes are treated in Chapters 2, 3, and 4. First in Chapter 2, we give explicit upper bounds that are based on the sphere-packing concepts or make use of known upper bounds on the maximum

cardinality of symmetric error correcting codes. For codes of relatively small lengths the best of these bounds are shown to be tight. Hence we know the exact value of the maximum cardinality in these cases. Then in Chapter 3, we treat upper bounds that can be obtained by solving an integer programming problem. In these programming problems the total number of codewords in a code is maximized over certain constraints on the weight distribution of the code. The integer programming bounds often improve on the explicit bounds, but are much harder to calculate. Finally in Chapter 4, we improve the integer programming bound in a number of specific cases by using some combinatorial arguments that seem hard to generalize.

Lower bounds on the maximum cardinality of asymmetric or unidirectional error correcting codes can be obtained by constructing codes, which is the subject of Chapters 5 and 6. In Chapter 5, we present a general construction method in which codes correcting up to t asymmetric or unidirectional errors are obtained by expurgating and puncturing an initial code that corrects up to t symmetric errors. Algorithms to optimize the cardinality of such codes and decoding aspects are also discussed. In Chapter 6 we apply this method and some trial and error techniques to obtain good codes of length up to 23 that correct up to 1, 2, 3, or 4 asymmetric or unidirectional errors.

Bounds on the cardinality of codes that are *linear* are discussed in Chapter 7. It is shown that any linear code correcting up to t unidirectional errors also corrects up to t symmetric errors. Therefore we only consider linear symmetric error correcting codes and linear asymmetric error correcting codes. Again the maximum cardinality can be exactly determined when the length is relatively small. For single and double error correction we show that the maximum cardinality of a linear asymmetric error correcting code exceeds the maximum cardinality of a linear symmetric error correcting code for only a finite number of lengths. On the construction side, we derive a class of linear codes of length 2^m correcting up to $2^{m-2}-1$ asymmetric errors for all even integers m not less than 4, the cardinality of which exceeds the largest cardinality of a comparable symmetric error correcting code.

In the appendices we give tables with bounds on the maximum number of codewords in a code of length n correcting up to t symmetric, up to t unidirectional, or up to t asymmetric errors as well as tables with bounds on the maximum dimension in a linear code of length n correcting up to t symmetric or up to t asymmetric errors, in all cases for $1 \leq t \leq 4$ and

$t \leq n \leq 23.$

Chapter 1

Introduction

1.1 The overall communication system

Error control coding has shown itself to be a powerful tool in obtaining efficient and reliable transmission of data over a noisy channel (see e.g. [20,31]). A simple model of a communication system in which error control coding is applied is shown in Figure 1.1. A source generates a message u which is an element of a message set $\mathcal{U}$. The number of possible messages, i.e. the cardinality $|\mathcal{U}|$ of the set $\mathcal{U}$, is denoted by M. The message u has to be transmitted to a user over a noisy binary channel. To this end the message u is encoded into a codeword $\mathbf{c}$ which is an element of a binary block code $\mathcal{C}$. Such a code is a subset of $(GF(2))^n$, the vector space of all binary vectors of length n. More details about these codes will be treated in Section 1.2. The encoding is performed by a bijection $f : \mathcal{U} \mapsto \mathcal{C}$. Hence

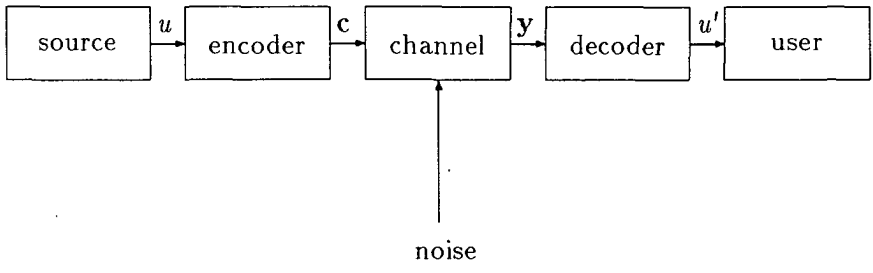


Figure 1.1: The overall communication system.

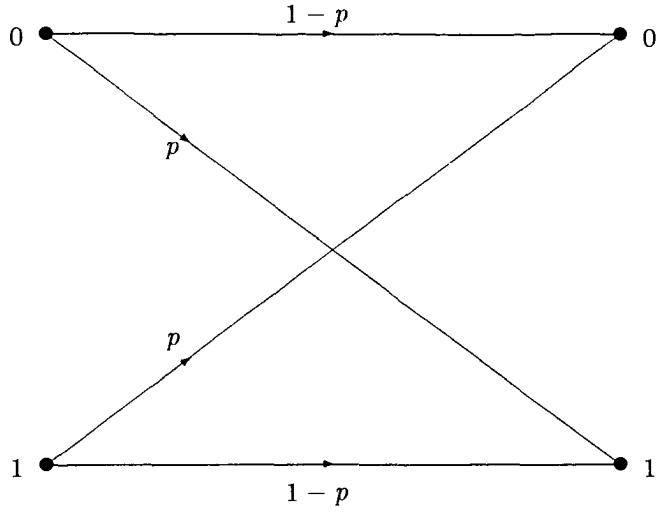


Figure 1.2: Binary Symmetric Channel.

$|C| = M$ and $n \geq \log_2(M)$. Next the codeword $\mathbf{c}$ is sent over a binary channel, on which are both the input alphabet and output alphabet $\{0, 1\}$. Because of channel noise, the received vector $\mathbf{y}$ may be different from $\mathbf{c}$. The number of coordinates in which $\mathbf{c}$ and $\mathbf{y}$ differ is called the number of errors made during transmission. The decoder must decide from $\mathbf{y}$ which message or, equivalently, which codeword was transmitted. This is performed by a surjection $g : (GF(2))^n \mapsto \mathcal{U}$. Then $u' = g(\mathbf{y})$ is an estimation of the original message u.

From the foregoing, it will be clear that the efficiency and reliability of such a system depend on the behavior of the channel, the choice of the code C, and the encoding and decoding functions f and g. A widely used channel model is the Binary Symmetric (Memoryless) Channel shown in Figure 1.2, on which both $0 \rightarrow 1$ and $1 \rightarrow 0$ cross-overs occur with equal probability p. Many good codes have been constructed for this channel, and efficient additional functions f and g have been developed (see e.g. [21]). In this thesis two other channel models are considered: the Binary Asymmetric Channel and the Binary Unidirectional Channel. These models will be discussed in Section 1.3.

To obtain reliable communication we demand that the decoder must

be able to indicate the correct codeword although some errors could be made during transmission. Let $\mathcal{B}(\mathbf{c},t)$ denote the set of all binary vectors of length n that can arise from the codeword $\mathbf{c}$ suffering t or less errors. We say that the code $\mathcal{C}$ is able to correct up to t errors if the sets $\mathcal{B}(\mathbf{c}_1,t)$ and $\mathcal{B}(\mathbf{c}_2,t)$ are disjoint for all $\mathbf{c}_1,\mathbf{c}_2 \in \mathcal{C}$ with $\mathbf{c}_1 \neq \mathbf{c}_2$. If a vector $\mathbf{y}$ is received that belongs to a certain $\mathcal{B}(\mathbf{c},t)$, then this vector can be decoded into $u' = f^{-1}(\mathbf{c})$. If a vector is received that does not belong to any $\mathcal{B}(\mathbf{c},t)$, then we can conclude that at least t+1 errors have been made. If the occurence of $t+1, t+2, \ldots, s$ errors (with $s \geq t$) always leads to a received vector outside the union of all the sets $\mathcal{B}(\mathbf{c},t)$, then we say that the code is also able to detect up to s errors: the code is t error correcting and s error detecting (t-EC s-ED). Necessary and sufficient conditions for a code to be t-EC s-ED will be derived in Sections 1.4 and 1.5 for various channels.

In the communication system considered in this section we speak of sending, receiving etc. Nevertheless, one should not think only of the transmission of messages from one place to another, but also of the storage of data ('transmission in time'). For of various reasons the data may be disturbed during storage in a memory system. Hence coding techniques are also often applied in these situations.

1.2 Binary block codes

As mentioned in Section 1.1, a binary block code $\mathcal{C}$ is a subset of $(GF(2))^n$. Therefore, we first discuss $(GF(2))^n$. Here $GF(2)$ is the field containing only two elements 0 and 1, on which the following addition + and multiplication · are defined:

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} \quad \text{and} \quad \begin{array}{c|cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} . \tag{1.1}$$

The n-dimensional vector space $(GF(2))^n$ (with $n \geq 1$) consists of all vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where each $x_i \in GF(2)$. In such a vector x_i is called the i^{th} symbol or i^{th} coordinate. Since each x_i is equal to 0 or 1, the cardinality of $(GF(2))^n$ is equal to 2^n. The number of coordinates that equal 1 in $\mathbf{x}$ is called the *weight* $w(\mathbf{x})$ of $\mathbf{x}$:

$$w(\mathbf{x}) = |\{i | x_i = 1\}|. \tag{1.2}$$

For two vectors $\mathbf{u}$ and $\mathbf{v}$ from $(GF(2))^n$, let $N(\mathbf{u},\mathbf{v})$ denote the number of coordinates where $\mathbf{u}$ equals 0 and $\mathbf{v}$ equals 1:

$$N(\mathbf{u},\mathbf{v}) = |\{i|u_i = 0 \wedge v_i = 1\}|. \tag{1.3}$$

If $N(\mathbf{u},\mathbf{v}) = 0$ the vector $\mathbf{u}$ is said to *cover* the vector $\mathbf{v}$ ($\mathbf{u} \geq \mathbf{v}$). If $\mathbf{u} \geq \mathbf{v}$ or $\mathbf{v} \geq \mathbf{u}$ the vectors $\mathbf{u}$ and $\mathbf{v}$ are said to be *ordered*, otherwise they are said to be *unordered*. The *Hamming distance* $d(\mathbf{u},\mathbf{v})$ between $\mathbf{u}$ and $\mathbf{v}$ is defined to be the sum of $N(\mathbf{u},\mathbf{v})$ and $N(\mathbf{v},\mathbf{u})$:

$$d(\mathbf{u},\mathbf{v}) = N(\mathbf{u},\mathbf{v}) + N(\mathbf{v},\mathbf{u}) = |\{i|u_i \neq v_i\}|. \tag{1.4}$$

For example we consider the vectors

$$\begin{aligned} \mathbf{u} &= 111111000000 \\ \mathbf{v} &= 110000111110 \end{aligned}$$

giving $w(\mathbf{u}) = 6$, $w(\mathbf{v}) = 7$, $N(\mathbf{u},\mathbf{v}) = 5$, $N(\mathbf{v},\mathbf{u}) = 4$, and $d(\mathbf{u},\mathbf{v}) = 5 + 4 = 9$.

It can easily be checked that $d(\mathbf{u},\mathbf{v})$ satisfies the demands to be a metric on $(GF(2))^n$:

$$\begin{array}{lll} \text{(i)} & d(\mathbf{u},\mathbf{v}) \geq 0 & \text{for all } \mathbf{u},\mathbf{v} \in (GF(2))^n \\ \text{(ii)} & d(\mathbf{u},\mathbf{v}) = 0 \Leftrightarrow \mathbf{u} = \mathbf{v} & \text{for all } \mathbf{u},\mathbf{v} \in (GF(2))^n \\ \text{(iii)} & d(\mathbf{u},\mathbf{v}) = d(\mathbf{v},\mathbf{u}) & \text{for all } \mathbf{u},\mathbf{v} \in (GF(2))^n \\ \text{(iv)} & d(\mathbf{u},\mathbf{v}) \leq d(\mathbf{u},\mathbf{w}) + d(\mathbf{w},\mathbf{v}) & \text{for all } \mathbf{u},\mathbf{v},\mathbf{w} \in (GF(2))^n \\ & & \textit{(triangle inequality).} \end{array} \tag{1.5}$$

The (minimum) Hamming distance d of a code $\mathcal{C}$ is now defined to be:

$$d = \min\{d(\mathbf{u},\mathbf{v})|\mathbf{u},\mathbf{v} \in \mathcal{C} \wedge \mathbf{u} \neq \mathbf{v}\}. \tag{1.6}$$

This distance is important with respect to the error correcting/detecting capability of the code.

So far we have met the following three parameters of a code $\mathcal{C}$:

n : the number of symbols in each codeword, which is called the *length* of the code;

M : the number of codewords, which is called the *cardinality* or *size* of the code;

d : the minimum Hamming distance between any two different codewords, which is called the *Hamming distance* of the code.

Further, let A_i denote the number of codewords of weight i in $\mathcal{C}$:

$$A_i = |\{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) = i\}| \text{ for } i = 0, 1, \ldots, n. \tag{1.7}$$

The numbers $A_0, A_1, \ldots, A_n$ are called the *weight distribution* of $\mathcal{C}$. A code in which all the codewords have the same weight w is called a *constant weight* code of weight w. The *rate* R of a code, defined by

$$R = \frac{\log_2(M)}{n}, \tag{1.8}$$

is a measure for the efficiency of the code. In general, we want codes with a high rate ('efficiency') and a large Hamming distance ('reliability'). Regretfully, these are conflicting goals. Very important in this respect is the following function:

$$A(n,d) \quad : \quad \text{the maximum number of codewords in a code of length } n \text{ and Hamming distance at least } d. \tag{1.9}$$

Both upper bounds $A^u(n,d)$ and lower bounds $A^l(n,d)$ on $A(n,d)$ have been investigated extensively (see e.g. [21]). When the best (smallest) known upper bound meets the best (largest) known lower bound the exact value of $A(n,d)$ has been determined. Two other important functions in coding theory are:

$$A(n,d,w) \quad : \quad \text{the maximum number of codewords in a code of length } n, \text{ Hamming distance at least } d, \text{ and constant weight } w; \tag{1.10}$$

$$T(w_1,n_1,w_2,n_2,d) \quad : \quad \text{the maximum number of codewords in a code of length } n_1+n_2 \text{ and Hamming distance at least d, where each codeword has exactly } w_1 \text{ ones in the first } n_1 \text{ coordinates and exactly } w_2 \text{ ones in the last } n_2 \text{ coordinates.} \tag{1.11}$$

Upper and lower bounds on $A(n,d,w)$ will be denoted by $A^u(n,d,w)$ and $A^l(n,d,w)$, respectively. Analogously, upper and lower bounds on $T(w_1,n_1,w_2,n_2,d)$ will be denoted by $T^u(w_1,n_1,w_2,n_2,d)$, and $T^l(w_1,n_1,w_2,n_2,d)$, respectively.

If $\mathcal{C}$ is a *subspace* of $(GF(2))^n$ the code is called *linear*. In this case the sum of any two codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ is again a codeword:

$$\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C} \Rightarrow \mathbf{c}_1 + \mathbf{c}_2 \in \mathcal{C}. \tag{1.12}$$

This property can be very useful with regard to encoding and decoding aspects. The dimension of the subspace will be denoted by k. As is well known from linear algebra, such a subspace can be generated by k basis vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k$:

$$\mathbf{c} \in \mathcal{C} \Leftrightarrow \mathbf{c} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_k\mathbf{b}_k \tag{1.13}$$

with $a_i \in GF(2)$. Hence the cardinality of the code equals 2^k. A generator matrix $\mathbf{G}$ of a linear code $\mathcal{C}$ is a $k \times n$ matrix, the rows of which form a basis of $\mathcal{C}$. Note that any linear code contains the all-zero vector, which will be denoted by $\mathbf{0}$ or 0^n. In a linear code $\mathcal{C}$ the Hamming distance equals the minimum weight over all codewords except the all-zero vector:

$$d = \min\{d(\mathbf{u}, \mathbf{v}) | \mathbf{u}, \mathbf{v} \in \mathcal{C} \wedge \mathbf{u} \neq \mathbf{v}\} = \min\{w(\mathbf{c}) | \mathbf{c} \in \mathcal{C} \wedge \mathbf{c} \neq \mathbf{0}\}. \tag{1.14}$$

For any vector $\mathbf{a} \in (GF(2))^n$, the set

$$\mathbf{a} + \mathcal{C} = \{\mathbf{a} + \mathbf{c} | \mathbf{c} \in \mathcal{C}\} \tag{1.15}$$

is called a *coset* of $\mathcal{C}$. The code $\mathcal{C}$ and any of its cosets have the same length n, cardinality M, and Hamming distance d.

Finally, two codes are called *equivalent* if they differ only in the order of the coordinates. Hence equivalent codes have the same parameters n, M, and d.

1.3 Symmetric, unidirectional, and asymmetric errors

As mentioned in Section 1.1 many codes have been designed for the correction and/or detection of errors caused by a binary symmetric channel as shown in Figure 1.2. The assumption that the $0 \rightarrow 1$ cross-over ('0-error') and the $1 \rightarrow 0$ cross-over ('1-error') occur with equal probability has shown to be quite reasonable in many applications. However, in some applications, such as optical communications, the errors have a highly asymmetric

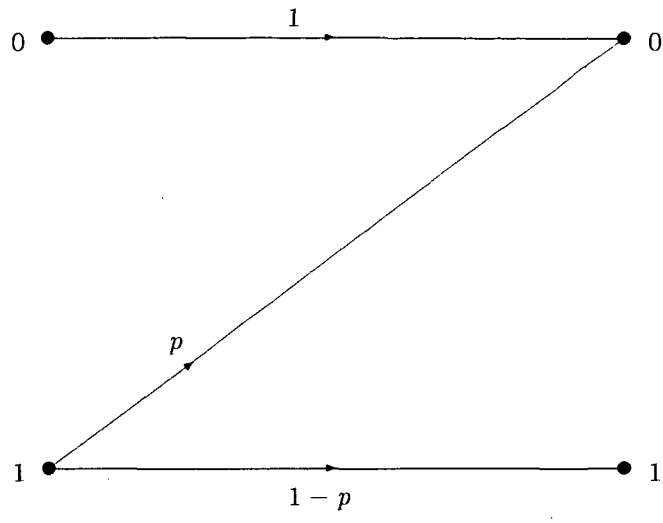


Figure 1.3: Binary Asymmetric Channel (Z-channel).

nature. Channels causing this kind of error can often be modeled by a *binary asymmetric channel* or *Z-channel* (see Figure 1.3), on which only 1-errors can occur. As an example, we mention the photon communication systems as described in [37], in which photons are used to transmit the information. Due to energy losses in the channel a photon may not be received. Since the number of received photons does not exceed the number of transmitted photons, the photon channel is an asymmetric channel.

Further, in some recently developed memory systems (cf. [5]), the errors appear to be of a unidirectional nature. These memory systems can be modeled by a *binary unidirectional channel*, which behaves for a certain codeword either like the Z-channel or like the inverted Z-channel, on which only 0-errors can occur (see Figure 1.4). For example (cf. [28]) we mention that the faults that affect address decoders often cause unidirectional errors, since this will result in either no access or multiple access. No access yields an all-zero vector readout, while multiple access causes the OR of several vectors to be read out. In the former case we only have 1-errors and in the latter case we only have 0-errors if the correct codeword is contained in the accessed set. Other sources of unidirectional errors are failures in one of the shift registers in a shift register type memory or failures in the

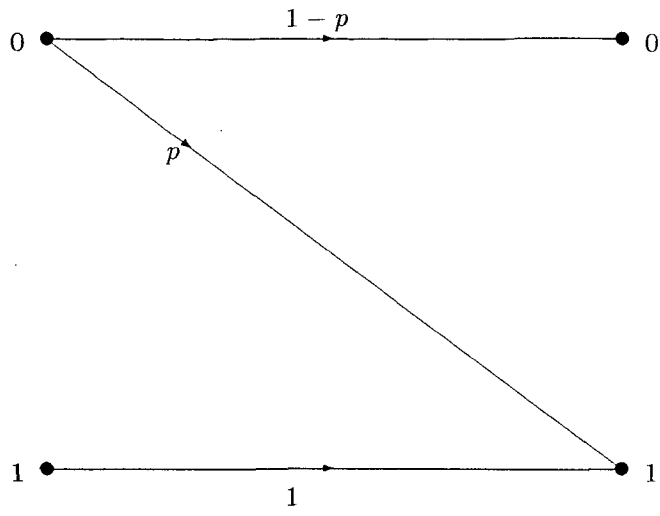


Figure 1.4: Inverted Z-channel.

power supply.

Based on the preceding statements, we shall now define the error types that will be considered. When sending a codeword $\mathbf{c} \in \mathcal{C}$ and receiving a vector $\mathbf{y} \in (GF(2))^n$, we say that $\mathbf{c}$ has suffered t *asymmetric errors* if $\mathbf{c}$ covers $\mathbf{y}$ and $d(\mathbf{c},\mathbf{y}) = t$, that $\mathbf{c}$ has suffered t *unidirectional errors* if $\mathbf{c}$ covers $\mathbf{y}$ or is covered by $\mathbf{y}$ and $d(\mathbf{c},\mathbf{y}) = t$, and that $\mathbf{c}$ has suffered t *symmetric errors* if $d(\mathbf{c},\mathbf{y}) = t$. In accordance with the three error types, we define three kinds of spheres with radius r for each $\mathbf{x} \in (GF(2))^n$:

$$S_{Sy}(\mathbf{x},r) = \{\mathbf{y} \in (GF(2))^n | d(\mathbf{x},\mathbf{y}) \leq r\} \quad (1.16)$$

$$S_U(\mathbf{x},r) = \{\mathbf{y} \in (GF(2))^n | d(\mathbf{x},\mathbf{y}) \leq r \wedge (\mathbf{x} \geq \mathbf{y} \vee \mathbf{y} \geq \mathbf{x})\} \quad (1.17)$$

$$S_{As}(\mathbf{x},r) = \{\mathbf{y} \in (GF(2))^n | d(\mathbf{x},\mathbf{y}) \leq r \wedge \mathbf{x} \geq \mathbf{y}\}. \quad (1.18)$$

For the sake of convenience we also define a super-sphere

$$S(\mathbf{x},r_1,r_2,r_3) = S_{Sy}(\mathbf{x},r_1) \cup S_U(\mathbf{x},r_2) \cup S_{As}(\mathbf{x},r_3) \quad (1.19)$$

for each $\mathbf{x} \in (GF(2))^n$ and $0 \leq r_1 \leq r_2 \leq r_3$. Each sphere $S_X(\mathbf{c},t)$ contains the vectors that can be received when codeword $\mathbf{c}$ is sent suffering t or fewer errors of type X (with $X = Sy$(mmetric), $X = U$(nidirectional), or

$X = As$(ymmetric), respectively). Hence we say that a code C can *correct* up to t errors of type X if the spheres $S_X(\mathbf{c}, t)$ are disjoint for any two distinct codewords. On the other hand, we say that a code can *detect* up to s errors of type X if the sphere $S_X(\mathbf{c}, s)$ does not contain codewords different from $\mathbf{c}$ for all $\mathbf{c} \in C$. Necessary and sufficient conditions are known for a code to be capable of correcting or detecting errors of each of the three types. However, sometimes a combination of correction and detection is required or even simultaneous correction and/or detection of errors of various types. For example, some authors (see e.g. [2,4,5,25]) considered codes correcting up to t symmetric errors and detecting all $(t+1$ or more) unidirectional errors, since it was observed that in some memory systems the number of unidirectional errors can be unlimited, while the number of symmetric errors is limited with high probability. A necessary and sufficient condition for this case was derived in [5]. To be able to deal with such cases it is interesting to look for necessary and sufficient conditions for all combinations of correction and detection capabilities for the three error types considered here.

We call a code t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED $(0 \leq t_1 \leq t_2 \leq t_3,\ 0 \leq d_1 \leq d_2 \leq d_3,\ t_i \leq d_i)$ if it can correct up to t_1 symmetric errors, up to t_2 unidirectional errors, and up to t_3 asymmetric errors, as well as detect from $t_1 + 1$ up to d_1 symmetric errors that are not of the unidirectional type, from $t_2 + 1$ up to d_2 unidirectional errors that are not of the asymmetric type, and from $t_3 + 1$ up to d_3 asymmetric errors. In the context of the spheres this means that

$$S(\mathbf{x}, t_1, t_2, t_3) \cap S(\mathbf{y}, d_1, d_2, d_3) = \emptyset \tag{1.20}$$

for any two distinct codewords $\mathbf{x}$ and $\mathbf{y}$. In the next section we derive necessary and sufficient conditions for a code to be t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED. Hence we can obtain necessary and sufficient conditions for correction and/or detection of any combination of symmetric/unidirectional/asymmetric errors by making appropriate choices for t_i and d_i.

1.4 Conditions for correction and/or detection capabilities

In the literature (see e.g. [5,7,19,21]) many necessary and sufficient conditions were derived on block codes to have certain error correcting and/or

detecting capabilities. Since in each derivation the same kinds of techniques were used, we tried to obtain general conditions including all combinations concerning symmetric, unidirectional, and asymmetric errors. The final result is given in Theorem 1.1.

Theorem 1.1 *A code $\mathcal{C}$ is t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED (with $0 \le t_1 \le t_2 \le t_3$, $0 \le d_1 \le d_2 \le d_3$, $t_i \le d_i$) if and only if all $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $N(\mathbf{a}, \mathbf{b}) \ge N(\mathbf{b}, \mathbf{a})$ satisfy*

$$\begin{cases} d(\mathbf{a},\mathbf{b}) \ge t_3 + d_2 + 1 \ \wedge d(\mathbf{a},\mathbf{b}) \ge t_2 + d_3 + 1 & \text{if } N(\mathbf{b},\mathbf{a}) = 0 \\ d(\mathbf{a},\mathbf{b}) \ge t_3 + d_1 + 1 \ \wedge d(\mathbf{a},\mathbf{b}) \ge t_1 + d_3 + 1 & \\ \qquad\qquad \wedge N(\mathbf{a},\mathbf{b}) \ge d_3 + 1 & \text{if } 1 \le N(\mathbf{b},\mathbf{a}) \le t_3 \\ d(\mathbf{a},\mathbf{b}) \ge t_3 + d_1 + 1 & \text{if } N(\mathbf{b},\mathbf{a}) \ge t_3 + 1 \end{cases}$$

Proof. A proof is given in Appendix A. □

Sometimes a code turns out to have stronger error correcting/detecting capabilities than it was originally designed for, as can be seen from the next theorem.

Theorem 1.2 *Any t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED code (with $0 \le t_1 \le t_2 \le t_3$, $0 \le d_1 \le d_2 \le d_3$, $t_i \le d_i$) is also a t'_1-SyEC t'_2-UEC t'_3-AsEC d'_1-SyED d'_2-UED d'_3-AsED code with*

$$\begin{aligned} t'_1 &= \max\{t_1, t_3 + d_1 - d_3\}, \\ t'_2 &= \max\{t_2, t_3 + d_2 - d_3\}, \\ t'_3 &= t_3, \\ d'_1 &= \max\{d_1, \min\{t_3 + 1, t_1 + d_3 - t_3\}\}, \\ d'_2 &= \max\{d_2, t_2 + d_3 - t_3\}, \\ d'_3 &= d_3. \end{aligned}$$

Proof. A proof is given in Appendix A. □

Many known results on error correcting/detecting codes appear as special cases of the general result stated in Theorem 1.1. In the next two corollaries we treat the cases of pure correction and pure detection.

Corollary 1.3 *A code $\mathcal{C}$ is t_1-SyEC t_2-UEC t_3-AsEC (with $0 \le t_1 \le t_2 \le t_3$) if and only if all $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $N(\mathbf{a}, \mathbf{b}) \ge N(\mathbf{b}, \mathbf{a})$ satisfy*

$$\begin{cases} d(\mathbf{a},\mathbf{b}) \ge t_2 + t_3 + 1 & \text{if } N(\mathbf{b},\mathbf{a}) = 0 \\ d(\mathbf{a},\mathbf{b}) \ge t_1 + t_3 + 1 \wedge N(\mathbf{a},\mathbf{b}) \ge t_3 + 1 & \text{if } N(\mathbf{b},\mathbf{a}) \ge 1 \end{cases}$$

Proof. Substitute $d_i = t_i$ $(i = 1, 2, 3)$ into Theorem 1.1. □

Corollary 1.4 *A code $\mathcal{C}$ is d_1-SyED d_2-UED d_3-AsED (with $0 \leq d_1 \leq d_2 \leq d_3$) if and only if all $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $N(\mathbf{a}, \mathbf{b}) \geq N(\mathbf{b}, \mathbf{a})$ satisfy*

$$\begin{cases} d(\mathbf{a}, \mathbf{b}) \geq d_3 + 1 & \text{if } N(\mathbf{b}, \mathbf{a}) = 0 \\ d(\mathbf{a}, \mathbf{b}) \geq d_1 + 1 & \text{if } N(\mathbf{b}, \mathbf{a}) \geq 1 \end{cases}$$

Proof. Substitute $t_i = 0$ $(i = 1, 2, 3)$ into Theorem 1.1. □

From Corollary 1.4 (and also from Theorem 1.2) it follows that any d_1-SyED d_2-UED d_3-AsED code is also a d_1-SyED d_3-UED d_3-AsED code. Hence we only have to consider d_1-SyED d_2-UED codes.

Other interesting results are stated in the next corollary and in [49].

Corollary 1.5 *A code $\mathcal{C}$ is t_1-SyEC t_2-UEC d_2-UED (with $0 \leq t_1 \leq t_2 \leq d_2$) if and only if all $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $N(\mathbf{a}, \mathbf{b}) \geq N(\mathbf{b}, \mathbf{a})$ satisfy*

$$\begin{cases} d(\mathbf{a}, \mathbf{b}) \geq t_2 + d_2 + 1 & \text{if } N(\mathbf{b}, \mathbf{a}) = 0 \\ d(\mathbf{a}, \mathbf{b}) \geq t_1 + d_2 + 1 \wedge N(\mathbf{a}, \mathbf{b}) \geq d_2 + 1 & \text{if } 1 \leq N(\mathbf{b}, \mathbf{a}) \leq t_2 \end{cases}$$

Proof. Substitute $t_3 = t_2$, $d_1 = t_1$, and $d_3 = d_2$ into Theorem 1.1. □

1.5 Single type error correction

Traditional coding theory is mainly focussed on codes correcting and/or detecting errors of the symmetric type. However, as mentioned before, in some modern communication systems the errors appear to be of other types. In order to be able to deal with such cases we have derived necessary and sufficient conditions for a code to be t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED. But it seems a very big step in the development of coding theory to study these codes in general. Therefore we concentrate in this thesis on codes correcting errors of one single type: t-SyEC codes, t-UEC codes, and t-AsEC codes. To this end we define three distances between two vectors $\mathbf{a}$ and $\mathbf{b}$ in $(GF(2))^n$:

$$\begin{cases} d_{Sy}(\mathbf{a}, \mathbf{b}) = N(\mathbf{a}, \mathbf{b}) + N(\mathbf{b}, \mathbf{a}) = d(\mathbf{a}, \mathbf{b}) \\ d_{As}(\mathbf{a}, \mathbf{b}) = 2 \max\{N(\mathbf{a}, \mathbf{b}), N(\mathbf{b}, \mathbf{a})\} \\ d_U(\mathbf{a}, \mathbf{b}) = \begin{cases} d_{Sy}(\mathbf{a}, \mathbf{b}) & \text{if } N(\mathbf{a}, \mathbf{b}) = 0 \vee N(\mathbf{b}, \mathbf{a}) = 0 \\ d_{As}(\mathbf{a}, \mathbf{b}) & \text{if } N(\mathbf{a}, \mathbf{b}) > 0 \wedge N(\mathbf{b}, \mathbf{a}) > 0 \end{cases} \end{cases} \tag{1.21}$$

It is obvious that

$$d_{Sy}(\mathbf{a},\mathbf{b}) \leq d_U(\mathbf{a},\mathbf{b}) \leq d_{As}(\mathbf{a},\mathbf{b}). \tag{1.22}$$

The distances $d_{Sy}(\mathbf{a},\mathbf{b})$ and $d_{As}(\mathbf{a},\mathbf{b})$ are related by

$$d_{As}(\mathbf{a},\mathbf{b}) = d_{Sy}(\mathbf{a},\mathbf{b}) + |w(\mathbf{a}) - w(\mathbf{b})|. \tag{1.23}$$

Hence

$$d_{Sy}(\mathbf{a},\mathbf{b}) = d_U(\mathbf{a},\mathbf{b}) = d_{As}(\mathbf{a},\mathbf{b}) \Leftrightarrow w(\mathbf{a}) = w(\mathbf{b}). \tag{1.24}$$

It can easily be checked that $d_{As}(\mathbf{a},\mathbf{b})$ is a metric (see (1.5)) on $(GF(2))^n$ just like $d_{Sy}(\mathbf{a},\mathbf{b})$. Note that $d_U(\mathbf{a},\mathbf{b})$ is *not* a metric on $(GF(2))^n$ (for $n \geq 3$), since it does not satisfy the triangle inequality, as can be seen from the next example ($n = 3$):

$$d_U(110,100) + d_U(100,001) = 1 + 2 = 3 < 4 = d_U(110,001).$$

For a code $\mathcal{C}$ we also define three distances:

$$\left\{ \begin{array}{rcl} d_{Sy} & = & \min\{d_{Sy}(\mathbf{a},\mathbf{b}) | \mathbf{a},\mathbf{b} \in \mathcal{C} \wedge \mathbf{a} \neq \mathbf{b}\} \\ d_U & = & \min\{d_U(\mathbf{a},\mathbf{b}) | \mathbf{a},\mathbf{b} \in \mathcal{C} \wedge \mathbf{a} \neq \mathbf{b}\} \\ d_{As} & = & \min\{d_{As}(\mathbf{a},\mathbf{b}) | \mathbf{a},\mathbf{b} \in \mathcal{C} \wedge \mathbf{a} \neq \mathbf{b}\} \end{array} \right. \tag{1.25}$$

The error correcting capability of a code $\mathcal{C}$ can be expressed in these three distances, as can be seen from the next corollary. The corollary simply recasts several well-known results (cf. [21,5,7]) in accordance with our notation.

Corollary 1.6 *Let $\mathcal{C}$ be a code of length n with distances d_{Sy}, d_U, and d_{As}.*

1. *$\mathcal{C}$ is t-SyEC if and only if $d_{Sy} \geq 2t + 1$;*
2. *$\mathcal{C}$ is t-UEC if and only if $d_U \geq 2t + 1$;*
3. *$\mathcal{C}$ is t-AsEC if and only if $d_{As} \geq 2t + 1$.*

Proof. These results follow from definitions (1.21) and (1.25) and by making the following substitutions into Corollary 1.3:

1. $t_1 = t_2 = t_3 = t$;
2. $t_1 = 0$ and $t_2 = t_3 = t$;

3. $t_1 = t_2 = 0$ and $t_3 = t$. □

In [7] it is mentioned that a code with $d_{As} \geq 2t + 1$ is able to correct t_0 or fewer 0-errors and t_1 or fewer 1-errors, where t_0 and t_1 are fixed and $t_0 + t_1 \leq t$. The correction of t asymmetric errors corresponds to the case that $t_0 = 0$ and $t_1 = t$.

As mentioned before one of the most basic problems in coding theory is to find the largest code of a given length with a certain error correcting capability. Therefore, we define for all $1 \leq t \leq n$:

$$A_{Sy}(n,t) : \text{the maximum number of codewords in a } t\text{-SyEC code of length } n; \quad (1.26)$$

$$A_U(n,t) : \text{the maximum number of codewords in a } t\text{-UEC code of length } n; \quad (1.27)$$

$$A_{As}(n,t) : \text{the maximum number of codewords in a } t\text{-AsEC code of length } n; \quad (1.28)$$

and

$$K_{Sy}(n,t) : \text{the largest dimension of a linear } t\text{-SyEC code of length } n; \quad (1.29)$$

$$K_U(n,t) : \text{the largest dimension of a linear } t\text{-UEC code of length } n; \quad (1.30)$$

$$K_{As}(n,t) : \text{the largest dimension of a linear } t\text{-AsEC code of length } n. \quad (1.31)$$

Of course,

$$A_{As}(n,t) \geq A_U(n,t) \geq A_{Sy}(n,t) = A(n,2t+1); \quad (1.32)$$

$$K_{As}(n,t) \geq K_U(n,t) \geq K_{Sy}(n,t); \quad (1.33)$$

$$K_{Sy}(n,t) \leq \lfloor \log_2(A_{Sy}(n,t)) \rfloor; \quad (1.34)$$

$$K_U(n,t) \leq \lfloor \log_2(A_U(n,t)) \rfloor; \quad (1.35)$$

$$K_{As}(n,t) \leq \lfloor \log_2(A_{As}(n,t)) \rfloor. \quad (1.36)$$

Here $\lfloor x \rfloor$ denotes the largest integer not exceeding the real number x. On the other hand, $\lceil x \rceil$ will denote the smallest integer not less than the real number x.

1.6 *Retrospect and prospect*

In this chapter we have described a simple model of a communication system in which data is sent over a noisy binary channel. In order to control the errors caused by the channel binary block codes are used. We have distinguished the following three error types: symmetric, unidirectional, and asymmetric. Necessary and sufficient conditions on a binary block code to correct/detect any combination of these three error types have been derived. Special attention has been paid to the correction of errors of one single type (SyEC, UEC, AsEC, respectively), since codes having such capabilities are the main subject of the rest of this thesis. In particular, we try to obtain upper and lower bounds on the maximum cardinality of t-AsEC codes and t-UEC codes of length n. These maximum cardinalities are denoted by $A_{As}(n,t)$ and $A_U(n,t)$, respectively.

In the next chapter we will derive some explicit upper bounds on $A_{As}(n,t)$ and $A_U(n,t)$. These will be obtained by applying the sphere-packing concept and by making use of known upper bounds on $A_{Sy}(n,t)$, the maximum cardinality of a t-SyEC code of length n. The best of these bounds will be shown to be tight for relatively small values of n.

Chapter 2

Explicit upper bounds

2.1 Sphere-packing bounds

Many upper bounds on $A_{Sy}(n,t)$ are known (see e.g. [1,21]). One of the oldest bounds is due to Hamming ([12]). Since in a t-SyEC code of length n and size M the spheres $S_{Sy}(\mathbf{c},t)$ must be disjoint for all codewords $\mathbf{c}$, he concluded that the product of M and the number of vectors in a sphere $S_{Sy}(\mathbf{c},t)$ is upper bounded by the number of vectors in $(GF(2))^n$:

$$2^n = |(GF(2))^n| \geq \sum_{\mathbf{c}\in\mathcal{C}} |S_{Sy}(\mathbf{c},t)| = M\left(\sum_{i=0}^{t}\left(\begin{array}{c} n \\ i \end{array}\right)\right). \qquad (2.1)$$

Varshamov (see [38]) applied this sphere-packing technique on t-AsEC codes $\mathcal{C}$ of length n and size M. He encountered the problem that the number of vectors in $S_{As}(\mathbf{c},t)$ depends on the weight of the codeword $\mathbf{c}$, which was not the case with $S_{Sy}(\mathbf{c},t)$. By observing that the inverted code $\bar{\mathcal{C}} = \{\mathbf{c}+\mathbf{1}|\mathbf{c}\in\mathcal{C}\}$ is also a t-AsEC code and by using the estimation

$$\begin{aligned} |S_{As}(\mathbf{c},t)| + |S_{As}(\bar{\mathbf{c}},t)| &= \sum_{i=0}^{t}\left(\left(\begin{array}{c} w(\mathbf{c}) \\ i \end{array}\right) + \left(\begin{array}{c} n-w(\mathbf{c}) \\ i \end{array}\right)\right) \\ &\geq \sum_{i=0}^{t}\left(\left(\begin{array}{c} \lfloor n/2\rfloor \\ i \end{array}\right) + \left(\begin{array}{c} \lfloor (n+1)/2\rfloor \\ i \end{array}\right)\right), \qquad (2.2) \end{aligned}$$

he derived

$$\begin{aligned} 2^{n+1} &= 2|(GF(2))^n| \geq \sum_{\mathbf{c}\in\mathcal{C}}(|S_{As}(\mathbf{c},t)| + |S_{As}(\bar{\mathbf{c}},t)|) \\ &\geq M\left(\sum_{i=0}^{t}\left(\left(\begin{array}{c} \lfloor n/2\rfloor \\ i \end{array}\right) + \left(\begin{array}{c} \lfloor (n+1)/2\rfloor \\ i \end{array}\right)\right)\right). \qquad (2.3) \end{aligned}$$

We meet the same problem if we apply the sphere-packing technique on t-UEC codes C of length n and size M. Using (2.2) we find

$$\begin{aligned} 2^n &= |(GF(2))^n| \geq \sum_{\mathbf{c}\in C} |S_U(\mathbf{c},t)| \\ &= \sum_{\mathbf{c}\in C}\left(1+\sum_{i=1}^{t}\left(\begin{array}{c} w(\mathbf{c}) \\ i \end{array}\right)+\left(\begin{array}{c} n-w(\mathbf{c}) \\ i \end{array}\right)\right) \\ &\geq M\left(1+\sum_{i=1}^{t}\left(\left(\begin{array}{c} \lfloor n/2\rfloor \\ i \end{array}\right)+\left(\begin{array}{c} \lfloor (n+1)/2\rfloor \\ i \end{array}\right)\right)\right). \end{aligned} \tag{2.4}$$

Hence we have now established (from (2.1), (2.4), and (2.3)) the following theorem.

