Weak Flip Codes and their Optimality on the Binary Erasure Channel

Hsuan-Yin Lin, Stefan M. Moser and Po-Ning Chen

17 June 2016

Abstract

This paper investigates fundamental properties of nonlinear binary codes by looking at the codebook matrix not row-wise (codewords), but column-wise. The family of weak flip codes is presented and shown to contain many beautiful properties. In particular the subfamily fair weak flip codes, which goes back to Berlekamp, Gallager, and Shannon and which was shown to achieve the error exponent with a fixed number of codewords $M$, can be seen as a generalization of linear codes to an arbitrary number of codewords. The fair weak flip codes are related to binary nonlinear Hadamard codes.

Based on the column-wise approach to the codebook matrix, the $r$-wise Hamming distance is introduced as a generalization to the well-known and widely used (pairwise) Hamming distance. It is shown that the minimum $r$-wise Hamming distance satisfies a generalized $r$-wise Plotkin bound. The $r$-wise Hamming distance structure of the nonlinear fair weak flip codes is analyzed and shown to be superior to many codes. In particular, it is proven that the fair weak flip codes achieve the $r$-wise Plotkin bound with equality for all $r$.

In the second part of the paper, these insights are applied to a binary erasure channel (BEC) with an arbitrary erasure probability $0 < \delta < 1$. An exact formula for the average error probability of an arbitrary (linear or nonlinear) code using maximum likelihood decoding is derived and shown to be expressible using only the $r$-wise Hamming distance structure of the code. For a number of codewords $M$ satisfying $M \leq 4$ and an arbitrary finite blocklength $n$, the globally optimal codes (in the sense of minimizing the average error probability) are found. For $M = 5$ or $M = 6$ and an arbitrary finite blocklength $n$, the optimal codes are conjectured. For larger $M$, observations regarding the optimal design are presented, e.g., that good codes have a large $r$-wise Hamming distance structure for all $r$.

Index Terms — Binary erasure channel (BEC), finite blocklength, generalized Plotkin bound, maximum likelihood (ML) decoder, minimum average error probability, optimal nonlinear code design, $r$-wise Hamming distance, weak flip codes.

The authors are with the Department of Electrical and Computer Engineering at National Chiao Tung University (NCTU), Hsinchu, Taiwan. Stefan M. Moser is also with ETH Zurich, Switzerland. This work has been supported by National Science Council under NSC 97–2221–E–009–003–MY3 and NSC 100–2221–E–009–006–MY3. This work was presented in parts at the 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton House, Monticello, IL, USA, Oct. 1–5, 2012 and the 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, Jun. 14–19, 2015.
1 Introduction

A goal in traditional coding theory is to find good codes that operate close to the ultimate limit of the channel capacity as introduced by Shannon [1]. Implicitly, by the definition of capacity, such codes are expected to have a large blocklength. Moreover, due to the potential simplifications and because such codes behave well for large blocklength, conventional coding theory often restricts itself to linear codes. It is also quite common to use the minimum Hamming distance and the weight enumerating function (WEF) as a design and quality criterion [2]. This is motivated by the equivalence of Hamming weight and Hamming distance for linear codes, and by the union bound that converts the global error probability into pairwise error probabilities.

In this work we would like to break away from these traditional simplifications and instead focus on an optimal design of codes for finite blocklength. Since for very short blocklength it is not realistic to transmit large quantities of information, we start by looking at codes with only a few codewords, so called ultrasmall block codes. Such codes have many practical applications. For example, in the situation of establishing an initial connection in a wireless link, the amount of information that needs to be transmitted during the setup of the link is limited to usually only a couple of bits. However, these bits need to be transmitted in very short time (e.g., blocklength in the range of $n = 20$ to $n = 30$) with the highest possible reliability [3].

Also in view of 5G wireless communication systems, very reliable codes with very low latency are asked for, which can only be found by restricting oneself to short packets [4].

Another important application of short codes appears in the context of “biological coding”. For example, there are attempts for future digital information storage systems that try to use DNA or DNA-based approaches to store data. There, very short and simple codes are needed to provide local integrity. While first architectures relied on a single-parity check code, more advanced systems try more elaborate schemes like simple Reed-Solomon codes [5]–[8].

As a last example we would like to mention molecular communication, where short messages are transmitted with the help of molecules that are transported by diffusion. Inherently, in such systems neither the blocklength and nor the number of codewords can be large [9].

While conventional coding theory in the sense of Shannon theory often focuses on stating important fundamental insights and properties like, e.g., at what rates it is possible to transmit information with an error probability that vanishes as the blocklength tends to infinity, we specifically turn our attention to the concrete code design, i.e., we are interested in actually finding a globally optimum code for a certain given channel and a given fixed blocklength.