Theorem 2.1 (Sphere-packing bounds) *For all $n \geq t \geq 1$ we have*

1. $$\begin{aligned} A_{Sy}(n,t) &\leq \frac{2^n}{\sum_{i=0}^{t}\left(\begin{array}{c} n \\ i \end{array}\right)} \\ &= \frac{2^n t!}{n^t}(1+o(n)); \end{aligned}$$

2. $$\begin{aligned} A_U(n,t) &\leq \frac{2^n}{1+\sum_{i=1}^{t}\left(\left(\begin{array}{c} \lfloor n/2\rfloor \\ i \end{array}\right)+\left(\begin{array}{c} \lfloor (n+1)/2\rfloor \\ i \end{array}\right)\right)} \\ &= 2^{t-1}\frac{2^n t!}{n^t}(1+o(n)); \end{aligned}$$

3. $$\begin{aligned} A_{As}(n,t) &\leq \frac{2^{n+1}}{\sum_{i=0}^{t}\left(\left(\begin{array}{c} \lfloor n/2\rfloor \\ i \end{array}\right)+\left(\begin{array}{c} \lfloor (n+1)/2\rfloor \\ i \end{array}\right)\right)} \\ &= 2^t\frac{2^n t!}{n^t}(1+o(n)). \end{aligned}$$

□

2.2 Combining bounds

Since bounds on $A_{Sy}(n,t)$ are well studied, it seems a good idea to try to use these bounds for obtaining bounds on $A_U(n,t)$ and $A_{As}(n,t)$. For the

asymmetric case this was established by Borden (see [18]), whose results are stated in the next two theorems.

Theorem 2.2 (Borden) *For $n \geq t \geq 1$ we have*

$$A_{As}(n,t) \leq (t+1)A_{Sy}(n,t).$$

□

Theorem 2.3 (Borden) *For $n \geq t \geq 1$ we have*

$$A_{As}(n,t) \leq A_{Sy}(n+t,t).$$

□

Using arguments similar to Borden's we can sharpen these bounds for the unidirectional case. The bound corresponding to the first Borden bound (Theorem 2.2) is given in Theorem 2.4, and the bound corresponding to the second Borden bound (Theorem 2.3) is given in Theorem 2.6. Lemma 2.5 is used to prove Theorem 2.6.

Theorem 2.4 *For $n \geq t \geq 1$ we have*

$$A_U(n,t) \leq tA_{Sy}(n,t).$$

Proof. Let $\mathcal{C}$ be a code of length n with $d_U \geq 2t+1$ and $|\mathcal{C}| = A_U(n,t)$. Define

$$\mathcal{C}_r = \{\mathbf{x} \in \mathcal{C} | w(\mathbf{x}) \equiv 2r \pmod{2t} \vee w(\mathbf{x}) \equiv 2r+1 \pmod{2t}\}$$

for $r = 0, 1, \ldots, t-1$. We shall prove that each $\mathcal{C}_r$ is a code with $d_{Sy} \geq 2t+1$. Let $\mathbf{a}, \mathbf{b} \in \mathcal{C}_r$ with $\mathbf{a} \neq \mathbf{b}$ and $w(\mathbf{b}) \geq w(\mathbf{a})$.

1. The case $N(\mathbf{b},\mathbf{a}) > 0$.
   Note that either $w(\mathbf{b}) - w(\mathbf{a}) \leq 1$ or $w(\mathbf{b}) - w(\mathbf{a}) \geq 2t-1$.

   (a) The case $w(\mathbf{b}) - w(\mathbf{a}) \leq 1$.
   Since $d_{As}(\mathbf{b},\mathbf{a}) \geq d_U(\mathbf{b},\mathbf{a}) \geq 2t+1$ and $d_{As}(\mathbf{b},\mathbf{a})$ is even, it follows that $d_{As}(\mathbf{b},\mathbf{a}) \geq 2t+2$. Hence it follows from (1.23) that

$$d_{Sy}(\mathbf{b},\mathbf{a}) = d_{As}(\mathbf{b},\mathbf{a}) - (w(\mathbf{b}) - w(\mathbf{a})) \geq 2t+2-1 = 2t+1.$$

(b) The case $w(\mathbf{b}) - w(\mathbf{a}) \geq 2t - 1$.
Since $N(\mathbf{a},\mathbf{b}) - N(\mathbf{b},\mathbf{a}) \geq 2t - 1$ and $N(\mathbf{b},\mathbf{a}) \geq 1$, it follows from (1.21) that

$$d_{Sy}(\mathbf{b},\mathbf{a}) = N(\mathbf{a},\mathbf{b}) + N(\mathbf{b},\mathbf{a}) \geq 2t - 1 + 2N(\mathbf{b},\mathbf{a}) \geq 2t + 1.$$

2. The case N(**b**,**a**)=0.
It follows from (1.21) that

$$d_{Sy}(\mathbf{b},\mathbf{a}) = d_U(\mathbf{b},\mathbf{a}) \geq 2t + 1.$$

In conclusion, each $\mathcal{C}_r$ is a t-SyEC code of length n , and so

$$|\mathcal{C}_r| \leq A_{Sy}(n,t)$$

for all $r = 0, 1, \ldots, t-1$. Hence

$$A_U(n,t) = |\mathcal{C}| = \sum_{r=0}^{t-1} |\mathcal{C}_r| \leq tA_{Sy}(n,t).$$

□

Note that in the case $t = 1$, Theorem 2.4 gives $A_U(n,1) \leq A_{Sy}(n,1)$, while it follows from (1.32) that $A_{Sy}(n,1) \leq A_U(n,1)$. Hence

$$A_{Sy}(n,1) = A_U(n,1) \tag{2.5}$$

for $n \geq 1$. This is not surprising at all, since there is no difference between a single symmetric error and a single unidirectional error.

Lemma 2.5 (Borden) *Let $i \in \{1,2,3,\ldots\}$. We define the function $\mathbf{u}_i$ which maps an integer z to a vector in $(GF(2))^i$ by*

$$\mathbf{u}_i(z) = (s(z), s(z+1), \ldots, s(z+i-1)),$$

where

$$s(z) = \begin{cases} 0 & \text{if } z \equiv 0,1,\ldots,i \pmod{2i+2} \\ 1 & \text{if } z \equiv i+1, i+2, \ldots, 2i+1 \pmod{2i+2} \end{cases}.$$

Then we have for all integers x and y that

$$d_{Sy}(\mathbf{u}_i(x), \mathbf{u}_i(y)) + d_{Sy}(\mathbf{u}_i(x+i+1), \mathbf{u}_i(y)) = i,$$

and for all integers x and y with $x \leq y \leq x+i+1$ that

$$y - x - 1 \leq d_{Sy}(\mathbf{u}_i(x), \mathbf{u}_i(y)) \leq y - x.$$

□

Theorem 2.6 *For $n \geq t \geq 1$ we have*

$$A_U(n,t) \leq A_{Sy}(n+t-1,t).$$

Proof. From Theorem 2.4 it follows that the theorem holds for $t = 1$. If $t \geq 2$, let $\mathcal{C}$ be a code of length n with $d_U \geq 2t+1$ and size $A_U(n,t)$. Let the function $\mathbf{u}_{t-1}$ be defined as in Lemma 2.5. We now construct a code $\mathcal{D}$ of length $n+t-1$ by lengthening each codeword $\mathbf{c} \in \mathcal{C}$ with $\mathbf{u}_{t-1}(w(\mathbf{c}))$:

$$\mathcal{D} = \{(\mathbf{c}, \mathbf{u}_{t-1}(w(\mathbf{c})))|\mathbf{c} \in \mathcal{C}\}.$$

We shall prove that $\mathcal{D}$ is a t-SyEC code. Let $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $w(\mathbf{a}) \geq w(\mathbf{b})$.

1. The case $N(\mathbf{a},\mathbf{b}) > 0$.
   In this case

   $$N(\mathbf{b},\mathbf{a}) = d_{As}(\mathbf{a},\mathbf{b})/2 = d_U(\mathbf{a},\mathbf{b})/2 \geq t+1$$

   and $N(\mathbf{a},\mathbf{b}) \geq 1$.

   (a) The case $N(\mathbf{a},\mathbf{b}) + N(\mathbf{b},\mathbf{a}) \leq 2t$ and $0 \leq N(\mathbf{b},\mathbf{a}) - N(\mathbf{a},\mathbf{b}) \leq t$. Then $w(\mathbf{b}) \leq w(\mathbf{a}) \leq w(\mathbf{b}) + t$. Hence it follows from Lemma 2.5 that

   $$\begin{aligned} & d_{Sy}(\mathbf{u}_{t-1}(w(\mathbf{a})), \mathbf{u}_{t-1}(w(\mathbf{b}))) \\ &\geq w(\mathbf{a}) - w(\mathbf{b}) - 1 \\ &= N(\mathbf{b},\mathbf{a}) - N(\mathbf{a},\mathbf{b}) - 1 \\ &= 2N(\mathbf{b},\mathbf{a}) - 1 - N(\mathbf{a},\mathbf{b}) - N(\mathbf{b},\mathbf{a}) \\ &\geq 2t + 2 - 1 - N(\mathbf{a},\mathbf{b}) - N(\mathbf{b},\mathbf{a}) \\ &= 2t + 1 - N(\mathbf{a},\mathbf{b}) - N(\mathbf{b},\mathbf{a}). \end{aligned}$$

   (b) The case $N(\mathbf{a},\mathbf{b}) + N(\mathbf{b},\mathbf{a}) \leq 2t$ and $t+1 \leq N(\mathbf{b},\mathbf{a}) - N(\mathbf{a},\mathbf{b}) \leq 2t-2$.
   Then $w(\mathbf{b}) + t + 1 \leq w(\mathbf{a}) \leq w(\mathbf{b}) + 2t - 2$. Hence $w(\mathbf{b}) \leq w(\mathbf{a}) - t - 1 \leq w(\mathbf{b}) + t - 3$, and so $w(\mathbf{b}) \leq w(\mathbf{a}) - t \leq w(\mathbf{b}) + t$. So it follows from Lemma 2.5 that

   $$\begin{aligned} & d_{Sy}(\mathbf{u}_{t-1}(w(\mathbf{a})), \mathbf{u}_{t-1}(w(\mathbf{b}))) \\ &= t - 1 - d_{Sy}(\mathbf{u}_{t-1}(w(\mathbf{a}) - t), \mathbf{u}_{t-1}(w(\mathbf{b}))) \end{aligned}$$

$$
\begin{aligned}
&\geq t-1-(w(\mathbf{a})-t-w(\mathbf{b}))\\
&= 2t-1-(w(\mathbf{a})-w(\mathbf{b}))\\
&= 2t-1-(N(\mathbf{b},\mathbf{a})-N(\mathbf{a},\mathbf{b}))\\
&= 2t-1-N(\mathbf{b},\mathbf{a})+N(\mathbf{a},\mathbf{b})\\
&= 2t-1-N(\mathbf{b},\mathbf{a})-N(\mathbf{a},\mathbf{b})+2N(\mathbf{a},\mathbf{b})\\
&\geq 2t+1-N(\mathbf{b},\mathbf{a})-N(\mathbf{a},\mathbf{b}).
\end{aligned}
$$

(c) The case $N(\mathbf{a},\mathbf{b})+N(\mathbf{b},\mathbf{a}) \geq 2t+1$. Then

$$
\begin{aligned}
&d_{Sy}(\mathbf{u}_{t-1}(w(\mathbf{a})),\mathbf{u}_{t-1}(w(\mathbf{b})))\\
&\quad \geq 2t+1-N(\mathbf{a},\mathbf{b})-N(\mathbf{b},\mathbf{a}).
\end{aligned}
$$

Hence

$$
\begin{aligned}
&d_{Sy}((\mathbf{a},\mathbf{u}_{t-1}(w(\mathbf{a}))),(\mathbf{b},\mathbf{u}_{t-1}(w(\mathbf{b}))))\\
&= d_{Sy}(\mathbf{a},\mathbf{b})+d_{Sy}(\mathbf{u}_{t-1}(w(\mathbf{a})),\mathbf{u}_{t-1}(w(\mathbf{b})))\\
&= N(\mathbf{a},\mathbf{b})+N(\mathbf{b},\mathbf{a})+d_{Sy}(\mathbf{u}_{t-1}(w(\mathbf{a})),\mathbf{u}_{t-1}(w(\mathbf{b})))\\
&\geq N(\mathbf{a},\mathbf{b})+N(\mathbf{b},\mathbf{a})+2t+1-N(\mathbf{a},\mathbf{b})-N(\mathbf{b},\mathbf{a})\\
&= 2t+1.
\end{aligned}
$$

2. The case $N(\mathbf{a},\mathbf{b})=0$.
In this case

$$
\begin{aligned}
&d_{Sy}((\mathbf{a},\mathbf{u}_{t-1}(w(\mathbf{a}))),(\mathbf{b},\mathbf{u}_{t-1}(w(\mathbf{b}))))\\
&\quad \geq d_{Sy}(\mathbf{a},\mathbf{b})=d_U(\mathbf{a},\mathbf{b})\geq 2t+1.
\end{aligned}
$$

Hence $\mathcal{D}$ is a t-SyEC code of length $n+t-1$. In conclusion,

$$
A_U(n,t)=|\mathcal{C}|=|\mathcal{D}|\leq A_{Sy}(n+t-1,t).
$$

□

Now by combining the results presented in Theorems 2.2, 2.3, 2.4, 2.6 and known upper bounds on $A_{Sy}(n,t)=A(n,2t+1)$ (see e.g. [21,1]) we can obtain upper bounds on $A_U(n,t)$ and $A_{As}(n,t)$. For example, combining Theorems 2.2, 2.4, and the Hamming bound from Theorem 2.1 we obtain

$$
A_U(n,t)\leq tA_{Sy}(n,t)\leq \frac{t2^n}{\sum_{i=0}^{t}\binom{n}{i}}=t\frac{2^n t!}{n^t}(1+o(n)) \qquad (2.6)
$$

and

$$A_{As}(n,t) \leq (t+1)A_{Sy}(n,t) \leq \frac{(t+1)2^n}{\sum_{i=0}^{t}\binom{n}{i}} = (t+1)\frac{2^n t!}{n^t}(1+o(n)). \quad (2.7)$$

Note that these bounds are better than the corresponding sphere-packing bounds from Theorem 2.1 for large values of n.

We once again state the bounds from Theorems 2.2, 2.3, 2.4, and 2.6:

$$A_{As}(n,t) \leq (t+1)A_{Sy}(n,t); \quad (2.8)$$

$$A_{As}(n,t) \leq A_{Sy}(n+t,t); \quad (2.9)$$

$$A_U(n,t) \leq tA_{Sy}(n,t); \quad (2.10)$$

$$A_U(n,t) \leq A_{Sy}(n+t-1,t). \quad (2.11)$$

When comparing the two bounds on $A_U(n,t)$ using the best known results on $A_{Sy}(n,t)$ in either case, we can conclude that the bounds equal for $t=1$, and that bound (2.11) is preferable for $t=2$ since $A_{Sy}(n+1,2) \leq 2A_{Sy}(n,2)$. However, for $t \geq 3$ bound (2.10) mostly appears to be the better one. Analogously comparing the bounds on $A_{As}(n,t)$, we can conclude that bound (2.9) is preferable for $t=1$ since $A_{Sy}(n+1,2) \leq 2A_{Sy}(n,2)$, but that for $t \geq 2$ bound (2.8) mostly appears to be the better one.

After having found some relations between $A_{As}(n,t)$ and $A_{Sy}(n,t)$ and between $A_U(n,t)$ and $A_{Sy}(n,t)$, we can also try to find relations between $A_{As}(n,t)$ and $A_U(n,t)$. Such a result is presented in Theorem 2.8.

Lemma 2.7 *Let $\mathcal{C}$ be a t-AsEC code of length n $(1 \leq t \leq n)$. Then each code $\mathcal{D}_i$ $(i = 0,1,\ldots,3t)$ of length n defined by*

$$\mathcal{D}_i = \{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) \equiv i, i+1, \ldots, i+t \pmod{3t+1}\}$$

is t-UEC.

Proof. Let $\mathbf{a},\mathbf{b} \in \mathcal{D}_i$ with $\mathbf{a} \neq \mathbf{b}$ and $w(\mathbf{a}) \geq w(\mathbf{b})$. Then either $w(\mathbf{a}) - w(\mathbf{b}) \leq t$ or $w(\mathbf{a}) - w(\mathbf{b}) \geq 2t+1$.

1. The case $w(\mathbf{a}) - w(\mathbf{b}) \leq t$. Since $\mathbf{a},\mathbf{b} \in \mathcal{C}$, it follows that $d_{As}(\mathbf{a},\mathbf{b}) \geq 2t+1$. Suppose $N(\mathbf{a},\mathbf{b}) = 0$ or $N(\mathbf{b},\mathbf{a}) = 0$, then

$$d_{As}(\mathbf{a},\mathbf{b}) = 2|w(\mathbf{a}) - w(\mathbf{b})| \leq 2t.$$

This contradicts $d_{As}(\mathbf{a},\mathbf{b}) \geq 2t+1$. Hence $N(\mathbf{a},\mathbf{b}) > 0$ and $N(\mathbf{b},\mathbf{a}) > 0$, and so

$$d_U(\mathbf{a},\mathbf{b}) = d_{As}(\mathbf{a},\mathbf{b}) \geq 2t+1.$$

2. The case $w(\mathbf{a}) - w(\mathbf{b}) \geq 2t+1$. In this case we have

$$d_U(\mathbf{a},\mathbf{b}) \geq d_{Sy}(\mathbf{a},\mathbf{b}) \geq w(\mathbf{a}) - w(\mathbf{b}) \geq 2t+1.$$

Hence $\mathcal{D}_i$ is t-UEC. □

Theorem 2.8 *For $n \geq t \geq 1$ we have*

$$A_{As}(n,t) \leq \frac{3t+1}{t+1} A_U(n,t).$$

Proof. Let $\mathcal{C}$ be a t-AsEC code of length n and size $A_{As}(n,t)$, and let the t-UEC codes $\mathcal{D}_i$ be defined as in Lemma 2.7 ($i = 0,1,\ldots,3t$). Since each codeword of $\mathcal{C}$ is also a codeword of exactly $t+1$ codes $\mathcal{D}_i$ ($|\{i|\mathbf{c} \in \mathcal{D}_i\}| = t+1$ for all $\mathbf{c} \in \mathcal{C}$), it follows that

$$(t+1)A_{As}(n,t) = (t+1)|\mathcal{C}| = \sum_{i=0}^{3t} |\mathcal{D}_i| \leq (3t+1)A_U(n,t).$$

□

Hence it can be concluded from (1.32) and Theorem 2.8 that

$$A_U(n,t) \leq A_{As}(n,t) \leq (3 - \frac{2}{t+1}) A_U(n,t) < 3A_U(n,t) \qquad (2.12)$$

for $n \geq t \geq 1$. For $t = 1$ Theorem 2.8 coincides with Theorem 2.2:

$$A_{As}(n,1) \leq 2A_U(n,1) = 2A_{Sy}(n,1) \qquad (2.13)$$

for $n \geq 1$.

2.3 Tight bounds for small lengths

In this section we give the exact values of $A_{Sy}(n,t)$, $A_U(n,t)$, and $A_{As}(n,t)$ for relatively small values of n. The results on $A_{Sy}(n,t)$ given in Theorem 2.9 are well known (cf. [21]) and are treated here for the sake of completeness.

Theorem 2.9 *We have*

1. $A_{Sy}(n,t) = 1$ *for* $t \leq n \leq 2t$ *and* $t \geq 1$;

2. $A_{Sy}(n,t) = 2$ *for* $2t+1 \leq n \leq 3t+1$ *and* $t \geq 1$;

3. $A_{Sy}(3t+2,t) = 4$ *for* $t \geq 1$.

Proof.

1. The code of length t containing only the all-zero word is t-SyEC. Further, a t-SyEC code of size 2 must have length at least $2t+1$ because of $d_{Sy} \geq 2t+1$. Hence

$$1 \leq A_{Sy}(t,t) \leq A_{Sy}(t+1,t) \leq \ldots \leq A_{Sy}(2t,t) \leq 1.$$

2. The code of length $2t+1$ containing the all-zero vector $\mathbf{0}$ and the all-one vector $\mathbf{1}$ is t-SyEC, since $d_{Sy} = 2t+1$. From the Plotkin bound (see e.g. [21]) it follows that the size of a t-SyEC code of length $3t+1$ does not exceed 2. Hence

$$2 \leq A_{Sy}(2t+1,t) \leq A_{Sy}(2t+2,t) \leq \ldots \leq A_{Sy}(3t+1,t) \leq 2.$$

3. The code of length $3t+2$ containing the four codewords

$$0^{3t+2}, 1^{2t+1}0^{t+1}, 1^t 0^{t+1} 1^{t+1}, 0^t 1^{2t+2}$$

is t-SyEC, since $d_{Sy} = 2t+1$. Hence

$$4 \leq A_{Sy}(3t+2,t) \leq 2A_{Sy}(3t+1,t) = 4.$$

□

The results on $A_U(n,t)$ and $A_{As}(n,t)$ as stated in Theorems 2.10 and 2.11 are derived by constructing codes of length n correcting up to t errors, the sizes of which reach the best corresponding upper bounds from the previous section.

Theorem 2.10 *We have*

1. $A_U(n,t) = 1$ *for* $t \leq n \leq t+1$ *and* $t \geq 1$;

2. $A_U(n,t) = 2$ *for* $t+2 \leq n \leq 2t+2$ *and* $t \geq 1$;

3. $A_U(2t+3,t) = 4$ *for* $t \geq 1$.

Proof.

1. From Theorems 2.9 and 2.6 it follows that

$$1 = A_{Sy}(t,t) \leq A_U(t,t) \leq A_U(t+1,t) \leq A_{Sy}(2t,t) = 1.$$

2. The code of length t+2 containing the two codewords 10^{t+1} and 01^{t+1} is t-UEC, since $d_U = 2t+2$. With Theorems 2.6 and 2.9 it now follows that

$$\begin{aligned} 2 &\leq A_U(t+2,t) \leq A_U(t+3,t) \leq \ldots \leq A_U(2t+2,t) \\ &\leq A_{Sy}(3t+1,t) = 2. \end{aligned}$$

3. The code of length $2t+3$ containing the four codewords

$$1100^{2t}, 0011^t0^t, 0110^t1^t, 1001^{2t}$$

is t-UEC, since $d_U = 2t+2$. Hence

$$4 \leq A_U(2t+3,t) \leq 2A_U(2t+2,t) = 4.$$

□

Theorem 2.11 *We have*

1. $A_{As}(t,t) = 1$ *for* $t \geq 1$;
2. $A_{As}(n,t) = 2$ *for* $t+1 \leq n \leq 2t+1$ *and* $t \geq 1$;
3. $A_{As}(2t+2,t) = 4$ *for* $t \geq 1$.

Proof.

1. From Theorems 2.9 and 2.3 it follows that

$$1 = A_{Sy}(t,t) \leq A_{As}(t,t) \leq A_{Sy}(2t,t) = 1.$$

2. The code of length $t+1$ containing the two codewords **0** and **1** is t-AsEC, since $d_{As} = 2t+2$. With Theorems 2.3 and 2.9 it now follows that

$$\begin{aligned} 2 &\leq A_{As}(t+1,t) \leq A_{As}(t+2,t) \leq \ldots \leq A_{As}(2t+1,t) \\ &\leq A_{Sy}(3t+1,t) = 2. \end{aligned}$$

n	$A_{Sy}(n,t)$	$A_U(n,t)$	$A_{As}(n,t)$
t	1	1	1
$t+1$	1	1	2
$t+2$	1	2	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2t$	1	2	2
$2t+1$	2	2	2
$2t+2$	2	2	4
$2t+3$	2	4	
$\vdots$	$\vdots$		
$3t+1$	2		
$3t+2$	4		

Table 2.1: $A_{Sy}(n,t), A_U(n,t), A_{As}(n,t)$ for $t \geq 2$ and relatively small n.

3. The code of length $2t+2$ containing the four codewords

$$0^{2t+2}, 0^{t+1}1^{t+1}, 1^{t+1}0^{t+1}, 1^{2t+2}$$

is t-AsEC, since $d_{As} = 2t+2$. Hence

$$4 \leq A_{As}(2t+2,t) \leq 2A_{As}(2t+1,t) = 4.$$

□

The results of this section are summarized in Tables 2.1 and B.1.

2.4 *Retrospect and prospect*

In this chapter we have met some explicit upper bounds on $A_U(n,t)$ and $A_{As}(n,t)$. The sphere-packing concept suffers from the fact that the number of vectors that can be received when sending a codeword $\mathbf{c}$ in which t or less unidirectional or asymmetric errors occur depends on the weight of $\mathbf{c}$. By making use of known upper bounds on $A_{Sy}(n,t)$ we obtain better results. To obtain explicit upper bounds in the unidirectional case we recommend the use of Theorem 2.6 if $t \leq 2$ and the use of Theorem 2.4 if

$t \geq 3$. To obtain explicit upper bounds in the asymmetric case we recommend the use of Theorem 2.3 if $t = 1$ and the use of Theorem 2.2 if $t \geq 2$. These bounds have been shown to be tight for relatively small values of n.

To improve the upper bounds found in this chapter we can apply the sphere-packing concept on the vectors of a certain weight instead of all vectors in $(GF(2))^n$. Then by collecting all the results on the weights $0, 1, \ldots n$ and by adding some other results on the maximum number of codewords of certain weights, we can formulate an integer programming problem, the solution of which gives the desired upper bound. These bounds, that we will treat in the next chapter, are usually better than the explicit bounds, but much harder to calculate.

Chapter 3

Integer programming bounds

3.1 Constraints on the weight distribution of AsEC and UEC codes

In order to obtain upper bounds on the maximum number of codewords in a SyEC code, a well-known approach is first to derive some constraints on the weight distribution of the code and then to collect these results to bound the total number of codewords. Goldbaum ([10]) used the same approach to derive upper bounds on the maximum number of codewords in an AsEC code. His constraints on the weight distribution A_i of a t-AsEC code C of length n are based on the sphere-packing concept from Section 2.1. Instead of applying this concept to the whole space $(GF(2))^n$ in one go, he applied the concept only to the vectors of length n having a certain weight i. To describe this idea, we first define

$$\mathcal{V}_i = \{\mathbf{x} \in (GF(2))^n | w(\mathbf{x}) = i\} \tag{3.1}$$

for $i = 0, 1, \ldots, n$, and

$$\mathcal{H}_i(\mathbf{v}) = \begin{cases} \{\mathbf{x} \in \mathcal{V}_i | \mathbf{x} \geq \mathbf{v}\} & \text{if } w(\mathbf{v}) < i \\ \{\mathbf{x} \in \mathcal{V}_i | \mathbf{v} \geq \mathbf{x}\} & \text{if } w(\mathbf{v}) \geq i \end{cases} \tag{3.2}$$

for $i = 0, 1, \ldots, n$ and $\mathbf{v} \in (GF(2))^n$. Hence

$$|\mathcal{H}_i(\mathbf{v})| = \begin{cases} \begin{pmatrix} n - w(\mathbf{v}) \\ i - w(\mathbf{v}) \end{pmatrix} & \text{if } w(\mathbf{v}) < i \\ \\ \begin{pmatrix} w(\mathbf{v}) \\ i \end{pmatrix} & \text{if } w(\mathbf{v}) \geq i \end{cases} \tag{3.3}$$

for $i = 0, 1, \ldots, n$ and $\mathbf{v} \in (GF(2))^n$. Now since the sets $S_{A_s}(\mathbf{c}, t)$ have to be disjoint for all codewords $\mathbf{c}$ in a t-AsEC code $\mathcal{C}$, Goldbaum concluded that

$$\begin{aligned}
\binom{n}{i} &= |\mathcal{V}_i| \\
&\geq \left| \left(\bigcup_{\mathbf{c} \in \mathcal{C}} S_{A_s}(\mathbf{c}, t) \right) \bigcap \mathcal{V}_i \right| \\
&= \left| \bigcup_{\substack{\mathbf{c} \in \mathcal{C} \\ i \leq w(\mathbf{c}) \leq i+t}} \mathcal{H}_i(\mathbf{c}) \right| \\
&= \sum_{\substack{\mathbf{c} \in \mathcal{C} \\ i \leq w(\mathbf{c}) \leq i+t}} |\mathcal{H}_i(\mathbf{c})| \\
&= \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j} \qquad (3.4)
\end{aligned}$$

for $i = 0, 1, \ldots, n$. Delsarte and Piret ([8]) gave a generalization of this result, as stated in the following theorem.

Theorem 3.1 (Delsarte and Piret) *In a t-AsEC code $\mathcal{C}$ of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\sum_{j=1}^{t-k} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{k} \binom{i+j}{j} A_{i+j} \leq \binom{n}{i}$$

for $i = 0, 1, \ldots, n$ and $k = 0, 1, \ldots, t$. □

Goldbaum's result (3.4) corresponds to the case $k = t$ in Theorem 3.1.

Delsarte and Piret ([8]) also derived another class of constraints on the weight distribution $A_0, A_1, \ldots, A_n$ of a t-AsEC code $\mathcal{C}$. For any integers s and i with $0 \leq s \leq i \leq n$, juxtaposition of a vector of length s and weight j to each codeword of weight $i - j$ $(0 \leq j \leq s)$ gives a code $\mathcal{C}'$ of length $n + s$, constant weight i, and Hamming distance at least $2t + 2$. Hence

$$\sum_{j=i-s}^{i} A_j = |\mathcal{C}'| \leq A(n+s, 2t+2, i) \qquad (3.5)$$

for $i = 0, 1, \ldots, n$ and $s = 0, 1, \ldots, i$. This result was extended by Kløve ([17]), as stated in the next theorem.

Theorem 3.2 (Kløve) *In a t-AsEC code C of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\sum_{j=i-s}^{i} A^l(s, 2t+2, i-j) A_j \leq A^u(n+s, 2t+2, i)$$

for $i = 0, 1, \ldots, n$ and $s = 0, 1, \ldots, i$. □

Delsarte and Piret's result (3.5) corresponds to the case that all lower bounds $A^l(s, 2t+2, i-j)$ are set equal to 1 in Theorem 3.2.

Kløve ([17]) now formulated an integer programming problem in which the sum $A_0 + A_1 + \cdots + A_n$ is maximized over constraints as stated in Theorems 3.1 and 3.2. The solution of this problem gives an upper bound on $A_{As}(n, t)$. By considering only a subset of the constraints Kløve derived an upper bound which is usually somewhat weaker, but much simpler to compute.

When applying the integer programming approach to derive upper bounds on $A_U(n, t)$, we can use the same constraints as in the asymmetric case, since any t-UEC code is also t-AsEC. However, it can be expected that some constraints can be strengthened. For the sphere-packing concept this turns out to be as stated in the next theorem.

Theorem 3.3 *In a t-UEC code C of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\sum_{j=1}^{t} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j} \leq \binom{n}{i}$$

for $i = 0, 1, \ldots, n$.

Proof. Proceeding as in (3.4) we find

$$\binom{n}{i} = |\mathcal{V}_i|$$

$$
\begin{aligned}
&\geq \left| \left(\bigcup_{\mathbf{c}\in\mathcal{C}} S_U(\mathbf{c},t) \right) \bigcap \mathcal{V}_i \right| \\
&= \left| \bigcup_{\substack{\mathbf{c}\in\mathcal{C} \\ i-t\leq w(\mathbf{c})\leq i+t}} \mathcal{H}_i(\mathbf{c}) \right| \\
&= \sum_{\substack{\mathbf{c}\in\mathcal{C} \\ i-t\leq w(\mathbf{c})\leq i+t}} |\mathcal{H}_i(\mathbf{c})| \\
&= \sum_{j=1}^{t} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j}.
\end{aligned}
$$

□

In Section 3.2 we will sharpen the constraints from Theorems 3.1 and 3.3. Integer programming problems which include these new constraints and lead to upper bounds on $A_{As(n,t)}$ and $A_U(n,t)$ are then stated in Section 3.3.

3.2 Sharpened constraints

3.2.1 The asymmetric case

The fundamental idea in the constraints as stated in Theorem 3.1 is that for a t-AsEC code the sets $\mathcal{H}_i(\mathbf{c})$ are disjoint for all codewords $\mathbf{c}$ of weight between $i+k-t$ and $i+k$. When considering also the codewords of weight $i+k+1$ it may occur that some of these sets overlap. Nevertheless, estimates of these overlaps can lead to sharper constraints on the weight distribution. The results are given in Theorem 3.4 for $0 \leq k \leq t-1$ and in Theorem 3.5 for $k = t$.

Theorem 3.4 *In a t-AsEC code C of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$
\begin{aligned}
&\sum_{j=1}^{t-k} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{k} \binom{i+j}{j} A_{i+j} \\
&\quad + \quad f_1(i,k,t) A_{i+k+1} \leq \binom{n}{i}
\end{aligned}
$$

for $i = 0, 1, \ldots, n-k-1$ *and* $k = 0, 1, \ldots, t-1$, *where*

$$f_1(i,k,t) = \begin{pmatrix} i+k+1 \\ i \end{pmatrix} - \begin{pmatrix} t+1 \\ k+1 \end{pmatrix} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor .$$

Proof. Let

$$\begin{aligned} \mathcal{E} &= \{\mathbf{c} \in \mathcal{C} | i-t+k \le w(\mathbf{c}) \le i+k\}, \\ \mathcal{E}_1 &= \{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) = i-t+k\}, \\ \mathcal{E}_2 &= \{\mathbf{c} \in \mathcal{C} | i-t+k+1 \le w(\mathbf{c}) \le i+k\}, \\ \mathcal{F} &= \{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) = i+k+1\}. \end{aligned}$$

First, we study the sets $\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$ for all $\mathbf{a}, \mathbf{b} \in \mathcal{E} \cup \mathcal{F}$, $\mathbf{a} \neq \mathbf{b}$.

1. If $\mathbf{a}, \mathbf{b} \in \mathcal{E}$ or $\mathbf{a}, \mathbf{b} \in \mathcal{F}$ or $\mathbf{a} \in \mathcal{E}_2, \mathbf{b} \in \mathcal{F}$, then

   $$\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b}) = \emptyset.$$

   For if the contrary holds, i.e. there exists a $\mathbf{v} \in \mathcal{V}_i$ such that $\mathbf{v} \in \mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$, then $d_{As}(\mathbf{a}, \mathbf{b}) \le 2t$, which contradicts the code $\mathcal{C}$ being t-AsEC.

2. If $\mathbf{a} \in \mathcal{E}_1$, $\mathbf{b} \in \mathcal{F}$, $\mathbf{b} \not\geq \mathbf{a}$, then

   $$\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b}) = \emptyset.$$

   For if the contrary holds, i.e. there exists a $\mathbf{v} \in \mathcal{V}_i$ such that $\mathbf{v} \in \mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$, then $\mathbf{b} \geq \mathbf{v} \geq \mathbf{a}$, which contradicts $\mathbf{b} \not\geq \mathbf{a}$.

3. If $\mathbf{a} \in \mathcal{E}_1$, $\mathbf{b} \in \mathcal{F}$, $\mathbf{b} \geq \mathbf{a}$, then $N(\mathbf{b}, \mathbf{a}) = 0$ and $N(\mathbf{a}, \mathbf{b}) = t+1$, and so

   $$|\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})| = |\mathbf{v} \in \mathcal{V}_i | \mathbf{b} \geq \mathbf{v} \geq \mathbf{a}\} | = \begin{pmatrix} t+1 \\ k+1 \end{pmatrix}.$$

When estimating the number of codewords in $\mathcal{E}_1$ that are covered by a particular codeword $\mathbf{f}$ in $\mathcal{F}$, note that these codewords form a constant weight code of length $i+k+1$, weight $i-t+k$ and Hamming distance at least $2t+2$, after deleting those coordinates where $\mathbf{f}$ equals 0. Hence

$$\begin{aligned} &|\{\mathbf{e} \in \mathcal{E}_1 | \mathbf{f} \geq \mathbf{e}\}| \\ &\le A(i+k+1, 2t+2, i-t+k) \\ &= A(i+k+1, 2t+2, t+1) \\ &= \left\lfloor \frac{i+k+1}{t+1} \right\rfloor \end{aligned}$$

for all $\mathbf{f} \in \mathcal{F}$.

In conclusion,

$$\begin{aligned}
\binom{n}{i} &= |\mathcal{V}_i| \\
&\geq \left| \left(\bigcup_{\mathbf{e}\in\mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \bigcup \left(\bigcup_{\mathbf{f}\in\mathcal{F}} \mathcal{H}_i(\mathbf{f}) \right) \right| \\
&= \left| \left(\bigcup_{\mathbf{e}\in\mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \right| + \left| \left(\bigcup_{\mathbf{f}\in\mathcal{F}} \mathcal{H}_i(\mathbf{f}) \right) \right| - \left| \left(\bigcup_{\mathbf{e}\in\mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \bigcap \left(\bigcup_{\mathbf{f}\in\mathcal{F}} \mathcal{H}_i(\mathbf{f}) \right) \right| \\
&= \sum_{\mathbf{e}\in\mathcal{E}} |\mathcal{H}_i(\mathbf{e})| + \sum_{\mathbf{f}\in\mathcal{F}} |\mathcal{H}_i(\mathbf{f})| - \sum_{\mathbf{f}\in\mathcal{F}} \sum_{\substack{\mathbf{e}\in\mathcal{E}_1 \\ \mathbf{f}\geq\mathbf{e}}} |\mathcal{H}_i(\mathbf{e}) \cap \mathcal{H}_i(\mathbf{f})| \\
&\geq \sum_{j=1}^{t-k} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{k} \binom{i+j}{j} A_{i+j} \\
&\quad + \binom{i+k+1}{k+1} A_{i+k+1} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor A_{i+k+1}.
\end{aligned}$$

□

Theorem 3.5 *In a t-AsEC code $\mathcal{C}$ of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\sum_{j=0}^{t} \binom{i+j}{j} A_{i+j} + \frac{f_2^l(i,t,n)}{\left\lfloor \frac{n-i}{t+1} \right\rfloor} A_{i+t+1} \leq \binom{n}{i}$$

for $i = 0, 1, \ldots, n-t-1$, where

$$f_2^l(i,t,n) = \binom{i+t+1}{i} - \sum_{j=0}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2).$$

Proof. Let

$$\begin{aligned}
\mathcal{E} &= \{\mathbf{c} \in \mathcal{C} | i \leq w(\mathbf{c}) \leq i+t\}, \\
\mathcal{F} &= \{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) = i+t+1\}.
\end{aligned}$$

First, we study the sets $\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$ for all $\mathbf{a}, \mathbf{b} \in \mathcal{E} \cup \mathcal{F}$, $\mathbf{a} \neq \mathbf{b}$.

1. If $\mathbf{a}, \mathbf{b} \in \mathcal{E}$, then
$$\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b}) = \emptyset.$$
For if the contrary holds, i.e. there exists a $\mathbf{v} \in \mathcal{V}_i$ such that $\mathbf{v} \in \mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$, then $d_{As}(\mathbf{a}, \mathbf{b}) \leq 2t$, which contradicts the code $\mathcal{C}$ being t-AsEC.

2. If $\mathbf{a} \in \mathcal{E} \cup \mathcal{F}$, $\mathbf{b} \in \mathcal{F}$, then $w(\mathbf{a}) = i + j$ (with $0 \leq j \leq t+1$) and $N(\mathbf{a}, \mathbf{b}) = t + 1 + s$ (with $s \geq 0$), for $N(\mathbf{a}, \mathbf{b}) \leq t$ would imply
$$d_{As}(\mathbf{a}, \mathbf{b}) = 2N(\mathbf{a}, \mathbf{b}) \leq 2t,$$
which contradicts the code $\mathcal{C}$ being t-AsEC. Hence we may assume without loss of generality that $\mathbf{a}$ and $\mathbf{b}$ look like
$$\begin{array}{rcl} \mathbf{b} & = & 1^{i-s}\ 1^{t+1+s}\ 0^{s+j}\ 0^{n-i-t-1-s-j} \\ \mathbf{a} & = & 1^{i-s}\ 0^{t+1+s}\ 1^{s+j}\ 0^{n-i-t-1-s-j} \end{array}$$
which shows that
$$|\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})| = |\{\mathbf{v} \in \mathcal{V}_i | \mathbf{a} \geq \mathbf{v} \wedge \mathbf{b} \geq \mathbf{v}\}| = \begin{cases} 1 & \text{if } s = 0 \\ 0 & \text{if } s > 0 \end{cases}.$$

When estimating the number of codewords $\mathbf{e}$ in $\mathcal{E}$ having $N(\mathbf{e}, \mathbf{f}) = t+1$ with a particular codeword $\mathbf{f}$ in $\mathcal{F}$, note that $N(\mathbf{e}, \mathbf{f}) = t + 1$ implies
$$\begin{array}{rcl} |\{k | e_k = 1 \wedge f_k = 1\}| & = & i \\ |\{k | f_k = 1\}| & = & i + t + 1 \\ |\{k | e_k = 1 \wedge f_k = 0\}| & = & j \\ |\{k | f_k = 0\}| & = & n - i - t - 1 \end{array}$$
if $w(\mathbf{e}) = i + j$. Since for any two different codewords $\mathbf{a}$ and $\mathbf{b}$ of equal weight
$$d_{Sy}(\mathbf{a}, \mathbf{b}) = d_{As}(\mathbf{a}, \mathbf{b}) - |w(\mathbf{a}) - w(\mathbf{b})| = d_{As}(\mathbf{a}, \mathbf{b}) \geq 2t + 2,$$
it follows that
$$\begin{array}{rcl} & & |\{\mathbf{e} \in \mathcal{E} | w(\mathbf{e}) = i + j \wedge N(\mathbf{e}, \mathbf{f}) = t + 1\}| \\ & \leq & T(i, i + t + 1, j, n - i - t - 1, 2t + 2) \\ & = & T(t + 1, i + t + 1, j, n - i - t - 1, 2t + 2) \\ & \leq & T^{\mathrm{u}}(t + 1, i + t + 1, j, n - i - t - 1, 2t + 2) \end{array}$$

for all $0 \leq j \leq t$ and $\mathbf{f} \in \mathcal{F}$.

When estimating the number of codewords in $\mathcal{F}$ that cover a particular vector $\mathbf{v}$ in $\mathcal{V}_i$, note that in each coordinate where $\mathbf{v}$ equals 0 at most one of the covering codewords of weight i+t+1 equals 1, due to their mutual distances. Hence

$$|\{\mathbf{f} \in \mathcal{F} | \mathbf{f} \geq \mathbf{v}\}| \leq \left\lfloor \frac{n-i}{t+1} \right\rfloor$$

for all $\mathbf{v} \in \mathcal{V}_i$.

Next consider the set $\mathcal{X}$ containing all the vectors in $\mathcal{V}_i$ that are not covered by a codeword $\mathbf{e} \in \mathcal{E}$:

$$\mathcal{X} = \mathcal{V}_i \setminus \left(\bigcup_{\mathbf{e} \in \mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right).$$

Let $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{A_{i+t+1}}$ be the elements of $\mathcal{F}$. For each $\mathbf{f}_r$ we consider the set $\mathcal{X}_r$ containing all the vectors in $\mathcal{V}_i$ that are covered by $\mathbf{f}_r$, but not covered by any codeword $\mathbf{e} \in \mathcal{E}$:

$$\mathcal{X}_r = \mathcal{H}_i(\mathbf{f}_r) \cap \mathcal{X}$$

for $r = 1, 2, \ldots, A_{i+t+1}$. These sets have the following properties:

1.

$$\begin{aligned} |\mathcal{X}| &= \left| \mathcal{V}_i \setminus \left(\bigcup_{\mathbf{e} \in \mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \right| \\ &= \binom{n}{i} - \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j}; \end{aligned}$$

2.

$$\begin{aligned} |\mathcal{X}_r| &= |\mathcal{H}_i(\mathbf{f}_r)| - \left| \mathcal{H}_i(\mathbf{f}_r) \cap \left(\bigcup_{\mathbf{e} \in \mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \right| \\ &= \binom{i+t+1}{i} - \sum_{\substack{\mathbf{e} \in \mathcal{E} \\ N(\mathbf{e}, \mathbf{f}_r) = t+1}} 1 \\ &\geq \binom{i+t+1}{i} \\ &\quad - \sum_{j=0}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2) \\ &= f_2^l(i, t, n), \end{aligned}$$

for all $1 \leq r \leq A_{i+t+1}$;

3.

$$\begin{aligned} |\{\mathcal{X}_r | \mathbf{x} \in \mathcal{X}_r\}| &= |\{\mathbf{f}_r \in \mathcal{F} | \mathbf{f}_r \geq \mathbf{x}\}| \\ &\leq \left\lfloor \frac{n-i}{t+1} \right\rfloor \end{aligned}$$

for all $\mathbf{x} \in \mathcal{X}$.

In conclusion,

$$\begin{aligned} & f_2'(i,t,n) \cdot A_{i+t+1} \\ &\leq \sum_{r=1}^{A_{i+t+1}} \sum_{\mathbf{x} \in \mathcal{X}_r} 1 \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\substack{r=1 \\ \mathbf{x} \in \mathcal{X}_r}}^{A_{i+t+1}} 1 \\ &\leq \left(\binom{n}{i} - \sum_{j=0}^{t} \binom{i+j}{i} A_{i+j} \right) \cdot \left\lfloor \frac{n-i}{t+1} \right\rfloor . \end{aligned}$$

□

Theorems 3.4 and 3.5 were obtained by considering the sets $\mathcal{H}_i(\mathbf{c})$ for all codewords $\mathbf{c}$ of weight between $i+k-t$ and $i+k$ in a t-AsEC code, and then including the codewords of weight $i+k+1$ into the discussion. When taking the codewords of weight $i+k-t-1$ instead of the codewords of weight $i+k+1$, similar results can be obtained. These are stated in Theorem 3.6 for $1 \leq k \leq t$ and in Theorem 3.7 for $k = 0$.

Theorem 3.6 *In a t-AsEC code $\mathcal{C}$ of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\begin{aligned} & f_1(n-i, t-k, t) A_{i+k-t-1} \\ &+ \sum_{j=1}^{t-k} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{k} \binom{i+j}{j} A_{i+j} \leq \binom{n}{i} \end{aligned}$$

for $i = t+1-k, t+2-k, \ldots, n$ and $k = 1, 2, \ldots, t$, where

$$f_1(i,k,t) = \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor .$$

Proof. This theorem can be proved in the same way as Theorem 3.4. □

Theorem 3.7 *In a t-AsEC code C of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\frac{f_2^l(n-i,t,n)}{\left\lfloor \frac{i}{t+1} \right\rfloor} A_{i-t-1} + \sum_{j=0}^{t} \binom{n-i+j}{j} A_{i-j} \leq \binom{n}{i}$$

for $i = t+1, t+2, \ldots, n$, *where*

$$f_2^l(i,t,n) = \binom{i+t+1}{i} - \sum_{j=0}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2).$$

Proof. This theorem can be proved in the same way as Theorem 3.5. □

3.2.2 The unidirectional case

We now try to sharpen the constraints from Theorem 3.3 by using arguments similar to the ones used in the previous subsection. Hence we consider not only the sets $\mathcal{H}_i(\mathbf{c})$ for all codewords $\mathbf{c}$ of weight in between $i-t$ and $i+t$ in a t-UEC code, but also for codewords of weight $i+t+1$. The result is given in Theorem 3.8. When including the codewords of weight $i-t-1$ instead of weight $i+t+1$, we obtain the result as stated in Theorem 3.9.

Theorem 3.8 *In a t-UEC code C of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\sum_{j=1}^{t} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j} + \frac{f_3^l(i,t,n)}{\left\lfloor \frac{n-i}{t+1} \right\rfloor} A_{i+t+1} \leq \binom{n}{i}$$

for $i = 0, 1, \ldots, n-t-1$, *where*

$$f_3^l(i,t,n) = \begin{pmatrix} i+t+1 \\ i \end{pmatrix} - \begin{pmatrix} 2t+1 \\ t+1 \end{pmatrix} A^u(i+t+1, 2t+2, 2t+1) - \sum_{j=1}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2).$$

Proof. Let

$$\begin{aligned} \mathcal{E} &= \{\mathbf{c} \in \mathcal{C} | i-t \le w(\mathbf{c}) \le i+t\}, \\ \mathcal{E}_1 &= \{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) = i-t\}, \\ \mathcal{E}_2 &= \{\mathbf{c} \in \mathcal{C} | i-t+1 \le w(\mathbf{c}) \le i\}, \\ \mathcal{E}_3 &= \{\mathbf{c} \in \mathcal{C} | i+1 \le w(\mathbf{c}) \le i+t\}, \\ \mathcal{F} &= \{\mathbf{c} \in \mathcal{C} | w(\mathbf{c}) = i+t+1\}. \end{aligned}$$

First, we study the sets $\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$ for all $\mathbf{a}, \mathbf{b} \in \mathcal{E} \cup \mathcal{F}$, $\mathbf{a} \neq \mathbf{b}$.

1. If $\mathbf{a}, \mathbf{b} \in \mathcal{E}$, then
$$\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b}) = \emptyset.$$
For if the contrary holds, i.e. there exists a $\mathbf{v} \in \mathcal{V}_i$ such that $\mathbf{v} \in \mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$, then $d_U(\mathbf{a}, \mathbf{b}) \le 2t$, which contradicts the code $\mathcal{C}$ being t-UEC.

2. If $\mathbf{a} \in \mathcal{E}_1$, $\mathbf{b} \in \mathcal{F}$, $\mathbf{b} \not\geq \mathbf{a}$, then
$$\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b}) = \emptyset.$$
For if the contrary holds, i.e. there exists a $\mathbf{v} \in \mathcal{V}_i$ such that $\mathbf{v} \in \mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$, then $\mathbf{b} \ge \mathbf{v} \ge \mathbf{a}$, which contradicts $\mathbf{b} \not\geq \mathbf{a}$.

3. If $\mathbf{a} \in \mathcal{E}_1$, $\mathbf{b} \in \mathcal{F}$, $\mathbf{b} \ge \mathbf{a}$, then $N(\mathbf{b}, \mathbf{a}) = 0$ and $N(\mathbf{a}, \mathbf{b}) = 2t+1$, and so
$$|\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})| = |\mathbf{v} \in \mathcal{V}_i | \mathbf{b} \ge \mathbf{v} \ge \mathbf{a}\}| = \begin{pmatrix} 2t+1 \\ t+1 \end{pmatrix}.$$

4. If $\mathbf{a} \in \mathcal{E}_2$, $\mathbf{b} \in \mathcal{F}$, then
$$\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b}) = \emptyset.$$
For if the contrary holds, i.e. there exists a $\mathbf{v} \in \mathcal{V}_i$ such that $\mathbf{v} \in \mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})$, then $\mathbf{b} \ge \mathbf{v} \ge \mathbf{a}$, and so
$$d_U(\mathbf{a}, \mathbf{b}) = w(\mathbf{b}) - w(\mathbf{a}) \le 2t,$$
which contradicts the code $\mathcal{C}$ being t-UEC.

5. If $\mathbf{a} \in \mathcal{E}_3 \cup \mathcal{F}$, $\mathbf{b} \in \mathcal{F}$, then $w(\mathbf{a}) = i + j$ (with $1 \leq j \leq t+1$) and $N(\mathbf{a},\mathbf{b}) = t + 1 + s$ (with $s \geq 0$), for $N(\mathbf{a},\mathbf{b}) \leq t$ would imply

$$d_U(\mathbf{a},\mathbf{b}) \leq 2N(\mathbf{a},\mathbf{b}) \leq 2t,$$

which contradicts the code $\mathcal{C}$ being t-UEC. Hence we may assume without loss of generality that $\mathbf{a}$ and $\mathbf{b}$ look like

$$\begin{array}{rcccccc} \mathbf{b} & = & 1^{i-s} & 1^{t+1+s} & 0^{s+j} & 0^{n-i-t-1-s-j} \\ \mathbf{a} & = & 1^{i-s} & 0^{t+1+s} & 1^{s+j} & 0^{n-i-t-1-s-j} \end{array}$$

which shows that

$$|\mathcal{H}_i(\mathbf{a}) \cap \mathcal{H}_i(\mathbf{b})| = |\{\mathbf{v} \in \mathcal{V}_i | \mathbf{a} \geq \mathbf{v} \wedge \mathbf{b} \geq \mathbf{v}\}| = \begin{cases} 1 & \text{if } s = 0 \\ 0 & \text{if } s > 0 \end{cases}.$$

When estimating the number of codewords in $\mathcal{E}_1$ that are covered by a particular codeword $\mathbf{f}$ in $\mathcal{F}$, note that these codewords form a constant weight code of length $i+t+1$, weight $i-t$ and Hamming distance at least $2t+2$, after deleting those coordinates where $\mathbf{f}$ equals 0. Hence

$$|\{\mathbf{e} \in \mathcal{E}_1 | \mathbf{f} \geq \mathbf{e}\}| \leq A(i+t+1, 2t+2, i-t) \leq A^u(i+t+1, 2t+2, i-t)$$

for all $\mathbf{f} \in \mathcal{F}$.

When estimating the number of codewords $\mathbf{e}$ in $\mathcal{E}_3$ having $N(\mathbf{e},\mathbf{f}) = t+1$ with a particular codeword $\mathbf{f}$ in $\mathcal{F}$, note that $N(\mathbf{e},\mathbf{f}) = t+1$ implies

$$\begin{array}{rcl} |\{k | e_k = 1 \wedge f_k = 1\}| & = & i \\ |\{k | f_k = 1\}| & = & i+t+1 \\ |\{k | e_k = 1 \wedge f_k = 0\}| & = & j \\ |\{k | f_k = 0\}| & = & n-i-t-1 \end{array}$$

if $w(\mathbf{e}) = i + j$. Since for any two different codewords $\mathbf{a}$ and $\mathbf{b}$ of equal weight

$$d_{Sy}(\mathbf{a},\mathbf{b}) = d_{As}(\mathbf{a},\mathbf{b}) - |w(\mathbf{a}) - w(\mathbf{b})| = d_{As}(\mathbf{a},\mathbf{b}) \geq 2t+2,$$

it follows that

$$\begin{array}{rcl} & & |\{\mathbf{e} \in \mathcal{E} | w(\mathbf{e}) = i+j \wedge N(\mathbf{e},\mathbf{f}) = t+1\}| \\ & \leq & T(i, i+t+1, j, n-i-t-1, 2t+2) \\ & = & T(t+1, i+t+1, j, n-i-t-1, 2t+2) \\ & \leq & T^u(t+1, i+t+1, j, n-i-t-1, 2t+2) \end{array}$$

for all $1 \leq j \leq t$ and $\mathbf{f} \in \mathcal{F}$.

When estimating the number of codewords in $\mathcal{F}$ that cover a particular vector $\mathbf{v}$ in $\mathcal{V}_i$, note that in each coordinate where $\mathbf{v}$ equals 0 at most one of the covering codewords of weight i+t+1 equals 1, because of their mutual distances. Hence

$$|\{\mathbf{f} \in \mathcal{F} | \mathbf{f} \geq \mathbf{v}\}| \leq \left\lfloor \frac{n-i}{t+1} \right\rfloor$$

for all $\mathbf{v} \in \mathcal{V}_i$.

Next consider the set $\mathcal{X}$ containing all the vectors in $\mathcal{V}_i$ that are unordered with all codewords $\mathbf{e} \in \mathcal{E}$:

$$\mathcal{X} = \mathcal{V}_i \setminus \left(\bigcup_{\mathbf{e} \in \mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right).$$

Let $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{A_{i+t+1}}$ be the elements of $\mathcal{F}$. For each $\mathbf{f}_r$ we consider the set containing all the vectors in $\mathcal{V}_i$ that are covered by $\mathbf{f}_r$, but are unordered with all codewords $\mathbf{e} \in \mathcal{E}$:

$$\mathcal{X}_r = \mathcal{H}_i(\mathbf{f}_r) \cap \mathcal{X}$$

for $r = 1, 2, \ldots, A_{i+t+1}$. These sets have the following properties:

1.

$$\begin{aligned} |\mathcal{X}| &= \left| \mathcal{V}_i \setminus \left(\bigcup_{\mathbf{e} \in \mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \right| \\ &= \binom{n}{i} - \sum_{j=1}^{t} \binom{n-i+j}{j} A_{i-j} - \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j}; \end{aligned}$$

2.

$$\begin{aligned} |\mathcal{X}_r| &= |\mathcal{H}_i(\mathbf{f}_r)| - \left| \mathcal{H}_i(\mathbf{f}_r) \cap \left(\bigcup_{\mathbf{e} \in \mathcal{E}} \mathcal{H}_i(\mathbf{e}) \right) \right| \\ &= \binom{i+t+1}{i} - \sum_{\substack{\mathbf{e} \in \mathcal{E}_1 \\ \mathbf{f}_r \geq \mathbf{e}}} \binom{2t+1}{t+1} - \sum_{\substack{\mathbf{e} \in \mathcal{E}_3 \\ N(\mathbf{e}, \mathbf{f}_r) = t+1}} 1 \\ &\geq \binom{i+t+1}{i} - \binom{2t+1}{t+1} A^u(i+t+1, 2t+2, i-t) \\ &\quad - \sum_{j=1}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2) \\ &= f_3^l(i,t,n), \end{aligned}$$

for all $1 \leq r \leq A_{i+t+1}$;

3.

$$\begin{aligned} |\{\mathcal{X}_r | \mathbf{x} \in \mathcal{X}_r\}| &= |\{\mathbf{f}_r \in \mathcal{F} | \mathbf{f}_r \geq \mathbf{x}\}| \\ &\leq \left\lfloor \frac{n-i}{t+1} \right\rfloor \end{aligned}$$

for all $\mathbf{x} \in \mathcal{X}$.

In conclusion,

$$\begin{aligned} & f_3^l(i,t,n) \cdot A_{i+t+1} \\ &\leq \sum_{r=1}^{A_{i+t+1}} \sum_{\mathbf{x} \in \mathcal{X}_r} 1 \\ &= \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\substack{r=1 \\ \mathbf{x} \in \mathcal{X}_r}}^{A_{i+t+1}} 1 \\ &\leq \left(\binom{n}{i} - \sum_{j=1}^{t} \binom{n-i+j}{j} A_{i-j} - \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j} \right) \cdot \left\lfloor \frac{n-i}{t+1} \right\rfloor . \end{aligned}$$

□

Theorem 3.9 *In a t-UEC code $\mathcal{C}$ of length n the weight distribution*

$$A_0, A_1, \ldots, A_n$$

satisfies

$$\begin{aligned} & \frac{f_3^l(n-i,t,n)}{\left\lfloor \frac{i}{t+1} \right\rfloor} A_{i-t-1} \\ &+ \sum_{j=1}^{t} \binom{n-i+j}{j} A_{i-j} + \sum_{j=0}^{t} \binom{i+j}{j} A_{i+j} \leq \binom{n}{i} \end{aligned}$$

for $i = t+1, t+2, \ldots, n$, where

$$\begin{aligned} f_3^l(i,t,n) &= \binom{i+t+1}{i} - \binom{2t+1}{t+1} A^u(i+t+1, 2t+2, 2t+1) \\ &\quad - \sum_{j=1}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2). \end{aligned}$$

Proof. This theorem can be proved in the same way as Theorem 3.8. □

3.2.3 Evaluation

All the constraints on a weight distributions A_j taken under consideration in this chapter are of the form

$$c_0 A_0 + c_1 A_1 + \ldots + c_n A_n \leq b,$$

where $c_0, c_1, \ldots, c_n$ and b are real numbers. Since $A_0, A_1, \ldots, A_n$ are non-negative integers, we call one constraint *sharper* than another if all c_j in the former constraint are not less than the c_j in the latter constraint, b in the former constraint does not exceed b in the latter constraint, and equality does not hold for at least one c_j or b. Hence it completely depends on the values of the functions

$$f_1(i,k,t) = \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor \tag{3.6}$$

$$f_2^l(i,t,n) = \binom{i+t+1}{i} - \sum_{j=0}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2) \tag{3.7}$$

$$f_3^l(i,t,n) = \binom{i+t+1}{i} - \binom{2t+1}{t+1} A^u(i+t+1, 2t+2, 2t+1) - \sum_{j=1}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2) \tag{3.8}$$

whether the constraints from the previous two subsections are sharper than the corresponding constraints from Theorems 3.1 and 3.3. Such a new constraint is sharper than the old one if and only if the function value concerned is positive. Hence evaluation of the new constraints can be done by studying the functions f_1, f_2^l, and f_3^l.

Evaluation of f_1

When examining the function $f_1(i,k,t)$ for $0 \leq i \leq n$ and $0 \leq k \leq t-1 \leq n-1$, we distinguish between the cases $i+k < t$ and $i+k \geq t$. If $i+k < t$, we find that

$$f_1(i,k,t) = \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor$$

$$
\begin{aligned}
&= \binom{i+k+1}{i} \\
&> 0. \qquad (3.9)
\end{aligned}
$$

If $i + k \geq t$, we find that

$$
\begin{aligned}
f_1(i,k,t) &= \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor \\
&\geq \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \frac{i+k+1}{t+1} \\
&= \frac{(i+k+1)!}{i!(k+1)!} - \frac{(t+1)!(i+k+1)}{(k+1)!(t-k)!(t+1)} \\
&= \frac{i+k+1}{k+1} \left(\frac{(i+k)!}{i!k!} - \frac{t!}{k!(t-k)!} \right) \\
&= \frac{i+k+1}{k+1} \left(\binom{i+k}{k} - \binom{t}{k} \right) \\
&\geq 0. \qquad (3.10)
\end{aligned}
$$

Note that equality holds everywhere in (3.10) if and only if

$$
i + k + 1 \equiv 0 \pmod{t+1} \wedge (k = 0 \vee i + k = t). \qquad (3.11)
$$

Hence the constraints from Theorem 3.4 equal the constraints from Theorem 3.1 in the cases satisfying (3.11), and they are sharper in all other cases. Also, it follows that the constraints from Theorem 3.6 equal the constraints from Theorem 3.1 in the cases satisfying

$$
n - i - k \equiv 0 \pmod{t+1} \wedge (k = t \vee k = n - i), \qquad (3.12)
$$

and that they are sharper in all other cases. For example, we consider $f_1(i,0,t)$:

$$
\begin{aligned}
f_1(i,0,t) &= (i+1) - (t+1) \cdot \left\lfloor \frac{i+1}{t+1} \right\rfloor \\
&= \begin{cases}
0 & \text{if } i \equiv t \pmod{t+1} \\
1 & \text{if } i \equiv 0 \pmod{t+1} \\
2 & \text{if } i \equiv 1 \pmod{t+1} \\
\vdots & \\
t & \text{if } i \equiv t-1 \pmod{t+1}
\end{cases}
\end{aligned}
$$

Evaluation of f_2^l

In order to examine the function $f_2^l(i,t,n)$ for $0 \leq i \leq n-t-1$ and $1 \leq t \leq n-1$, we first consider the function

$$f_2(i,t,n) = \begin{pmatrix} i+t+1 \\ i \end{pmatrix} - \sum_{j=0}^{t} T(t+1, i+t+1, j, n-i-t-1, 2t+2), \quad (3.13)$$

on which $f_2^l(i,t,n)$ is a lower bound. Since the function $T(t+1, i+t+1, j, n-i-t-1, 2t+2)$ is *nondecreasing* in n, the function $f_2(i,t,n)$ is *nonincreasing* in n. So if we are able to find a n^* such that $f_2(i,t,n^*)$ is nonpositive, then $f_2(i,t,n)$ is nonpositive for all $n > n^*$. To this end we define the code $\mathcal{C} = \{\mathbf{c}_r\}$ of size

$$M = \begin{pmatrix} i+t+1 \\ t+1 \end{pmatrix},$$

length $i+t+1+Mt$, and Hamming distance $2t+2$ by

$$\begin{array}{lllllllll}
\mathbf{c}_1 & = & \mathbf{v}_1 & 1^t & 0^t & 0^t & 0^t & \ldots & 0^t & 0^t \\
\mathbf{c}_2 & = & \mathbf{v}_2 & 0^t & 1^t & 0^t & 0^t & \ldots & 0^t & 0^t \\
\mathbf{c}_3 & = & \mathbf{v}_3 & 0^t & 0^t & 1^t & 0^t & \ldots & 0^t & 0^t \\
\ldots & & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\mathbf{c}_M & = & \mathbf{v}_M & 0^t & 0^t & 0^t & 0^t & \ldots & 0^t & 1^t
\end{array}$$

where $\mathbf{v}_r$ runs through all the vectors of length $i+t+1$ and weight $t+1$. It follows from this code that

$$T(t+1, i+t+1, t, t\begin{pmatrix} i+t+1 \\ t+1 \end{pmatrix}, 2t+2) \geq |\mathcal{C}| = \begin{pmatrix} i+t+1 \\ t+1 \end{pmatrix},$$

and so we have for all lengths

$$n \geq n^* = i+t+1+t\begin{pmatrix} i+t+1 \\ t+1 \end{pmatrix},$$

that

$f_2^l(i,t,n)$

$$
\begin{aligned}
&\leq f_2(i,t,n) \\
&\leq f_2(i,t,n^{'}) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \sum_{j=0}^{t} T(t+1, i+t+1, j, n^{'} - i - t - 1, 2t+2) \\
&\leq \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - T(t+1, i+t+1, t, n^{'} - i - t - 1, 2t+2) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - T(t+1, i+t+1, t, t \left(\begin{array}{c} i+t+1 \\ t+1 \end{array} \right), 2t+2) \\
&\leq \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left(\begin{array}{c} i+t+1 \\ t+1 \end{array} \right) \\
&= 0.
\end{aligned}
$$

On the other hand, we find for the smallest possible length

$$n = i + t + 1$$

that

$$
\begin{aligned}
&f_2(i,t,n) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \sum_{j=0}^{t} T(t+1, i+t+1, j, n - i - t - 1, 2t+2) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \sum_{j=0}^{t} T(t+1, i+t+1, j, 0, 2t+2) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - T(t+1, i+t+1, 0, 0, 2t+2) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left\lfloor \frac{i+t+1}{t+1} \right\rfloor \\
&\geq 0,
\end{aligned}
$$

where equality holds if and only if $i = 0$. Hence we can conclude that the constraints from Theorem 3.5 do not improve the constraints from Theorem 3.1 ($k = t$) if $i = 0$ or if n is relatively large with regard to t and i. The constraints from Theorem 3.5 certainly are sharper than the constraints from Theorem 3.1 ($k = t$) for $i > 0$ and $n = i + t + 1$ when choosing

$$f_2^t(i,t,n) = \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left\lfloor \frac{i+t+1}{t+1} \right\rfloor,$$

and they may be sharper for $i > 0$ and n not too large with regard to t and i. It can also be concluded that the constraints from Theorem 3.7 may only be sharper than the constraints from Theorem 3.1 ($k = 0$) if $i < n$ and n is not too large with regard to t and $n - i$. We illustrate the foregoing by showing an example in which $i = t = 1$ and $n \geq 3$. It can easily be checked that

$$T(2, 3, 0, n-3, 4) = 1$$

and

$$T(2, 3, 1, n-3, 4) = \begin{cases} 0 & \text{if } n = 3 \\ 1 & \text{if } n = 4 \\ 2 & \text{if } n = 5 \\ 3 & \text{if } n \geq 6 \end{cases}.$$

So

$$\begin{aligned} f_2(1, 1, n) &= \binom{3}{1} - \sum_{j=0}^{1} T(2, 3, j, n-3, 4) \\ &= \begin{cases} 2 & \text{if } n = 3 \\ 1 & \text{if } n = 4 \\ 0 & \text{if } n = 5 \\ -1 & \text{if } n \geq 6 \end{cases}. \end{aligned}$$

Hence for $i = t = 1$ the constraints from Theorem 3.5 are stronger than the constraints from Theorem 3.1 ($k = 1$) if $3 \leq n \leq 4$, they are equally strong if $n = 5$, but they are weaker if $n \geq 6$.

Evaluation of f_3^l

In order to examine the function $f_3^l(i, t, n)$ for $0 \leq i \leq n - t - 1$ and $1 \leq t \leq n - 1$, we follow the same strategy as in the examination of $f_2^l(i, t, n)$. Hence we first consider the function

$$\begin{aligned} f_3(i, t, n) &= \binom{i+t+1}{i} - \binom{2t+1}{t+1} A(i+t+1, 2t+2, 2t+1) \\ &\quad - \sum_{j=1}^{t} T(t+1, i+t+1, j, n-i-t-1, 2t+2). \end{aligned}$$

on which $f_3^l(i,t,n)$ is a lower bound. This function $f_3(i,t,n)$ is *nonincreasing* in n and is nonpositive for all

$$n \geq n^* = i + t + 1 + t \left(\begin{array}{c} i+t+1 \\ t+1 \end{array} \right).$$

For the smallest possible length

$$n = i + t + 1$$

we find that

$$\begin{aligned}
& f_3(i,t,n) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left(\begin{array}{c} 2t+1 \\ t+1 \end{array} \right) A(i+t+1, 2t+2, 2t+1) \\
&\quad - \sum_{j=1}^{t} T(t+1, i+t+1, j, n-i-t-1, 2t+2) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left(\begin{array}{c} 2t+1 \\ t+1 \end{array} \right) A(i+t+1, 2t+2, 2t+1) \\
&\quad - \sum_{j=1}^{t} T(t+1, i+t+1, j, 0, 2t+2) \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left(\begin{array}{c} 2t+1 \\ t+1 \end{array} \right) A(i+t+1, 2t+2, 2t+1) \\
&\geq \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \left(\begin{array}{c} 2t+1 \\ t+1 \end{array} \right) \cdot \frac{i+t+1}{2t+1} \cdot \frac{i+t}{2t} \cdot \cdots \cdot \frac{i+1}{t+1} \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \frac{(2t+1)!(i+t+1)!t!}{t!(t+1)!i!(2t+1)!} \\
&= \left(\begin{array}{c} i+t+1 \\ i \end{array} \right) - \frac{(i+t+1)!}{(t+1)!i!} \\
&= 0,
\end{aligned}$$

where we have used a well-known upper bound for constant weight codes (see e.g. [21]) to estimate $A(i+t+1, 2t+2, 2t+1)$. Hence we can conclude that the constraints from Theorem 3.8 certainly are not weaker than the constraints from Theorem 3.3 when n is relatively small with regard to t and i, and that they certainly are not stronger when n is relatively large with regard to t and i. This also holds for the constraints from Theorem 3.9

when substituting i by $n-i$. Again, we illustrate the foregoing by showing an example, in which $i = 3$, $t = 1$, and $n \geq 5$. It can easily be checked that

$$A(5,4,3) = 2$$

and

$$T(2,5,1,n-5,4) = \begin{cases} 0 & \text{if } n = 5 \\ 2 & \text{if } n = 6 \\ 4 & \text{if } n = 7 \\ 6 & \text{if } n = 8 \\ 8 & \text{if } n = 9 \\ 10 & \text{if } n \geq 10 \end{cases}.$$

So

$$\begin{aligned} & f_3(3,1,n) \\ &= \begin{pmatrix} 5 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 2 \end{pmatrix} A(5,4,3) - \sum_{j=1}^{1} T(2,5,j,n-5,4) \\ &= 10 - 6 - T(2,5,1,n-5,4) \\ &= \begin{cases} 4 & \text{if } n = 5 \\ 2 & \text{if } n = 6 \\ 0 & \text{if } n = 7 \\ -2 & \text{if } n = 8 \\ -4 & \text{if } n = 9 \\ -6 & \text{if } n \geq 10 \end{cases} \end{aligned}$$

Hence for $i = 3$ and $t = 1$ the constraints from Theorem 3.8 are stronger than the constraints from Theorem 3.3 if $5 \leq n \leq 6$, they are equally strong if $n = 7$, but they are weaker if $n \geq 8$.

3.3 Integer programming problems

When considering for example 1-AsEC codes of length 9, it follows from the constraints of Section 3.1 that the weight distribution

$$A_0, A_1, \ldots, A_9$$

of such a code C satisfies

$$A_0 + A_1 \leq A(10,4,1) = 1 \text{ (Theorem 3.2 with } i = 1, s = 1)$$

$$
\begin{aligned}
A_2 &\leq A(9,4,2) = 4 \text{ (Theorem 3.2 with } i=2, s=0) \\
A_3 &\leq A(9,4,3) = 12 \text{ (Theorem 3.2 with } i=3, s=0) \\
A_2 + A_3 &\leq A(10,4,3) = 13 \text{ (Theorem 3.2 with } i=3, s=1) \\
A_4 &\leq A(9,4,4) = 18 \text{ (Theorem 3.2 with } i=4, s=0) \\
A_5 &\leq A(9,4,5) = 18 \text{ (Theorem 3.2 with } i=5, s=0) \\
A_6 &\leq A(9,4,6) = 12 \text{ (Theorem 3.2 with } i=6, s=0) \\
A_7 &\leq A(9,4,7) = 4 \text{ (Theorem 3.2 with } i=7, s=0) \\
A_6 + A_7 &\leq A(10,4,7) = 13 \text{ (Theorem 3.2 with } i=7, s=1) \\
A_8 + A_9 &\leq A(10,4,9) = 1 \text{ (Theorem 3.2 with } i=9, s=1) \\
A_3 + 4A_4 &\leq \begin{pmatrix} 9 \\ 3 \end{pmatrix} = 84 \text{ (Theorem 3.1 with } i=3, k=1) \\
4A_5 + A_6 &\leq \begin{pmatrix} 9 \\ 6 \end{pmatrix} = 84 \text{ (Theorem 3.1 with } i=6, k=0)
\end{aligned}
$$

Hence

$$
\begin{aligned}
|\mathcal{C}| &= (A_0 + A_1) + (A_2 + A_3) + A_4 + A_5 + (A_6 + A_7) + (A_8 + A_9) \\
&\leq 1 + 13 + 18 + 18 + 13 + 1 \\
&= 64.
\end{aligned}
$$

Including also the sharpened constraints from Section 3.2 we obtain

$$
\begin{aligned}
A_2 + A_3 + 4A_4 &\leq \begin{pmatrix} 9 \\ 3 \end{pmatrix} = 84 \text{ (Theorem 3.6 with } i=3, k=1) \\
4A_5 + A_6 + A_7 &\leq \begin{pmatrix} 9 \\ 6 \end{pmatrix} = 84 \text{ (Theorem 3.4 with } i=6, k=0)
\end{aligned}
$$

which gives

$$
\begin{aligned}
A_2 + A_3 + A_4 &\leq 30 \\
A_5 + A_6 + A_7 &\leq 30
\end{aligned}
$$

and so

$$
\begin{aligned}
|\mathcal{C}| &= (A_0 + A_1) + (A_2 + A_3 + A_4) + (A_5 + A_6 + A_7) + (A_8 + A_9) \\
&\leq 1 + 30 + 30 + 1 \\
&= 62.
\end{aligned}
$$

Since this holds for any 1-AsEC code of length 9 we have thus established the upper bound

$$A_{As}(9,1) \leq 62.$$

This upper bound turns out to be tight, since Delsarte and Piret ([8]) have found a 1-AsEC code of length 9 and size 62.

Of cource, this technique can be applied for any n and t to obtain upper bounds on the size of t-AsEC or t-UEC codes of length n. In fact, we are solving an integer programming problem in which the total number of codewords is maximized over several constraints on the weight distribution. We now state these integer programming problems, the solutions of which give upper bounds on $A_{As}(n,t)$ (Theorem 3.10) and $A_U(n,t)$ (Theorem 3.11).

Theorem 3.10 *For $n > t \geq 1$ let*

$$I^u_{As}(n,t) = \max \sum_{r=0}^{n} Z_r,$$

where the maximum is taken over the following constraints:

1. *(a) Z_r are nonnegative integers for $r = 0,1,\ldots,n$.*
   (b) $Z_0 = 1$ and $Z_r = 0$ for $r = 1,2,\ldots,t$.
   (c) $Z_n = 1$ and $Z_{n-r} = 0$ for $r = 1,2,\ldots,t$.
2. $$\sum_{j=i-s}^{i} A^l(s,2t+2,i-j)Z_j \leq A^u(n+s,2t+2,i)$$
   for $i = t+1,t+2,\ldots,n-t-1$ and $s = 0,1,\ldots,i-t-1$.
3. *(a)*
   $$\sum_{j=1}^{t-k} \binom{n-i+j}{j} Z_{i-j} + \sum_{j=0}^{k} \binom{i+j}{j} Z_{i+j} + f_1(i,k,t)Z_{i+k+1} \leq \binom{n}{i}$$
   for $i = t-k,t-k+1,\ldots,n-k-1$ and $k = 0,1,\ldots,t-1$, where
   $$f_1(i,k,t) = \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor.$$

(b)

$$\sum_{j=0}^{t} \binom{i+j}{j} Z_{i+j} + \max\{0, \frac{f_2^l(i,t,n)}{\left\lfloor \frac{n-i}{t+1} \right\rfloor}\} Z_{i+t+1} \leq \binom{n}{i}$$

for $i = 0, 1, \ldots, n-t-1$, *where*

$$\begin{aligned} f_2^l(i,t,n) &= \binom{i+t+1}{i} \\ &\quad - \sum_{j=0}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2). \end{aligned}$$

(c)

$$\begin{aligned} & f_1(n-i, t-k, t) Z_{i+k-t-1} \\ & + \sum_{j=1}^{t-k} \binom{n-i+j}{j} Z_{i-j} + \sum_{j=0}^{k} \binom{i+j}{j} Z_{i+j} \leq \binom{n}{i} \end{aligned}$$

for $i = t+1-k, t+2-k, \ldots, n-k$ *and* $k = 1, 2, \ldots, t$, *where*

$$f_1(i,k,t) = \binom{i+k+1}{i} - \binom{t+1}{k+1} \cdot \left\lfloor \frac{i+k+1}{t+1} \right\rfloor.$$

(d)

$$\max\{0, \frac{f_2^l(n-i,t,n)}{\left\lfloor \frac{i}{t+1} \right\rfloor}\} Z_{i-t-1} + \sum_{j=0}^{t} \binom{n-i+j}{j} Z_{i-j} \leq \binom{n}{i}$$

for $i = t+1, t+2, \ldots, n$, *where*

$$\begin{aligned} f_2^l(i,t,n) &= \binom{i+t+1}{i} \\ &\quad - \sum_{j=0}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2). \end{aligned}$$

Then

$$A_{As}(n,t) \leq I_{As}^u(n,t).$$

Proof. Let $\mathcal{C}^*$ be a t-AsEC code of length n and size $A_{As}(n,t)$. Let Z_r be the number of codewords in $\mathcal{C}^*$ of weight r:

$$Z_r = |\{\mathbf{c} \in \mathcal{C}^* | w(\mathbf{c}) = r\}|$$

for $r = 0, 1, \ldots, n$. Hence the Z_r are nonnegative integers. Kløve ([17]) has shown that removing all codewords of weight m with $m \leq t$ or $m \geq n - t$ from a t-AsEC of length n and then including $\mathbf{0}$ and $\mathbf{1}$ does not decrease the size nor the distance d_{As} of the code. Hence we may assume $Z_0 = Z_n = 1$ and $Z_r = Z_{n-r} = 0$ for $r = 1, 2, \ldots t$. It follows from Theorems 3.2, 3.1, 3.4, 3.5, 3.6, and 3.7 that the Z_r also satisfy the other constraints. Hence

$$A_{As}(n,t) = |\mathcal{C}^*| = \sum_{r=0}^{n} Z_r \leq I^u_{As}(n,t).$$

□

Theorem 3.11 *For $n \geq t \geq 1$ let*

$$I^u_U(n,t) = \max \sum_{r=0}^{n} Z_r,$$

where the maximum is taken over the following constraints:

1. *Z_r are nonnegative integers for $r = 0, 1, \ldots, n$.*
2. $$\sum_{j=i-s}^{i} A^l(s, 2t+2, i-j) Z_j \leq A^u(n+s, 2t+2, i)$$

   for $i = 0, 1, \ldots, n$ and $s = 0, 1, \ldots, i$.
3. *(a)*

$$\sum_{j=1}^{t} \binom{n-i+j}{j} Z_{i-j} + \sum_{j=0}^{t} \binom{i+j}{j} Z_{i+j} + \max\{0, \frac{f^l_3(i,t,n)}{\left\lfloor \frac{n-i}{t+1} \right\rfloor}\} Z_{i+t+1} \leq \binom{n}{i}$$

for $i = 0, 1, \ldots, n-t-1$, *where*

$$f_3^l(i,t,n) = \begin{pmatrix} i+t+1 \\ i \end{pmatrix} - \begin{pmatrix} 2t+1 \\ t+1 \end{pmatrix} A^u(i+t+1, 2t+2, 2t+1) - \sum_{j=1}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2).$$

(b)

$$\max\{0, \frac{f_3^l(n-i,t,n)}{\left\lfloor \frac{i}{t+1} \right\rfloor}\} Z_{i-t-1} + \sum_{j=1}^{t} \begin{pmatrix} n-i+j \\ j \end{pmatrix} Z_{i-j} + \sum_{j=0}^{t} \begin{pmatrix} i+j \\ j \end{pmatrix} Z_{i+j} \leq \begin{pmatrix} n \\ i \end{pmatrix}$$

for $i = t+1, t+2, \ldots, n$, *where*

$$f_3^l(i,t,n) = \begin{pmatrix} i+t+1 \\ i \end{pmatrix} - \begin{pmatrix} 2t+1 \\ t+1 \end{pmatrix} A^u(i+t+1, 2t+2, 2t+1) - \sum_{j=1}^{t} T^u(t+1, i+t+1, j, n-i-t-1, 2t+2).$$

Then

$$A_U(n,t) \leq I_U^u(n,t).$$

Proof. Let C^* be a t-UEC code of length n and size $A_U(n,t)$. Let Z_r be the number of codewords in C^* of weight r:

$$Z_r = |\{\mathbf{c} \in C^* | w(\mathbf{c}) = r\}|$$

for $r = 0, 1, \ldots, n$. Hence the Z_r are nonnegative integers. It follows from Theorems 3.2, 3.3, 3.8, and 3.9 that the Z_r also satisfy the other constraints. Hence

$$A_U(n,t) = |C^*| = \sum_{r=0}^{n} Z_r \leq I_U^u(n,t).$$

□

The superscript ^{u} in both integer programming bounds $I_{A_s}^u(n,t)$ and $I_U^u(n,t)$ indicates that the actual value of these bounds depends on the choices for the values of functions like $A^l(s, 2t+2, i-j)$, $A^u(n+s, 2t+2, i)$, and $T^u(t+1, i+t+1, j, n-i-t-1, 2t+2)$ that appear in the constraints. To obtain the best results, we take the largest known values for all lower bounds and the smallest known values for all upper bounds in the constraints (see e.g. [21,1,29]). Enlargement of the knowledge about these bounds can affect the values of $I_{A_s}^u(n,t)$ and $I_U^u(n,t)$.

The integer programming bounds $I_{A_s}^u(n,t)$ and $I_U^u(n,t)$ provide the best possible or best known upper bounds on $A_{A_s}(n,t)$ and $A_U(n,t)$ for many small values of n and t (see Tables B.1, B.2, B.3, B.4). For large values of n computational problems arise concerning the solving of the integer programming problem. Furthermore, the bounds on constant weight codes are less well studied for lengths of 25 or more. Therefore, the integer programming approach appears to be less appropriate for large lengths.

3.4 *Retrospect and prospect*

In this chapter we have described a way of finding upper bounds on $A_{A_s}(n,t)$ and $A_U(n,t)$ by solving an integer programming problem. In this integer programming problem the total number of codewords in a AsEC or UEC code is maximized over some constraints on the weight distribution in such a code. Known techniques to obtain these constraints are based on constant weight and sphere-packing arguments as treated in Section 3.1. The latter can be sharpened for many cases as shown in Section 3.2. Collecting these results gives the integer programming bounds as stated in Section 3.3, which usually improve the explicit bounds from the previous chapter, but are much harder to calculate, especially when n is large.

In the next chapter we will see that for some values of n and t the integer programming bounds can be improved by applying some extra combinatorial arguments on the weight distribution of a t-AsEC or t-UEC code of length n. Since it seems hard to generalize these arguments, they will be demonstrated by showing several examples.