In this paper, we reintroduce a class of codes, called fair weak flip codes, that have many beautiful properties similar to those of binary linear codes. However, while binary linear codes are very much limited since they can only exist if the number of codewords $M$ happens to be an integer-power of 2, our class of codes exists for arbitrary $M$. We will investigate these “quasi-linear” codes and show that they satisfy the Plotkin bound.

Fair weak flip codes are related to a class of binary nonlinear codes that are con-
structed with the help of Hadamard matrices and Levenshtein’s theorem \cite{10, Ch. 2}. These binary nonlinear Hadamard codes also meet the Plotkin bound. As a matter of fact, if for the parameters \( (M, n) \) of a given fair weak flip code there exists a Hadamard code, then these two codes are equivalent.\(^3\) In this sense we can consider the fair weak flip codes to be a subclass of Hadamard codes. Note, however, that there is no guarantee that for every choice of parameters \( (M, n) \) for which fair weak flip codes exist, there also exists a corresponding Hadamard code.

Moreover, also note that while Levenshtein’s method is only concerned with an optimal pairwise Hamming distance structure, we will show that fair weak flip codes are globally optimal (i.e., they are the best with respect to error probability and not only to pairwise Hamming distance, and they are best among all codes, linear or nonlinear). We prove this global optimality in the case of the number of codewords \( M \leq 4 \), and conjecture it for \( M \geq 5 \).

We also introduce a generalization to the Hamming distance, the \( r \)-wise Hamming distance, and we prove that the exact average error probability of an arbitrary binary code on the binary erasure channel (BEC) can be fully characterized using the \( r \)-wise Hamming distances only. Furthermore, we propose a Plotkin-type bound on the \( r \)-wise Hamming distances for binary codes.

Our definition of the \( r \)-wise Hamming distance is related to the \( r \)th generalized Hamming weight introduced in \cite{11} and used to investigate a code’s security performance on the wire-tap channel of Type II. Note, however, that \cite{11} restricts itself to linear codes only. Indeed, the \( r \)th generalized Hamming weight is defined by the minimum support of any \( r \)-dimensional subcode of a given linear code of dimension \( k \) (where a support of a linear code is defined as the number of positions where not all codewords are zero), and thus only describes subsets of codewords that form a linear subcode.

On the other hand, our \( r \)-wise Hamming distance is defined for linear and nonlinear codes and characterizes the relation of any subset of \( r \) codewords. Since an arbitrary subset of codewords from a linear code can be either linear or nonlinear, this leads to an essential distinction of our work from previous works \cite{12}–\cite{14}.

We further define a class of codes called weak flip codes that contains the fair weak flip codes as a special case. We prove that some particular weak flip codes are optimal for the BEC for \( M \leq 4 \) and for any finite blocklength \( n \). For \( M \geq 5 \), we believe that for certain blocklengths the codes which maximize all the minimum \( r \)-wise Hamming distances (including the pairwise Hamming distance) are best among all possible codes. Evidence for this claim will be presented for the case of \( M = 8 \).

This work is an extension of our previous work \cite{15} and of \cite{16}, \cite{17}, where we study ultrasmall block codes for the situation of general binary-input binary-output channels and where we derive the optimal code design for the two special cases of the Z-channel (ZC) and the binary symmetric channel (BSC). We will also briefly compare our findings here with these channels, especially with the symmetric BSC.

The foundations of our insights lie in a powerful way of creating and analyzing both linear and nonlinear block codes. As is customary, we use the codebook matrix containing the codewords in its rows to describe our codes.\(^4\) However, for our code construction and performance analysis, we are looking at this codebook matrix not row-wise, but column-wise. All our proofs and also our definitions of the new \( r \)-wise

\(^3\)For a precise definition of equivalence see Remark 8 below.

\(^4\)The codebook matrix is not to be confused with a generator matrix that can be used to describe linear codes.
Hamming distance and the “quasi-linear” codes are fully based on this new approach. (This is another fundamental difference between our results and the binary nonlinear Hadamard codes that are constructed based on Hadamard matrices and Levenshtein’s theorem [10].)

The remainder of this paper is structured as follows. After some comments about our notation, we will introduce the channel model and review some common definitions in Section 2. In Section 3 we introduce the family of weak flip codes including its subfamily of fair weak flip codes and compare it to the well-known binary nonlinear Hadamard codes. Section 4 reviews some previous results. The main results are then summarized and discussed in Sections 5 and 6: Section 5 provides the definition of the r-wise Hamming distance and discusses the quasi-linear properties of weak flip codes, and in Section 6 the BEC and its optimal codes are presented. We conclude in Section 7. Some of the lengthy proofs from Section 6 are postponed to the appendix.