Chapter 4

Combinatorial upper bounds

4.1 More constraints on the weight distribution of AsEC codes

In the previous chapter upper bounds on $A_{As}(n,t)$ were obtained by deriving constraints on the weight distribution

$$A_0, A_1, \ldots, A_n$$

of a t-AsEC code of length n. Such an upper bound often turned out to be the best known, in which case there are two possibilities:

1. the largest known lower bound on $A_{As}(n,t)$ equals this upper bound;
2. the largest known lower bound on $A_{As}(n,t)$ is less than this upper bound.

If the former holds, the problem of finding $A_{As}(n,t)$ for this n and t has been solved. If the latter holds, we still do not know the exact value of $A_{As}(n,t)$. Hence we can try to construct a larger code in order to increase the lower bound, or to lower the upper bound, or both. For all options the constraints on the weight distribution from the previous chapter might be helpful, as they indicate how the weight distribution of a larger code should look. In some cases this can lead to the actual construction of such a code, for example by trial and error, which will be treated in Chapter 6. In other cases the nonexistence of such a code can be proved, which is the subject of this chapter. The constraints on the weight distribution from the following theorem form an important tool to derive this nonexistence.

Theorem 4.1 *Let C be a t-AsEC code of length n with weight distribution*

$$A_0, A_1, \ldots, A_n.$$

Let s and i be integers such that $0 < s \leq i \leq n$. Define

$$\begin{aligned} D &= \sum_{j=i-s}^{i} A_j, \\ E &= \sum_{j=i-s}^{i} jA_j, \\ q &= \lfloor E/n \rfloor, \\ r &= E - nq, \\ S_k &= \sum_{j=i-s}^{k+i-s-1} A_j \text{ for } k = 1, 2, \ldots, s, \\ S &= \sum_{j=1}^{s} S_j(S_j - 1). \end{aligned}$$

Then

$$nq(q-1) + 2rq + S \leq D(D-1)(i-t-1).$$

Proof. We define a subcode C' of C by

$$C' = \{\mathbf{c} \in C | i - s \leq w(\mathbf{c}) \leq i\}.$$

Then we extend C' by adding a tail $0^{s-i+m}1^{i-m}$ of length s to each codeword of weight m $(i - s \leq m \leq i)$. Let $\mathbf{X} = (x_{ab})$ (with $1 \leq a \leq D$ and $1 \leq b \leq n + s$) be a $D \times (n + s)$ matrix of these extended codewords as shown in Figure 4.1. Each row then has weight i and the Hamming distance between two different rows is at least $2t + 2$. Next we consider the sum P of the inner products (over the real numbers) of the rows:

$$P = \sum_{f=1}^{D} \sum_{\substack{g=1 \\ g \neq f}}^{D} \sum_{h=1}^{n+s} x_{fh} x_{gh}.$$

Since the Hamming distance between two different rows is at least $2t + 2$, their inner product is at most $i - t - 1$. Evaluating the sum P 'rowwise', we can thus conclude that

$$P \leq D(D-1)(i-t-1).$$

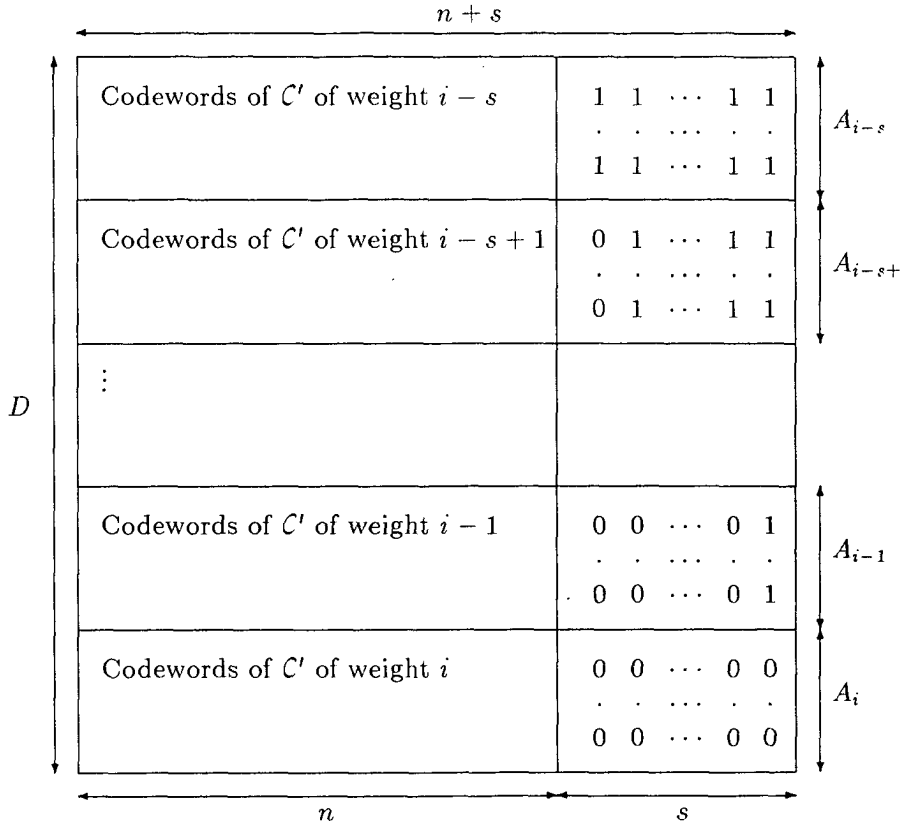


Figure 4.1: Matrix $\mathbf{X}$.

On the other hand, we can also evaluate the sum P 'columnwise':

$$P = \sum_{h=1}^{n+s} \sum_{f=1}^{D} \sum_{\substack{g=1 \\ g \neq f}}^{D} x_{fh} x_{gh}.$$

Let y_h denote the number of 1's in column h of $\mathbf{X}$:

$$y_h = |\{f | x_{fh} = 1\}|$$

for $h = 1, 2, \ldots, n+s$. Then this column contributes $y_h(y_h - 1)$ to the sum P. Hence

$$P = \sum_{h=1}^{n+s} y_h(y_h - 1).$$

From Figure 4.1 it is clear that $y_h = S_{h-n}$ for $h = n+1, n+2, \ldots, n+s$. Hence

$$\begin{aligned} P &= \sum_{h=1}^{n+s} y_h(y_h - 1) \\ &= \sum_{h=1}^{n} y_h(y_h - 1) + \sum_{h=n+1}^{n+s} y_h(y_h - 1) \\ &= \sum_{h=1}^{n} y_h(y_h - 1) + \sum_{k=1}^{s} S_k(S_k - 1) \\ &= \sum_{h=1}^{n} y_h^2 - \sum_{h=1}^{n} y_h + S \end{aligned}$$

Note that $\sum_{h=1}^{n} y_h$ equals the total number of 1's in $\mathcal{C}'$, so

$$\sum_{h=1}^{n} y_h = \sum_{j=i-s}^{i} jA_j = E.$$

The minimum of $\sum_{h=1}^{n} z_h^2$ subject to $\sum_{h=1}^{n} z_h = E = nq + r$ and the z_h's being nonnegative integers, is attained when $z_1 = z_2 = \ldots = z_r = q+1$ and $z_{r+1} = z_{r+2} = \ldots = z_n = q$. This minimum then is given by $r(q+1)^2 + (n-r)q^2$. Hence

$$\begin{aligned} D(D-1)(i-t-1) &\geq P \\ &= \sum_{h=1}^{n} y_h^2 - \sum_{h=1}^{n} y_h + S \\ &\geq (r(q+1)^2 + (n-r)q^2) - (nq+r) + S \\ &= nq(q-1) + 2rq + S. \end{aligned}$$

□

These constraints are nonlinear in A_i, and therefore not suitable for inclusion in the integer programming problems from Section 3.3. How we can apply these constraints to sharpen upper bounds on $A_{As}(n,t)$ and $A_U(n,t)$ will be explained in the next two sections by showing several examples, in which also some other combinatorial arguments will be used.

4.2 Some combinatorial bounds on the size of AsEC codes

In this section, let C^* be a t-AsEC code of length n and size $A_{As}(n,t)$, where $n > t \geq 1$. Kløve ([17]) showed that in this code we may assume $A_0 = A_n = 1$ and $A_j = A_{n-j} = 0$ for $j = 1, 2, \ldots, t$. With this in mind we can derive the new upper bounds.

Theorem 4.2 *We have*

$$A_{As}(14,3) \leq 34.$$

Proof. Let C^* be a 3-AsEC code of length 14 and size $A_{As}(14,3)$.

1. From Theorem 3.2 ($s = 0, i = 4,5,6$; $s = 1, i = 5,6$) it follows that $A_4 \leq 3$, $A_5 \leq 4$, $A_6 \leq 7$, $A_4 + A_5 \leq 6$, and $A_5 + A_6 \leq 10$. Thus $A_4 + A_5 + A_6 \leq 13$. Suppose $A_4 + A_5 + A_6 = 13$, then $A_4 = A_5 = 3$ and $A_6 = 7$. This contradicts Theorem 4.1 ($s = 1, i = 5$). Hence $A_4 + A_5 + A_6 \leq 12$.

2. Proceeding as in the previous case we can prove that $A_8 + A_9 + A_{10} \leq 12$

3. From Theorem 3.2 ($s = 0, i = 7$) it follows that $A_7 \leq 8$.

In conclusion,

$$A_{As}(14,3) = |C^*| = \sum_{j=0}^{14} A_j \leq 1 + 12 + 8 + 12 + 1 = 34.$$

□

Theorem 4.3 *We have*

$$A_{As}(16,3) \leq 90.$$

Proof. Let C^* be a 3-AsEC code of length 16 and size $A_{As}(16,3)$. From Theorem 3.2 ($s = 0, i = 4$; $s = 1, i = 6, 8, 10, 12$) it follows that

$$|C^*| = \sum_{j=0}^{16} A_j \leq 1 + 4 + 17 + 34 + 27 + 7 + 1 = 91.$$

Suppose $|C^*| = 91$, it then follows from Theorem 3.2 ($s = 0, i = 4, 12$; $s = 1, i = 5, 6, \ldots, 12$) that $A_4 = 4$ and $A_5 = 3$. However, it is easy to see that $A_4 = 4$ implies $A_5 = 0$, which contradicts $A_5 = 3$. Hence

$$A_{As}(16,3) = |C^*| \leq 90.$$

□

Theorem 4.4 *We have*

$$A_{As}(10,2) \leq 18.$$

Proof. Let C^* be a 2-AsEC code of length 10 and size $A_{As}(10,2)$. From Theorem 3.2 ($s = 0, i = 3, 4, 5, 6, 7$; $s = 1, i = 4, 7$) it follows that $A_3 \leq 3$, $A_4 \leq 5$, $A_5 \leq 6$, $A_6 \leq 5$, $A_7 \leq 3$, $A_3 + A_4 \leq 6$, and $A_6 + A_7 \leq 6$. We distinguish between two cases for A_5: $A_5 \leq 4$ and $A_5 \geq 5$.

1. In the case $A_5 \leq 4$, we have

$$|C^*| = \sum_{j=0}^{10} A_j \leq 1 + 6 + 4 + 6 + 1 = 18.$$

2. In the case $A_5 \geq 5$, suppose $A_3 + A_4 = 6$. Then $A_3 = 3 - x$ and $A_4 = 3 + x$ with $x \in \{0, 1, 2\}$. We will now derive a contradiction for each value of x.

(a) In the case $x = 0$, we have $A_3 = 3$ and $A_4 = 3$. Without loss of generality we may assume that the codewords of weight 3 and 4 resemble the following vectors $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_6$:

$$\begin{array}{lcccccccccc}
\mathbf{c}_1 & = & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{c}_2 & = & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
\mathbf{c}_3 & = & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
\mathbf{c}_4 & = & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\mathbf{c}_5 & = & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
\mathbf{c}_6 & = & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \quad .
\end{array}$$

Considering only the codewords of weight 5, we note that each column contains two or three 1's, since $A(9,6,4) = A(9,6,5) = 3$. Hence there is a codeword $\mathbf{c}$ of weight 5 whose last coordinate is equal to 1. For this codeword $\mathbf{c}$ we have

$$d_{As}(\mathbf{c}, \mathbf{c}_4) + d_{As}(\mathbf{c}, \mathbf{c}_5) + d_{As}(\mathbf{c}, \mathbf{c}_6) = 16,$$

which implies that $d_{As}(\mathbf{c}, \mathbf{c}_j) \leq 4$ for at least one $j \in \{4, 5, 6\}$. This contradicts the code being 2-AsEC.

(b) In the case $x = 1$, we have $A_3 = 2$ and $A_4 = 4$. By evaluating the sum of the inner products of the codewords of weight i and weight j $(3 \leq i \leq j \leq 5)$ in two ways ('rowwise' and 'columnwise'), we conclude that the inner product of a codeword of weight i and a different codeword of weight j is equal to 0 if $i = j = 3$, is equal to 1 if $j = 4$ and $i = 3, 4$, is equal to 2 if $i = j = 5$, and is less than 3 if $j = 5$ and $i = 3, 4$. Hence we may assume without loss of generality that the codewords of weight 3,4, and 5 resemble the following vectors $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{11}$ (or $\mathbf{c}_{12}$):

$$\begin{array}{lcccccccccc}
\mathbf{c}_1 & = & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{c}_2 & = & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
\mathbf{c}_3 & = & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
\mathbf{c}_4 & = & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
\mathbf{c}_5 & = & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\mathbf{c}_6 & = & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
\mathbf{c}_7 & = & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\mathbf{c}_8 & = & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\mathbf{c}_9 & = & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\mathbf{c}_{10} & = & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\mathbf{c}_{11} & = & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
(\mathbf{c}_{12} & = & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \quad).
\end{array}$$

Let $\mathbf{c}$ be a codeword of weight 5 whose first coordinate is equal to 1. At least two of the coordinates c_2, c_3, c_6 of $\mathbf{c}$ are equal to 1 since the inner product of $\mathbf{c}$ and $\mathbf{c}_j$ must be less than 3 for $j = 3, 4$. Furthermore, the vector $\mathbf{c}' = (c_2, c_3, c_6)$ is different from 110 and 111 since the inner product of $\mathbf{c}$ and $\mathbf{c}_1$ must be less than 3. Finally, the vectors $\mathbf{c}'$ must be different for all $\mathbf{c}$, since the inner products of these codewords $\mathbf{c}$ taken two at a time are equal to 2. Hence $\mathbf{c}'$ is equal to 101 or 011, $A_5 = 5$, and the first coordinate of $\mathbf{c}_{11}$ is equal to 0.

Let $\mathbf{x}$ be $\mathbf{c}_9$, $\mathbf{c}_{10}$, or $\mathbf{c}_{11}$ and suppose $x_6 = 1$. Then $x_4 = x_5 = 1$ since the inner product of $\mathbf{x}$ and $\mathbf{c}_j$ must be less than 3 for $j = 5, 6$. Hence $d_{A_s}(\mathbf{x}, \mathbf{c}_2) = 4$, which contradicts the code being 2-AsEC. Thus $x_6 = 0$.

Consider the code C' which is formed by taking $\mathbf{c}_9$, $\mathbf{c}_{10}$, and $\mathbf{c}_{11}$ and then deleting the first and the sixth column. C' is a constant weight code of length 8, size 3, Hamming distance at least 6, and weight 5. This contradicts $A(8, 6, 5) = 2$.

(c) In the case $x = 2$, we have $A_3 = 1$ and $A_4 = 5$. This contradicts Theorem 4.1 ($s = 1, i = 4$).

Thus it follows that $A_3 + A_4 \leq 5$. In a similar way we can prove that $A_6 + A_7 \leq 5$. Hence

$$|C^*| = \sum_{j=0}^{10} A_j \leq 1 + 5 + 6 + 5 + 1 = 18.$$

In conclusion,

$$A_{A_s}(10, 2) = |C^*| \leq 18.$$

□

Theorem 4.5 *We have*

$$A_{A_s}(16, 4) \leq 16.$$

Proof. Let C^* be a 4-AsEC code of length 16 and size $A_{A_s}(16, 4)$. From Theorem 3.2 ($s = 1, i = 6, 8$; $s = 2, i = 11$) it follows that

$$|C^*| = \sum_{j=0}^{16} A_j \leq 1 + 3 + 6 + 6 + 1 = 17.$$

Suppose $|C^*| = 17$.

1. From Theorem 3.2 $(s = 2, i = 7)$ it follows that $A_5 + A_6 + A_7 \leq 6$. Suppose $A_5 + A_6 + A_7 \leq 5$, it then follows from Theorem 3.2 $(s = 1, i = 9, 11)$ that $|C^*| \leq 1 + 5 + 6 + 3 + 1 = 16$, which contradicts $|C^*| = 17$. Hence $A_5 + A_6 + A_7 = 6$.

2. Proceeding as in the previous case we can prove that $A_9 + A_{10} + A_{11} = 6$, $A_7 + A_8 = 6$, and $A_8 + A_9 = 6$.

Hence $A_8 = 17 - 1 - 6 - 6 - 1 = 3$ and $A_7 = A_9 = 6 - 3 = 3$, which contradicts Theorem 4.1 $(s = 2, i = 9)$. In conclusion,

$$A_{As}(16, 4) = |C^*| \leq 16.$$

□

Theorem 4.6 *We have*

$$A_{As}(17, 4) \leq 26.$$

Proof. Let C^* be a 4-AsEC code of length 17 and size $A_{As}(17, 4)$.

1. From Theorem 3.2 $(s = 1, i = 6, 8)$[1] it follows that $A_5 + A_6 \leq 4$ and $A_7 + A_8 \leq 9$. Hence $A_5 + A_6 + A_7 + A_8 \leq 13$. Suppose $A_5 + A_6 + A_7 + A_8 = 13$, then $A_5 + A_6 = 4$ and $A_7 + A_8 = 9$. From Theorem 3.2 $(s = 0, i = 8)$ it follows that $A_8 \leq 6$. We now derive a contradiction for each possible value of A_8.

   (a) Suppose $A_8 = 6$, then $A_7 = 9 - 6 = 3$. This contradicts Theorem 4.1 $(s = 1, i = 8)$.

   (b) Suppose $A_8 = 5$, then $A_7 = 9 - 5 = 4$. From Theorem 3.2 $(s = 0, i = 5, 6)$ it follows that $A_5 \leq 3$ and $A_6 \leq 3$. Hence $A_6 \geq 1$, which contradicts Theorem 4.1 $(s = 2, i = 8)$.

   (c) Suppose $A_8 \leq 4$, it then follows from Theorem 3.2 $(s = 2, i = 7)$ that $A_5 + A_6 + A_7 + A_8 \leq 8 + 4 = 12$. This contradicts $A_5 + A_6 + A_7 + A_8 = 13$

   Hence $A_5 + A_6 + A_7 + A_8 \leq 12$.

2. Proceeding as in the previous case we can prove that $A_9 + A_{10} + A_{11} + A_{12} \leq 12$.

[1] Note that $A(18, 10, 6)$ is equal to 4 (not 3, as in [29, table VI]).

In conclusion,

$$A_{As}(17,4) = |C^*| = \sum_{j=0}^{17} A_j \leq 1 + 12 + 12 + 1 = 26.$$

□

Two other examples on improving the integer programming bounds by combinatorial arguments similar to those used in the proofs of the preceding theorems were found by Böinck ([3]). His results are stated in Theorems 4.7 and 4.8. Proofs are included since the results have remained unpublished thus far.

Theorem 4.7 (Böinck) *We have*

$$A_{As}(19,4) \leq 74.$$

Proof. Let C^* be a 4-AsEC code of length 19 and size $A_{As}(19,4)$.

1. From Theorem 3.2 ($s = 0, i = 5,6,7$; $s = 1, i = 6,7$) it follows that $A_5 \leq 3$, $A_6 \leq 4$, $A_7 \leq 8$, $A_5 + A_6 \leq 5$, and $A_6 + A_7 \leq 10$. Thus $A_5 + A_6 + A_7 \leq 13$. Suppose $A_5 + A_6 + A_7 = 13$, then $A_5 = 3$, $A_6 = 2$, and $A_7 = 8$. This contradicts Theorem 4.1 ($s = 1, i = 6$). Hence $A_5 + A_6 + A_7 \leq 12$.

2. Proceeding as in the previous case we can prove that $A_{12} + A_{13} + A_{14} \leq 12$.

3. From Theorem 3.2 ($s = 1, i = 9,11$) it follows that $A_8 + A_9 \leq 24$ and $A_{10} + A_{11} \leq 24$.

In conclusion,

$$A_{As}(19,4) = |C^*| = \sum_{j=0}^{19} A_j \leq 1 + 12 + 24 + 24 + 12 + 1 = 74.$$

□

Theorem 4.8 (Böinck) *We have*

$$A_{As}(20,4) \leq 133.$$

Proof. Let C^* be a 4-AsEC code of length 20 and size $A_{As}(20,4)$.

1. From Theorem 3.2 ($s = 0, i = 5,6,7$; $s = 1, i = 7$) it follows that $A_5 \leq 4$, $A_6 \leq 5$, $A_7 \leq 10$, and $A_6 + A_7 \leq 13$. By applying Theorem 4.1 ($s = 1, i = 6$) we find that $A_5 = 4$ implies $A_6 = 0$ and that $A_5 = 3$ implies $A_6 \leq 2$. Hence $A_5 + A_6 + A_7 \leq 15$.

2. From Theorem 3.2 ($s = 1, i = 9,11,13,15$) it follows that $A_8 + A_9 \leq 39$, $A_{10} + A_{11} \leq 44$, $A_{12} + A_{13} \leq 26$, and $A_{14} + A_{15} \leq 7$.

In conclusion,

$$A_{As}(20,4) = |C^*| = \sum_{j=0}^{20} A_j \leq 1 + 15 + 39 + 44 + 26 + 7 + 1 = 133.$$

□

The integer programming bound $I^u_{As}(n,t)$ from Theorem 3.10 gives $A_{As}(14,4) \leq I^u_{As}(14,4) = 9$. Helgesen ([13]) showed that the shortest possible length of a code containing 9 codewords and correcting up to 4 asymmetric errors is equal to 15. Hence $A_{As}(14,4) \leq 8$. We can also derive this result using arguments based on Theorems 3.2 and 4.1, as we did in the proofs of the preceding theorems.

Theorem 4.9 (Helgesen) *We have*

$$A_{As}(14,4) \leq 8.$$

Proof. Let C^* be a 4-AsEC code of length 14 and size $A_{As}(14,4)$. From Theorem 3.2 ($s = 2, i = 7$; $s = 1, i = 9$)[2] it follows that

$$|C^*| = \sum_{j=0}^{14} A_j \leq 1 + 4 + 3 + 1 = 9.$$

Suppose $|C^*| = 9$. We will derive a contradiction in several steps.

1. Define $J = \{5,6,7,8,9\}$. From Theorem 3.2 ($s = 0, i = 5,6,7,8,9$) it follows that $A_j \leq 2$ for all $j \in J$. Suppose $A_j = 0$ for some $j \in J$. It then is easy to check that $|C^*| \leq 8$ by employing Theorem 3.2 ($s = 0, i = 5,6,7,8,9$; $s = 1, i = 6,7,8,9$; $s = 2, i = 7,8,9$), which contradicts $|C^*| = 9$. Thus $1 \leq A_j \leq 2$ for all $j \in J$.

[2] Note that $A(16,10,7)$ is equal to 4 (not 3, as in [21, fig. 3], [1, table IID], [11, table IV]).

t	n	$I^u_{A_s}(n,t)$ (Theorem 3.10)	combinatorial bound	reference
2	10	20	18	Theorem 4.4
3	14	36	34	Theorem 4.2
3	16	91	90	Theorem 4.3
4	14	9	8	Theorem 4.9
4	16	17	16	Theorem 4.5
4	17	28	26	Theorem 4.6
4	19	76	74	Theorem 4.7
4	20	135	133	Theorem 4.8

Table 4.1: Some combinatorial upper bounds on $A_{A_s}(n,t)$.

2. Suppose $A_7 = 2$. It follows from Theorem 3.2 $(s = 2, i = 7, 9)$ that $A_5 + A_6 \leq 4 - 2 = 2$ and $A_8 + A_9 \leq 4 - 2 = 2$. Hence $|C^*| \leq 1 + 2 + 2 + 2 + 1 = 8$, which contradicts $|C^*| = 9$. Thus $A_7 = 1$.

3. From Theorem 3.2 $(s = 2, i = 7)$ it follows that $A_5 + A_6 + A_7 \leq 4$. Suppose $A_5 + A_6 + A_7 \leq 3$. It then follows from Theorem 3.2 $(s = 1, i = 9)$ that $|C^*| \leq 1 + 3 + 3 + 1 = 8$, which contradicts $|C^*| = 9$. Thus $A_5 + A_6 + A_7 = 4$.

4. It follows from the preceding steps that $A_5 = 2$, $A_6 = 1$, $A_7 = 1$ or $A_5 = 1$, $A_6 = 2$, $A_7 = 1$, which contradicts Theorem 4.1 $(s = 2, i = 7)$. Thus $|C^*| \neq 9$.

In conclusion,

$$A_{A_s}(14,4) = |C^*| \leq 8.$$

□

In Table 4.1 we summarize the results found in this section.

4.3 Some combinatorial bounds on the size of UEC codes

In this section, let C^* be a t-UEC code of length n and size $A_U(n,t)$, where $n > t \geq 1$. We give three examples on improving the integer programming bound $I^u_U(n,t)$ from Theorem 3.11 by using combinatorial arguments.

Theorem 4.10 *We have*

$$A_U(9,2) \leq 10.$$

Proof. Let C^* be a 2-UEC code of length 9 and size $A_U(9,2)$.

1. From Theorem 3.2 ($s = 2, i = 2$; $s = 0, i = 3,4$; $s = 1, i = 4$) it follows that $A_0 + A_1 + A_2 \leq 1$, $A_3 \leq 3$, $A_4 \leq 3$, and $A_3 + A_4 \leq 5$. Thus $A_0 + A_1 + A_2 + A_3 + A_4 \leq 1 + 5 = 6$. Suppose $A_0 + A_1 + A_2 + A_3 + A_4 = 6$, then $A_0 + A_1 + A_2 = 1$ and $A_3 + A_4 = 5$. Since $A_3 = 3$ implies $A_4 = 0$, we have $A_3 = 2$ and $A_4 = 3$. Without loss of generality we may assume the codewords of weight 3 and 4 look like the following vectors $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_5$:

$$\begin{array}{lcccccccccc} \mathbf{c}_1 & = & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{c}_2 & = & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \mathbf{c}_3 & = & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ \mathbf{c}_4 & = & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \mathbf{c}_5 & = & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}.$$

   Let $\mathbf{c}$ be the codeword of weight less than 3. Then $|\{j | c_j = u_j = 1\}| = 0$ for $\mathbf{u} = \mathbf{c}_1, \mathbf{c}_2$. Hence $\mathbf{c}_j \geq c$ for at least one $j \in \{3,4,5\}$, in which case $d_U(\mathbf{c}_j, \mathbf{c}) = d_{Sy}(\mathbf{c}_j, \mathbf{c}) \leq 4$. This contradicts the code being 2-UEC. Hence $A_0 + A_1 + A_2 + A_3 + A_4 \leq 5$.

2. Proceeding as in the previous case we can prove that $A_5 + A_6 + A_7 + A_8 + A_9 \leq 5$.

In conclusion,

$$A_U(9,2) = |C^*| = \sum_{j=0}^{9} A_j \leq 5 + 5 = 10.$$

□

Theorem 4.11 *We have*

$$A_U(11,3) \leq 7.$$

Proof. Let C^* be a 3-UEC code of length 11 and size $A_U(11,3)$. From Theorem 3.2 ($s = 0, i = 4,5,6,7$; $s = 1, i = 5,7$; $s = 2, i = 6,7$; $s = 3, i = 3,11$) it follows that $A_4 \leq 2$, $A_5 \leq 2$, $A_6 \leq 2$, $A_7 \leq 2$, $A_4 + A_5 \leq 3$,

$A_6 + A_7 \leq 3$, $A_4 + A_5 + A_6 \leq 4$, $A_5 + A_6 + A_7 \leq 4$, $A_0 + A_1 + A_2 + A_3 \leq 1$, and $A_8 + A_9 + A_{10} + A_{11} \leq 1$. Thus $A_0 + A_1 + \ldots + A_{11} \leq 8$. Suppose $A_0 + A_1 + \ldots + A_{11} = 8$, then $A_0 + A_1 + A_2 + A_3 = 1$, $A_8 + A_9 + A_{10} + A_{11} = 1$, $A_4 = A_7 = 2$, and $A_5 = A_6 = 1$. Without loss of generality we may assume that the codewords of weight 4 and 5 look like the following vectors $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$:

$$\begin{array}{lcccccccccccc} \mathbf{c}_1 & = & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{c}_2 & = & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \mathbf{c}_3 & = & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} .$$

Let $\mathbf{c}$ be the codeword of weight less than 4. Then $|\{j | c_j = u_j = 1\}| = 0$ for $\mathbf{u} = \mathbf{c}_1, \mathbf{c}_2$. Hence $\mathbf{c}_3 \geq \mathbf{c}$, which implies $d_U(\mathbf{c}_3, \mathbf{c}) = d_{SY}(\mathbf{c}_3, \mathbf{c}) \leq 5$. This contradicts the code being 3-UEC. Hence $A_0 + A_1 + \ldots + A_{11} \neq 8$. In conclusion,

$$A_U(11,3) = |\mathcal{C}^*| = \sum_{j=0}^{11} A_j \leq 7.$$

□

Theorem 4.12 *We have*

$$A_U(12,3) \leq 10.$$

Proof. Let $\mathcal{C}^*$ be a 3-UEC code of length 12 and size $A_U(12,3)$. From Theorem 3.2 ($s = 0, i = 4,5,6,7,8$; $s = 1, i = 5,6,7,8$; $s = 3, i = 3,12$) it follows that $A_4 \leq 3$, $A_5 \leq 3$, $A_6 \leq 4$, $A_7 \leq 3$, $A_8 \leq 3$, $A_4 + A_5 \leq 3$, $A_5 + A_6 \leq 4$, $A_6 + A_7 \leq 4$, $A_7 + A_8 \leq 3$, $A_0 + A_1 + A_2 + A_3 \leq 1$, and $A_9 + A_{10} + A_{11} + A_{12} \leq 1$. From Theorem 3.3 ($i = 1, 11$) it follows that $12A_0 + A_1 + 2A_2 + 3A_3 + 4A_4 \leq 12$ and $4A_8 + 3A_9 + 2A_{10} + A_{11} + 12A_{12} \leq 12$. Hence $A_4 = 3$ implies $A_0 = A_1 = A_2 = A_3 = 0$. Thus

$$\begin{aligned} &(A_0 + A_1 + A_2 + A_3 + A_4) + (A_5 + A_6) \\ &+ (A_7 + A_8) + (A_9 + A_{10} + A_{11} + A_{12}) \\ &\leq 3 + 4 + 3 + 1 = 11. \end{aligned}$$

Suppose $A_0 + A_1 + \ldots + A_{12} = 11$, then $A_0 + A_1 + A_2 + A_3 = 1$, $A_9 + A_{10} + A_{11} + A_{12} = 1$, $A_4 = A_8 = 2$, $A_5 = A_7 = 1$, and $A_6 = 3$. Let $\mathbf{Y} = (y_{ab})$

(with $1 \leq a \leq 11$ and $1 \leq b \leq 12$) be a 11×12 matrix of these codewords:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & = & \{ \mathbf{y}_1 \text{ of weight less than 4} \\ \mathbf{Y}_2 & = & \begin{cases} \mathbf{y}_2 \text{ of weight 4} \\ \mathbf{y}_3 \text{ of weight 4} \\ \mathbf{y}_4 \text{ of weight 5} \end{cases} \\ \mathbf{Y}_3 & = & \begin{cases} \mathbf{y}_5 \text{ of weight 6} \\ \mathbf{y}_6 \text{ of weight 6} \\ \mathbf{y}_7 \text{ of weight 6} \end{cases} \\ \mathbf{Y}_4 & = & \begin{cases} \mathbf{y}_8 \text{ of weight 7} \\ \mathbf{y}_9 \text{ of weight 8} \\ \mathbf{y}_{10} \text{ of weight 8} \end{cases} \\ \mathbf{Y}_5 & = & \{ \mathbf{y}_{11} \text{ of weight more than 8} \end{bmatrix}.$$

The matrix $\mathbf{Y}_2$ is equivalent to

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

or equivalent to

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

In the former case it is impossible to choose a matrix $\mathbf{Y}_1$. Hence the latter case holds. Analogously it follows that $\mathbf{Y}_4$ is equivalent to

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

Matrix $\mathbf{Y}_3$ is equivalent to

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

Now it easily follows that the weight of $\mathbf{y}_1$ equals 2 and the weight of $\mathbf{y}_{11}$ equals 10.

Next we consider the sum Q of the inner products of the rows of $\mathbf{Y}_2$ and the rows of $\mathbf{Y}_3$:

$$Q = \sum_{i=2}^{4} \sum_{j=5}^{7} \sum_{k=1}^{11} y_{ik} y_{jk}.$$

As the inner product of a row of $\mathbf{Y}_2$ and a row of $\mathbf{Y}_3$ is at most 2, we find by evaluating this sum 'rowwise' that

$$Q \leq 3 \cdot 3 \cdot 2 = 18.$$

By observing that $\mathbf{Y}_3$ contains six columns of weight 2 and six of weigth 1 and that $\mathbf{Y}_2$ contains two columns of weight 2, nine of weight 1, and one of weight 0, we find for the 'columnwise' evaluation that

$$Q \geq 1 \cdot 0 \cdot 2 + 5 \cdot 1 \cdot 2 + 4 \cdot 1 \cdot 1 + 2 \cdot 2 \cdot 1 = 0 + 10 + 4 + 4 = 18.$$

Hence the inner product of a row of $\mathbf{Y}_2$ and a row of $\mathbf{Y}_3$ must be exactly 2, the columns of weight 2 in $\mathbf{Y}_2$ must face columns of weight 1 in $\mathbf{Y}_3$, and the column of weight 0 in $\mathbf{Y}_2$ must face a column of weight 2 in $\mathbf{Y}_3$. Using similar arguments to the rows of $\mathbf{Y}_4$ and $\mathbf{Y}_3$, we can conclude that the inner product of $\mathbf{y}_8$ and a row of $\mathbf{Y}_3$ must be exactly 3, that the inner product of $\mathbf{y}_9$ or $\mathbf{y}_{10}$ and a row of $\mathbf{Y}_3$ must be exactly 4, that the columns of weight 1 in $\mathbf{Y}_4$ must face columns of weight 2 in $\mathbf{Y}_3$, and that the column of weight 3 in $\mathbf{Y}_4$ must face a column of weight 1 in $\mathbf{Y}_3$.

By starting with $\mathbf{Y}_2$ and then (partially) filling in $\mathbf{Y}_1$, $\mathbf{Y}_3$, $\mathbf{Y}_4$, and $\mathbf{Y}_5$ we find for the matrix $\mathbf{Y}$ without loss of generality:

$$\mathbf{Y} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
\cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & \cdot & 1 & 1 & 1 & 0 \\
\cdot & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}.$$

Now since the inner product of $\mathbf{y}_9$ and $\mathbf{y}_5$ must be equal to 4 and since $\mathbf{y}_9$ is not allowed to cover $\mathbf{y}_3$, the first coordinate of $\mathbf{y}_9$ must be equal to 1.

t	n	$I_U^u(n,t)$ (Theorem 3.11)	combinatorial bound	reference
2	9	12	10	Theorem 4.10
3	11	8	7	Theorem 4.11
3	12	11	10	Theorem 4.12

Table 4.2: Some combinatorial upper bounds on $A_U(n,t)$.

But then $\mathbf{y}_9$ covers $\mathbf{y}_4$ and so $d_U(\mathbf{y}_9, \mathbf{y}_4) = 8 - 5 = 3$ which contradicts the code being 3-UEC. Hence $A_0 + A_1 + \ldots + A_{12} \neq 11$. In conclusion,

$$A_U(12,3) = |C| = \sum_{j=0}^{12} A_j \leq 10.$$

□

In Table 4.2 we summarize the results found in this section.

4.4 *Retrospect and prospect*

In this chapter we have sharpened some of the integer programming bounds on $A_{As}(n,t)$ and $A_U(n,t)$ from the previous chapter by using some extra combinatorial arguments. The main argument has been given in Theorem 4.1 in Section 4.1. Combining Theorems 4.1 and 3.2 is mostly the key to the new bounds. Examples have been given in Section 4.2 for the asymmetric case and in Section 4.3 for the unidirectional case.

This chapter completes the part of the thesis on *upper bounds* on $A_{As}(n,t)$ and $A_U(n,t)$. The best known upper bounds in the ranges $1 \leq t \leq 4$ and $t \leq n \leq 23$ can be found in Tables B.1, B.2, B.3, and B.4. The next two chapters will be devoted to deriving *lower bounds* on $A_{As}(n,t)$ and $A_U(n,t)$ by constructing AsEC and UEC codes.

Chapter 5

Construction method

5.1 Expurgation and puncturing

Lower bounds on $A_{As}(n,t)$ and $A_U(n,t)$ can be obtained by constructing codes. Some well-known construction methods for AsEC codes are the 'prefix/suffix' constructions of Kim and Freiman ([15,16]), and the 'group-theoretic' constructions of Varshamov ([40,22]) and Constantin and Rao ([7,24,14,9]).

In [8] Delsarte and Piret introduced the idea of constructing a t-AsEC code by modifying an initial code with good (Hamming) distance properties by successive judicious deletions of coordinates and vectors. Shiozaki ([34]) presented a construction method in which a t-AsEC code of length $n-1$ is obtained by expurgating and puncturing a t-SyEC code of length n. In this chapter we generalize Shiozaki's method in such a way that a t-AsEC or t-UEC code of length $n-m$ is obtained by expurgating and puncturing a t-SyEC code of length n.

In this first section we give the construction method itself. In Section 5.2 we present algorithms to optimize the cardinalities of the codes obtained by this method. In Section 5.3 we consider some decoding aspects of the codes. In each section we start by treating the subject in general, then we continue by splitting up in the asymmetric case and the unidirectional case, and finally we finish by giving some examples.

General case

We start by describing the expurgating and puncturing technique in

general. We consider a t-SyEC code $\mathcal{C}_1$ of length n and vectors

$$\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{n-m} \tag{5.1}$$

of length m, where $1 \leq m \leq n - 2t$. Let $\mathbf{x} = (\mathbf{x}', \mathbf{x}'')$ be the $(n-m, m)$ partition of any $\mathbf{x} \in (GF(2))^n$. We define

$$\mathcal{T}_i(\mathbf{s}) = \{\mathbf{c} \in \mathcal{C}_1 | w(\mathbf{c}') = i \wedge \mathbf{c}'' = \mathbf{s}\} \tag{5.2}$$

for $i = 0, 1, \ldots, n-m$ and $\mathbf{s} \in (GF(2))^m$. We define the code $\mathcal{Y}$ of length n by

$$\mathcal{Y} = \bigcup_{i=0}^{n-m} \mathcal{T}_i(\mathbf{a}_i). \tag{5.3}$$

The code $\mathcal{C}_2$ of length $n-m$ is now formed by taking all the codewords of $\mathcal{Y}$ and then deleting the last m coordinates of each codeword. The size of the code $\mathcal{C}_2$ equals

$$|\mathcal{C}_2| = |\mathcal{Y}| = \sum_{i=0}^{n-m} |\mathcal{T}_i(\mathbf{a}_i)|. \tag{5.4}$$

The whole concept is sketched in Figure 5.1.

Asymmetric case

If the vectors $\mathbf{a}_i$ satisfy

$$d_{Sy}(\mathbf{a}_j, \mathbf{a}_{j+1}) \leq 1 \text{ for all } 0 \leq j \leq n-m-1, \tag{5.5}$$

then the code $\mathcal{C}_2$ is t-AsEC. This can be shown as follows. Let $\mathbf{u}, \mathbf{v} \in \mathcal{C}_2$ with $\mathbf{u} \neq \mathbf{v}$. Because of (5.5) it follows that

$$d_{Sy}(\mathbf{a}_{w(\mathbf{u})}, \mathbf{a}_{w(\mathbf{v})}) \leq |w(\mathbf{u}) - w(\mathbf{v})|.$$

Therefore,

$$\begin{aligned} d_{As}(\mathbf{u}, \mathbf{v}) &= d_{Sy}(\mathbf{u}, \mathbf{v}) + |w(\mathbf{u}) - w(\mathbf{v})| \\ &\geq 2t + 1 - d_{Sy}(\mathbf{a}_{w(\mathbf{u})}, \mathbf{a}_{w(\mathbf{v})}) + |w(\mathbf{u}) - w(\mathbf{v})| \\ &\geq 2t + 1 - |w(\mathbf{u}) - w(\mathbf{v})| + |w(\mathbf{u}) - w(\mathbf{v})| \\ &= 2t + 1. \end{aligned}$$

Hence $\mathcal{C}_2$ is indeed t-AsEC if the $\mathbf{a}_i$ satisfy (5.5).

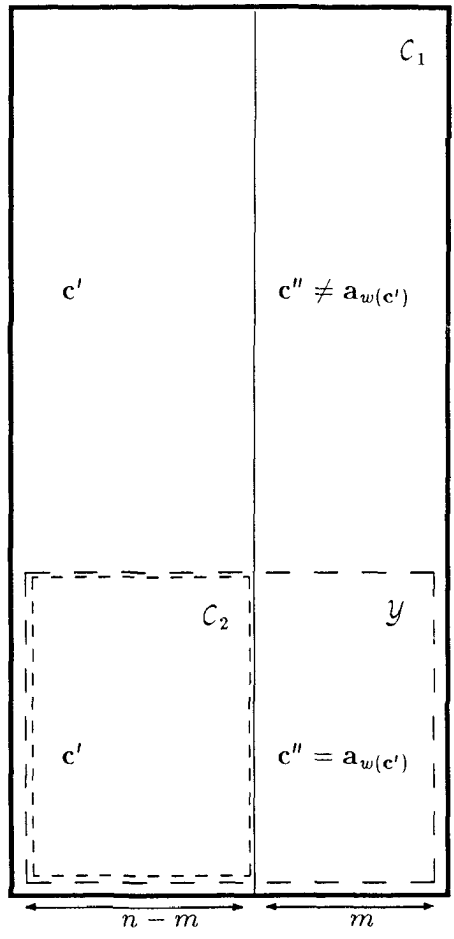


Figure 5.1: Construction method diagram.

Shiozaki's method corresponds to the situation that $m = 1$, in which case (5.5) is satisfied for any choice for the $\mathbf{a}_i$.

Unidirectional case

If the vectors $\mathbf{a}_i$ satisfy both

$$d_{Sy}(\mathbf{a}_j, \mathbf{a}_{j+1}) \leq 1 \text{ for all } 0 \leq j \leq n-m-1 \tag{5.6}$$

and

$$\mathbf{a}_j = \mathbf{a}_{j+2t} \text{ for all } 0 \leq j \leq n-m-2t, \tag{5.7}$$

then the code $\mathcal{C}_2$ is t-UEC. This can be shown as follows. For convenience we first define

$$\mathbf{a}_{n-m+j} = \mathbf{a}_{n-m+j-2t} \text{ for all } 1 \leq j \leq 2t \tag{5.8}$$

in addition to (5.1). Let $\mathbf{u}, \mathbf{v} \in \mathcal{C}_2$ with $\mathbf{u} \neq \mathbf{v}$ and $w(\mathbf{v}) \geq w(\mathbf{u})$. We distinguish between the cases $N(\mathbf{v}, \mathbf{u}) > 0$ and $N(\mathbf{v}, \mathbf{u}) = 0$.

1. The case $N(\mathbf{v}, \mathbf{u}) > 0$. Since $w(\mathbf{v}) \geq w(\mathbf{u})$, it follows that $N(\mathbf{u}, \mathbf{v}) \geq N(\mathbf{v}, \mathbf{u}) > 0$, and so it follows from (5.6) that

$$\begin{aligned} d_U(\mathbf{u}, \mathbf{v}) &= d_{As}(\mathbf{u}, \mathbf{v}) \\ &= d_{Sy}(\mathbf{u}, \mathbf{v}) + |w(\mathbf{u}) - w(\mathbf{v})| \\ &\geq 2t + 1 - d_{Sy}(\mathbf{a}_{w(\mathbf{u})}, \mathbf{a}_{w(\mathbf{v})}) + |w(\mathbf{u}) - w(\mathbf{v})| \\ &\geq 2t + 1 - |w(\mathbf{u}) - w(\mathbf{v})| + |w(\mathbf{u}) - w(\mathbf{v})| \\ &= 2t + 1. \end{aligned}$$

2. The case $N(\mathbf{v}, \mathbf{u}) = 0$. In this case

$$d_U(\mathbf{u}, \mathbf{v}) = d_{Sy}(\mathbf{u}, \mathbf{v}) = w(\mathbf{v}) - w(\mathbf{u}) \geq 2t + 1 - m.$$

Suppose $w(\mathbf{v}) - w(\mathbf{u}) = 2t + 1 - k$ with $1 \leq k \leq \min\{m, 2t+1\}$, then

$$\begin{aligned} k &\leq d_{Sy}(\mathbf{a}_{w(\mathbf{v})}, \mathbf{a}_{w(\mathbf{u})}) \\ &= d_{Sy}(\mathbf{a}_{w(\mathbf{u})+2t+1-k}, \mathbf{a}_{w(\mathbf{u})}) \\ &= d_{Sy}(\mathbf{a}_{w(\mathbf{u})+2t+1-k}, \mathbf{a}_{w(\mathbf{u})+2t}) \\ &\leq |w(\mathbf{u}) + 2t + 1 - k - w(\mathbf{u}) - 2t| \\ &= k - 1 \text{ (contradiction).} \end{aligned}$$

Thus $d_U(\mathbf{u}, \mathbf{v}) = w(\mathbf{v}) - w(\mathbf{u}) \geq 2t + 1$.

Hence C_2 is t-UEC if the $\mathbf{a}_i$ satisfy (5.6) and (5.7).

Note that in applying this construction method on a t-SyEC code C_1 to obtain a t-UEC code C_2, we only have to provide the vectors

$$\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{2t-1}$$

because of the periodicity in the $\mathbf{a}_i$. These vectors satisfy

$$d_{Sy}(\mathbf{a}_j, \mathbf{a}_{j+1}) \leq 1 \text{ for all } 0 \leq j \leq 2t-2 \tag{5.9}$$

and

$$d_{Sy}(\mathbf{a}_{2t-1}, \mathbf{a}_0) \leq 1. \tag{5.10}$$

Hence it is quite natural to define

$$S_i(\mathbf{s}) = \bigcup_{j=0}^{\lfloor (n-m-i)/2t \rfloor} T_{i+2tj}(\mathbf{s}) \tag{5.11}$$

for $i = 0, 1, \ldots, 2t-1$ and $\mathbf{s} \in (GF(2))^m$. With (5.4) it then follows that

$$|C_2| = \sum_{i=0}^{n-m} |T_i(\mathbf{a}_i)| = \sum_{i=0}^{2t-1} |S_i(\mathbf{a}_i)|. \tag{5.12}$$

Examples

Let C_1 be the Golay code of length 23, size 4096, and Hamming distance 7 (see e.g. [21]). This is a linear 3-SyEC code with generator matrix

$$\left[\begin{array}{c|cccccccccc} & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ \mathbf{I}_{12} & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right], \tag{5.13}$$

where $\mathbf{I}_{12}$ denotes the 12×12 identity matrix. Further we choose $m = 2$. The cardinalities of the sets $T_i(\mathbf{s})$ and $S_i(\mathbf{s})$ are now shown in Table 5.1.

i	$\lvert\mathcal{T}_i(00)\rvert$	$\lvert\mathcal{T}_i(10)\rvert$	$\lvert\mathcal{T}_i(01)\rvert$	$\lvert\mathcal{T}_i(11)\rvert$
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	21
6	0	56	56	56
7	120	120	120	0
8	210	0	0	0
9	0	0	0	280
10	0	336	336	336
11	336	336	336	0
12	280	0	0	0
13	0	0	0	210
14	0	120	120	120
15	56	56	56	0
16	21	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	1
$\sum$	1024	1024	1024	1024

i	$\lvert\mathcal{S}_i(00)\rvert$	$\lvert\mathcal{S}_i(10)\rvert$	$\lvert\mathcal{S}_i(01)\rvert$	$\lvert\mathcal{S}_i(11)\rvert$
0	281	56	56	56
1	120	120	120	210
2	210	120	120	120
3	56	56	56	281
4	21	336	336	336
5	336	336	336	21
$\sum$	1024	1024	1024	1024

Table 5.1: Cardinalities of the sets $\mathcal{T}_i(\mathrm{s})$ and $\mathcal{S}_i(\mathrm{s})$ for the Golay code with $m = 2$.

	asymmetric case		unidirectional case		
i	$\mathbf{a}_i$	$\lvert T_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert T_i(\mathbf{a}_i)\rvert$	$\lvert S_i(\mathbf{a}_i)\rvert$
0	00	1	00	1	281
1	00	0	01	0	120
2	00	0	00	0	210
3	00	0	10	0	56
4	01	0	10	0	336
5	11	21	10	0	336
6	10	56	00	0	
7	00	120	01	120	
8	00	210	00	210	
9	10	0	10	0	
10	10	336	10	336	
11	00	336	10	336	
12	00	280	00	280	
13	01	0	01	0	
14	01	120	00	0	
15	01	56	10	56	
16	00	21	10	0	
17	00	0	10	0	
18	00	0	00	0	
19	00	0	01	0	
20	10	0	00	0	
21	11	1	10	0	
$\sum$		1558		1339	1339

Table 5.2: A possible choice for the $\mathbf{a}_i$ in applying the construction method to the Golay code with $m = 2$.

In Table 5.2 we give a possible choice for the $\mathbf{a}_i$ in the asymmetric and a possible choice for the $\mathbf{a}_i$ in the unidirectional case. In this way we obtain a 3-AsEC code of length 21 and size 1558 and a 3-UEC code of length 21 and size 1339, respectively. The largest possible 3-SyEC code of length 21 contains only 1024 codewords.

5.2 Optimization of the cardinality

The size of a code C_2 constructed by the 'expurgating/puncturing' technique from the preceding section depends on the choice for the $\mathbf{a}_i$. In this section we study how to choose the $\mathbf{a}_i$ to obtain the largest code C_2 from a given code C_1 with m fixed.

General case

The problem of finding optimal $\mathbf{a}_i$ in the construction methods can be solved by using a dynamic programming approach. For both the asymmetric and the unidirectional case we will give a shortest-route algorithm similar to Viterbi's decoding algorithm for convolutional codes (see e.g. [23]).

Asymmetric case

In the asymmetric case we have to determine $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{n-m}$ keeping in mind that (5.5) holds. An optimal choice is achieved by the following algorithm:

1. Initially, set

$$\begin{aligned} M_0(\mathbf{s}) &:= |\overline{T}_0(\mathbf{s})| \text{ for all } \mathbf{s} \in (GF(2))^m; \\ \mathbf{b}_0(\mathbf{s}) &:= \mathbf{s} \text{ for all } \mathbf{s} \in (GF(2))^m; \\ j &:= 1. \end{aligned}$$

2. For each $\mathbf{s} \in (GF(2))^m$, find a $\mathbf{t} \in (GF(2))^m$ such that

   (a) $d_{Sy}(\mathbf{s}, \mathbf{t}) \leq 1$;

   (b) $M_{j-1}(\mathbf{t}) = \max\{M_{j-1}(\mathbf{u}) | \mathbf{u} \in (GF(2))^m \wedge d_{Sy}(\mathbf{s}, \mathbf{u}) \leq 1\}$.

Then set

$$\begin{aligned} M_j(\mathbf{s}) &:= M_{j-1}(\mathbf{t}) + |\mathcal{T}_j(\mathbf{s})|; \\ \mathbf{b}_j(\mathbf{s}) &:= (\mathbf{b}_{j-1}(\mathbf{t}), \mathbf{s}) \ (\textit{concatenation}). \end{aligned}$$

3. If $j = n - m$, then go to *step* 4. Otherwise, set $j := j + 1$ and go to *step* 2.

4. Find an $\mathbf{f} \in (GF(2))^m$ such that

$$M_{n-m}(\mathbf{f}) = \max\{M_{n-m}(\mathbf{u}) | \mathbf{u} \in (GF(2))^m\}.$$

The size of the largest code $\mathcal{C}_2$ from $\mathcal{C}_1$ is equal to $M_{n-m}(\mathbf{f})$. This maximum can be obtained in the following way. Let $\mathbf{a}_0$ be equal to the first m coordinates of $\mathbf{b}_{n-m}(\mathbf{f})$, let $\mathbf{a}_1$ be equal to the next m coordinates of $\mathbf{b}_{n-m}(\mathbf{f})$, etc. It can easily checked that

$$2^{-m}|\mathcal{C}_1| \leq M_{n-m}(\mathbf{f}) \leq |\mathcal{C}_1|. \tag{5.14}$$

Note that for $m = 1$ the algorithm can be reduced to simply setting

$$\mathbf{a}_i = \begin{cases} 0 & \text{if } |\mathcal{T}_i(0)| \geq |\mathcal{T}_i(1)| \\ 1 & \text{if } |\mathcal{T}_i(0)| < |\mathcal{T}_i(1)| \end{cases}$$

for $i = 0, 1, \ldots, n - m$.

Unidirectional case

In the unidirectional case we only have to determine $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{2t-1}$ keeping in mind that (5.9) and (5.10) hold. For convenience we define

$$\mathcal{S}_{2t}(\mathbf{s}) = \emptyset \tag{5.15}$$

for all $\mathbf{s} \in (GF(2))^m$ in addition to (5.11). To make sure that (5.9) and (5.10) are satisfied we use an algorithm similar to the one used in the asymmetric case, where we force the path to start ($j = 0$) and finish ($j = 2t$) in the same vector $\mathbf{v} \in (GF(2))^m$. Now considering all possible $\mathbf{v}$ gives the following algorithm:

1. Initially, set

$$\begin{aligned} M_0(\mathbf{v}, \mathbf{v}) &:= |\mathcal{S}_0(\mathbf{v})| \text{ for all } \mathbf{v} \in (GF(2))^m; \\ \mathbf{b}_0(\mathbf{v}, \mathbf{v}) &:= \mathbf{v} \text{ for all } \mathbf{v} \in (GF(2))^m; \\ j &:= 1. \end{aligned}$$

2. For all $\mathbf{s},\mathbf{v} \in (GF(2))^m$ such that $d_{Sy}(\mathbf{s},\mathbf{v}) \leq t - |t-j|$, find a $\mathbf{t} \in (GF(2))^m$ such that

   (a) $d_{Sy}(\mathbf{s},\mathbf{t}) \leq 1$;

   (b) $M_{j-1}(\mathbf{t},\mathbf{v}) = \max\{M_{j-1}(\mathbf{u},\mathbf{v}) | \mathbf{u} \in (GF(2))^m \wedge d_{Sy}(\mathbf{s},\mathbf{u}) \leq 1 \wedge d_{Sy}(\mathbf{u},\mathbf{v}) \leq t - |t-j+1|\}$.

   Then set

$$\begin{aligned} M_j(\mathbf{s},\mathbf{v}) &:= M_{j-1}(\mathbf{t},\mathbf{v}) + |S_j(\mathbf{s})|; \\ \mathbf{b}_j(\mathbf{s},\mathbf{v}) &:= (\mathbf{b}_{j-1}(\mathbf{t},\mathbf{v}),\mathbf{s}) \ (\textit{concatenation}). \end{aligned}$$

3. If $j = 2t$, then go to *step* 4. Otherwise, set $j := j+1$ and go to *step* 2.

4. Find an $\mathbf{f} \in (GF(2))^m$ such that

$$M_{2t}(\mathbf{f},\mathbf{f}) = \max\{M_{2t}(\mathbf{u},\mathbf{u}) | \mathbf{u} \in (GF(2))^m\}.$$

The size of the largest code $\mathcal{C}_2$ from $\mathcal{C}_1$ is equal to $M_{2t}(\mathbf{f},\mathbf{f})$. This maximum can be obtained in the following way. Let $\mathbf{a}_0$ be equal to the first m coordinates of $\mathbf{b}_{2t}(\mathbf{f},\mathbf{f})$, let $\mathbf{a}_1$ be equal to the next m coordinates of $\mathbf{b}_{2t}(\mathbf{f},\mathbf{f})$, etc. It can easily checked that

$$2^{-m}|\mathcal{C}_1| \leq M_{2t}(\mathbf{f},\mathbf{f}) \leq |\mathcal{C}_1|. \tag{5.16}$$

Note that for $m = 1$ the algorithm can be reduced to simply setting

$$\mathbf{a}_i = \begin{cases} 0 & \text{if } |S_i(0)| \geq |S_i(1)| \\ 1 & \text{if } |S_i(0)| < |S_i(1)| \end{cases}$$

for $i = 0, 1, \ldots, 2t-1$.

Examples

Again we choose $m = 2$ and take for $\mathcal{C}_1$ the 3-SyEC Golay code of length 23 and size 4096. For the asymmetric case the values of $M_j(\mathbf{s})$ that follow from the optimization algorithm are given in Table 5.3, while for the unidirectional case the values of $M_j(\mathbf{s},\mathbf{v})$ are given in Table 5.4. In both cases an example of an optimal path is marked by $*$-symbols. Hence we obtain a 3-AsEC code of length 21 and size 1628 and a 3-UEC code of length 21 and size 1474, respectively. The cardinalities of these codes exceed the cardinalities of the codes obtained by choosing the $\mathbf{a}_i$ as in Table 5.2.

j	$M_j(00)$	$M_j(10)$	$M_j(01)$	$M_j(11)$
0	* 1	0	0	0
1	1	1	* 1	0
2	1	1	1	* 1
3	1	1	1	* 1
4	1	1	1	* 1
5	1	1	1	* 22
6	1	78	* 78	78
7	198	198	* 198	78
8	408	198	198	* 198
9	408	408	408	* 478
10	408	* 814	814	814
11	* 1150	1150	1150	814
12	* 1430	1150	1150	1150
13	1430	* 1430	1430	1360
14	1430	* 1550	1550	1550
15	* 1606	1606	1606	1550
16	* 1627	1606	1606	1606
17	1627	1627	* 1627	1606
18	1627	1627	1627	* 1627
19	1627	1627	1627	* 1627
20	1627	1627	1627	* 1627
21	1627	1627	1627	* 1628

Table 5.3: M_j(s) in the asymmetric optimization algorithm for the Golay code with $m = 2$.

j	$M_j(00,00)$	$M_j(10,00)$	$M_j(01,00)$	$M_j(11,00)$
0	* 281	–	–	–
1	401	* 401	401	–
2	611	* 521	521	521
3	667	667	667	* 802
4	688	1138	* 1138	1138
5	1474	1474	* 1474	–
6	* 1474	–	–	–

j	$M_j(00,10)$	$M_j(10,10)$	$M_j(01,10)$	$M_j(11,10)$
0	–	56	–	–
1	176	176	–	266
2	386	386	386	386
3	442	442	442	667
4	463	1003	1003	1003
5	1339	1339	–	1024
6	–	1339	–	–

j	$M_j(00,01)$	$M_j(10,01)$	$M_j(01,01)$	$M_j(11,01)$
0	–	–	56	–
1	176	–	176	266
2	386	386	386	386
3	442	442	442	667
4	463	1003	1003	1003
5	1339	–	1339	1024
6	–	–	1339	–

j	$M_j(00,11)$	$M_j(10,11)$	$M_j(01,11)$	$M_j(11,11)$
0	–	–	–	56
1	–	176	176	266
2	386	386	386	386
3	442	442	442	667
4	463	1003	1003	1003
5	–	1339	1339	1024
6	–	–	–	1339

Table 5.4: $M_j(\mathbf{s}, \mathbf{v})$ in the unidirectional optimization algorithm for the Golay code with $m = 2$.

5.3 Decoding aspects

In this section we study some decoding aspects for the codes C_2 from Section 5.1. If the code C_1 has nice decoding properties, which will often be the case, we can use these to decode C_2.

General case

We assume that a decoding algorithm for C_1 is available. When receiving a vector $\mathbf{y}'$ from $(GF(2))^{n-m}$, we can estimate the codeword $\mathbf{c}'$ from C_2 that was sent as follows:

1. Lengthen the vector $\mathbf{y}'$ with a tail $\mathbf{y}''$ of length m to obtain a vector $\mathbf{y} = (\mathbf{y}', \mathbf{y}'')$ of length n.
2. Apply the decoding algorithm for C_1 on the vector $\mathbf{y}$, which results in a codeword $\mathbf{x}$ from C_1.
3. Delete the last m coordinates of $\mathbf{x}$ to obtain a vector $\mathbf{x}'$, that we will use as an estimation for $\mathbf{c}'$.

As $(\mathbf{c}', \mathbf{a}_{w(\mathbf{c}')})$ is a codeword from C_1 it follows immediately that

$$d_{Sy}((\mathbf{c}', \mathbf{a}_{w(\mathbf{c}')}), (\mathbf{y}', \mathbf{y}'')) \leq t \Longrightarrow \mathbf{x}' = \mathbf{c}'. \tag{5.17}$$

Asymmetric case

When choosing

$$\mathbf{y}'' = \mathbf{a}_j \text{ with } j = \min\{w(\mathbf{y}') + t, n - m\}, \tag{5.18}$$

the preceding decoding procedure always gives the correct codeword in the case that t or less errors occured during transmission over an asymmetric channel. We can prove this as follows. Since the errors are of the asymmetric type we have $\mathbf{c}' \geq \mathbf{y}'$ and $w(\mathbf{c}') - t \leq w(\mathbf{y}') \leq w(\mathbf{c}')$. Hence we have for $w(\mathbf{y}') \leq n - m - t$ that

$$\begin{aligned} & d_{Sy}((\mathbf{c}', \mathbf{a}_{w(\mathbf{c}')}), (\mathbf{y}', \mathbf{y}'')) \\ &= d_{Sy}(\mathbf{c}', \mathbf{y}') + d_{Sy}(\mathbf{a}_{w(\mathbf{c}')}, \mathbf{a}_{w(\mathbf{y}')+t}) \end{aligned}$$

$$\begin{aligned} &\leq w(\mathbf{c}') - w(\mathbf{y}') + 1 \cdot |w(\mathbf{y}') + t - w(\mathbf{c}')| \\ &= w(\mathbf{c}') - w(\mathbf{y}') + w(\mathbf{y}') + t - w(\mathbf{c}') \\ &= t, \end{aligned}$$

and for $w(\mathbf{y}') > n - m - t$ that

$$\begin{aligned} &d_{Sy}((\mathbf{c}', \mathbf{a}_{w(\mathbf{c}')}), (\mathbf{y}', \mathbf{y}'')) \\ &\quad = d_{Sy}(\mathbf{c}', \mathbf{y}') + d_{Sy}(\mathbf{a}_{w(\mathbf{c}')}, \mathbf{a}_{n-m}) \\ &\quad \leq w(\mathbf{c}') - w(\mathbf{y}') + 1 \cdot |n - m - w(\mathbf{c}')| \\ &\quad = w(\mathbf{c}') - w(\mathbf{y}') + n - m - w(\mathbf{c}') \\ &\quad < t. \end{aligned}$$

The assertion now follows from (5.17).

If $\mathbf{x}' \not\geq \mathbf{y}'$, then we can conclude that at least $t + 1$ asymmetric errors must have occured.

Unidirectional case

When choosing

$$\mathbf{y}'' = \mathbf{a}_j \text{ with } j = w(\mathbf{y}') + t - \lfloor (w(\mathbf{y}') + t)/2t \rfloor \cdot 2t, \tag{5.19}$$

the preceding decoding procedure always gives the correct codeword in the case that t or less errors occured during transmission over a unidirectional channel. We can prove this as follows. Remember that

$$\mathbf{a}_j = \mathbf{a}_{j-2t} \text{ for all } 2t \leq j \leq n - m + 2t.$$

We distinguish between two cases: $\mathbf{c}' \geq \mathbf{y}'$ and $\mathbf{y}' \geq \mathbf{c}'$.

1. The case $\mathbf{c}' \geq \mathbf{y}'$. In this case $w(\mathbf{c}') - t \leq w(\mathbf{y}') \leq w(\mathbf{c}')$. Hence

$$\begin{aligned} &d_{Sy}((\mathbf{c}', \mathbf{a}_{w(\mathbf{c}')}), (\mathbf{y}', \mathbf{y}'')) \\ &\quad = d_{Sy}(\mathbf{c}', \mathbf{y}') + d_{Sy}(\mathbf{a}_{w(\mathbf{c}')}, \mathbf{a}_j) \\ &\quad = d_{Sy}(\mathbf{c}', \mathbf{y}') + d_{Sy}(\mathbf{a}_{w(\mathbf{c}')}, \mathbf{a}_{w(\mathbf{y}')+t}) \\ &\quad \leq w(\mathbf{c}') - w(\mathbf{y}') + 1 \cdot |w(\mathbf{y}') + t - w(\mathbf{c}')| \\ &\quad = w(\mathbf{c}') - w(\mathbf{y}') + w(\mathbf{y}') + t - w(\mathbf{c}') \\ &\quad = t. \end{aligned}$$

2. The case $\mathbf{y}' \geq \mathbf{c}'$. In this case $w(\mathbf{c}') \leq w(\mathbf{y}') \leq w(\mathbf{c}') + t$. Hence

$$\begin{aligned} & d_{Sy}((\mathbf{c}', \mathbf{a}_{w(\mathbf{c}')}), (\mathbf{y}', \mathbf{y}'')) \\ &= d_{Sy}(\mathbf{c}', \mathbf{y}') + d_{Sy}(\mathbf{a}_{w(\mathbf{c}')}, \mathbf{a}_j) \\ &= d_{Sy}(\mathbf{c}', \mathbf{y}') + d_{Sy}(\mathbf{a}_{w(\mathbf{c}')+2t}, \mathbf{a}_{w(\mathbf{y}')+t}) \\ &\leq w(\mathbf{y}') - w(\mathbf{c}') + 1 \cdot |w(\mathbf{c}') + 2t - w(\mathbf{y}') - t| \\ &= w(\mathbf{y}') - w(\mathbf{c}') + w(\mathbf{c}') + t - w(\mathbf{y}') \\ &= t. \end{aligned}$$

The assertion now follows from (5.17).

If $\mathbf{x}' \not\geq \mathbf{y}'$ and $\mathbf{y}' \not\geq \mathbf{x}'$, then we can conclude that at least $t + 1$ unidirectional errors must have occured.

Examples

Once again we choose $m = 2$ and take for $\mathcal{C}_1$ the 3-SyEC Golay code of length 23 and size 4096 with generator matrix as shown in (5.13). A decoding algorithm for this code can be found in [27]. We choose the $\mathbf{a}_i$ as shown in Table 5.2. The code $\mathcal{C}_2$ of length 21 obtained in this way is 3-AsEC and of size 1558 in the asymmetric case and 3-UEC and of size 1339 in the unidirectional case. We consider various examples for the received vector $\mathbf{y}'$.

1. The asymmetric case.

   (a)

$$\begin{aligned} \mathbf{y}' &= 100000000000110000000 \Rightarrow \mathbf{y}'' = \mathbf{a}_{3+3} = 10 \Rightarrow \\ \mathbf{y} &= 100000000000011000000010 \Rightarrow \\ \mathbf{x} &= 100000000000011011100010 \Rightarrow \\ \mathbf{x}' &= 10000000000110111000 \end{aligned}$$

   (b)

$$\begin{aligned} \mathbf{y}' &= 000100000000010100111 \Rightarrow \mathbf{y}'' = \mathbf{a}_{6+3} = 10 \Rightarrow \\ \mathbf{y} &= 000100000000001010011110 \Rightarrow \\ \mathbf{x} &= 000100000000001011011100 \Rightarrow \\ \mathbf{x}' &= 00010000000010110111 \end{aligned}$$

(c)

$$\begin{aligned} \mathbf{y}' &= 100000000001111111111 \Rightarrow \mathbf{y}'' = \mathbf{a}_{11+3} = 01 \Rightarrow \\ \mathbf{y} &= 100000000001111111111101 \Rightarrow \\ \mathbf{x} &= 000000000001111111111111 \Rightarrow \\ \mathbf{x}' &= 00000000001111111111 \end{aligned}$$

Since $\mathbf{x}' \not\geq \mathbf{y}'$ we conclude that at least 4 asymmetric errors have occured.

2. The unidirectional case.

(a)

$$\begin{aligned} \mathbf{y}' &= 0110000100001101100 10 \Rightarrow \mathbf{y}'' = \mathbf{a}_{8+3-6} = 10 \Rightarrow \\ \mathbf{y} &= 011000010000110110010 10 \Rightarrow \\ \mathbf{x} &= 011000000000110110010 01 \Rightarrow \\ \mathbf{x}' &= 01100000000110110010 \end{aligned}$$

(b)

$$\begin{aligned} \mathbf{y}' &= 1111100000000001101011 \Rightarrow \mathbf{y}'' = \mathbf{a}_{10+3-12} = 01 \Rightarrow \\ \mathbf{y} &= 1111100000000001101011 01 \Rightarrow \\ \mathbf{x} &= 0101000000000001101011 01 \Rightarrow \\ \mathbf{x}' &= 010100000000001101011 \end{aligned}$$

(c)

$$\begin{aligned} \mathbf{y}' &= 0000000010101100100 11 \Rightarrow \mathbf{y}'' = \mathbf{a}_{7+3-6} = 10 \Rightarrow \\ \mathbf{y} &= 000000001010110010011 10 \Rightarrow \\ \mathbf{x} &= 000000000110110010011 10 \Rightarrow \\ \mathbf{x}' &= 00000000110110010011 \end{aligned}$$

Since $\mathbf{x}' \not\geq \mathbf{y}'$ and $\mathbf{y}' \not\geq \mathbf{x}'$ we can conclude that at least 4 unidirectional errors have occured.

5.4 *Retrospect and prospect*

In this chapter we have given a construction method in which t-AsEC and t-UEC codes can be obtained by expurgating and puncturing an *initial* t-SyEC code. The method itself has been described in Section 5.1. Shortest-route algorithms to optimize the cardinalities of the codes obtained by the method have been presented in Section 5.2. Finally, for decoding the codes it has been shown in Section 5.3 that we can use a decoding algorithm for the initial t-SyEC code with some small extra operations.

In the next chapter we will use this method to obtain t-AsEC and t-UEC codes of length n that contain more codewords than the largest known t-SyEC code of length n. We will restrict ourselves to the area $n \leq 23$ and $t \leq 4$.

Chapter 6

Constructions

6.1 Trial and error

In this chapter we try to construct AsEC and UEC codes that improve comparable SyEC codes in size. To this end we will use the construction method from the previous chapter. Further we will also use some *trial and error* techniques. These techniques are described in this first section. Then in Section 6.2 we construct t-AsEC codes of length n, while in Section 6.3 we construct t-UEC codes of length n. In both cases we focus on the area $n \leq 23$ and $t \leq 4$, since this is the area covered by most tables on the size of optimal t-SyEC of length n (see e.g. [21,29]).

A first trial and error technique concerns the actual construction of codes. From the constraints on the weight distribution of an AsEC or UEC code as presented in Chapter 3 we can get some idea of how a code whose size approximates the corresponding integer programming bound should look. For example, it follows from Theorem 3.2 $(s = 0, i = 3,4,5,6; s = 1, i = 4,6; s = 2, i = 2,9)$ that for a 2-AsEC code of length 9 the weight distribution satisfies $A_3 \leq 3$, $A_4 \leq 3$, $A_5 \leq 3$, $A_6 \leq 3$, $A_3 + A_4 \leq 5$, $A_5 + A_6 \leq 5$, $A_0 + A_1 + A_2 \leq 1$, and $A_7 + A_8 + A_9 \leq 1$. Hence $A_{As}(9,2) \leq 1 + 5 + 5 + 1 = 12$. If a 2-AsEC code of length 9 and size 12 actually exists, then it must hold that $A_0 + A_1 + A_2 = 1$, $A_3 + A_4 = 5$, $A_5 + A_6 = 5$, and $A_7 + A_8 + A_9 = 1$. From $A_3 + A_4 = 5$ it follows that $A_3 = 3, A_4 = 2$ or $A_3 = 2, A_4 = 3$. In the former case we can assume without loss of generality that the codewords of weight 3 look like the following vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$:

$$\begin{aligned} \mathbf{v}_1 &= 111000000 \\ \mathbf{v}_2 &= 000111000 \\ \mathbf{v}_3 &= 000000111. \end{aligned}$$

Now it is easy to see that no vector $\mathbf{v}_4$ of weight 4 can be chosen such that $d_{As}(\mathbf{v}_4, \mathbf{v}_j) \geq 5$ for all $j = 1, 2, 3$. Hence $A_3 = 2$ and $A_4 = 3$, which gives without loss of generality:

$$\begin{aligned} \mathbf{v}_1 &= 111000000 \\ \mathbf{v}_2 &= 000111000 \\ \mathbf{v}_3 &= 100100011 \\ \mathbf{v}_4 &= 010010101 \\ \mathbf{v}_5 &= 001001110. \end{aligned}$$

By using similar arguments for the codewords of weight 5 and 6, and by watching the distances between the weight ≤ 4 and weight ≥ 5 codewords, it follows that we can add the following vectors to the code:

$$\begin{aligned} \mathbf{v}_6 &= 000111111 \\ \mathbf{v}_7 &= 111000111 \\ \mathbf{v}_8 &= 011110010 \\ \mathbf{v}_9 &= 101011001 \\ \mathbf{v}_{10} &= 110101100. \end{aligned}$$

Finally adding the all-zero and all-one vectors of length 9 gives the desired 2-AsEC code of length 9 and size 12.

Another way of applying trial and error techniques is trying to add vectors to an existing t-AsEC or t-UEC code without disturbing $d_{As} \geq 2t + 1$ or $d_U \geq 2t + 1$.

A third technique concerns the transposing of coordinates in the initial code in the construction method from the previous chapter. Although equivalent codes have the same properties in many respects, it often does matter which coordinates form the last m positions when applying the construction method.

6.2 Some AsEC codes

For single asymmetric error correcting (1-AsEC) codes various construction methods are available. An overview has been given in [18]. For the 1-AsEC Constantin-Rao codes of length n (see e.g. [7]) it has been proved that their size equals at least $2^n/(n+1)$, which is also the Hamming *upper* bound (see Theorem 2.1) on the size of 1-SyEC codes of length n. Hence a 1-AsEC code of length n improving the largest 1-SyEC code of length n can be obtained for all n for which the Hamming bound is not sharp, i.e. $n \neq 2^j - 1$. Also for some $n = 2^j - 1$ better codes are known, for example Delsarte and Piret's 1-AsEC code of length 7 and size 18 (see [8]).

The theory of multiple asymmetric error correcting codes is less well developed. Therefore we will now study 2-AsEC, 3-AsEC, and 4-AsEC codes. In [13] Helgesen constructed codes of sizes up to 12 correcting almost any fixed number of asymmetric errors and having minimal block length. In the context of this section this means that he has found optimal codes of length less than 10 for $t = 2$, less than 13 for $t = 3$, and less than 16 for $t = 4$. We too give some codes having the same parameters (length, size, asymmetric error correcting capability) as the corresponding Helgesen codes. These codes were found independently and are listed in this thesis for the sake of completeness since Helgesen's results have remained unpublished thus far.

6.2.1 2-AsEC codes

In [8] Delsarte and Piret presented some 2-AsEC codes of length less than 15 obtained by expurgation and puncturing techniques. We now give 2-AsEC codes of length $n = 9, 12$ improving their codes, and besides we also give 2-AsEC codes of length $15 \leq n \leq 23$.

Length 9

In [18] it was stated that Delsarte and Piret's code $\mathcal{C}_9$ of length 9 and size 12 ([8]) that was presented as being 2-AsEC does *not* correct up to 2 asymmetric errors since the d_{A_s} distance between the codewords 6 and 8 is only 4. Correct 2-AsEC codes of length 9 and size 12 have been derived by Helgesen ([13]) and in Section 6.1 of this thesis. Table D.1 once again lists the codewords of the latter code.

Length 12

In [8] Delsarte and Piret constructed a 2-AsEC code $\mathcal{C}_{13}$ of length 13 and size 98. They stated that a column i $(1 \leq i \leq 13)$ exists in which the symbol 0 appears for 50 vectors. Hence a code $\mathcal{C}_{12}$ of length 12 and size 50 could be obtained by taking all the codewords having coordinate i equal to 0 and then deleting column i. Further investigation of $\mathcal{C}_{13}$ leads to the conclusion that there are 12 columns in which the symbol 0 appears for 50 vectors and 1 column j $(1 \leq j \leq 13)$ in which the symbol 0 appears for 45 vectors. Hence we can construct a larger code of length 12 by taking all the vectors having coordinate j equal to 1 and then deleting column j. Further, we can add the all-zero vector to this code while keeping the code 2-AsEC. Hence we have constructed a 2-AsEC code of length 12 and size $98 - 45 + 1 = 54$.

Lengths 15,16,17,18

Sloane *et al.* ([35]) constructed 2-SyEC codes of length 19 and size 2048 by dividing the Hamming code of length 15, size 2048, and Hamming distance 3 into eight cosets of the Preparata code of length 15, size 256, and Hamming distance 5, and then attaching to each coset a different codeword of the even-weight code of length 4, size 8, and Hamming distance 2. Let $\mathcal{S}_{19}$ be a code obtained in this way with $\mathbf{0} \in \mathcal{S}_{19}$. By applying the asymmetric construction method with $\mathcal{C}_1 = \mathcal{S}_{19}$, we obtain a 2-AsEC code of length 18 and size 1217 if $m = 1$, of length 17 and size 647 if $m = 2$, of length 16 and size 364 if $m = 3$, and of length 15 and size 266 if $m = 4$. Table E.1 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

It can easily be seen from Table E.1 that we can add the all-one vector to the code of length 18 while keeping this code 2-AsEC. Hence we have a 2-AsEC code length 18 and size 1218.

Lengths 19,20

Sloane *et al.* ([35]) also constructed 2-SyEC codes of length 20 and size 2560 by considering five cosets of the Preparata code (length 15, size 256, Hamming distance 5) in the Hamming code (length 15, size 2048, Hamming distance 3) and five cosets of the repetition code (length 5, size

2, Hamming distance 5) in the code

$$\{\ 00000, 11000, 10100, 10010, 10001,\\ 11111, 00111, 01011, 01101, 01110\ \}$$

(length 5, size 10, Hamming distance 2), and then concatenating the vectors of coset i of the Preparata code in every possible way with the vectors of coset i of the repetition code ($1 \leq i \leq 5$). Let S_{20} be a code obtained in this way with $\mathbf{0} \in S_{20}$. The weight distribution of S_{20} as well as the weight distribution of S_{19} are given in Table 6.1.

Since S_{19} contains no codewords of weight $1, 2, 3, 4$ it follows that it holds for all vectors $\mathbf{a} \in (GF(2))^{19}$ of weight 3 and codewords $\mathbf{c} \in S_{19}$ that $d_{As}(\mathbf{a}, \mathbf{c}) \geq 6$ if $w(\mathbf{c}) \neq 5$ and that $d_{As}(\mathbf{a}, \mathbf{c}) \geq 4$ if $w(\mathbf{c}) = 5$. This distance equals 4 if and only if $w(\mathbf{c}) = 5$ and $\mathbf{c} \geq \mathbf{a}$. Each $\mathbf{c} \in S_{19}$ of weight 5 covers exactly 10 vectors of weight 3, and no vector of weight 3 is covered by two or more codewords of weight 5. Therefore,

$$|\{\mathbf{a} \in (GF(2))^{19} | w(\mathbf{a}) = 3 \wedge d_{As}(\mathbf{a}, \mathbf{c}) \geq 6\ \forall\ \mathbf{c} \in S_{19}\}|\\ = \begin{pmatrix} 19 \\ 3 \end{pmatrix} - 72 \begin{pmatrix} 5 \\ 3 \end{pmatrix} = 249 > 0.$$

Hence we can add at least one vector of weight 3 (which is not covered by a codeword of weight 5) to S_{19} while keeping the code 2-AsEC. Furthermore we can also add the all-one vector while keeping the code 2-AsEC. In this way we can construct a 2-AsEC code of length 19 containing at least $2048 + 2 = 2050$ codewords.

Considering the weight distribution of S_{20}, we note that

$$|\{\mathbf{a} \in (GF(2))^{20} | w(\mathbf{a}) = 3 \wedge d_{As}(\mathbf{a}, \mathbf{c}) \geq 6\ \forall\ \mathbf{c} \in S_{20}\}|\\ = \begin{pmatrix} 20 \\ 3 \end{pmatrix} - 63 \begin{pmatrix} 5 \\ 3 \end{pmatrix} = 510 > 0.$$

Hence we can add a vector $\mathbf{v}$ of weight 3 to S_{20} while keeping the code 2-AsEC. Furthermore,

$$|\{\mathbf{a} \in (GF(2))^{20} | w(\mathbf{a}) = 3 \wedge d_{As}(\mathbf{a}, \mathbf{v}) \geq 6 \wedge d_{As}(\mathbf{a}, \mathbf{c}) \geq 6\ \forall\ \mathbf{c} \in S_{20}\}|\\ \geq \begin{pmatrix} 17 \\ 3 \end{pmatrix} - 63 \begin{pmatrix} 5 \\ 3 \end{pmatrix} = 50 > 0.$$

Hence we can add at least one other vector of weight 3 while keeping the code 2-AsEC. We can also add the complements of these two vectors while

i	$\mathcal{S}_{19}$ A_i	$\mathcal{S}_{20}$ A_i	$\mathcal{W}_{23}$ A_i
0	1	1	1
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	72	63	84
6	160	150	252
7	128	147	445
8	210	207	890
9	448	430	1620
10	432	564	2268
11	240	430	2632
12	168	207	2632
13	120	147	2268
14	48	150	1620
15	16	63	890
16	5	0	445
17	0	0	252
18	0	0	84
19	0	0	0
20		1	0
21			0
22			0
23			1
$\sum$	2048	2560	16384

Table 6.1: Weight distribution of the 2-SyEC codes $\mathcal{S}_{19}$, $\mathcal{S}_{20}$, and $\mathcal{W}_{23}$.

keeping the code 2-AsEC. In this way we can construct a 2-AsEC code of length 20 containing at least $2560 + 4 = 2564$ codewords.

Of course, these are very minor improvements. However, we can conclude from the foregoing that the codes S_{19} and S_{20} which might be optimal in the sense of having the largest number of codewords in a 2-SyEC code of length 19 (20), are certainly *not* optimal in the sense of having the largest number of codewords in a 2-AsEC code of length 19 (20).

Lengths 21,22

Wagner ([42]) constructed a 2-SyEC linear code $\mathcal{W}_{23}$ of length 23 and dimension 14. We consider a code $\mathcal{W}'_{23}$ which is equivalent to $\mathcal{W}$. This code $\mathcal{W}'_{23}$ has generator matrix

$$\left[\begin{array}{c|c} & 111000011 \\ & 100110011 \\ & 101010110 \\ & 010111001 \\ & 100001011 \\ & 010100011 \\ & 001000111 \\ \mathbf{I}_{14} & 100100110 \\ & 010001101 \\ & 101011000 \\ & 110110100 \\ & 111101010 \\ & 011011111 \\ & 101111101 \end{array}\right]. \tag{6.1}$$

By applying the asymmetric construction method with $C_1 = \mathcal{W}'_{23}$, we obtain a 2-AsEC code of length 22 and size 8450 if $m = 1$, and of length 21 and size 4251 if $m = 2$. Table E.2 shows a possible choice for the $\mathbf{a}_i$ that yields these results. When the construction method is applied on the Wagner code $\mathcal{W}$ itself, with generator matrix as stated in [42] or [6], the resulting 2-AsEC codes only have sizes 8322 and 4200, respectively.