As a convention in coding theory, vectors (denoted by boldface Roman letters, e.g., \( \mathbf{x} \)) are row-vectors. However, for simplicity of notation and to avoid a large number of transpose-signs, we slightly misuse this notational convention for one special case: any vector \( c \) is a column-vector. It should be always clear from the context because these vectors are used to build codebook matrices and are therefore also conceptually quite different from the transmitted codeword \( x \) or the received sequence \( y \).

Moreover, we use a bar \( \bar{x} \) to denote the flipped version of \( x \), i.e., \( \bar{x} = x \oplus 1 \) (where \( \oplus \) denotes the componentwise XOR operation and where \( 1 \) is the all-one vector). We use capital letters for random quantities, e.g., \( X \), and small letters for their deterministic counterparts, e.g., \( x \); constants are depicted by Greek letters, small Romans, or a special font, e.g., \( M \); sets are denoted by calligraphic letters, e.g., \( \mathcal{M} \); and \( |\mathcal{M}| \) denotes the cardinality of the set \( \mathcal{M} \).

2 Channel Model and Coding Schemes

In this work, we consider the well-known binary erasure channel (BEC) given in Figure 1. The BEC is a discrete memoryless channel (DMC) with a binary input alphabet \( \mathcal{X} = \{0, 1\} \) and a ternary output alphabet \( \mathcal{Y} = \{0, 1, 2\} \), and with a conditional channel...
law
\[
P_{Y|X}(y|x) = \begin{cases} 
1 - \delta & \text{if } y = x, \ x \in \{0, 1\}, \\
\delta & \text{if } y = 2, \ x \in \{0, 1\}.
\end{cases} \tag{1}
\]

Here \(0 \leq \delta < 1\) is called the erasure probability.

While we focus on the BEC, we will sometimes briefly compare our results with the situation of the binary symmetric channel (BSC), particularly in view of [17]. The BSC is a binary-input, binary-output DMC with conditional channel law
\[
P_{Y|X}(y|x) = \begin{cases} 
1 - \epsilon & \text{if } y = x, \ x \in \{0, 1\}, \\
\epsilon & \text{if } y = 1 - x, \ x \in \{0, 1\}.
\end{cases} \tag{2}
\]

where \(0 \leq \epsilon < \frac{1}{2}\) is called crossover probability.

We next review some common definitions.

**Definition 1.** An \((M, n)\) coding scheme for a DMC \((X, Y, P_{Y|X})\) consists of the message set \(M = \{1, 2, \ldots, M\}\), a codebook \(C(M, n)\) with \(M\) length-\(n\) codewords \(x_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,n}) \in X^n, \ m \in M\), an encoder that maps every message \(m\) into its corresponding codeword \(x_m\), and a decoder that makes a decoding decision \(g(y) \in M\) for every received \(n\)-vector \(y \in Y^n\).

The set of codewords \(C(M, n)\) is called \((M, n)\) codebook or simply \((M, n)\) code. Sometimes we follow the custom of traditional coding theory and use three parameters:\footnote{Actually, it is usual to have them ordered as \((n, M, d)\), but for consistency, we will stick to \((M, n)\) or \((M, n, d)\).} \((M, n, d)\) code, where the third parameter \(d\) denotes the minimum Hamming distance \(d_{\text{min}}(C(M, n))\), i.e., the minimum number of components in which any two codewords differ.

We assume that the \(M\) possible messages are equally likely and \(g\) is the maximum likelihood (ML) decoder\footnote{Under the assumption of equally likely messages, the ML decoding rule is equivalent to the maximum a posteriori (MAP) decoding rule, i.e., for a given code and DMC, it minimizes the average error probability as defined in (11) among all possible decoders.}
\[
g(y) \triangleq \arg\max_{1 \leq m \leq M} P_{Y|X}(y|x_m), \tag{3}
\]
where in case that there are several \(m\) achieving the maximum, an arbitrary one of them is chosen.

**Definition 2.** For a given code \(C(M, n)\) we define the decoding region \(D_{m}(M, n)\) corresponding to the \(m\)th codeword \(x_m\) as
\[
D_{m}(M, n) \triangleq \{y : g(y) = m\}. \tag{4}
\]

Note that in Definition 2, all decoding regions must be disjoint, and their union must be equal to \(Y^n\)
\[
D_{m}(M, n) \cap D_{m'}(M, n) = \emptyset, \quad 1 \leq m < m' \leq M, \tag{5}
\]
\[
\bigcup_{m \in M} D_{m}(M, n) = Y^n. \tag{6}
\]
As mentioned above, there does not necessarily exist a unique \( m \) such that for a given \( y \),

\[
P_Y|X(y|x_m) = \max_{1 \leq m' \leq M} P_Y|X(y|x_{m'}),
\]

i.e., certain received vectors \( y \) could be assigned to different decoding regions without changing the performance of the coding scheme. In the following we define closed decoding regions that break the condition (5).

**Definition 3.** The closed decoding region \( D^{(M,n)}_m \) corresponding to the \( m \)th codeword \( x_m \) is defined as

\[
D^{(M,n)}_m \triangleq \{ y : P_Y|X(y|x_m) = \max_{1 \leq m' \leq M} P_Y|X(y|x_{m'}) \}, \quad m \in M.
\]

Note that \( D^{(M,n)}_m \subseteq D^{(M,n)}_m \).

**Definition 4.** For an \((M,n)\) code, given that message \( m \) (and hence the \( m \)th codeword \( x_m \)) has been sent, we define \( \lambda_m \) to be the corresponding probability of a decoding error under the ML decoder \( g \):

\[
\lambda_m(C^{(M,n)}) \triangleq \Pr[g(Y) \neq m | X = x_m] = \sum_{y \in D^{(M,n)}_m} P_Y|X(y|x_m).
\]

The average error probability \( P_e \) of an \((M,n)\) code is defined as

\[
P_e(C^{(M,n)}) \triangleq \frac{1}{M} \sum_{m=1}^{M} \lambda_m(C^{(M,n)}).
\]

Sometimes it will be more convenient to focus on the probability of not making any error, denoted success probability \( \psi_m \):

\[
\psi_m(C^{(M,n)}) \triangleq \Pr[g(Y) = m | X = x_m] = \sum_{y \in D^{(M,n)}_m} P_Y|X(y|x_m) = \Pr[Y \in D^{(M,n)}_m | X = x_m].
\]

The definition of the average success probability \( P_c \) follows accordingly.

Our ultimate goal is to find the structure of a code that minimizes the average error probability among all codes based on the ML decoding rule.

**Definition 5.** A code \( C^{(M,n)} \) is called optimal and denoted by \( C^{(M,n)*} \) if

\[
P_e(C^{(M,n)*}) \leq P_e(C^{(M,n)})
\]

for any (linear or nonlinear) code \( C^{(M,n)} \).

\(^7\)The subscript “c” stands for “correct.”
3 Preliminaries

3.1 Average Error (and Success) Probability over the BEC

We start with the following definitions.

Definition 6. The Hamming distance $d_H(x_m, x_m')$ between two binary length-$n$ vectors $x_m$ and $x_m'$ is defined as the number of positions $j$ where $x_{m,j} \neq x_{m',j}$. The Hamming weight of a binary length-$n$ vector $x$ is defined as $w_H(x) \triangleq d_H(x, \mathbf{0})$.

Definition 7. By $N(\alpha|y)$ we denote the number of occurrences of a symbol $\alpha \in \mathcal{Y}$ in a received vector $y$, and $\mathcal{I}(\alpha|y)$ is defined as the set of indices $j$ such that $y_j = \alpha$. Thus, $N(\alpha|y) = |\mathcal{I}(\alpha|y)|$. Moreover, we use $x_{m,\mathcal{I}(\alpha|y)}$ (respectively, $y_{\mathcal{I}(\alpha|y)}$) to describe a vector of length $N(\alpha|y)$ containing the components $x_{m,j}$ (respectively, $y_j$) where $j \in \mathcal{I}(\alpha|y)$. We also write $x_{m,\mathcal{I}(\alpha|y)} \cup x_{m,\mathcal{I}(\mathcal{Y}\setminus\{\alpha\}|y)}$ for the complete vector $x_m$, where the “union”-operation implicitly reorders the indices in the usual ascending order.

The error probability when transmitting uniformly picked codewords from code $\mathcal{C}^{(M,n)}$ over the BEC can be written as follows:

$$
P_e(\mathcal{C}^{(M,n)}) = \frac{1}{M} \sum_{m=1}^{M} \sum_{\substack{y \in \mathcal{Y}^n \\|g(y)\neq m}} (1 - \delta)^{N(0|y)} \delta^{N(2|y)} \mathcal{I}\left\{d_H(x_{m,\mathcal{I}(0|y)}), y_{\mathcal{I}(0|y)} = 0\right\}.
$$

(16)

where $\mathcal{I}\{$STATEMENT$\}$ denotes the indicator function whose value is 1 if the STATEMENT is correct and 0 otherwise.