Length 23

It follows from the weight distribution of the Wagner code $\mathcal{W}_{23}$ as shown in Table 6.1 (see [6]), that this code can be enlarged in a way similar to that

in which $\mathcal{S}_{20}$ was enlarged. This gives a 2-AsEC code of length 23 containing at least $2^{14} + 4 = 16388$ codewords, which shows that the Wagner code, that might be an optimal 2-SyEC code of length 23, is certainly *not* an optimal 2-AsEC code of length 23.

6.2.2 3-AsEC codes

We now give 3-AsEC codes of length $n = 11$ and $13 \leq n \leq 21$.

Lengths 11,13,14

Table D.2 lists the codewords of 3-AsEC codes of length 11 and size 8, of length 13 and size 18, and of length 14 and size 30, that have been found by trial and error.

Length 15

Let $\mathcal{B}_{15}$ be the linear 3-SyEC BCH code of length 15 and dimension 5, with generator matrix

$$\begin{bmatrix} 111011001010000 \\ 011101100101000 \\ 001110110010100 \\ 000111011001010 \\ 000011101100101 \end{bmatrix}. \tag{6.2}$$

We can add the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_6$ from Table F.1 together with their complements to this code $\mathcal{B}_{15}$ while keeping the code 3-AsEC. Thus we have constructed a 3-AsEC code of length 15 and size $32 + 12 = 44$.

Lengths 16,17,18,19,20,21

Let $\mathcal{G}_{23}$ be the linear 3-SyEC Golay code of length 23 and dimension 12, with generator matrix as shown in (5.13). By applying the asymmetric construction method with $\mathcal{C}_1 = \mathcal{G}_{23}$, we obtain a 3-AsEC code of length 21 and size 1628 if $m = 2$, of length 20 and size 860 if $m = 3$, of length 19 and size 450 if $m = 4$, of length 18 and size 234 if $m = 5$, of length 17 and size 122 if $m = 6$, and of length 16 and size 66 if $m = 7$. Tables E.3 and E.4 show a possible choice for the $\mathbf{a}_i$ that yields these results.

Length 23

Trying to enlarge the code $\mathcal{G}_{23}$ in a way similar to that in which $\mathcal{S}_{19}$, $\mathcal{S}_{20}$, and $\mathcal{W}_{23}$ were enlarged, we note that

$$|\{\mathbf{a} \in (GF(2))^{23} | w(\mathbf{a}) = 4 \wedge d_{A_s}(\mathbf{a}, \mathbf{c}) \geq 8 \; \forall \; \mathbf{c} \in \mathcal{G}_{23}\}|$$
$$= \begin{pmatrix} 23 \\ 4 \end{pmatrix} - A_7 \begin{pmatrix} 7 \\ 4 \end{pmatrix} = 8855 - 253 \cdot 35 = 0.$$

Hence we cannot add a vector of weight 4 to $\mathcal{G}_{23}$ while keeping the code 3-AsEC. A question which is still unresolved is whether $\mathcal{G}_{23}$, which is optimal in the sense of having the largest number of codewords in a 3-SyEC code of length 23, is also optimal in the sense of having the largest number of codewords in a 3-AsEC code of length 23.

6.2.3 4-AsEC codes

We next give 4-AsEC codes of length $13 \leq n \leq 23$.

Lengths 13,14,15,16

Tables D.3 and D.4 list the codewords of 4-AsEC codes of length 13 and size 6, of length 14 and size 8, of length 15 and size 12, and of length 16 and size 16 that have been found by trial and error.

Lengths 17,18,19

Let $\mathcal{L}_{19}$ be the 4-SyEC code of length 19 and size 40 that contains the all-zero vector, the all-one vector, and the 19 cyclic shifts of both

$$\mathbf{b} = 1100111101010000110$$

and its complement $\overline{\mathbf{b}}$. By applying the asymmetric construction method with $\mathcal{C}_1 = \mathcal{L}_{19}$, we obtain a 4-AsEC code $\mathcal{L}_{18}$ of length 18 and size 30 if $m = 1$, and a 4-AsEC code $\mathcal{L}_{17}$ of length 17 and size 20 if $m = 2$. Table E.5 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

Table F.3 lists vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_6$ that can be added to $\mathcal{L}_{17}$ while keeping this code 4-AsEC. Thus we have constructed a 4-AsEC code of length 17 and size $20 + 6 = 26$.

Table F.3 also lists vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ that can be added together with their complements to $\mathcal{L}_{18}$ or to the code $\mathcal{L}_{19}$ while keeping these codes

4-AsEC. Thus we have constructed 4-AsEC codes of length 18 and size $30 + 6 = 36$ and of length 19 and size $40 + 6 = 46$.

Lengths 20,21

By lengthening all codewords from $\mathcal{L}_{19}$ of odd weight with a 1 and all codewords of even weight with a 0 we obtain a code $\mathcal{L}_{20}$ of length 20, size 40, and Hamming distance 10.

By taking the code of length 23, size 48, and Hamming distance 11 that contains the all-zero vector, the all-one vector, and the 23 cyclic shifts of both

$$\mathbf{b} = 11111010110011001010000$$

and its complement $\overline{\mathbf{b}}$, and then deleting the last two columns of this code, we obtain a code $\mathcal{L}_{21}$ of length 21, size 48, and Hamming distance 9.

Table F.3 lists vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_7$ that can be added together with their complements to $\mathcal{L}_{20}$ or $\mathcal{L}_{21}$ while keeping these codes 4-AsEC. Thus we have constructed 4-AsEC codes of length 20 and size $40 + 14 = 54$ and of length 21 and size $48 + 14 = 62$.

Lengths 22,23

In [36] van Tilborg constructed a linear code of length 24, dimension 7, and Hamming distance 10 with generator matrix

$$\begin{bmatrix} 001101101011011001000000 \\ 101010111111101101100000 \\ 010100011110100111110000 \\ 000111101101010010111000 \\ 000010010101111000011100 \\ 001001001011100100001110 \\ 000001101110110011000111 \end{bmatrix}. \tag{6.3}$$

Deleting the last column of this code gives a code $\mathcal{T}_{23}$ of length 23, dimension 7, and Hamming distance 9.

By applying the asymmetric construction method with $\mathcal{C}_1 = \mathcal{T}_{23}$, we obtain a 4-AsEC code $\mathcal{T}_{22}$ of length 22 and size 83 if $m = 1$. Table E.5 shows a possible choice for the $\mathbf{a}_i$ that yields this result.

Table F.4 lists vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_5$ that can be added to $\mathcal{T}_{22}$ or $\mathcal{T}_{23}$ while keeping these codes 4-AsEC. Thus we have constructed a 4-AsEC codes of length 22 and size $83+5 = 88$ and of length 23 and size $2^7+5 = 133$.

6.3 Some UEC codes

Since any 1-UEC code is also 1-SyEC code and vice versa, we use the well-studied single symmetric error correcting codes also for single unidirectional error correction.

The theory of multiple unidirectional error correcting codes is less well developed. Therefore we will now study 2-UEC, 3-UEC, and 4-UEC codes.

6.3.1 2-UEC codes

We now give 2-UEC codes of length $8 \leq n \leq 14$, $16 \leq n \leq 18$, and $21 \leq n \leq 22$.

Lengths 8,9

Table D.5 lists the codewords of 2-UEC codes of length 8 and size 6 and of length 9 and size 10, that have been found by trial and error.

Length 10

Let $\mathcal{L}_{11}$ be the 2-SyEC code of length 11 and size 24 that contains the all-zero vector, the all-one vector, and the 11 cyclic shifts of both

$$\mathbf{b} = 11011100010$$

and its complement $\overline{\mathbf{b}}$. By applying the unidirectional construction method with $\mathcal{C}_1 = \mathcal{L}_{11}$, we obtain a 2-UEC code of length 10 and size 16 if $m = 1$. Table E.6 shows a possible choice for the $\mathbf{a}_i$ that yields this result.

Lengths 11,12,13,14

Let $\mathcal{R}_{16}$ be the linear Reed-Muller code of length 16, dimension 5, and Hamming distance 8, with generator matrix

$$\begin{bmatrix} 1111111111111111 \\ 0000000011111111 \\ 0000111100001111 \\ 0011001100110011 \\ 0101010101010101 \end{bmatrix}. \tag{6.4}$$

We can now construct the Nordstrom-Robinson code $\mathcal{N}_{16}$ of length 16, size 256, and Hamming distance 6, by shifting $\mathcal{R}_{16}$ over the following eight vectors $\mathbf{b}_j$ ($\mathcal{N}_{16} = \cup_{j=0}^{7}\{\mathbf{c} + \mathbf{b}_j | \mathbf{c} \in \mathcal{R}_{16}\}$):

$$\begin{aligned}
\mathbf{b}_0 &= 0000000000000000 \\
\mathbf{b}_1 &= 0011010110010000 \\
\mathbf{b}_2 &= 0101100100110000 \\
\mathbf{b}_3 &= 1001001101010000 \\
\mathbf{b}_4 &= 1000010000101110 \\
\mathbf{b}_5 &= 0100001010001110 \\
\mathbf{b}_6 &= 0010100001001110 \\
\mathbf{b}_7 &= 0001000100011110.
\end{aligned}$$

Deleting the last column of $\mathcal{N}_{16}$ we obtain a code $\mathcal{N}_{15}$ of length 15, size 256, and Hamming distance 5. The weight distributions of $\mathcal{N}_{15}$, $\mathcal{N}_{16}$, and $\mathcal{R}_{16}$ are shown in Table 6.2.

We now apply the unidirectional construction method with $\mathcal{C}_1 = \mathcal{N}_{15}$ and $m = 1$. The cardinalities of $\mathcal{T}_i(0)$ and $\mathcal{T}_i(1)$ are also shown in Table 6.2. Choosing $\mathbf{a}_0 = 1, \mathbf{a}_1 = 0, \mathbf{a}_2 = 0, \mathbf{a}_3 = 0$ gives a code $\mathcal{C}_2'$ containing 176 codewords. Choosing $\mathbf{a}_0 = 1, \mathbf{a}_1 = 0, \mathbf{a}_2 = 0, \mathbf{a}_3 = 1$ gives a code $\mathcal{C}_2''$ containing also 176 codewords. Note that the Hamming distance of any two distinct codewords of weight 8 in $\mathcal{N}_{16}$ is at least 8, since these codewords are also contained in $\mathcal{R}_{16}$. From Table 6.2 it follows that $\mathcal{N}_{16}$ contains no codewords of weight 7 or 9. Hence the Hamming distance of any two distinct codewords of weight 7 or 8 in $\mathcal{N}_{15}$ is at least 7. We can now easily check that $\mathcal{C}_2' \cup \mathcal{C}_2''$ is a 2-UEC code $\mathcal{N}_{14}$ of length 14 and size $|\mathcal{C}_2'| + |\mathcal{C}_2'' \setminus \mathcal{C}_2'| = |\mathcal{C}_2'| + |\mathcal{T}_7(1)| = 176 + 8 = 184$.

Let $\mathbf{u} = (\mathbf{u}_m', \mathbf{u}_m'')$ denote the $(14-m, m)$ partition of any $\mathbf{u} \in (GF(2))^{14}$ for $m = 1, 2, \ldots, 5$. We define

$$\mathcal{N}_m(\mathbf{s}) = \{\mathbf{c}_m' | \mathbf{c} \in \mathcal{N}_{14} \wedge \mathbf{c}_m'' = \mathbf{s}\}$$

for $\mathbf{s} \in (GF(2))^m$ and $m = 1, 2, \ldots 5$. In Table 6.3 we give the cardinality of $\mathcal{N}_5(\mathbf{s})$ for all $\mathbf{s} \in (GF(2))^5$. From this table it follows that $\mathcal{N}_3(010)$ is a 2-UEC code of length 11 and size 26, $\mathcal{N}_2(10)$ is a 2-UEC code of length 12 and size 52, and $\mathcal{N}_1(0)$ is a 2-UEC code of length 13 and size 92.

Lengths 16,17,18

By applying the unidirectional construction method with $\mathcal{C}_1 = \mathcal{S}_{19}$ (see

	$\mathcal{R}_{16}$	$\mathcal{N}_{16}$	$\mathcal{N}_{15}$		
i	A_i	A_i	A_i	$\lvert\mathcal{T}_i(0)\rvert$	$\lvert\mathcal{T}_i(1)\rvert$
0	1	1	1	1	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	14
5	0	0	42	28	28
6	0	112	70	42	7
7	0	0	15	8	8
8	30	30	15	7	42
9	0	0	70	28	28
10	0	112	42	14	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	1
15	0	0	1		
16	1	1			
$\sum$	32	256	256	128	128

Table 6.2: Weight distribution of the codes $\mathcal{R}_{16}$, $\mathcal{N}_{16}$, and $\mathcal{N}_{15}$.

Subsection 6.2.1), we obtain a 2-UEC code of length 18 and size 1216 if $m = 1$, of length 17 and size 640 if $m = 2$, and of length 16 and size 352 if $m = 3$. Table E.6 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

Lengths 21,22

By applying the unidirectional construction method with $C_1 = \mathcal{W}'_{23}$ (see Subsection 6.2.1), we obtain a 2-UEC code of length 22 and size 8448 if $m = 1$, and of length 21 and size 4224 if $m = 2$. Table E.7 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

6.3.2 3-UEC codes

We now give 3-UEC codes of length $11 \leq n \leq 14$ and $16 \leq n \leq 22$.

Lengths 11,12

Table D.6 lists the codewords of 3-UEC codes of length 11 and size 7 and of length 12 and size 10, that have been found by trial and error.

Lengths 13,14

By applying the unidirectional construction method with $C_1 = \mathcal{B}_{15}$ (see Subsection 6.2.2), we obtain a 3-UEC code of length 14 and size 22 if $m = 1$, and of length 13 and size 14 if $m = 2$. Table E.8 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

Lengths 16,17,18,19,20,21,22

We consider a code $\mathcal{G}'_{23}$ which is equivalent to the Golay code $\mathcal{G}_{23}$ of length 23, dimension 12, and Hamming distance 7. This code $\mathcal{G}'_{23}$ has

s	$\lvert\mathcal{N}_5(s)\rvert$	$\lvert\mathcal{G}_5(s)\rvert$
00000	4	35
10000	4	47
01000	6	47
11000	6	44
00100	6	47
10100	6	44
01100	4	44
11100	4	53
00010	6	38
10010	6	53
01010	8	53
11010	6	44
00110	6	53
10110	8	44
01110	6	44
11110	6	47
00001	6	47
10001	6	44
01001	6	44
11001	8	53
00101	8	44
10101	6	53
01101	6	53
11101	6	38
00011	4	53
10011	4	44
01011	6	44
11011	6	47
00111	6	44
10111	6	47
01111	4	47
11111	4	35
$\sum$	184	1474

Table 6.3: Cardinalities of $\mathcal{N}_5(s)$ and $\mathcal{G}_5(s)$.

generator matrix

$$\left[\begin{array}{c|c} & 11011100010 \\ & 01110110001 \\ & 10101111000 \\ & 01011011100 \\ & 00110101110 \\ \mathbf{I}_{12} & 00001110111 \\ & 10010011011 \\ & 11000101101 \\ & 11100010110 \\ & 01101001011 \\ & 10111000101 \\ & 11111111111 \end{array}\right]. \tag{6.5}$$

By applying the unidirectional construction method with $\mathcal{C}_1 = \mathcal{G}'_{23}$, we obtain a 3-UEC code $\mathcal{G}'_{22}$ of length 22 and size 2588 if $m = 1$, and a 3-UEC code $\mathcal{G}'_{21}$ of length 21 and size 1474 if $m = 2$. Table E.8 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

Let $\mathbf{u} = (\mathbf{u}'_m, \mathbf{u}''_m)$ denote the $(21-m, m)$ partition of any $\mathbf{u} \in (GF(2))^{21}$ for $m = 1, 2, \ldots, 5$. We define

$$\mathcal{G}_m(\mathbf{s}) = \{\mathbf{c}'_m | \mathbf{c} \in \mathcal{G}'_{21} \wedge \mathbf{c}''_m = \mathbf{s}\}$$

for $\mathbf{s} \in (GF(2))^m$ and $m = 1, 2, \ldots 5$. In Table 6.3 we give the cardinality of $\mathcal{G}_5(\mathbf{s})$ for all $\mathbf{s} \in (GF(2))^5$. From this table it follows that $\mathcal{G}_5(01010)$ is a 3-UEC code of length 16 and size 53, $\mathcal{G}_4(1010)$ is a 3-UEC code of length 17 and size 97, $\mathcal{G}_3(010)$ is a 3-UEC code of length 18 and size 188, $\mathcal{G}_2(10)$ is a 3-UEC code of length 19 and size 376, and $\mathcal{G}_1(0)$ is a 3-UEC code of length 20 and size 737.

6.3.3 4-UEC codes

We now give 4-UEC codes of length $13 \leq n \leq 18$ and $n = 22$.

Lengths 13,14,15

Table D.7 lists the codewords of 4-UEC codes of length 13 and size 6, of length 14 and size 8, and of length 15 and size 10 that have been found by trial and error.

Lengths 16,17,18

By applying the unidirectional construction method with $\mathcal{C}_1 = \mathcal{L}_{19}$ (see Subsection 6.2.3), we obtain a 4-UEC code of length 18 and size 28 if $m = 1$, of length 17 and size 18 if $m = 2$, and of length 16 and size 11 if $m = 3$. Table E.9 shows a possible choice for the $\mathbf{a}_i$ that yields these results.

Length 22

By applying the unidirectional construction method with $\mathcal{C}_1 = \mathcal{T}_{23}$ (see Subsection 6.2.3), we obtain a 4-UEC code of length 22 and size 82 if $m = 1$. Table E.9 shows a possible choice for the $\mathbf{a}_i$ that yields this result.

6.4 *Retrospect and prospect*

In this chapter we have applied the construction method from Chapter 5 to obtain t-AsEC codes and t-UEC codes of length n in the area $t \leq 4$ and $n \leq 23$, that are larger than comparable SyEC codes. Optimal choices for the tails $\mathbf{a}_i$ in the method are given in Appendix E. Sometimes only minor improvements have been obtained, as for example the 2-AsEC code of length 21 and size 4251 (Subsection 6.2.1) compared with the largest known 2-SyEC code of length 21 that contains 4096 codewords. However in other cases, the number of codewords almost doubled, for example when comparing the 3-AsEC code of length 17 and size 122 (Subsection 6.2.2) with the largest known 3-SyEC code of length 17 that contains 64 codewords. Often the final result depends on the arrangement of the columns in the initial code. As seen for the Wagner code (Subsection 6.2.1), the asymmetric construction method applied with $m = 2$ to the code $\mathcal{W}_{23}$ gives a 2-AsEC code of length 21 and size 4200, while the method applied with $m = 2$ to the equivalent code $\mathcal{W}'_{23}$ gives a 2-AsEC code of length 21 and size 4251. We are not sure if the arrangement of the columns as in the latter code is an optimal one for the construction method. To find this out would probably require a detailed study of the Wagner code, i.e. in general the initial code $\mathcal{C}_1$.

Further, in Section 6.1 we have sketched a way to construct a code by trial and error with the use of constraints on the weight distribution that follow from Chapter 3. This technique seems to be particularly appropriate to derive optimal t-UEC and t-AsEC codes of length n just outside the area

covered by Theorems 2.10 and 2.11, i.e. $n \approx 2t + 4$. Examples are listed in Appendix D.

Finally, we have enlarged existing AsEC codes by adding vectors without disturbing the asymmetric error correcting capability of the code. Examples of such vectors corresponding to some codes from Section 6.2 are listed in Appendix F. Again it would require detailed knowledge about the original code to optimize this technique.

The sizes of most of the codes that have been constructed in this chapter appear as lower bound on $A_{As}(n,t)$ and $A_U(n,t)$ in the Tables B.1, B.2, B.3, and B.4.

Almost all of the AsEC and UEC codes constructed thus far in this thesis are not linear. In the next chapter we will focus on linear AsEC codes in comparison with the well-studied linear SyEC codes. The unidirectional case need not be studied, since any linear t-UEC code turns out to be also t-SyEC.

Chapter 7

Linear asymmetric error correcting codes

7.1 Linear codes of relatively small lengths

As mentioned in Section 1.2 we call a code of length n linear if it is a subspace of $(GF(2))^n$. Linear SyEC codes have been studied extensively (see e.g. [21,36]). Bounds on the maximum dimension $K_{Sy}(n,t)$ of a linear t-SyEC code of length n can be found in [41]. For relatively small n the values of $K_{Sy}(n,t)$ are known exactly, as stated in the following theorem.

Theorem 7.1 *We have*

1. $K_{Sy}(n,t) = 0$ *for* $t \leq n \leq 2t$ *and* $t \geq 1$;
2. $K_{Sy}(n,t) = 1$ *for* $2t+1 \leq n \leq 3t+1$ *and* $t \geq 1$;
3. $K_{Sy}(n,t) = 2$ *for* $3t+2 \leq n \leq 3t+1+\lceil (2t+1)/4 \rceil$ *and* $t \geq 1$;
4. $K_{Sy}(n,t) = 3$ *for* $n = 3t+2+\lceil (2t+1)/4 \rceil$ *and* $t \geq 1$.

Proof. The Griesmer lower bound

$$\sum_{i=0}^{k-1} \lceil d/(2^i) \rceil$$

for the minimal length of a linear code with dimension k and Hamming distance d is tight for $1 \leq k \leq 3$ (see e.g. [36]). The theorem now follows by substituting $d = 2t+1$ and $k = 1,2,3$. □

Since any linear code contains the all-zero vector $\mathbf{0}$, it follows that in a linear t-UEC code codewords of weight $1, 2, \ldots, 2t$ do not occur. Because of (1.14) it follows that such a code is also t-SyEC. With (1.33) we get:

$$K_{Sy}(n,t) = K_U(n,t) \leq K_{As}(n,t) \tag{7.1}$$

for all $1 \leq t \leq n$.

The values of $K_{As}(n,t)$ can also be determined exactly for relatively small n, as stated in the next theorem.

Theorem 7.2 *We have*

1. $K_{As}(n,t) = 0$ *for* $n = t$ *and* $t \geq 1$;
2. $K_{As}(n,t) = 1$ *for* $t+1 \leq n \leq 2t+1$ *and* $t \geq 1$;
3. $K_{As}(n,t) = 2$ *for* $2t+2 \leq n \leq 3t+2$ *and* $t \geq 1$;
4. $K_{As}(n,t) = 3$ *for* $n = 3t+3$ *and* $t \geq 1$.

Proof. The linear codes of dimension k and length $k(t+1)$ having generator matrices consisting of $t+1$ repetitions of the identity matrix $\mathbf{I}_k$ are t-AsEC, and thus $K_{As}(k(t+1),t) \geq k$ for all $k \geq 1$. With (1.33), (1.36) and the Theorems 7.1 and 2.11 it now follows that:

1.

$$0 = K_{Sy}(t,t) \leq K_{As}(t,t) \leq \lfloor \log_2(A_{As}(t,t)) \rfloor = \lfloor \log_2(1) \rfloor = 0;$$

2.

$$\begin{aligned} 1 &\leq K_{As}(t+1,t) \leq K_{As}(t+2,t) \leq \ldots \leq K_{As}(2t+1,t) \\ &\leq \lfloor \log_2(A_{As}(2t+1,t)) \rfloor = \lfloor \log_2(2) \rfloor = 1; \end{aligned}$$

3.

$$2 \leq K_{As}(2t+2,t) \leq K_{As}(2t+3,t) \leq \ldots \leq K_{As}(3t+2,t);$$

4.

$$3 \leq K_{As}(3t+3,t) \leq K_{As}(3t+2,t) + 1.$$

Hence the only thing left to prove is that $K_{As}(3t+2,t) \leq 2$. Suppose the contrary holds, i.e. there is a linear t-AsEC code C of length $3t+2$ and dimension 3. Without loss of generality we may assume that three basis vectors of C look like

$$\begin{array}{rcccccccc} \mathbf{b}_1 & = & 1^\alpha & 0^\beta & 0^\gamma & 1^\delta & 1^\varsigma & 0^\eta & 1^\theta \\ \mathbf{b}_2 & = & 0^\alpha & 1^\beta & 0^\gamma & 1^\delta & 0^\varsigma & 1^\eta & 1^\theta \\ \mathbf{b}_3 & = & 0^\alpha & 0^\beta & 1^\gamma & 0^\delta & 1^\varsigma & 1^\eta & 1^\theta \end{array}$$

with

$$\alpha+\beta+\gamma+\delta+\varsigma+\eta+\theta = 3t+2 \quad (*1)$$

and $w(\mathbf{b}_1) \leq w(\mathbf{b}_2) \leq w(\mathbf{b}_3)$, i.e.

$$\alpha+\delta+\varsigma \leq \beta+\delta+\eta \leq \gamma+\varsigma+\eta. \quad (*2)$$

Because of $(*2)$ we have

$$\begin{array}{rcll} d_{As}(\mathbf{b}_1,\mathbf{b}_2) & = & 2(\beta+\eta) \geq 2t+2 & (*3) \\ d_{As}(\mathbf{b}_1,\mathbf{b}_3) & = & 2(\gamma+\eta) \geq 2t+2 & (*4) \\ d_{As}(\mathbf{b}_2,\mathbf{b}_3) & = & 2(\gamma+\varsigma) \geq 2t+2. & (*5) \end{array}$$

Now $(*1,*3,*5)$ give

$$\begin{array}{rcll} \alpha+\delta+\theta & = & 3t+2-(\beta+\eta)-(\gamma+\varsigma) & \\ & \leq & 3t+2-(t+1)-(t+1)=t. & (*6) \end{array}$$

Because of $(*6)$ we have

$$\begin{array}{rcll} d_{As}(\mathbf{b}_1+\mathbf{b}_2,\mathbf{b}_2) & = & 2(\alpha+\varsigma) \geq 2t+2 & (*7) \\ d_{As}(\mathbf{b}_1+\mathbf{b}_3,\mathbf{b}_3) & = & 2(\varsigma+\theta) \geq 2t+2. & (*8) \end{array}$$

Since $\alpha+\beta \leq t$ would imply that $d_{As}(\mathbf{b}_1+\mathbf{b}_2,\mathbf{b}_3) = 2(\gamma+\theta) \geq 2t+2$ and so with $(*3,*7)$ that $(\gamma+\theta)+(\beta+\eta)+(\alpha+\varsigma) \geq (t+1)+(t+1)+(t+1) = 3t+3$ which contradicts $(*1)$, we have

$$\alpha+\beta \geq t+1. \quad (*9)$$

Finally, it follows from $(*4,*8,*9)$ that

$$\begin{array}{rcl} \alpha+\beta+\gamma+\delta+\varsigma+\eta+\theta & \geq & \\ (\gamma+\eta)+(\varsigma+\theta)+(\alpha+\beta) & \geq & \\ (t+1)+(t+1)+(t+1) & = & 3t+3, \end{array}$$

n	$K_{Sy}(n,t)$	$K_{As}(n,t)$
t	0	0
$t+1$	0	1
$\vdots$	$\vdots$	$\vdots$
$2t$	0	1
$2t+1$	1	1
$2t+2$	1	2
$\vdots$	$\vdots$	$\vdots$
$3t+1$	1	2
$3t+2$	2	2
$3t+3$	2	3
$\vdots$	$\vdots$	
$3t+1+\lceil(2t+1)/4\rceil$	2	
$3t+2+\lceil(2t+1)/4\rceil$	3	

Table 7.1: $K_{Sy}(n,t)$ and $K_{As}(n,t)$ for $t \geq 2$ and relatively small n.

which contradicts $(*1)$. Hence no linear t-AsEC code of length $3t+2$ and dimension 3 exists, and so $K_{As}(3t+2,t) \leq 2$. □

The results of Theorems 7.1 and 7.2 are summarized in Tables 7.1 and C.1.

Just as we were interested in cases in which $A_{As}(n,t) > A_{Sy}(n,t)$ in the preceding chapters, we are interested in n and t for which $K_{As}(n,t) > K_{Sy}(n,t)$ in this chapter. Theorems 7.1 and 7.2 give us the following result.

Corollary 7.3 *We have*

$$K_{As}(n,t) = K_{Sy}(n,t) + 1$$

for $t \geq 1$ *and* $t+1 \leq n \leq 3t+1+\lceil(2t+1)/4\rceil$, $n \neq 2t+1, 3t+2$. □

Varshamov ([39]) was rather pessimistic of finding good linear AsEC codes. The reason why is treated in Section 7.2. For $t = 1, 2$ all n for which $K_{As}(n,t) > K_{Sy}(n,t)$ are determined in Section 7.3. Finally in Section 7.4 it is proved that $K_{As}(2^m, 2^{m-2}-1) > K_{Sy}(2^m, 2^{m-2}-1) = m+1$ for all $m \geq 4$, m even, by constructing a class of linear $(2^{m-2}-1)$-AsEC codes of length 2^m and dimension $m+2$.

7.2 Results of Varshamov

In [39] Varshamov started to investigate linear codes that correct asymmetric errors. The following lemma is the clue to his results.

Lemma 7.4 (Varshamov) *In any linear t-AsEC code $\mathcal{C}$ all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ with $t+1 \leq w(\mathbf{u}) \leq 2t$ satisfy*

$$|\{i|u_i = 1 \wedge v_i = 1\}| \neq t.$$

□

Using this lemma he derived the following theorem.

Theorem 7.5 (Varshamov) *For all $t \geq 1$ and $n \geq 3t$ we have*

$$K_{As}(n,t) - K_{As}(n-2t,t) = 2t \Rightarrow K_{As}(n,t) = K_{Sy}(n,t).$$

□

Finally, the observation that the condition $K_{As}(n,t) - K_{As}(n-2t,t) = 2t$ is satisfied for almost all integers n when t is fixed gave the following corollary.

Corollary 7.6 (Varshamov) *For fixed $t \geq 1$ we have*

$$|\{n|t \leq n \leq N \wedge K_{Sy}(n,t) = K_{As}(n,t)\}| = N - O(\log N).$$

□

Hence it was concluded that $K_{As}(n,t)$ equals $K_{Sy}(n,t)$ almost everywhere.

7.3 Single and double error correcting codes

From Corollary 7.3 it follows that $K_{As}(n,t) = K_{Sy}(n,t) + 1$ for $n = 2,4$ if $t = 1$ and for $n = 3,4,6,7,9$ if $t = 2$. We will show that these are the only cases in which $K_{As}(n,t) > K_{Sy}(n,t)$ for $t = 1,2$. To this end we will first give a generalization of Lemma 7.4.

Lemma 7.7 *In any linear t-AsEC code $\mathcal{C}$ all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ with $\mathbf{u} \neq \mathbf{0}$ satisfy*

$$|\{i|u_i = 1 \wedge v_i = 1\}| \geq t+1 \vee |\{i|u_i = 1 \wedge v_i = 1\}| \leq w(\mathbf{u}) - t - 1.$$

Proof. Since C is linear the vector $\mathbf{u}+\mathbf{v}$ is a codeword. Without loss of generality we may assume that $\mathbf{u}, \mathbf{v}$, and $\mathbf{u}+\mathbf{v}$ look like:

$$\begin{aligned} \mathbf{u} &= 1^{\alpha}\ 1^{\beta}\ 0^{\gamma}\ 0^{\delta} \\ \mathbf{v} &= 1^{\alpha}\ 0^{\beta}\ 1^{\gamma}\ 0^{\delta} \\ \mathbf{u}+\mathbf{v} &= 0^{\alpha}\ 1^{\beta}\ 1^{\gamma}\ 0^{\delta}. \end{aligned}$$

Hence

$$d_{A_s}(\mathbf{v}, \mathbf{u}+\mathbf{v}) = 2\max\{\alpha, \beta\} = 2\max\{\alpha, w(\mathbf{u})-\alpha\} \geq 2t+2,$$

which implies $\alpha \geq t+1$ or $w(\mathbf{u})-\alpha \geq t+1$. □

We will also need the following lemma.

Lemma 7.8 *We have*

$$K_{Sy}(n+t+1, t) \geq K_{Sy}(n, t)+1$$

for all $1 \leq t \leq n$.

Proof. Let C be a linear t-SyEC code of length n and dimension $k = K_{Sy}(n, t)$. Without loss of generality we may assume that a basis of C looks like $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{k-1}, \mathbf{b}_k = 1^{2t+1}0^{n-2t-1}$. We add a tail 0^{t+1} to each of these k vectors, and we define an extra basis vector to be $\mathbf{b}_{k+1} = 1^t 0^{n-t} 1^{t+1}$. The code C' with this new basis is a t-SyEC code of length $n+t+1$ and dimension $K_{Sy}(n, t)+1$. Hence $K_{Sy}(n+t+1, t) \geq K_{Sy}(n, t)+1$. □

The existence of a linear t-AsEC code whose dimension exceeds the dimension of the largest linear t-SyEC code of the same length turns out to have important consequences.

Theorem 7.9 *A linear t-AsEC code C of length n and dimension k with $n \geq 2t+1$ and $k = K_{A_s}(n, t) \geq K_{Sy}(n, t)+1$ contains at least one codeword $\mathbf{c}$ with $t+1 \leq w(\mathbf{c}) \leq 2t$.*

If C contains a codeword $\mathbf{a}$ with $t+1 \leq w(\mathbf{a}) \leq \min\{t+2, 2t\}$, then

$$K_{A_s}(n-t-1, t) \geq K_{A_s}(n, t)-1$$

and

$$K_{A_s}(n-t-1, t) \geq K_{Sy}(n-t-1, t)+1.$$

Proof. Since $\mathbf{0} \in \mathcal{C}$ and since the code is t-AsEC but not t-SyEC, $\mathcal{C}$ contains at least one codeword $\mathbf{c}$ of weight $t+1 \leq w(\mathbf{c}) \leq 2t$.

In the case $\mathcal{C}$ contains a codeword $\mathbf{a}$ with $t+1 \leq w(\mathbf{a}) \leq \min\{t+2, 2t\}$ we may assume without loss of generality that a basis of $\mathcal{C}$ looks like $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{k-1}, \mathbf{b}_k = 1^{w(\mathbf{a})}0^{n-w(\mathbf{a})}$, where the first $t+1$ coordinates of all $\mathbf{b}_i$ $(1 \leq i \leq k-1)$ equal 0^{t+1} due to Lemma 7.7. We now omit $\mathbf{b}_k$ from the basis and delete the first $t+1$ coordinates of the other $k-1$ basis vectors. The code $\mathcal{C}'$ with this new basis is a t-AsEC code of length $n-t-1$ and dimension $k-1$. Hence

$$K_{As}(n-t-1,t) \geq \dim(\mathcal{C}') = \dim(\mathcal{C}) - 1 = K_{As}(n,t) - 1.$$

Suppose $K_{As}(n-t-1,t) = K_{Sy}(n-t-1,t)$, then

$$K_{Sy}(n-t-1,t) = K_{As}(n-t-1,t) \geq K_{As}(n,t) - 1 \geq K_{Sy}(n,t),$$

which contradicts Lemma 7.8. Hence

$$K_{As}(n-t-1,t) \geq K_{Sy}(n-t-1,t) + 1.$$

□

Corollary 7.10 *If* $K_{As}(n,1) \geq K_{Sy}(n,1) + 1$ *for a certain* $n \geq 3$, *then*

$$K_{As}(n-2,1) \geq K_{As}(n,1) - 1$$

and

$$K_{As}(n-2,1) \geq K_{Sy}(n-2,1) + 1.$$

Proof. This follows immediately from Theorem 7.9, as a 1-AsEC code of length n and dimension $K_{As}(n,1) \geq K_{Sy}(n,1) + 1$ contains at least one codeword $\mathbf{a}$ of weight 2. □

Corollary 7.11 *If* $K_{As}(n,2) \geq K_{Sy}(n,2) + 1$ *for a certain* $n \geq 5$, *then*

$$K_{As}(n-3,2) \geq K_{As}(n,2) - 1$$

and

$$K_{As}(n-3,2) \geq K_{Sy}(n-3,2) + 1.$$

Proof. This follows immediately from Theorem 7.9, as a 2-AsEC code of length n and dimension $K_{As}(n,2) \geq K_{Sy}(n,2) + 1$ contains at least one codeword $\mathbf{a}$ of weight 3 or 4. □

We are now ready for the final results.

Theorem 7.12 *We have*

$$K_{As}(n,1) = K_{Sy}(n,1) \text{ for all } n \geq 1, n \neq 2,4.$$

Proof. Assuming that $K_{As}(m,1) = K_{Sy}(m,1)$ for a certain $m \geq 1$, it then follows that also $K_{As}(m+2,1) = K_{Sy}(m+2,1)$. For if the contrary would hold, i.e. $K_{As}(m+2,1) \geq K_{Sy}(m+2,1) + 1$, Corollary 7.10 (with $n = m+2$) is contradicted.

We now need two starting points. These can be obtained from Theorems 7.1 and 7.2. First, since $K_{As}(1,1) = K_{Sy}(1,1)$, we have $K_{As}(n,1) = K_{Sy}(n,1)$ for all $n \geq 1$, n odd. Finally, it follows from $K_{As}(6,1) = K_{Sy}(6,1)$ that $K_{As}(n,1) = K_{Sy}(n,1)$ for all $n \geq 6$, n even. □

Theorem 7.13 *We have*

$$K_{As}(n,2) = K_{Sy}(n,2) \text{ for all } n \geq 2, n \neq 3,4,6,7,9.$$

Proof. Assuming that $K_{As}(m,2) = K_{Sy}(m,2)$ for a certain $m \geq 2$, it then follows that also $K_{As}(m+3,2) = K_{Sy}(m+3,2)$. For if the contrary would hold, i.e. $K_{As}(m+3,2) \geq K_{Sy}(m+3,2) + 1$, Corollary 7.11 (with $n = m+3$) is contradicted.

We now need three starting points. First, since $K_{As}(2,2) = K_{Sy}(2,2)$ (see Theorems 7.1 and 7.2), we have $K_{As}(n,2) = K_{Sy}(n,2)$ for all $n \geq 2$, $n \equiv 2 \pmod 3$. Further, it follows from Corollary 7.11 (with $n = 10$) and $K_{Sy}(10,2) = 3$ (see e.g. [41]) that $K_{As}(10,2) \geq 4$ would imply $K_{As}(7,2) \geq 4 - 1 = 3$. Since $K_{As}(7,2) = 2$ (see Theorem 7.2), we have $K_{As}(10,2) = K_{Sy}(10,2) = 3$, and so $K_{As}(n,2) = K_{Sy}(n,2)$ for all $n \geq 10$, $n \equiv 1 \pmod 3$. Finally, it follows from Corollary 7.11 (with $n = 12$) and $K_{Sy}(12,2) = 4$ (see e.g. [41]) that $K_{As}(12,2) \geq 5$ would imply $K_{As}(9,2) \geq 5 - 1 = 4$. Since $K_{As}(9,2) = 3$ (see Theorem 7.2), we have $K_{As}(12,2) = K_{Sy}(12,2) = 4$, and so $K_{As}(n,2) = K_{Sy}(n,2)$ for all $n \geq 12$, $n \equiv 0 \pmod 3$. □

7.4 A class of linear AsEC codes

In this section we present a class of linear AsEC codes that are superior to comparable SyEC codes.

Description

The codes can be described by taking a first-order Reed-Muller code and one of its cosets. A generator matrix $\mathbf{G}_m$ of the first-order Reed-Muller code $\mathcal{R}(1,m)$ with $m \geq 1$ can be defined as (cf. [21]):

$$\mathbf{G}_m = \begin{pmatrix} \mathbf{G}_{m-1} & \mathbf{G}_{m-1} \\ 0\cdots 0 & 1\cdots 1 \end{pmatrix} \text{ with } \mathbf{G}_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{7.2}$$

These codes $\mathcal{R}(1,m)$ have length 2^m, dimension $m+1$, and Hamming distance 2^{m-1}. All codewords in $\mathcal{R}(1,m)$ have weight 2^{m-1}, with exception of the all-one and all-zero vector, as shown in Table 7.2.

For a certain m we call the rows of $\mathbf{G}_m$ (from above) $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_m$, respectively. Let $\mathbf{u} \cdot \mathbf{x}$ be defined as the vector $(u_1 \cdot x_1, u_2 \cdot x_2, \ldots, u_n \cdot x_n)$. The vector $\mathbf{z}_m$ is now defined as

$$\mathbf{z}_m = \mathbf{v}_1 \cdot \mathbf{v}_2 + \mathbf{v}_3 \cdot \mathbf{v}_4 + \cdots + \mathbf{v}_{m-1} \cdot \mathbf{v}_m \tag{7.3}$$

for $m \geq 2$, m even.

For all $m \geq 2$, m even, we consider the code $\mathcal{R}^*_m$ consisting of the union of $\mathcal{R}(1,m)$ and its coset $\mathbf{z}_m + \mathcal{R}(1,m) = \{\mathbf{z}_m + \mathbf{c} | \mathbf{c} \in \mathcal{R}(1,m)\}$:

$$\mathcal{R}^*_m = (\mathcal{R}(1,m)) \cup (\mathbf{z}_m + \mathcal{R}(1,m)). \tag{7.4}$$

It is easy to check (by induction) that $\mathcal{R}^*_m$ is a linear code of length 2^m, dimension $m+2$, and Hamming distance $2^{m-1} - 2^{m/2-1}$, having generator matrix

$$\mathbf{G}^*_m = \begin{bmatrix} \mathbf{G}_m \\ \mathbf{z}_m \end{bmatrix} \tag{7.5}$$

	$\mathcal{R}(1,m)$	$\mathbf{z}_m + \mathcal{R}(1,m)$	$\mathcal{R}^*_m$
i	A_i	A_i	A_i
0	1	0	1
$2^{m-1} - 2^{m/2-1}$	0	2^m	2^m
2^{m-1}	$2^{m+1} - 2$	0	$2^{m+1} - 2$
$2^{m-1} + 2^{m/2-1}$	0	2^m	2^m
2^m	1	0	1

Table 7.2: Weight distributions of $\mathcal{R}(1,m)$, $\mathbf{z}_m + \mathcal{R}(1,m)$, and $\mathcal{R}^*_m$.

and weight distribution as shown in Table 7.2.