The success probability accordingly is given by:

$$
P_s(\mathcal{C}^{(M,n)}) = \frac{1}{M} \sum_{m=1}^{M} \sum_{\substack{y \in \mathcal{Y}^n \\|g(y)\neq m}} (1 - \delta)^{N(0|y)} \delta^{N(2|y)} \mathcal{I}\left\{d_H(x_{m,\mathcal{I}(0|y)}), y_{\mathcal{I}(0|y)} = 0\right\}.
$$

(17)

3.2 Column-Wise Description of General Binary Codes

Usually, a general codebook $\mathcal{C}^{(M,n)}$ with $M$ codewords and with blocklength $n$ is written as an $M \times n$ codebook matrix where the $M$ rows correspond to the $M$ codewords:

$$
\mathcal{C}^{(M,n)} = \begin{pmatrix}
\vdots \\
\mathbf{x}_1 \\
\vdots \\
\mathbf{x}_M
\end{pmatrix} = \begin{pmatrix}
\mathbf{c}_1 \\
\mathbf{c}_2 \\
\vdots \\
\mathbf{c}_n
\end{pmatrix}.
$$

(18)

In our approach, we prefer to consider the codebook matrix column-wise rather than row-wise [17]. We denote the length-$M$ column-vectors of the codebook by $\mathbf{c}_j$, $j \in \{1, \ldots, n\}$.

Remark 8. Since we assume equally likely messages, any permutation of rows only changes the assignment of codewords to messages and has therefore no impact on the performance. We thus consider two codes with permuted rows as being equal (this agrees with the concept of a code being a set of codewords, where the ordering of the...
codewords is irrelevant). Furthermore, since we only consider memoryless channels, any permutation of the columns of \( \mathcal{C}^{(M,n)} \) will lead to another code with identical error probability. We say that such two codes are equivalent. We would like to emphasize that two codes being equivalent is not the same as two codes being equal. However, as we are mainly interested in the performance of a code, we usually treat two equivalent codes as being the same.

Due to the symmetry of the BEC\(^8\) we have an additional equivalence in the codebook design (compare also with the BSC [17]).

**Lemma 9.** Consider an arbitrary code \( \mathcal{C}^{(M,n)} \) to be used on the BEC and consider an arbitrary \( M \)-vector \( \mathbf{c} \). Construct a new length-\((n + 1)\) code \( \overline{\mathcal{C}}^{(M,n+1)} \) by appending \( c \) to the codebook matrix of \( \mathcal{C}^{(M,n)} \) and another new length-\((n + 1)\) code \( \overline{\mathcal{C}}^{(M,n+1)} \) by appending the flipped vector \( \overline{\mathbf{c}} = \mathbf{c} \oplus \mathbf{1} \) to the codebook matrix of \( \mathcal{C}^{(M,n)} \). Then the performance of these two new codes are identical:

\[
P_e(\mathcal{C}^{(M,n+1)}) = P_e(\overline{\mathcal{C}}^{(M,n+1)}).
\]  

(19)

Note that Lemma 9 cannot be generalized further, i.e., for some \( \mathcal{C}^{(M,n)} \), appending a vector \( \tilde{\mathbf{c}} \) other than \( \overline{\mathbf{c}} \) may result in a length-\((n + 1)\) code \( \tilde{\mathcal{C}}^{(M,n+1)} \) that is not equivalent to \( \mathcal{C}^{(M,n+1)} \).

Next we define a convenient numbering system for the possible columns of the codebook matrix of binary codes.

**Definition 10.** For fixed \( M \) and \( b_m \in \{0,1\} \), \( m \in \mathcal{M} \), we describe the column vector \((b_1 b_2 \cdots b_M)^T\) by its reverse binary representation of nonnegative integers

\[
j = \sum_{m=1}^{M} b_m 2^{M-m},
\]

(20)

and write \( \mathbf{c}^{(M)}_j \triangleq (b_1 b_2 \cdots b_M)^T \). For example, \( \mathbf{c}^{(5)}_{12} = (0 1 1 0 0)^T \) and \( \mathbf{c}^{(5)}_3 = (0 0 0 1 1)^T \).

Due to Lemma 9, we discard any column starting with a one, i.e., we require \( b_1 = 0 \). Moreover, as it will never help to improve the performance, we exclude the all-zero column. Hence, the set of all possible candidate columns of general binary codes can be restricted to

\[
\mathcal{C}^{(M)} \triangleq \left\{ \mathbf{c}^{(M)}_1, \mathbf{c}^{(M)}_2, \ldots, \mathbf{c}^{(M)}_{2^{M-1}-1} \right\},
\]

(21)

For a given codebook and for any

\[
j \in \mathcal{J} \triangleq \{1, \ldots, 2^{M-1} - 1\},
\]

(22)

let \( t_j \) denote the number of the corresponding candidate columns \( \mathbf{c}^{(M)}_j \) appearing in the codebook matrix of \( \mathcal{C}^{(M,n)} \). Because of Remark 8, the ordering of the candidate columns is irrelevant, and any binary code with blocklength

\[
n = \sum_{j=1}^{2^{M-1}-1} t_j
\]

(23)

---

\(^8\)The symmetry property here is identical to the symmetry definitions in [18, p. 94]. Hence, it is not surprising that Lemma 9 also holds for general binary-input symmetric channels.
can therefore be fully described by the parameter vector
\[ t \triangleq [t_1, t_2, \ldots, t_{2^M-1-1}] . \] (24)

We say that such a code has a type vector (or simply type) \( t \), and write \( C_{t_1, \ldots, t_{2^M-1-1}}^{(M,n)} \) or \( C_t^{(M,n)} \).

**Example 11.** For \( M = 3, 4 \), the candidate columns sets are
\[
\begin{align*}
C^{(3)} &= \left\{ c_1^{(3)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},
                 c_2^{(3)} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},
                 c_3^{(3)} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}; \\
C^{(4)} &= \left\{ c_1^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},
                 c_2^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix},
                 c_3^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix},
                 c_4^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix},
                 c_5^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix},
                 c_6^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix},
                 c_7^{(4)} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right\} .
\end{align*}
\] (25)

A codebook \( C_t^{(4,7)} \) of type \( t = [2, 0, 2, 0, 2, 1, 0] \) is equivalent to all columns permutations of the following codebook:
\[
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & \end{pmatrix} .
\] (27)

### 3.3 Weak Flip Codes

We next introduce some special families of binary codes.

**Definition 12.** Given an integer \( M \geq 2 \), a length-\( M \) candidate column is called a weak flip column and denoted \( c^{(M)}_{\text{weak}} \) if its first component is 0 and its Hamming weight equals to \( \lfloor \frac{M}{2} \rfloor \) or \( \lceil \frac{M}{2} \rceil \). The collection of all possible weak flip columns is called weak flip candidate columns set and is denoted by \( C_{\text{weak}}^{(M)} \). The remaining, nonweak flip candidate columns are collected in \( C_{\text{nonweak}}^{(M)} \), i.e., \( C^{(M)} = C_{\text{weak}}^{(M)} \cup C_{\text{nonweak}}^{(M)} \).

We see that a weak flip column contains an almost equal number of zeros and ones. For the remainder of this paper, we introduce the following shorthands:
\[
J \triangleq 2^{M-1} - 1, \quad \bar{\ell} \triangleq \left\lceil \frac{M}{2} \right\rceil, \quad \ell \triangleq \left\lfloor \frac{M}{2} \right\rfloor, \quad L \triangleq \left( 2\ell - 1 \right) .
\] (28)

Recall the corresponding sets \( M \) given in Definition 1 and \( J \) given in (22).

---

\(^9\)Note that sometimes, for the sake of convenience, we will omit the superscripts \((M,n)\) or \((M)\).
Lemma 13. The cardinality of the weak flip candidate columns set is

\[ |C^{(M)}_{\text{weak}}| = L, \]  

(29)

and the cardinality of the nonweak flip candidate columns set is

\[ |C^{(M)}_{\text{nonweak}}| = J - L. \]  

(30)

Proof: If \( M = 2\bar{\ell} \), then we have \( (2\bar{\ell}-1) \) possible choices of weak flip columns, while if \( M = 2\bar{\ell} - 1 \), we have \( (2\bar{\ell}-2) + (2\bar{\ell}-1) = 2\bar{\ell}-1 \) choices. This proves (29). Since in total we have \( J \) candidate columns, (30) follows directly from (29). It can also be computed as

\[ |C^{(M)}_{\text{nonweak}}| = \sum_{h=1}^{\bar{\ell}-1} \binom{M-1}{h} + \sum_{h=\bar{\ell}+1}^{L} \binom{M-1}{h} = J - L. \]  

(31)

Remark 14. The above lemma assures that the cardinalities of the weak flip candidate columns set for \( M = 2\bar{\ell} - 1 \) and of the weak flip candidate columns set for \( M = 2\bar{\ell} \) are both the same for any positive integer \( \bar{\ell} \) and are both given by \( (2\bar{\ell}-1) \). Actually, if we take \( C^{(2\bar{\ell}-1)}_{\text{weak}} \) and we append as the last bit a one to all its weak flip columns of weight \( \ell = \bar{\ell} - 1 \) and a zero to the other weak flip columns of weight \( \bar{\ell} \), we obtain \( C^{(2\bar{\ell})}_{\text{weak}} \). Hence, \( C^{(2\bar{\ell}-1)}_{\text{weak}} \) can be obtained from \( C^{(2\bar{\ell})}_{\text{weak}} \) by removing the last bit from all column vectors. See for example (34) and (35) below.