Example

For example, we treat the case $m = 4$. The code $\mathcal{R}(1,4)$ of length 16, dimension 5, and Hamming distance 8, has weight distribution $A_0 = 1$, $A_8 = 30$, $A_{16} = 1$, and generator matrix

$$\mathbf{G}_4 = \begin{bmatrix} 1111111111111111 \\ 0101010101010101 \\ 0011001100110011 \\ 0000111100001111 \\ 0000000011111111 \end{bmatrix}. \tag{7.6}$$

Hence

$$\mathbf{z}_4 = (0001000100011110), \tag{7.7}$$

and so

$$\mathbf{G}_4^* = \begin{bmatrix} 1111111111111111 \\ 0101010101010101 \\ 0011001100110011 \\ 0000111100001111 \\ 0000000011111111 \\ 0001000100011110 \end{bmatrix}. \tag{7.8}$$

The code $\mathcal{R}_4^*$ has length 16, dimension $5 + 1 = 6$, Hamming distance $8 - 2 = 6$, and weight distribution $A_0 = 1$, $A_6 = 16$, $A_8 = 30$, $A_{10} = 16$, $A_{16} = 1$.

Error correction capabilities

Since the Hamming distance of $\mathcal{R}(1,m)$ equals 2^{m-1}, this code is $(2^{m-2}-1)$-SyEC for all $m \geq 3$. Hence $\mathcal{R}(1,m)$ is also $(2^{m-2}-1)$-AsEC for all $m \geq 3$. Since the Hamming distance of $\mathcal{R}_m^*$ equals $2^{m-1} - 2^{m/2-1}$, this code is only $(2^{m-2} - 2^{m/2-2} - 1)$-SyEC for $m \geq 4$, m even. But $\mathcal{R}_m^*$ is still $(2^{m-2} - 1)$-AsEC for all $m \geq 4$, m even! This can be shown as follows. Let $\mathbf{u}$ and $\mathbf{v}$ be two different codewords of $\mathcal{R}_m^*$. If $\mathbf{u}, \mathbf{v} \in \mathcal{R}(1,m)$ or $\mathbf{u}, \mathbf{v} \in \mathcal{R}(1,m) + \mathbf{z}_m$, then

$$d_{As}(\mathbf{u},\mathbf{v}) \geq d_{Sy}(\mathbf{u},\mathbf{v}) \geq 2^{m-1}.$$

If $\mathbf{u} \in \mathcal{R}(1,m)$ and $\mathbf{v} \in \mathcal{R}(1,m) + \mathbf{z}_m$, then the weight distributions of $\mathcal{R}(1,m)$ and $\mathcal{R}(1,m) + \mathbf{z}_m$ give

$$d_{As}(\mathbf{u},\mathbf{v}) = d_{Sy}(\mathbf{u},\mathbf{v}) + |w(\mathbf{u}) - w(\mathbf{v})| \geq 2^{m-1} - 2^{m/2-1} + 2^{m/2-1} = 2^{m-1}.$$

Hence the d_{As} distance of $\mathcal{R}_m^*$ is at least 2^{m-1}, and so the code is $(2^{m-2}-1)$-AsEC.

For comparison we also consider $(2^{m-2}-1)$-SyEC codes of length 2^m with $m \geq 3$. It follows from the $\mathcal{R}(1,m)$ codes that the largest possible dimension of such codes is at least $m+1$. On the other hand, there is a result by Logačev (see e.g. [36]), in which it is claimed that that the minimum length of a linear code of dimension k and Hamming distance d with $3 \leq d \leq 2^{k-2}-2$ is at least $1 + \sum_{i=0}^{k-1} \lceil d/2^i \rceil$. Hence the minimum length of a linear $(2^{m-2}-1)$-SyEC code of dimension $m+2$ is at least 2^m+1, and so the largest dimension of a linear $(2^{m-2}-1)$-SyEC code of length 2^m is at most $m+1$.

Thus we can summarize the results of this section in the following theorem.

Theorem 7.14 *We have*

$$\begin{array}{lcll} K_{As}(2^m, 2^{m-2}-1) & \geq & m+2 & \text{for } m \geq 4,\ m \text{ even;} \\ K_{Sy}(2^m, 2^{m-2}-1) & = & m+1 & \text{for } m \geq 3. \end{array}$$

□

Table F.2 lists vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ that can be added together with their complements to $\mathcal{R}_4^*$ while keeping the code 3-AsEC. Thus we have constructed a 3-AsEC code of length 16 and size $64 + 8 = 72$. Hence $A_{As}(16,3) \geq 72$.

7.5 *Retrospect and prospect*

In this chapter we have considered linear AsEC codes in comparison with linear SyEC codes. Linear UEC codes do not have to be involved since any linear t-UEC code is also t-SyEC. In Section 7.1 we have given the exact values for the maximum dimension $K_{Sy}(n,t)$ of a linear t-SyEC code of length n and for the maximum dimension $K_{As}(n,t)$ of a linear t-AsEC code of length n for relatively small n, yielding some (trivial) cases in

which $K_{As}(n,t) > K_{Sy}(n,t)$. Some of Varshamov's results concerning linear AsEC codes have been described in Section 7.2, giving rather pessimistic expectations on finding good linear AsEC codes. Varshamov's assertion

> '$K_{As}(n,t)$ equals $K_{Sy}(n,t)$ almost everywhere'

has even be strengthened for $t = 1$ and $t = 2$ to

> '$K_{As}(n,t)$ equals $K_{Sy}(n,t)$ everywhere, with the exception of only a finite number of values of n'

in Section 7.3. It is still an open question as to whether the latter assertion holds for $t \geq 3$.

Somewhat more optimistic expectations on finding linear AsEC codes that are superior to comparable SyEC codes follow from the results that have been presented in Section 7.4. The class of linear $(2^{m-2}-1)$-AsEC codes $\mathcal{R}_m^*$ of length 2^m and dimension $m+2$ ($m \geq 4$, m even) indicates that $K_{As}(n,t) \geq K_{Sy}(n,t) + 1$ in some nontrivial cases. Another open question is whether there are n and t satisfying $K_{As}(n,t) \geq K_{Sy}(n,t) + 2$.

Considerations like the ones made in Theorem 7.9 might give a clue when trying to answer these open questions, since the necessary presence of codewords of weight between $t+1$ and $2t$ in a linear t-AsEC code of length n and dimension at least $K_{Sy}(n,t) + 1$ has shown to be of great importance when applying shortening techniques on such a code.

Tables of bounds on $K_{Sy}(n,t)$ and $K_{As}(n,t)$ for $n \leq 23$ and $t \leq 4$ are given in Tables C.1 and C.2 in Appendix C.

Bibliography

[1] M.R. Best, A.E. Brouwer, F.J. MacWilliams, A.M. Odlyzko, and N.J.A. Sloane, "Bounds for binary codes of length less than 25", IEEE Trans. Inf. Theory, vol. IT-24, pp. 81-93, Jan. 1978.

[2] M. Blaum and H.C.A. van Tilborg, "On t-error correcting/all unidirectional error detecting codes", to appear in IEEE Trans. Comput.

[3] F.J.H. Böinck, private communication.

[4] B. Bose and D.K. Pradhan, "Optimal unidirectional error detecting/correcting codes", IEEE Trans. Comput., vol. C-31, pp. 564-568, June 1982.

[5] B. Bose and T.R.N. Rao, "Theory of unidirectional error correcting/detecting codes", IEEE Trans. Comput., vol. C-31, pp. 521-530, June 1982.

[6] A.E. Brouwer, P. Delsarte, and P. Piret, "On the (23,14,5) Wagner code", IEEE Trans. Inf. Theory, vol. IT-26, pp. 742-743, Nov. 1980.

[7] S.D. Constantin and T.R.N. Rao, "On the theory of binary asymmetric error-correcting codes", Inform. Control, vol. 40, pp. 20-36, 1979.

[8] P. Delsarte and P. Piret, "Bounds and constructions for binary asymmetric error-correcting codes", IEEE Trans. Inf. Theory, vol. IT-27, pp. 125-128, Jan. 1981.

[9] P. Delsarte and P. Piret, "Spectral enumerators for certain additive-error-correcting codes over integer alphabets", Inform. Control, vol. 48, pp. 193-210, 1981.

[10] I.Ya. Goldbaum, "Estimate for the number of signals in codes correcting nonsymmetric errors" (in Russian), Automat. Telemekh., vol. 32,

pp. 94-97, 1971 (English translation: Automat. Rem. Control, vol. 32, pp. 1783-1785, 1971).

[11] R.L. Graham and N.J.A. Sloane, "Lower bounds for constant weight codes", IEEE Trans. Inf. Theory, vol. IT-26, pp. 37-43, Jan. 1980.

[12] R.W. Hamming, "Error detecting and error correcting codes", Bell Syst. Techn. J., vol. 29, pp. 147-160, 1950.

[13] C. Helgesen, "Asymmetric error-correcting codes with minimal block length", unpublished manuscript, Bergen, Norway, 1984.

[14] T. Helleseth and T. Kløve, "On group-theoretic codes for asymmetric channels", Inform. Control, vol. 49, pp. 1-9, 1981.

[15] W.H. Kim and C.V. Freiman, "Single error-correcting codes for asymmetric binary channels", IRE Trans. Inf. Theory, vol. IT-5, pp. 62-66, June 1959.

[16] W.H. Kim and C.V. Freiman, "Multiple error-correcting codes for a binary asymmetric channel", IEEE Trans. Circuit Theory, vol. CT-6, Special Supplement on International Symposium on Circuit and Information Theory, pp. 71-78, 1959.

[17] T. Kløve, "Upper bounds on codes correcting asymmetric errors", IEEE Trans. Inf. Theory, vol. IT-27, pp. 128-131, Jan. 1981.

[18] T. Kløve, "Error correcting codes for the asymmetric channel", Rep. 18-09-07-81, Dept. Mathematics, University of Bergen, Norway, July 1981.

[19] D.J. Lin and B. Bose, "Theory and design of t-error correcting and d ($d > t$)-unidirectional error detecting (t-EC d-UED) codes", IEEE Trans. Comput., vol. C-37, pp. 433-439, April 1988.

[20] S. Lin and D.J. Costello, Jr., *Error control coding: fundamentals and applications*, Englewood Cliffs, NJ : Prentice-Hall, 1983.

[21] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, Amsterdam : North-Holland, 1977.

[22] R.J. McEliece, "Comment on 'A class of codes for asymmetric channels and a problem from the additive theory of numbers'", IEEE Trans. Inf. Theory, vol. IT-19, p. 137, Jan. 1973.

[23] R.J. McEliece, *The theory of information and coding*, Reading, MA : Addison-Wesley, 1977.

[24] R.J. McEliece and E.R. Rodemich, "The Constantin-Rao construction for binary asymmetric error-correcting codes", Inform. Control, vol. 44, pp. 187-196, 1980.

[25] D. Nikolos, N. Gaitanis, and G. Philokyprou, "Systematic t-error correcting/all unidirectional error detecting codes", IEEE Trans. Comput., vol. C-35, pp. 394-402, May 1986.

[26] E. van Os, "Packing density of codes" (in Dutch), M.Sc. Thesis, Department of Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, May 1987.

[27] V. Pless, "Decoding the Golay codes", IEEE Trans. Inf. Theory, vol. IT-32, pp. 561-567, July 1986.

[28] D.K. Pradhan, "A new class of error-correcting/detecting codes for fault-tolerant computer application", IEEE Trans. Comput., vol. C-29, pp.471-481, June 1980.

[29] C.L.M. van Pul, "On bounds on codes", M.Sc. Thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands, August 1982.

[30] T.R.N. Rao and A.S. Chawla, "Asymmetric error codes for some LSI semi-conductor memories", Proc. Annual Southeastern Symp. Syst. Theory, pp. 170-171, 1975.

[31] T.R.N. Rao and E. Fujiwara, *Error-control coding for computer systems*, Englewood Cliffs, NJ : Prentice-Hall, 1989.

[32] A.M. Romanov, "New binary codes of minimal distance 3" (in Russian), Problemy Peredachi Informatsii, vol. 19, pp. 101-102, 1983.

[33] Y. Saitoh, K. Yamaguchi, and H. Imai, "Some new binary codes correcting asymmetric/unidirectional errors", submitted to IEEE trans. Inf. Theory.

[34] A. Shiozaki, "Construction for binary asymmetric error-correcting codes", IEEE Trans. Inf. Theory, vol. IT-28, pp. 787-789, Sept. 1982.

[35] N.J.A. Sloane, S.M. Reddy, and C.L. Chen, "New binary codes", IEEE Trans. Inf. Theory, vol. IT-18, pp. 503-510, July 1972.

[36] H.C.A. van Tilborg, "The smallest length of binary 7-dimensional linear codes with prescribed minimum distance", Discrete Mathematics, vol. 33, pp. 197-207, 1981.

[37] T. Uyematsu, K. Yamazaki, O. Hirota, M. Nakagawa, and K. Sakaniwa, "Effect of asymmetric error correcting codes in photon communication systems", Transactions IEICE, vol. E71, no. 9, pp. 850-857, September 1988.

[38] R.R. Varshamov, "Estimate of the number of signals in codes with correction of nonsymmetric errors" (in Russian), Automat. Telemekh., vol. 25, pp. 1628-1629, 1964 (English translation: Automat. Rem. Control, vol. 25, pp. 1468-1469, 1964).

[39] R.R. Varshamov, "Some features of linear codes that correct asymmetric errors" (in Russian), Doklady Akad. Nauk. SSSR 157 no.3, pp. 546-548, 1964 (English translation: Soviet Physics-Doklady 9, pp.538-540, Jan. 1965).

[40] R.R. Varshamov, "A class of codes for asymmetric channels and a problem from the additive theory of numbers", IEEE Trans. Inf. Theory, vol. IT-19, pp. 92-95, Jan. 1973.

[41] T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes", IEEE Trans. Inf. Theory, vol. IT-33, pp. 665-680, September 1987.

[42] T.J. Wagner, "A search technique for quasi-perfect codes", Inform. Control, vol. 9, pp. 94-99, 1966.

[43] J.H. Weber, C. de Vroedt, and D.E. Boekee, "A construction method for codes correcting asymmetric errors", Proceedings Eighth Symposium on Information Theory in the Benelux, Deventer, The Netherlands, pp. 203-207, May 1987.

[44] J.H. Weber, C. de Vroedt, and D.E. Boekee, "New upper bounds on the size of codes correcting asymmetric errors", IEEE Trans. Inf. Theory, vol. IT-33, pp. 434-437, May 1987.

[45] J.H. Weber, C. de Vroedt, and D.E. Boekee, "Bounds on the size of codes correcting unidirectional errors", Proceedings Ninth Symposium on Information Theory in the Benelux, Mierlo, The Netherlands, pp. 9-15, May 1988.

[46] J.H. Weber, C. de Vroedt, and D.E. Boekee, "Construction methods for codes correcting asymmetric or unidirectional errors", Abstracts of papers 1988 IEEE International Symposium on Information Theory, Kobe, Japan, pp. 120-121, June 1988.

[47] J.H. Weber, C. de Vroedt, and D.E. Boekee, "Codes correcting unidirectional errors", Proceedings 1988 Beijing International Workshop on Information Theory, Beijing, P.R. China, pp. BI7.1-BI7.4, July 1988.

[48] J.H. Weber, C. de Vroedt, and D.E. Boekee, "Bounds and constructions for binary codes of length less than 24 and asymmetric distance less than 6", IEEE Trans. Inf. Theory, vol. IT-34, pp. 1321-1331, September 1988.

[49] J.H. Weber, C. de Vroedt, and D.E. Boekee, "Conditions on block codes for correction/detection of errors of various types", Proceedings Tenth Symposium on Information Theory in the Benelux, Houthalen, Belgium, pp. 31-36, May 1989.

[50] J.H. Weber, C. de Vroedt, and D.E. Boekee, "Bounds and constructions for codes correcting unidirectional errors", IEEE Trans. Inf. Theory, vol. IT-35, July 1989.

[51] J.H. Weber, C. de Vroedt, and D.E. Boekee, "On linear codes correcting asymmetric errors", Proceedings Fourth Joint Swedish-USSR International Workshop on Information Theory, Gotland, Sweden, pp. 310-314, August-September 1989.

Appendix A

Derivation of the conditions for error correction and/or detection capabilities

In this appendix we prove the Theorems 1.1 and 1.2 that were stated in Section 1.4.

Theorem 1.1 A code $\mathcal{C}$ is t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED (with $0 \leq t_1 \leq t_2 \leq t_3$, $0 \leq d_1 \leq d_2 \leq d_3$, $t_i \leq d_i$) if and only if all $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $N(\mathbf{a},\mathbf{b}) \geq N(\mathbf{b},\mathbf{a})$ satisfy

$$\begin{cases} d(\mathbf{a},\mathbf{b}) \geq t_3 + d_2 + 1 & \wedge d(\mathbf{a},\mathbf{b}) \geq t_2 + d_3 + 1 & \text{if } N(\mathbf{b},\mathbf{a}) = 0 \\ d(\mathbf{a},\mathbf{b}) \geq t_3 + d_1 + 1 & \wedge d(\mathbf{a},\mathbf{b}) \geq t_1 + d_3 + 1 & \\ & \wedge N(\mathbf{a},\mathbf{b}) \geq d_3 + 1 & \text{if } 1 \leq N(\mathbf{b},\mathbf{a}) \leq t_3 \\ d(\mathbf{a},\mathbf{b}) \geq t_3 + d_1 + 1 & & \text{if } N(\mathbf{b},\mathbf{a}) \geq t_3 + 1 \end{cases}$$

Proof. From the definition of a t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED code it follows that we have to prove that for all $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ with $\mathbf{a} \neq \mathbf{b}$ and $N(\mathbf{a},\mathbf{b}) \geq N(\mathbf{b},\mathbf{a})$:

$$S(\mathbf{a}, t_1, t_2, t_3) \cap S(\mathbf{b}, d_1, d_2, d_3) = \emptyset \wedge S(\mathbf{b}, t_1, t_2, t_3) \cap S(\mathbf{a}, d_1, d_2, d_3) = \emptyset$$

$$\Longleftrightarrow$$

$$\begin{cases} d(\mathbf{a},\mathbf{b}) \geq t_3 + d_2 + 1 & \wedge d(\mathbf{a},\mathbf{b}) \geq t_2 + d_3 + 1 & \text{if } N(\mathbf{b},\mathbf{a}) = 0 \\ d(\mathbf{a},\mathbf{b}) \geq t_3 + d_1 + 1 & \wedge d(\mathbf{a},\mathbf{b}) \geq t_1 + d_3 + 1 & \\ & \wedge N(\mathbf{a},\mathbf{b}) \geq d_3 + 1 & \text{if } 1 \leq N(\mathbf{b},\mathbf{a}) \leq t_3 \\ d(\mathbf{a},\mathbf{b}) \geq t_3 + d_1 + 1 & & \text{if } N(\mathbf{b},\mathbf{a}) \geq t_3 + 1 \end{cases}$$

Without loss of generality we may assume that $\mathbf{a}$ and $\mathbf{b}$ satisfy

$$\begin{cases} a_i = b_i & \text{for } 1 \leq i \leq \alpha \\ a_i > b_i & \text{for } \alpha + 1 \leq i \leq \alpha + \beta \\ a_i < b_i & \text{for } \alpha + \beta + 1 \leq i \leq \alpha + \beta + \gamma = n \end{cases}$$

with $\alpha = n - N(\mathbf{a},\mathbf{b}) - N(\mathbf{b},\mathbf{a})$, $\beta = N(\mathbf{b},\mathbf{a})$, and $\gamma = N(\mathbf{a},\mathbf{b})$.

"$\Rightarrow$" Define $\mathbf{z} \in (GF(2))^n$ as

$$\begin{cases} z_i = a_i & \text{for } 1 \leq i \leq \alpha \\ z_i = b_i & \text{for } \alpha + 1 \leq i \leq \alpha + \beta \\ z_i = a_i & \text{for } \alpha + \beta + 1 \leq i \leq \alpha + \beta + \mu \\ z_i = b_i & \text{for } \alpha + \beta + \mu + 1 \leq i \leq \alpha + \beta + \gamma = n \end{cases}$$

where μ will be filled in in accordance with the case under consideration.

1. The case $N(\mathbf{b},\mathbf{a}) = 0$. Suppose $d(\mathbf{a},\mathbf{b}) \leq t_2 + d_3$ or $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_2$.

   (a) If $d(\mathbf{a},\mathbf{b}) \leq t_2 + d_3$,
   then $\mathbf{z}$ (with $\mu = \min\{\gamma, d_3\}$) $\in S_U(\mathbf{a}, t_2) \cap S_{As}(\mathbf{b}, d_3)$.

(b) If $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_2$,
then $\mathbf{z}$ (with $\mu = \min\{\gamma, t_3\}$) $\in S_U(\mathbf{a}, d_2) \cap S_{As}(\mathbf{b}, t_3)$.

2. The case $1 \leq N(\mathbf{b},\mathbf{a}) \leq t_3$. Suppose $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_1$ or $d(\mathbf{a},\mathbf{b}) \leq t_1 + d_3$ or $N(\mathbf{a},\mathbf{b}) \leq d_3$.

(a) If $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_1$ and $t_3 \leq \gamma$,
then $\mathbf{z}$ (with $\mu = t_3$) $\in S_{Sy}(\mathbf{a}, d_1) \cap S_{As}(\mathbf{b}, t_3)$.

(b) If $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_1$ and $t_3 > \gamma$,
then $\mathbf{z}$ (with $\mu = \gamma$) $\in S_{As}(\mathbf{a}, t_3) \cap S_{As}(\mathbf{b}, t_3)$.

(c) If $d(\mathbf{a},\mathbf{b}) \leq t_1 + d_3$ and $d_3 \leq \gamma$,
then $\mathbf{z}$ (with $\mu = d_3$) $\in S_{Sy}(\mathbf{a}, t_1) \cap S_{As}(\mathbf{b}, d_3)$.

(d) If $d(\mathbf{a},\mathbf{b}) \leq t_1 + d_3$ and $d_3 > \gamma$,
then $\mathbf{z}$ (with $\mu = \gamma$) $\in S_{As}(\mathbf{a}, t_3) \cap S_{As}(\mathbf{b}, d_3)$.

(e) If $N(\mathbf{a},\mathbf{b}) \leq d_3$,
then $\mathbf{z}$ (with $\mu = \gamma$) $\in S_{As}(\mathbf{a}, t_3) \cap S_{As}(\mathbf{b}, d_3)$.

3. The case $N(\mathbf{b},\mathbf{a}) \geq t_3 + 1$. Suppose $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_1$.

(a) If $d(\mathbf{a},\mathbf{b}) \leq t_3 + d_1$,
then $\mathbf{z}$ (with $\mu = t_3$) $\in S_{Sy}(\mathbf{a}, d_1) \cap S_{As}(\mathbf{b}, t_3)$.

Hence we have shown that

$$\mathbf{z} \in S(\mathbf{a}, t_1, t_2, t_3) \cap S(\mathbf{b}, d_1, d_2, d_3) \vee \mathbf{z} \in S(\mathbf{b}, t_1, t_2, t_3) \cap S(\mathbf{a}, d_1, d_2, d_3)$$

for each case, which contradicts the assumption that these two intersections of sets are both empty.

"$\Leftarrow$" Suppose there is a $\mathbf{z} \in (GF(2))^n$ such that

$$\mathbf{z} \in S(\mathbf{a}, t_1, t_2, t_3) \cap S(\mathbf{b}, d_1, d_2, d_3) \vee \mathbf{z} \in S(\mathbf{b}, t_1, t_2, t_3) \cap S(\mathbf{a}, d_1, d_2, d_3).$$

Again, we shall find a contradiction for each case. This will only be shown for

$$\mathbf{z} \in S(\mathbf{a}, t_1, t_2, t_3) \cap S(\mathbf{b}, d_1, d_2, d_3),$$

since it can be shown in a completely analogous way for

$$\mathbf{z} \in S(\mathbf{b}, t_1, t_2, t_3) \cap S(\mathbf{a}, d_1, d_2, d_3).$$

1. The case $N(\mathbf{b},\mathbf{a}) = 0$.

   (a) If $N(\mathbf{a},\mathbf{z}) = 0$,
   then $d(\mathbf{a},\mathbf{b}) = N(\mathbf{a},\mathbf{b}) \leq N(\mathbf{z},\mathbf{b}) \leq d(\mathbf{z},\mathbf{b}) \leq d_3$.

   (b) If $N(\mathbf{a},\mathbf{z}) \geq 1$,
   then $d(\mathbf{a},\mathbf{b}) \leq d(\mathbf{a},\mathbf{z}) + d(\mathbf{z},\mathbf{b}) \leq t_2 + d_3$.

2. The case $1 \leq N(\mathbf{b},\mathbf{a}) \leq t_3$.

   (a) If $N(\mathbf{a},\mathbf{z}) \geq 1$ and $N(\mathbf{z},\mathbf{a}) \geq 1$,
   then $d(\mathbf{a},\mathbf{b}) \leq d(\mathbf{a},\mathbf{z}) + d(\mathbf{z},\mathbf{b}) \leq t_1 + d_3$.

   (b) If $N(\mathbf{a},\mathbf{z}) = 0$,
   then $N(\mathbf{a},\mathbf{b}) \leq N(\mathbf{z},\mathbf{b}) \leq d(\mathbf{z},\mathbf{b}) \leq d_3$.

   (c) If $N(\mathbf{z},\mathbf{a}) = 0$ and $N(\mathbf{z},\mathbf{b}) \geq 1$ and $N(\mathbf{b},\mathbf{z}) \geq 1$,
   then $d(\mathbf{a},\mathbf{b}) \leq d(\mathbf{a},\mathbf{z}) + d(\mathbf{z},\mathbf{b}) \leq t_2 + d_1$.

   (d) If $N(\mathbf{z},\mathbf{a}) = 0$ and $N(\mathbf{z},\mathbf{b}) = 0$,
   then $N(\mathbf{a},\mathbf{b}) \leq N(\mathbf{a},\mathbf{z}) \leq d(\mathbf{a},\mathbf{z}) \leq t_2$.

   (e) If $N(\mathbf{z},\mathbf{a}) = 0$ and $N(\mathbf{b},\mathbf{z}) = 0$,
   then $N(\mathbf{b},\mathbf{a}) \leq N(\mathbf{b},\mathbf{z}) = 0$.

3. The case $N(\mathbf{b},\mathbf{a}) \geq t_3 + 1$.

   (a) If $N(\mathbf{b},\mathbf{z}) \geq 1$ and $N(\mathbf{z},\mathbf{b}) \geq 1$,
   then $d(\mathbf{a},\mathbf{b}) \leq d(\mathbf{a},\mathbf{z}) + d(\mathbf{z},\mathbf{b}) \leq t_3 + d_1$.

   (b) If $N(\mathbf{z},\mathbf{b}) = 0$,
   then $N(\mathbf{b},\mathbf{a}) \leq N(\mathbf{a},\mathbf{b}) \leq N(\mathbf{a},\mathbf{z}) \leq d(\mathbf{a},\mathbf{z}) \leq t_3$.

   (c) If $N(\mathbf{b},\mathbf{z}) = 0$,
   then $N(\mathbf{b},\mathbf{a}) \leq N(\mathbf{z},\mathbf{a}) \leq d(\mathbf{z},\mathbf{a}) \leq t_3$. □

Theorem 1.2 Any t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED code (with $0 \le t_1 \le t_2 \le t_3$, $0 \le d_1 \le d_2 \le d_3$, $t_i \le d_i$) is also a t_1'-SyEC t_2'-UEC t_3'-AsEC d_1'-SyED d_2'-UED d_3'-AsED code with

$$\begin{aligned} t_1' &= \max\{t_1, t_3 + d_1 - d_3\}, \\ t_2' &= \max\{t_2, t_3 + d_2 - d_3\}, \\ t_3' &= t_3, \\ d_1' &= \max\{d_1, \min\{t_3 + 1, t_1 + d_3 - t_3\}\}, \\ d_2' &= \max\{d_2, t_2 + d_3 - t_3\}, \\ d_3' &= d_3. \end{aligned}$$

Proof. First, observe that $t_2' + d_3 = t_3 + d_2'$. Next, since

$$\begin{aligned} 0 &\le t_1 \le t_1' = \max\{t_1, t_3 + d_1 - d_3\} \le \max\{t_2, t_2' - d_2' + d_1\} \\ &\le t_2' = \max\{t_2, t_3 + d_2 - d_3\} \le t_3 = t_3', \\ 0 &\le d_1 \le d_1' \le \max\{d_1, d_3 + t_1 - t_3\} \le \max\{d_2, d_2' - t_2' + t_1\} \\ &\le d_2' = \max\{d_2, d_3 + t_2 - t_3\} \le d_3 = d_3', \\ t_1' &= \max\{t_1, t_3 + d_1 - d_3\} \le d_1 \le d_1', \\ t_2' &= \max\{t_2, t_3 + d_2 - d_3\} \le d_2 \le d_2', \\ t_3' &= t_3 \le d_3 = d_3', \end{aligned}$$

we may apply Theorem 1.1 to obtain necessary and sufficient conditions for a code to be t_1'-SyEC t_2'-UEC t_3'-AsEC d_1'-SyED d_2'-UED d_3'-AsED. Finally, we now show that these conditions are implied by the necessary and sufficient conditions for a code to be t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED.

1. The case $N(\mathbf{b}, \mathbf{a}) = 0$.

   (a) If $t_3 + d_2 \ge t_2 + d_3$,
   then $d(\mathbf{a}, \mathbf{b}) \ge t_3 + d_2 + 1 = t_3' + d_2' + 1 = t_2' + d_3' + 1$.

   (b) If $t_3 + d_2 < t_2 + d_3$,
   then $d(\mathbf{a}, \mathbf{b}) \ge t_2 + d_3 + 1 = t_2' + d_3' + 1 = t_3' + d_2' + 1$.

2. The case $1 \le N(\mathbf{b}, \mathbf{a}) \le t_3$.

   (a) If $t_3 + d_1 \ge t_1 + d_3$,
   then $d(\mathbf{a}, \mathbf{b}) \ge t_3 + d_1 + 1 = t_3' + d_1' + 1 = t_1' + d_3' + 1$ and $N(\mathbf{a}, \mathbf{b}) \ge d_3 + 1 = d_3' + 1$.

(b) If $t_3 + d_1 < t_1 + d_3$,
then $d(\mathbf{a}, \mathbf{b}) \geq t_1 + d_3 + 1 = t'_1 + d'_3 + 1 \geq t'_3 + d'_1 + 1$ and $N(\mathbf{a}, \mathbf{b}) \geq d_3 + 1 = d'_3 + 1$.

3. The case $N(\mathbf{b}, \mathbf{a}) \geq t_3 + 1$.

(a) If $t_3 + 1 \leq d_1$ or $t_1 + d_3 \leq t_3 + d_1$,
then $d(\mathbf{a}, \mathbf{b}) \geq t_3 + d_1 + 1 = t'_3 + d'_1 + 1$.

(b) If $t_3 + 1 > d_1$ and $t_1 + d_3 > t_3 + d_1$,
then $d(\mathbf{a}, \mathbf{b}) \geq 2t_3 + 2 \geq t'_3 + d'_1 + 1$. □

Appendix B

Bounds for optimal SyEC, UEC, and AsEC codes

In this appendix we present bounds on the largest cardinality $A_{Sy}(n,t)$ of a t-SyEC code of length n, on the largest cardinality $A_U(n,t)$ of a t-UEC code of length n, and on the largest cardinality $A_{As}(n,t)$ of a t-AsEC code of length n, all in the area $t \leq 4$ and $n \leq 23$. Upper bounds for such codes were discussed in Chapters 2, 3, and 4, while constructive lower bounds were treated in Chapters 5 and 6.

In Tables B.1, B.2, B.3, and B.4 the bounds on $A_{Sy}(n,t)$ are all taken from the tables in [29], except the lower bounds for $t = 1$ and $16 \leq n \leq 18$. Recently, van Os ([26]) has shown that $A_{Sy}(18,1) \geq 10496$ (hence $A_{Sy}(17,1) \geq 5248$), and Romanov ([32]) has shown that $A_{Sy}(16,1) \geq 2720$.

All upper and many lower bounds on $A_U(n,t)$ and $A_{As}(n,t)$ in Tables B.1, B.2, B.3, and B.4 follow from results presented in this thesis. We now list these results, where the letters correspond to the references in the tables.

g1 $A_{Sy}(n,t) = A_U(n,t) \leq A_{As}(n,t)$ for $1 = t \leq n$
(see 1.32 and 2.5).

gt $A_{Sy}(n,t) \leq A_U(n,t) \leq A_{As}(n,t)$ for $2 \leq t \leq n$
(see 1.32).

us $A_U(n,t) = \begin{cases} 1 & \text{for } t \leq n \leq t+1 \text{ and } t \geq 1 \\ 2 & \text{for } t+2 \leq n \leq 2t+2 \text{ and } t \geq 1 \\ 4 & \text{for } n = 2t+3 \text{ and } t \geq 1 \end{cases}$
(Theorem 2.10).

as $A_{As}(n,t) = \begin{cases} 1 & \text{for } n = t \text{ and } t \geq 1 \\ 2 & \text{for } t+1 \leq n \leq 2t+1 \text{ and } t \geq 1 \\ 4 & \text{for } n = 2t+2 \text{ and } t \geq 1 \end{cases}$
(Theorem 2.11).

B1 $A_{As}(n,t) \leq (t+1)A_{Sy}(n,t)$ for $1 \leq t \leq n$
(Theorem 2.2).

B2 $A_{As}(n,t) \leq A_{Sy}(n+t,t)$ for $1 \leq t \leq n$
(Theorem 2.3).

B3 $A_{U}(n,t) \leq tA_{Sy}(n,t)$ for $1 \leq t \leq n$
(Theorem 2.4).

B4 $A_{U}(n,t) \leq A_{Sy}(n+t-1,t)$ for $1 \leq t \leq n$
(Theorem 2.6).

B5 $(3t+1)A_{U}(n,t) \geq (t+1)A_{As}(n,t)$ for $1 \leq t \leq n$
(Theorem 2.8).

ia $A_{As}(n,t) \leq I^{u}_{As}(n,t)$ for $1 \leq t \leq n$
(Asymmetric Integer Programming Bound, Theorem 3.10).

iu $A_{U}(n,t) \leq I^{u}_{U}(n,t)$ for $1 \leq t \leq n$
(Unidirectional Integer Programming Bound, Theorem 3.11).

sa Upper bound on $A_{As}(n,t)$ derived by sharpening the Asymmetric Integer Programming Bound using combinatorial arguments
(Section 4.2).

su Upper bound on $A_{U}(n,t)$ derived by sharpening the Unidirectional Integer Programming Bound using combinatorial arguments
(Section 4.3).

ca Lower bound on $A_{As}(n,t)$ derived by construction of a t-AsEC code of length n
(Section 6.2, Section 7.4).

cu Lower bound on $A_{U}(n,t)$ derived by construction of a t-UEC code of length n
(Section 6.3).

Other lower bounds follow from codes that can be found in literature:

Va Code obtained by Varshamov ([40]).

CR Code obtained by Constantin and Rao ([7]).

KF Code obtained by Kim and Freiman ([15]).

DP Code obtained by Delsarte and Piret ([8]).

Sh Code obtained by Shiozaki ([34]); Shiozaki's method can be considered as the $m = 1$ case of the asymmetric construction method presented in Chapter 5 of this thesis.

rV Code obtained repetition of the 1-AsEC Varshamov code of length 5 and size 6.

rK Code obtained repetition of the 1-AsEC Kim-Freiman code of length 6 and size 12.

Very recently Saitoh *et al.* ([33]) have announced new construction methods for AsEC and UEC codes. These codes result in numerous new lower bounds on $A_{As}(n,t)$ and $A_U(n,t)$ in the area $14 \leq n \leq 23$ and $3 \leq t \leq 6$.

n	$A_{Sy}(n,1)$	$A_{As}(n,1)$
1	1	${}^{as}1^{as}$
2	1	${}^{as}2^{as}$
3	2	${}^{as}2^{as}$
4	2	${}^{as}4^{as}$
5	4	${}^{Va}6^{ia}$
6	8	${}^{KF}12^{ia}$
7	16	${}^{DP}18^{ia}$
8	20	${}^{DP}36^{ia}$
9	40	${}^{DP}62^{ia}$
10	72 – 79	${}^{DP}108$ – 117^{ia}
11	144 – 158	${}^{DP}174$ – 210^{ia}
12	256	${}^{Va}316$ – 410^{ia}
13	512	${}^{Va}586$ – 786^{ia}
14	1024	${}^{Va}1096$ – 1500^{ia}
15	2048	${}^{Va}2048$ – 2828^{ia}
16	2720 – 3276	${}^{Va}3856$ – 5430^{ia}
17	5248 – 6552	${}^{CR}7296$ – 10374^{ia}
18	10496 – 13104	${}^{Va}13798$ – 19898^{ia}
19	20480 – 26208	${}^{Va}26216$ – 38008^{ia}
20	36864 – 43690	${}^{Va}49940$ – 73174^{ia}
21	73728 – 87380	${}^{Va}95326$ – 140798^{ia}
22	147456 – 173784	${}^{Va}182362$ – 271953^{ia}
23	294912 – 344636	${}^{Va}349536$ – 523586^{ia}

Table B.1: Bounds on $A_{Sy}(n,1)$ and $A_{As}(n,1)$ for $1 \leq n \leq 23$.

n	$A_{Sy}(n,2)$	$A_U(n,2)$	$A_{As}(n,2)$
2	1	${}^{us}1^{us}$	${}^{as}1^{as}$
3	1	${}^{us}1^{us}$	${}^{as}2^{as}$
4	1	${}^{us}2^{us}$	${}^{as}2^{as}$
5	2	${}^{us}2^{us}$	${}^{as}2^{as}$
6	2	${}^{us}2^{us}$	${}^{as}4^{as}$
7	2	${}^{us}4^{us}$	${}^{as}4^{ia}$
8	4	${}^{cu}6^{B4}$	${}^{DP}7^{ia}$
9	6	${}^{cu}10^{su}$	${}^{ca}12^{ia}$
10	12	${}^{cu}16 - 18^{gt}$	${}^{DP}18^{sa}$
11	24	${}^{cu}26 - 32^{gt}$	${}^{DP}30 - 32^{ia}$
12	32	${}^{cu}52 - 61^{iu}$	${}^{ca}54 - 63^{ia}$
13	64	${}^{cu}92 - 114^{gt}$	${}^{DP}98 - 114^{ia}$
14	128	${}^{cu}184 - 218^{gt}$	${}^{DP}186 - 218^{ia}$
15	256	${}^{gt}256 - 340^{B4}$	${}^{ca}266 - 398^{ia}$
16	$256 - 340$	${}^{cu}352 - 680^{B4}$	${}^{ca}364 - 739^{ia}$
17	$512 - 680$	${}^{cu}640 - 1277^{iu}$	${}^{ca}647 - 1279^{ia}$
18	$1024 - 1288$	${}^{cu}1216 - 2372^{B4}$	${}^{ca}1218 - 2380^{ia}$
19	$2048 - 2372$	${}^{gt}2048 - 4096^{B4}$	${}^{ca}2050 - 4242^{ia}$
20	$2560 - 4096$	${}^{gt}2560 - 6942^{B4}$	${}^{ca}2564 - 8069^{ia}$
21	$4096 - 6942$	${}^{cu}4224 - 13774^{B4}$	${}^{ca}4251 - 14374^{ia}$
22	$8192 - 13774$	${}^{cu}8448 - 24106^{B4}$	${}^{ca}8450 - 26679^{ia}$
23	$16384 - 24106$	${}^{gt}16384 - 48212^{B3}$	${}^{ca}16388 - 50200^{ia}$

Table B.2: Bounds on $A_{Sy}(n,2)$, $A_U(n,2)$, and $A_{As}(n,2)$ for $2 \leq n \leq 23$.

n	$A_{Sy}(n,3)$	$A_U(n,3)$	$A_{As}(n,3)$
3	1	${}^{us}1^{us}$	${}^{as}1^{as}$
4	1	${}^{us}1^{us}$	${}^{as}2^{as}$
5	1	${}^{us}2^{us}$	${}^{as}2^{as}$
6	1	${}^{us}2^{us}$	${}^{as}2^{as}$
7	2	${}^{us}2^{us}$	${}^{as}2^{as}$
8	2	${}^{us}2^{us}$	${}^{as}4^{as}$
9	2	${}^{us}4^{us}$	${}^{as}4^{ia}$
10	2	${}^{us}4^{B4}$	${}^{rV}6^{ia}$
11	4	${}^{cu}7^{su}$	${}^{ca}8^{ia}$
12	4	${}^{cu}10^{su}$	${}^{rK}12^{ia}$
13	8	${}^{cu}14 - 18^{gt}$	${}^{ca}18^{ia}$
14	16	${}^{cu}22 - 34^{gt}$	${}^{ca}30 - 34^{sa}$
15	32	${}^{gt}32 - 50^{gt}$	${}^{ca}44 - 50^{ia}$
16	36 – 37	${}^{cu}53 - 90^{gt}$	${}^{ca}72 - 90^{sa}$
17	64 – 72	${}^{cu}97 - 168^{gt}$	${}^{ca}122 - 168^{ia}$
18	128 – 144	${}^{cu}188 - 320^{gt}$	${}^{ca}234 - 320^{ia}$
19	256 – 279	${}^{cu}376 - 616^{gt}$	${}^{ca}450 - 616^{ia}$
20	512	${}^{cu}737 - 1142^{iu}$	${}^{ca}860 - 1144^{ia}$
21	1024	${}^{cu}1474 - 2134^{gt}$	${}^{ca}1628 - 2134^{ia}$
22	2048	${}^{cu}2588 - 4114^{iu}$	${}^{Sh}3072 - 4116^{ia}$
23	4096	${}^{gt}4096 - 7346^{gt}$	${}^{gt}4096 - 7346^{ia}$

Table B.3: Bounds on $A_{Sy}(n,3)$, $A_U(n,3)$, and $A_{As}(n,3)$ for $3 \leq n \leq 23$.

n	$A_{Sy}(n,4)$	$A_U(n,4)$	$A_{As}(n,4)$
4	1	$^{us}1^{us}$	$^{as}1^{as}$
5	1	$^{us}1^{us}$	$^{as}2^{as}$
6	1	$^{us}2^{us}$	$^{as}2^{as}$
7	1	$^{us}2^{us}$	$^{as}2^{as}$
8	1	$^{us}2^{us}$	$^{as}2^{as}$
9	2	$^{us}2^{us}$	$^{as}2^{as}$
10	2	$^{us}2^{us}$	$^{as}4^{as}$
11	2	$^{us}4^{us}$	$^{as}4^{ia}$
12	2	$^{us}4^{gt}$	$^{as}4^{ia}$
13	2	$^{cu}6^{gt}$	$^{ca}6^{ia}$
14	4	$^{cu}8^{gt}$	$^{ca}8^{sa}$
15	4	$^{cu}10 - 12^{gt}$	$^{ca}12^{ia}$
16	6	$^{cu}11 - 16^{gt}$	$^{ca}16^{sa}$
17	10	$^{cu}18 - 26^{gt}$	$^{ca}26^{sa}$
18	20	$^{cu}28 - 44^{gt}$	$^{ca}36 - 44^{ia}$
19	40	$^{gt}40 - 74^{gt}$	$^{ca}46 - 74^{sa}$
20	$40 - 48$	$^{gt}40 - 133^{gt}$	$^{ca}54 - 133^{sa}$
21	$48 - 88$	$^{gt}48 - 229^{gt}$	$^{ca}62 - 229^{ia}$
22	$64 - 150$	$^{cu}82 - 423^{gt}$	$^{ca}88 - 423^{ia}$
23	$128 - 280$	$^{gt}128 - 745^{gt}$	$^{ca}133 - 745^{ia}$

Table B.4: Bounds on $A_{Sy}(n,4)$, $A_U(n,4)$, and $A_{As}(n,4)$ for $4 \leq n \leq 23$.