Definition 15. A weak flip code \( C^{(M,n)}_{\text{weak}} \) is constructed only by weak flip columns. Since in its type (24) all positions corresponding to nonweak flip columns are zero, we use a reduced type vector:

\[ t_{\text{weak}} \triangleq [t_{j_1}, t_{j_2}, \ldots, t_{j_L}], \]  

(32)

where

\[ \sum_{w=1}^{L} t_{j_w} = n \]  

(33)

with \( j_w, w = 1, \ldots, L \), representing the numbers of the candidate columns that are weak flip columns.

For \( M = 2 \) or \( M = 3 \), all candidate columns are also weak flip columns (note that \( 2^{M-1} - 1 = \binom{M-1}{\bar{\ell}} \)). For \( M = 4 \), \( t_{\text{weak}} = [t_3, t_5, t_6] \). A similar definition can be given also for larger \( M \); however, one needs to be aware that the number of weak flip candidate columns is increasing fast. For \( M = 5 \) or \( M = 6 \), we have ten weak flip candidate
columns:

\[
C_{\text{weak}}^{(5)} = \left\{ c_3^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, c_5^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, c_6^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, c_7^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, c_9^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, c_10^{(5)} \triangleq \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, c_11^{(5)} \triangleq \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, c_12^{(5)} \triangleq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, c_13^{(5)} \triangleq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, c_14^{(5)} \triangleq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right\},
\]

and

\[
C_{\text{weak}}^{(6)} = \left\{ c_7^{(6)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, c_{11}^{(6)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, c_{13}^{(6)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, c_{14}^{(6)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, c_{19}^{(6)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, c_{21}^{(6)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, c_{22}^{(6)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, c_{25}^{(6)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, c_{26}^{(6)} \triangleq \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, c_{28}^{(6)} \triangleq \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\},
\]

respectively.

We will next introduce a special subclass of weak flip codes that, as we will see in Section 5.2, possesses particularly beautiful properties.

**Definition 16.** A weak flip code is called *fair* if it is constructed by an equal number of all possible weak flip candidate columns in $C_{\text{weak}}^{(M)}$. Note that by definition the blocklength of a fair weak flip code is always an integer-multiple of $L$.

Fair weak flip codes have been used by Shannon *et al.* [19] for the derivation of error exponents, although the codes were not named at that time. Note that in [19] the error exponents are defined when blocklength $n$ goes to infinity, but in this work we consider finite $n$.

### 3.4 Hadamard Codes

In this section, we review the family of *Hadamard codes* and investigate its relation to weak flip codes and fair weak flip codes. We follow the definition of [10, Ch. 2].

**Definition 17.** For an even integer $m$, a *(normalized)* Hadamard matrix $H_m$ of order $m$ is an $m \times m$ matrix with entries $+1$ and $-1$ and with the first row and column being all $+1$, such that

\[
H_m H_m^\top = m I_m,
\]
if such a matrix exists. Here $I_m$ is the identity matrix of size $m$. If the entries $+1$ are replaced by 0 and the entries $-1$ by 1, $H_m$ is changed into the binary Hadamard matrix $A_m$.

Note that a necessary condition for the existence of $H_m$ (and the corresponding $A_m$) is that $m$ is 1, 2, or a multiple of 4 [10, Ch. 2].

**Definition 18.** The binary Hadamard matrix $A_m$ gives rise to three families of Hadamard codes:

1. The $(m, m - 1, \frac{m}{2})$ Hadamard code $H_{1,m}$ consists of the rows of $A_m$ with the first column deleted. Moreover, the codewords in $H_{1,m}$ that begin with 0 form the $(\frac{m}{2}, m - 2, \frac{m}{2})$ Hadamard code $H_{1,m}'$ if the initial zero is deleted.

2. The $(2m, m - 1, \frac{m}{2} - 1)$ Hadamard code $H_{2,m}$ consists of $H_{1,m}$ together with the complements of all its codewords.

3. The $(2m, m, \frac{m}{2})$ Hadamard code $H_{3,m}$ consists of the rows of $A_m$ and their complements.

Further Hadamard codes can be created by an arbitrary combination of the codebook matrices of different Hadamard codes.

**Example 19.** Consider the $(8, 7, 4)$ Hadamard code

$$H_{1,8} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \quad (37)$$

From this code, an $(8, 35, 20)$ Hadamard code can be constructed by simply concatenating $H_{1,8}$ five times.

Note that since the rows of $H_m$ are orthogonal, so are the columns of $H_m$, and thus it follows that each column of the corresponding matrix $A_m$ has a Hamming weight $\frac{m}{2}$. Moreover, by definition the first row of a binary Hadamard matrix is the all-zero row. Hence, we see that all Hadamard codes are weak flip codes, i.e., the family of weak flip codes is a superset of the family of Hadamard codes.