Appendix C

Bounds for optimal linear SyEC and AsEC codes

In this appendix we present the exact values for the largest dimension $K_{Sy}(n,t)$ of a linear t-SyEC code of length n and bounds on the largest dimension $K_{As}(n,t)$ of a linear t-AsEC code of length n in the area $t \leq 4$ and $n \leq 23$. Such codes were discussed in Chapter 7. In Tables C.1 and C.2 the values of $K_{Sy}(n,t)$ can all be derived from the tables in [41]. For $n \leq 3t+3$ the values of $K_{As}(n,t)$ follow from Theorem 7.2. For $3t+4 \leq n \leq 23$ the bounds on $K_{As}(n,t)$ were obtained as follows:

1. The case $t = 1$. We have $K_{As}(n,1) = K_{Sy}(n,1)$ for $7 \leq n \leq 23$ from Theorem 7.12.

2. The case $t = 2$. We have $K_{As}(n,2) = K_{Sy}(n,2)$ for $10 \leq n \leq 23$ from Theorem 7.13.

3. The case $t = 3$. We have

   $$5 = K_{Sy}(15,3) \leq K_{As}(15,3) \leq \lfloor \log_2(A_{As}(15,3)) \rfloor \leq \lfloor \log_2(50) \rfloor = 5.$$

   Further

   $$6 \leq K_{As}(16,3) \leq K_{As}(15,3) + 1 = 6$$

   by Theorem 7.14 and

   $$10 = K_{Sy}(21,3) \leq K_{As}(21,3) \leq K_{As}(15,3) + 6 = 11.$$

   Suppose $K_{As}(21,3) = 11$, then $K_{As}(21,3) - K_{As}(15,3) = 11 - 5 = 6$ and $K_{As}(21,3) = 11 > 10 = K_{Sy}(21,3)$, which contradicts Theorem 7.5. Hence

   $$K_{As}(21,3) = 10.$$

All other bounds for $13 \leq n \leq 23$ now follow from simple arguments as $K_{As}(n,3) \geq K_{Sy}(n,3)$ and $K_{As}(n,3) \leq K_{As}(n-1,3)+1$.

4. The case $t = 4$. All bounds for $16 \leq n \leq 23$ follow from simple arguments as $K_{As}(n,4) \geq K_{As}(n-1,4)$, $K_{As}(n,4) \geq K_{Sy}(n,4)$, and $K_{As}(n,4) \leq \lfloor \log_2(A_{As}(n,4)) \rfloor$.

n	$K_{Sy}(n,1)$	$K_{As}(n,1)$	$K_{Sy}(n,2)$	$K_{As}(n,2)$
1	0	0		
2	0	1	0	0
3	1	1	0	1
4	1	2	0	1
5	2	2	1	1
6	3	3	1	2
7	4	4	1	2
8	4	4	2	2
9	5	5	2	3
10	6	6	3	3
11	7	7	4	4
12	8	8	4	4
13	9	9	5	5
14	10	10	6	6
15	11	11	7	7
16	11	11	8	8
17	12	12	9	9
18	13	13	9	9
19	14	14	10	10
20	15	15	11	11
21	16	16	12	12
22	17	17	13	13
23	18	18	14	14

Table C.1: Exact values of $K_{Sy}(n,t)$ and $K_{As}(n,t)$ for $1 \leq t \leq 2$ and $t \leq n \leq 23$.

n	$K_{Sy}(n,3)$	$K_{As}(n,3)$	$K_{Sy}(n,4)$	$K_{As}(n,4)$
3	0	0		
4	0	1	0	0
5	0	1	0	1
6	0	1	0	1
7	1	1	0	1
8	1	2	0	1
9	1	2	1	1
10	1	2	1	2
11	2	2	1	2
12	2	3	1	2
13	3	$3-4$	1	2
14	4	$4-5$	2	2
15	5	5	2	3
16	5	6	2	$3-4$
17	6	$6-7$	3	$3-4$
18	7	$7-8$	3	$3-5$
19	8	$8-9$	4	$4-6$
20	9	$9-10$	5	$5-7$
21	10	10	5	$5-7$
22	11	11	6	$6-8$
23	12	12	7	$7-9$

Table C.2: Exact values of $K_{Sy}(n,t)$ and bounds for $K_{As}(n,t)$ for $3 \leq t \leq 4$ and $t \leq n \leq 23$.

Appendix D

Trial and error codes

In this appendix we present codes obtained by trial and error, in a way described in Section 6.1. These codes are listed in Tables D.1-D.7.

	$n = 9$
$\mathbf{c}_1$	000000000
$\mathbf{c}_2$	111000000
$\mathbf{c}_3$	000111000
$\mathbf{c}_4$	100100011
$\mathbf{c}_5$	010010101
$\mathbf{c}_6$	001001110
$\mathbf{c}_7$	011110010
$\mathbf{c}_8$	101011001
$\mathbf{c}_9$	110101100
$\mathbf{c}_{10}$	111000111
$\mathbf{c}_{11}$	000111111
$\mathbf{c}_{12}$	111111111

Table D.1: 2-AsEC code of length 9 and size 12.

	$n = 11$	$n = 13$	$n = 14$
$\mathbf{c}_1$	00000000000	0000000000000	00000000000000
$\mathbf{c}_2$	11110000000	0001001010100	10010010000001
$\mathbf{c}_3$	00001111000	0000110101000	01001000010100
$\mathbf{c}_4$	10000001111	1100010010001	00100101001000
$\mathbf{c}_5$	00110110101	1010001100010	11111100000000
$\mathbf{c}_6$	11011011001	0111111000000	11000011001100
$\mathbf{c}_7$	11101100110	0110000111100	11000000110011
$\mathbf{c}_8$	11111111111	0001100110011	00110010101010
$\mathbf{c}_9$		0000011001111	00110001010101
$\mathbf{c}_{10}$		1101010101010	00001110011001
$\mathbf{c}_{11}$		1100101100101	00001101100110
$\mathbf{c}_{12}$		1011001011001	10101011110000
$\mathbf{c}_{13}$		1010110010110	10101000001111
$\mathbf{c}_{14}$		0011101101110	10010110010110
$\mathbf{c}_{15}$		0101110011101	10010101101001
$\mathbf{c}_{16}$		1111011110100	01100111000011
$\mathbf{c}_{17}$		1000111111011	01100100111100
$\mathbf{c}_{18}$		1111111111111	01011010100101
$\mathbf{c}_{19}$			01011001011010
$\mathbf{c}_{20}$			00000011111111
$\mathbf{c}_{21}$			00111100110011
$\mathbf{c}_{22}$			00111111001100
$\mathbf{c}_{23}$			11001110101010
$\mathbf{c}_{24}$			11001101010101
$\mathbf{c}_{25}$			11110001100110
$\mathbf{c}_{26}$			11110010011001
$\mathbf{c}_{27}$			01111001111101
$\mathbf{c}_{28}$			10011111011011
$\mathbf{c}_{29}$			11100110110111
$\mathbf{c}_{30}$			11111111111111

Table D.2: 3-AsEC codes of length 11 and size 8, of length 13 and size 18, and of length 14 and size 30.

	$n = 13$	$n = 14$
$\mathbf{c}_1$	0000000000000	00000000000000
$\mathbf{c}_2$	1111100000000	11111000000000
$\mathbf{c}_3$	0000000011111	00000111110000
$\mathbf{c}_4$	0011111111000	10000100001111
$\mathbf{c}_5$	1100011100111	01101010111001
$\mathbf{c}_6$	1111111111111	01110111001110
$\mathbf{c}_7$		10011001110111
$\mathbf{c}_8$		11111111111111

Table D.3: 4-AsEC codes of length 13 and size 6 and of length 14 and size 8.

	$n = 15$	$n = 16$
$\mathbf{c}_1$	000000000000000	0000000000000000
$\mathbf{c}_2$	111110000000000	0101000000010101
$\mathbf{c}_3$	000001111100000	0010100011001000
$\mathbf{c}_4$	100001000011110	0000011100100010
$\mathbf{c}_5$	011000110011001	1111111000000000
$\mathbf{c}_6$	000110001110101	1100000111110000
$\mathbf{c}_7$	110100101101010	1010000100001111
$\mathbf{c}_8$	011011010100110	0011001011100011
$\mathbf{c}_9$	100111110001101	0000111110011100
$\mathbf{c}_{10}$	101010011111011	0111100101011001
$\mathbf{c}_{11}$	011101101010111	1001110101100110
$\mathbf{c}_{12}$	111111111111111	1100011000111011
$\mathbf{c}_{13}$		1110100011111110
$\mathbf{c}_{14}$		0101011111110101
$\mathbf{c}_{15}$		1011111110001011
$\mathbf{c}_{16}$		1111111111111111

Table D.4: 4-AsEC codes of length 15 and size 12 and of length 16 and size 16.

	$n = 8$	$n = 9$
$\mathbf{c}_1$	00100100	000000011
$\mathbf{c}_2$	01001001	111000000
$\mathbf{c}_3$	11110000	000111000
$\mathbf{c}_4$	10001110	100100110
$\mathbf{c}_5$	00111011	010010101
$\mathbf{c}_6$	11010111	110011010
$\mathbf{c}_7$		101110001
$\mathbf{c}_8$		011110110
$\mathbf{c}_9$		110101101
$\mathbf{c}_{10}$		101011111

Table D.5: 2-UEC codes of length 8 and size 6 and of length 9 and size 10.

	$n = 11$	$n = 12$
$\mathbf{c}_1$	11000000000	000001100000
$\mathbf{c}_2$	00111100000	111100000000
$\mathbf{c}_3$	10100011100	000000001111
$\mathbf{c}_4$	01010010011	000111010001
$\mathbf{c}_5$	10001101011	011011001010
$\mathbf{c}_6$	01111001110	010100110110
$\mathbf{c}_7$	11011111101	101101000111
$\mathbf{c}_8$		110111011100
$\mathbf{c}_9$		001110111011
$\mathbf{c}_{10}$		111011110111

Table D.6: 3-UEC codes of length 11 and size 7 and of length 12 and size 10.

	$n = 13$	$n = 14$	$n = 15$
$\mathbf{c}_1$	1110000000000	11000000000000	111110000000000
$\mathbf{c}_2$	0001111100000	00111110000000	000001111100000
$\mathbf{c}_3$	1001000011110	10100001111000	100001000011110
$\mathbf{c}_4$	0110110011001	01010001000111	011000110011001
$\mathbf{c}_5$	1100101100111	10011100110110	000110001110101
$\mathbf{c}_6$	0111011111111	01101010101101	110100101101010
$\mathbf{c}_7$		00110111011011	011011010100110
$\mathbf{c}_8$		11001111111111	100111110001101
$\mathbf{c}_9$			101010011111011
$\mathbf{c}_{10}$			011101101010111

Table D.7: 4-UEC codes of length 13 and size 6, of length 14 and size 8, and of length 15 and size 10.

Appendix E

Optimal choices for the tails in the construction method

In Section 5.1 a construction method was presented to obtain AsEC or UEC codes. The size of such a code depends on the choice for the tails $\mathbf{a}_i$ in the method. In Section 5.2 algorithms were given to get optimal choices for a given initial code C_1 and a fixed tail-length m. In this appendix we list in Tables E.1-E.9 such optimal choices for certain initial codes as treated in Sections 6.2 and 6.3. Further we also provide the cardinalities of the corresponding sets $\mathcal{T}_i(\mathbf{a}_i)$ or $\mathcal{S}_i(\mathbf{a}_i)$. Note that $|\mathcal{T}_i(\mathbf{a}_i)|$ equals the number of codewords of weight i in the resulting code C_2.

	$C_1 = S_{19}$							
	$m = 1$		$m = 2$		$m = 3$		$m = 4$	
i	$\mathbf{a}_i$	$\lvert T_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert T_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert T_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert T_i(\mathbf{a}_i)\rvert$
0	0	1	00	1	000	1	0000	1
1	0	0	01	0	010	0	0100	0
2	0	0	11	0	110	0	1100	0
3	0	0	11	5	110	5	1100	5
4	1	15	10	10	100	5	1000	0
5	0	57	00	47	000	42	0000	42
6	0	115	00	85	000	70	0000	70
7	1	105	10	60	100	30	0000	15
8	1	198	10	120	100	60	0000	15
9	0	250	00	130	000	70	0000	70
10	0	222	00	102	000	42	0000	42
11	1	114	10	36	100	18	0100	0
12	1	75	10	30	100	15	1100	5
13	0	45	00	15	100	5	1000	0
14	0	15	00	5	000	0	0000	0
15	1	5	00	1	000	1	0000	1
16	0	0	00	0	000	0		
17	0	0	00	0				
18	0	0						
$\sum$		1217		647		364		266

Table E.1: An optimal choice for the $\mathbf{a}_i$ in applying the asymmetric construction method to the 2-SyEC code S_{19} with $m = 1, 2, 3, 4$.

	$C_1 = \mathcal{W}'_{23}$			
	$m = 1$		$m = 2$	
i	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$
0	0	1	00	1
1	0	0	01	0
2	0	0	11	0
3	0	0	11	3
4	1	20	01	17
5	0	64	00	49
6	0	188	00	137
7	0	320	01	220
8	1	660	01	425
9	0	960	00	565
10	0	1308	00	713
11	0	1408	01	704
12	1	1308	01	595
13	0	960	01	395
14	0	660	11	220
15	0	320	11	137
16	1	188	01	51
17	0	64	00	15
18	0	20	00	3
19	0	0	01	0
20	0	0	11	0
21	0	0	11	1
22	1	1		
$\sum$		8450		4251

Table E.2: An optimal choice for the $\mathbf{a}_i$ in applying the asymmetric construction method to the 2-SyEC code $\mathcal{W}'_{23}$ with $m = 1, 2$.

	$\mathcal{C}_1 = \mathcal{G}_{23}$							
	$m=2$		$m=3$		$m=4$		$m=5$	
i	$\mathbf{a}_i$	$\lvert\overline{\mathcal{T}}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\overline{\mathcal{T}}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\overline{\mathcal{T}}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\overline{\mathcal{T}}_i(\mathbf{a}_i)\rvert$
0	00	1	000	1	0000	1	01111	0
1	01	0	001	0	0001	0	01111	0
2	11	0	011	0	0011	0	01111	0
3	11	0	111	0	0111	0	01111	1
4	11	0	111	5	0111	4	01111	3
5	11	21	011	16	0111	12	11111	0
6	01	56	011	40	1111	0	11111	18
7	01	120	111	0	1111	48	11110	30
8	11	0	111	120	1110	72	11100	42
9	11	280	110	160	1100	88	11000	46
10	10	336	100	176	1000	88	10000	42
11	00	336	000	160	0000	72	00000	30
12	00	280	000	120	0000	48	00000	18
13	10	0	100	0	1000	0	10000	0
14	10	120	100	40	1000	12	10000	3
15	00	56	000	16	0000	4	00000	1
16	00	21	000	5	0000	1	00000	0
17	01	0	001	0	0000	0	00000	0
18	11	0	011	0	0000	0	00000	0
19	11	0	111	0	0000	0		
20	11	0	111	1				
21	11	1						
$\sum$		1628		860		450		234

Table E.3: An optimal choice for the $\mathbf{a}_i$ in applying the asymmetric construction method to the 3-SyEC code $\mathcal{G}_{23}$ with $m = 2, 3, 4, 5$.

	$\mathcal{C}_1 = \mathcal{G}_{23}$			
	$m = 6$		$m = 7$	
i	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$
0	110111	0	1111101	0
1	110111	0	1111101	0
2	110111	1	1111101	1
3	111111	0	1111111	0
4	111111	0	1111111	2
5	111111	6	1111110	4
6	111110	12	1111100	8
7	111100	18	1101100	12
8	111000	24	1101000	12
9	110000	24	1001000	12
10	100000	18	1000000	8
11	000000	12	0000000	4
12	000000	6	0000000	2
13	100000	0	0000010	0
14	100000	1	0000010	1
15	000000	0	0000000	0
16	000000	0	0000000	0
17	000000	0		
$\sum$		122		66

Table E.4: An optimal choice for the $\mathbf{a}_i$ in applying the asymmetric construction method to the 3-SyEC code $\mathcal{G}_{23}$ with $m = 6, 7$.

	$\mathcal{C}_1 = \mathcal{L}_{19}$				$\mathcal{C}_1 = \mathcal{T}_{23}$	
	$m = 1$		$m = 2$		$m = 1$	
i	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{T}_i(\mathbf{a}_i)\rvert$
0	0	1	00	1	0	1
1	0	0	01	0	0	0
2	0	0	11	0	0	0
3	0	0	11	0	0	0
4	0	0	11	0	0	0
5	0	0	11	0	0	0
6	0	0	11	0	0	0
7	0	0	11	4	0	0
8	1	9	10	5	1	8
9	0	10	00	5	0	15
10	0	9	00	4	0	17
11	0	0	01	0	0	7
12	0	0	11	0	1	13
13	0	0	11	0	1	9
14	0	0	11	0	0	7
15	0	0	11	0	0	1
16	0	0	11	0	1	3
17	0	0	11	1	0	2
18	1	1			0	0
19					0	0
20					0	0
21					0	0
22					0	0
$\sum$		30		20		83

Table E.5: An optimal choice for the $\mathbf{a}_i$ in applying the asymmetric construction method to the 4-SyEC code $\mathcal{L}_{19}$ with $m = 1, 2$ and the 4-SyEC code $\mathcal{T}_{23}$ with $m = 1$.

	$\mathcal{C}_1 = \mathcal{L}_{11}$		$\mathcal{C}_1 = \mathcal{S}_{19}$					
	$m = 1$		$m = 1$		$m = 2$		$m = 3$	
i	$\mathbf{a}_i$	$\lvert\mathcal{S}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{S}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{S}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{S}_i(\mathbf{a}_i)\rvert$
0	1	5	1	288	01	160	001	80
1	0	6	0	352	00	192	000	112
2	0	5	0	352	00	192	000	112
3	0	0	1	224	01	96	001	48
$\sum$		16		1216		640		352

Table E.6: An optimal choice for the $\mathbf{a}_i$ in applying the unidirectional construction method to the 2-SyEC code $\mathcal{L}_{11}$ with $m = 1$ and to the 2-SyEC code $\mathcal{S}_{19}$ with $m = 1, 2, 3$.

	$\mathcal{C}_1 = \mathcal{W}'_{23}$			
	$m = 1$		$m = 2$	
i	$\mathbf{a}_i$	$\lvert\mathcal{S}_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert\mathcal{S}_i(\mathbf{a}_i)\rvert$
0	1	2176	01	1088
1	0	2048	00	1024
2	0	2176	00	1088
3	0	2048	00	1024
$\sum$		8448		4224

Table E.7: An optimal choice for the $\mathbf{a}_i$ in applying the unidirectional construction method to the 2-SyEC code $\mathcal{W}'_{23}$ with $m = 1, 2$.

	$\mathcal{C}_1 = \mathcal{B}_{15}$				$\mathcal{C}_1 = \mathcal{G}'_{23}$			
	$m = 1$		$m = 2$		$m = 1$		$m = 2$	
i	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$
0	1	7	01	4	0	617	00	281
1	0	8	01	4	0	176	01	120
2	0	7	00	3	0	330	01	120
3	0	0	01	0	0	176	11	281
4	0	0	01	0	1	617	01	336
5	0	0	11	3	0	672	01	336
$\sum$		22		14		2588		1474

Table E.8: An optimal choice for the $\mathbf{a}_i$ in applying the unidirectional construction method to the 3-SyEC code $\mathcal{B}_{15}$ with $m = 1, 2$ and to the 3-SyEC code $\mathcal{G}'_{23}$ with $m = 1, 2$.

	$\mathcal{C}_1 = \mathcal{L}_{19}$						$\mathcal{C}_1 = \mathcal{T}_{23}$	
	$m = 1$		$m = 2$		$m = 3$		$m = 1$	
i	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$	$\mathbf{a}_i$	$\lvert S_i(\mathbf{a}_i)\rvert$
0	1	9	10	5	001	3	1	11
1	0	10	10	5	000	2	0	17
2	0	9	00	4	000	2	0	17
3	0	0	10	0	001	0	0	7
4	0	0	10	0	011	0	1	13
5	0	0	10	0	111	0	1	9
6	0	0	10	0	111	2	0	7
7	0	0	11	4	011	2	0	1
$\sum$		28		18		11		82

Table E.9: An optimal choice for the $\mathbf{a}_i$ in applying the unidirectional construction method to the 4-SyEC code $\mathcal{L}_{19}$ with $m = 1, 2, 3$ and to the 4-SyEC code $\mathcal{T}_{23}$ with $m = 1$.

Appendix F

Enlargement of some AsEC codes

As seen in Chapter 6 it is sometimes possible to add vectors to a t-AsEC code while keeping this code t-AsEC. In this appendix we list in Tables F.1-F.4 such vectors for some 3-AsEC and 4-AsEC codes that were treated in Section 6.2 and Section 7.4. With some effort it can be checked that these vectors indeed do not affect the asymmetric error correcting capability. To this end various properties of the original code can be used, as for example the nonoccurence of certain weights in the code, the code being cyclic, etc.

	$n = 15$
$\mathbf{v}_1$	111100000000000
$\mathbf{v}_2$	000011110000000
$\mathbf{v}_3$	000000001111000
$\mathbf{v}_4$	100001001000110
$\mathbf{v}_5$	000100010010011
$\mathbf{v}_6$	001000100001101

Table F.1: Vectors that can be added to $\mathcal{B}_{15}$ while keeping the code 3-AsEC.

	$n = 16$
$\mathbf{v}_1$	1000010000100001
$\mathbf{v}_2$	0100001000011000
$\mathbf{v}_3$	0010000110000100
$\mathbf{v}_4$	0001100001000010

Table F.2: Vectors that can be added to $\mathcal{R}_4^*$ while keeping the code 3-AsEC.

	$n = 17$
$\mathbf{v}_1$	00011000001011000
$\mathbf{v}_2$	11000000010100001
$\mathbf{v}_3$	00100101100000100
$\mathbf{v}_4$	01101111100111101
$\mathbf{v}_5$	10111011011111010
$\mathbf{v}_6$	11010110111001111
	$n = 18$
$\mathbf{v}_1$	000010000011110000
$\mathbf{v}_2$	000000001000001111
$\mathbf{v}_3$	111000110000000000
	$n = 19$
$\mathbf{v}_1$	1111100000000000000
$\mathbf{v}_2$	0000011111000000000
$\mathbf{v}_3$	0000000000111110000
	$n = 20$
$\mathbf{v}_1$	11111100000000000000
$\mathbf{v}_2$	10000011111000000000
$\mathbf{v}_3$	01000000100111100000
$\mathbf{v}_4$	00010010000100011100
$\mathbf{v}_5$	00100001000010010011
$\mathbf{v}_6$	00001000010001001010
$\mathbf{v}_7$	00000100001000100101
	$n = 21$
$\mathbf{v}_1$	111111000000000000000
$\mathbf{v}_2$	100000111110000000000
$\mathbf{v}_3$	010000100001111000000
$\mathbf{v}_4$	001000010001000111000
$\mathbf{v}_5$	000100001000100100110
$\mathbf{v}_6$	000010000100010001001
$\mathbf{v}_7$	000001000010001010100

Table F.3: Vectors that can be added to $\mathcal{L}_n$ ($17 \leq n \leq 21$) while keeping the code 4-AsEC.

	$n = 22$
$\mathbf{v}_1$	1111111111111111111111
$\mathbf{v}_2$	1111100000000000000000
$\mathbf{v}_3$	0000000000000000011111
$\mathbf{v}_4$	0000011010001010000000
$\mathbf{v}_5$	0000000000110101100000

	$n = 23$
$\mathbf{v}_1$	11111111111111111111111
$\mathbf{v}_2$	00000000000000000011111
$\mathbf{v}_3$	00000000000001111100000
$\mathbf{v}_4$	00000000111110000000000
$\mathbf{v}_5$	00011111000000000000000

Table F.4: Vectors that can be added to $\mathcal{T}_n$ ($22 \leq n \leq 23$) while keeping the code 4-AsEC.

Samenvatting

Dit proefschrift handelt over foutenverbeterende codes voor betrouwbare verzending of opslag van gegevens in een communicatiesysteem dat gebruik maakt van een binair kanaal. De meeste klassen van codes zijn ontworpen met het oog op gebruik voor een symmetrisch kanaal, waarbij $0 \rightarrow 1$ overgangen met dezelfde waarschijnlijkheid optreden als $1 \rightarrow 0$ overgangen (*symmetrische fouten*). In bepaalde toepassingen, zoals optische communicatie, is de kans op een fout waarbij een 1 in een 0 overgaat echter beduidend groter dan de kans op een fout waarbij een 0 in een 1 overgaat. Dergelijke toepassingen kunnen gemodelleerd worden door middel van een asymmetrisch kanaal, waarbij alleen overgangen van het type $1 \rightarrow 0$ optreden (*asymmetrische fouten*). Verder gedragen sommige recent ontwikkelde geheugensystemen zich als een unidirectioneel kanaal, waarbij zowel $1 \rightarrow 0$ als $0 \rightarrow 1$ overgangen op kunnen treden, maar alle fouten van hetzelfde type zijn bij de verzending van een zeker codewoord (*unidirectionele fouten*).

Codes welke symmetrische fouten verbeteren zijn uitgebreid bestudeerd. Natuurlijk kunnen deze codes ook gebruikt worden om asymmetrische of unidirectionele fouten te verbeteren. Er mag echter verwacht worden dat er asymmetrische of unidirectionele foutenverbeterende codes geconstrueerd kunnen worden welke met minder redundantie toekunnen dan vergelijkbare symmetrische foutenverbeterende codes. Het voornaamste doel van dit proefschrift is het verkrijgen van boven- en ondergrenzen voor de maximale cardinaliteit van codes ter lengte n welke t of minder asymmetrische of unidirectionele fouten verbeteren.

In Hoofdstuk 1 schetsen we het communicatiesysteem, geven we een korte inleiding in binaire blokcodes, en behandelen we de symmetrische, unidirectionele, en asymmetrische fouttypes. Verder leiden we algemene voorwaarden af voor de capaciteiten van blokcodes wat betreft de correctie en/of detectie van fouten, waarbij we de nadruk leggen op correctie van een enkel fouttype.

Bovengrenzen voor de maximale cardinaliteit van asymmetrische of unidirectionele foutenverbeterende codes worden behandeld in de Hoofdstukken 2, 3 en 4. Eerst geven we in Hoofdstuk 2 expliciete bovengrenzen welke gebaseerd zijn op het principe van bolstapeling of gebruik maken van bekende bovengrenzen voor de maximale cardinaliteit van symmetrische foutenverbeterende codes. Voor codes met een betrekkelijk kleine lengte wordt aangetoond dat deze grenzen scherp zijn. In deze gevallen is de maximale cardinaliteit dus precies bekend. Vervolgens behandelen we in Hoofdstuk 3 bovengrenzen welke verkregen kunnen worden door het oplossen van een geheeltallig programmeringsprobleem. In deze programmeringsproblemen wordt het totale aantal codewoorden in een code gemaximaliseerd onder bepaalde voorwaarden voor de gewichtsverdeling van de code. Vaak verbeteren deze geheeltallige programmeringsgrenzen de expliciete grenzen, maar ze zijn veel moeilijker te berekenen. Tenslotte verbeteren we de geheeltallige programmeringsgrens in een aantal specifieke gevallen door gebruik te maken van enkele combinatorische argumenten, welke moeilijk te generaliseren lijken.

Ondergrenzen voor de maximale cardinaliteit van asymmetrische of unidirectionele foutenverbeterende codes kunnen verkregen worden door het construeren van codes, hetgeen het onderwerp van de Hoofdstukken 5 en 6 is. In Hoofdstuk 5 presenteren we een methode waarbij codes welke t of minder asymmetrische of unidirectionele fouten verbeteren geconstrueerd worden door het schrappen en afbreken van codewoorden in een code welke t of minder symmetrische fouten verbetert. Algoritmes ter optimalisatie van de cardinaliteit van dergelijke codes en decodeeraspecten komen ook aan de orde. In Hoofdstuk 6 passen we deze methode en tevens enkele zogenaamde 'trial and error' technieken toe om goede codes ter lengte 23 of minder te krijgen welke tot en met 1, 2, 3 of 4 asymmetrische of unidirectionele fouten verbeteren.

Grenzen voor codes welke *lineair* zijn komen aan de orde in Hoofdstuk 7. Er wordt aangetoond dat elke lineaire code welke t of minder unidirectionele fouten verbetert ook t of minder symmetrische fouten verbetert. Daarom beschouwen we alleen lineaire symmetrische foutenverbeterende codes en lineare asymmetrische foutenverbeterende codes. Wederom kan de maximale cardinaliteit exact bepaald worden wanneer de lengte betrekkelijk klein is. Voor correctie van enkele en dubbele fouten laten we zien dat de maximale cardinaliteit van een lineaire asymmetrische foutenverbeterende code de maximale cardinaliteit van een lineaire symmetrische foutenverbeterende code alleen overtreft voor een eindig aantal lengtes.

Wat betreft constructies leiden we voor alle even getallen m groter of gelijk aan 4 een klasse van lineaire codes ter lengte 2^m af welke $2^{m-2}-1$ of minder asymmetrische fouten verbeteren en waarvan de cardinaliteit de grootste cardinaliteit van een vergelijkbare symmetrische foutenverbeterende code overtreft.

In de appendices geven we zowel tabellen met grenzen voor het maximale aantal codewoorden in een code ter lengte n welke t of minder symmetrische, unidirectionele, of asymmetrische fouten verbetert als tabellen met grenzen voor de maximale dimensie van een lineaire code ter lengte n welke t of minder symmetrische of asymmetrische fouten verbetert, in alle gevallen voor $1 \leq t \leq 4$ en $t \leq n \leq 23$.

Tables with bounds on
$A_{SV}(n,t)$, $A_U(n,t)$ and $A_{As}(n,t)$
for $1 \leq t \leq 4$ and $t \leq n \leq 23$
(reprinted from Appendix B)

n	$A_{SV}(n,1)$	$A_{As}(n,1)$
1	1	$^{as}1^{as}$
2	1	$^{as}2^{as}$
3	2	$^{as}2^{as}$
4	2	$^{as}4^{as}$
5	4	$^{Va}6^{ia}$
6	8	$^{KF}12^{ia}$
7	16	$^{DP}18^{ia}$
8	20	$^{DP}36^{ia}$
9	40	$^{DP}62^{ia}$
10	72 – 79	$^{DP}108$ – 117^{ia}
11	144 – 158	$^{DP}174$ – 210^{ia}
12	256	$^{Va}316$ – 410^{ia}
13	512	$^{Va}586$ – 786^{ia}
14	1024	$^{Va}1096$ – 1500^{ia}
15	2048	$^{Va}2048$ – 2828^{ia}
16	2720 – 3276	$^{Va}3856$ – 5430^{ia}
17	5248 – 6552	$^{CR}7296$ – 10374^{ia}
18	10496 – 13104	$^{Va}13798$ – 19898^{ia}
19	20480 – 26208	$^{Va}26216$ – 38008^{ia}
20	36864 – 43690	$^{Va}49940$ – 73174^{ia}
21	73728 – 87380	$^{Va}95326$ – 140798^{ia}
22	147456 – 173784	$^{Va}182362$ – 271953^{ia}
23	294912 – 344636	$^{Va}349536$ – 523586^{ia}

n	$A_{Sy}(n,2)$	$A_U(n,2)$	$A_{As}(n,2)$
2	1	$^{us}1^{us}$	$^{as}1^{as}$
3	1	$^{us}1^{us}$	$^{as}2^{as}$
4	1	$^{us}2^{us}$	$^{as}2^{as}$
5	2	$^{us}2^{us}$	$^{as}2^{as}$
6	2	$^{us}2^{us}$	$^{as}4^{as}$
7	2	$^{us}4^{us}$	$^{as}4^{ia}$
8	4	$^{cu}6^{B4}$	$^{DP}7^{ia}$
9	6	$^{cu}10^{su}$	$^{ca}12^{ia}$
10	12	$^{cu}16 - 18^{gt}$	$^{DP}18^{sa}$
11	24	$^{cu}26 - 32^{gt}$	$^{DP}30 - 32^{ia}$
12	32	$^{cu}52 - 61^{iu}$	$^{ca}54 - 63^{ia}$
13	64	$^{cu}92 - 114^{gt}$	$^{DP}98 - 114^{ia}$
14	128	$^{cu}184 - 218^{gt}$	$^{DP}186 - 218^{ia}$
15	256	$^{gt}256 - 340^{B4}$	$^{ca}266 - 398^{ia}$
16	256 – 340	$^{cu}352 - 680^{B4}$	$^{ca}364 - 739^{ia}$
17	512 – 680	$^{cu}640 - 1277^{iu}$	$^{ca}647 - 1279^{ia}$
18	1024 – 1288	$^{cu}1216 - 2372^{B4}$	$^{ca}1218 - 2380^{ia}$
19	2048 – 2372	$^{gt}2048 - 4096^{B4}$	$^{ca}2050 - 4242^{ia}$
20	2560 – 4096	$^{gt}2560 - 6942^{B4}$	$^{ca}2564 - 8069^{ia}$
21	4096 – 6942	$^{cu}4224 - 13774^{B4}$	$^{ca}4251 - 14374^{ia}$
22	8192 – 13774	$^{cu}8448 - 24106^{B4}$	$^{ca}8450 - 26679^{ia}$
23	16384 – 24106	$^{gt}16384 - 48212^{B3}$	$^{ca}16388 - 50200^{ia}$

n	$A_{Sy}(n,3)$	$A_{U}(n,3)$	$A_{As}(n,3)$
3	1	${}^{us}1^{us}$	${}^{as}1^{as}$
4	1	${}^{us}1^{us}$	${}^{as}2^{as}$
5	1	${}^{us}2^{us}$	${}^{as}2^{as}$
6	1	${}^{us}2^{us}$	${}^{as}2^{as}$
7	2	${}^{us}2^{us}$	${}^{as}2^{as}$
8	2	${}^{us}2^{us}$	${}^{as}4^{as}$
9	2	${}^{us}4^{us}$	${}^{as}4^{ai}$
10	2	${}^{us}4^{B4}$	${}^{rV}6^{ai}$
11	4	${}^{cu}7^{su}$	${}^{ca}8^{ai}$
12	4	${}^{cu}10^{su}$	${}^{rK}12^{ai}$
13	8	${}^{cu}14 - 18^{gt}$	${}^{ca}18^{ai}$
14	16	${}^{cu}22 - 34^{gt}$	${}^{ca}30 - 34^{sa}$
15	32	${}^{gt}32 - 50^{gt}$	${}^{ca}44 - 50^{ia}$
16	36 - 37	${}^{cu}53 - 90^{gt}$	${}^{ca}66 - 90^{sa}$
17	64 - 72	${}^{cu}97 - 168^{gt}$	${}^{ca}122 - 168^{ia}$
18	128 - 144	${}^{cu}188 - 320^{gt}$	${}^{ca}234 - 320^{ia}$
19	256 - 279	${}^{cu}376 - 616^{gt}$	${}^{ca}450 - 616^{ia}$
20	512	${}^{cu}737 - 1142^{iu}$	${}^{ca}860 - 1144^{ia}$
21	1024	${}^{cu}1474 - 2134^{gt}$	${}^{ca}1628 - 2134^{ia}$
22	2048	${}^{cu}2588 - 4114^{iu}$	${}^{Sh}3072 - 4116^{ia}$
23	4096	${}^{gt}4096 - 7346^{gt}$	${}^{gt}4096 - 7346^{ia}$

n	$A_{Sy}(n,4)$	$A_U(n,4)$	$A_{As}(n,4)$
4	1	${}^{us}1^{us}$	${}^{as}1^{as}$
5	1	${}^{us}1^{us}$	${}^{as}2^{as}$
6	1	${}^{us}2^{us}$	${}^{as}2^{as}$
7	1	${}^{us}2^{us}$	${}^{as}2^{as}$
8	1	${}^{us}2^{us}$	${}^{as}2^{as}$
9	2	${}^{us}2^{us}$	${}^{as}2^{as}$
10	2	${}^{us}2^{us}$	${}^{as}4^{as}$
11	2	${}^{us}4^{us}$	${}^{as}4^{ia}$
12	2	${}^{us}4^{gt}$	${}^{as}4^{ia}$
13	2	${}^{cu}6^{gt}$	${}^{ca}6^{ia}$
14	4	${}^{cu}8^{gt}$	${}^{ca}8^{sa}$
15	4	${}^{cu}10 - 12^{gt}$	${}^{ca}12^{ia}$
16	6	${}^{cu}11 - 16^{gt}$	${}^{ca}16^{sa}$
17	10	${}^{cu}18 - 26^{gt}$	${}^{ca}26^{sa}$
18	20	${}^{cu}28 - 44^{gt}$	${}^{ca}36 - 44^{ia}$
19	40	${}^{gt}40 - 74^{gt}$	${}^{ca}46 - 74^{sa}$
20	40 – 48	${}^{gt}40 - 133^{gt}$	${}^{ca}54 - 133^{sa}$
21	48 – 88	${}^{gt}48 - 229^{gt}$	${}^{ca}62 - 229^{ia}$
22	64 – 150	${}^{cu}82 - 423^{gt}$	${}^{ca}88 - 423^{ia}$
23	128 – 280	${}^{gt}128 - 745^{gt}$	${}^{ca}133 - 745^{ia}$

Acknowledgements

I would like to thank all the people that have contributed to this thesis in one way or another.

First of all, I am indebted to my colleagues of the Information Theory Group of the Department of Electrical Engineering of the Delft University of Technology for their support and for providing a stimulating environment. Especially I thank my successive room-mates, Inald Lagendijk, Jan van der Lubbe, and Hendro Handojo, for their pleasant company.

I also express my gratitude towards the students of this group. Particularly I thank Bert Visser for his contributions to the optimization and decoding aspects of the construction method as presented in Chapter 5.

The interest shown by the regular participants in the weekly colloquia on 'coding theory' and 'combinatorial structures' at the Department of Mathematics and Informatics of the Delft University of Technology is highly appreciated.

Thanks are also due to Frank Böinck, Hideki Imai, Yuichi Saitoh, and Tomohiko Uyematsu for valuable correspondence.

I am grateful to Mrs. J.B. Zaat-Jones for her conscientious correction of the English text.

Last but not least I wish to thank all of my family, friends, and fellow soccer-players of Excelsior '20 for their support and for providing the indispensable relaxation.

Curriculum Vitae

Jacobus Hendricus (Jos) Weber was born in Schiedam, The Netherlands, on June 19th, 1961. In May 1979 he obtained the Gymnasium β diploma from the Spieringshoek Lyceum in Schiedam. Subsequently he enrolled in the Department of Mathematics and Informatics of the Delft University of Technology, from which he received the degree of Ingenieur (cum laude) in November 1985.

From January 1984 to July 1984 he was a teacher in informatics at the Protestant School for Secondary Economic and Administrative Education in Rotterdam. In December 1985 he joined the Information Theory Group of the Department of Electrical Engineering of the Delft University of Technology, where he was employed as a research assistant till May 1989, and as an assistent professor afterwards.

His main research interest is coding theory.