On the other hand, fair weak flip codes can be seen as a “subset” of Hadamard codes because for all parameters $(M, n)$ for which fair weak flip codes and also Hadamard codes exist, a Hadamard code can be constructed that is also a fair weak flip code. The problem with this statement lies in the fact that the Hadamard codes rely on the existence of Hadamard matrices, which in general is not guaranteed, i.e., it is difficult to predict whether for a given pair $(M, n)$, a Hadamard code exists or not. This is in stark contrast to weak flip codes (which exist for all $M$ and $n$) and fair weak flip codes (which exist for all $M$ and for all $n$ being a multiple of $L$).

We also remark that a Hadamard code of parameters $(M, n)$, for which fair weak flip codes exist, is not necessarily equivalent to a fair weak flip code.

\[ \text{Recall that the third parameter in } (M, n, d) \text{ denotes the minimum Hamming distance.} \]
Example 20. We continue with Example 19 and note that the \((8, 35, 20)\) Hadamard code that is constructed by five repetitions of the matrix \(H_{1,8}\) given in (37) is actually not a fair weak flip code since we have not used all possible weak flip candidate columns. However, it is possible to find five different \((8, 7, 4)\) Hadamard codes that combine to an \((8, 35, 20)\) fair weak flip code. Recall that the \((8, 35, 20)\) fair weak flip code is composed of all \(\binom{7}{4} = 35\) different weak flip columns.

Note that two Hadamard matrices are equivalent if one can be obtained from the other by permuting rows and columns and by multiplying rows and columns by \(-1\). In other words, Hadamard codes can actually be constructed from different sets of weak flip candidate columns.

3.5 Linear Codes

In conventional coding theory, *linear codes* form an important class of error correcting codes that have been shown to possess powerful algebraic properties. We review them here only briefly and refer to the vast existing literature for more details (e.g., see [2], [10]).

Definition 21. Let \(M = 2^k\), where \(k \in \mathbb{N} \triangleq \{1, 2, 3, \ldots\}\). The binary code \(C_{\text{lin}}^{(M,n)}\) is linear if its codewords span a \(k\)-dimensional subspace of the \(n\)-dimensional vector space over the channel input alphabet.

One of the most important property of a linear code is as follows.

Proposition 22. Let \(C_{\text{lin}}\) be linear and let \(x_m \in C_{\text{lin}}\) be given. Then the code obtained by adding \(x_m\) to each codeword of \(C_{\text{lin}}\) is equal to \(C_{\text{lin}}\).

Linear codes are weak flip codes:

Proposition 23. A linear \((M,n)\) binary code always contains the all-zero codeword, and each column of its codebook matrix has Hamming weight \(\frac{M}{2}\). Thus, every linear code is a weak flip code.

Note that linear codes only exist if \(M = 2^k\), while weak flip codes are defined for any \(M\). Also note that the converse of Proposition 23 does not necessarily hold, i.e., even if \(M = 2^k\) for some \(k \in \mathbb{N}\), a weak flip code \(C^{(M,n)}\) is not necessarily linear. In summary, we have the following relations among linear, weak flip, and arbitrary \((M,n)\) codes:

\[
\left\{ C_{\text{lin}}^{(M,n)} \right\} \subset \left\{ C_{\text{weak}}^{(M,n)} \right\} \subset \left\{ C^{(M,n)} \right\}.
\]  

(38)

Next, we will derive the set \(C_{\text{lin}}^{(M)}\) of all possible length-\(M\) candidate columns for the codebook matrices of binary linear codes with \(M = 2^k\) codewords. Being a subspace, linear codes are usually represented by a generator matrix \(G_{k \times n}\). We now apply our column-wise point-of-view to the construction of generator matrices.\(^{11}\) The generator matrix \(G_{k \times n}\) consists of \(n\) column vectors \(c_j\) of length \(k\) similar to (18). Note that since the generator matrix is a basis of the code subspace, the only useless column is the all-zero column, i.e., there are totally

\[K \triangleq 2^k - 1 = M - 1\]  

(39)

\(^{11}\)The authors in [20] have also used this approach to exhaustively examine all possible linear codes.
possible candidate columns for $G_{k \times n}$: $c_{j}^{(k)} \triangleq (b_{1} b_{2} \cdots b_{k})^{T}$, where $j = \sum_{i=1}^{k} b_{i} 2^{k-i}$ and where $b_{1}$ is not necessarily equal to zero. Let $U_{k}^{T}$ be an auxiliary $k \times K$ matrix consisting of all possible $K$ candidate columns for the generator matrix: $U_{k}^{T} = (c_{1}^{(k)} \cdots c_{K}^{(k)})$. This matrix $U_{k}^{T}$ then allows us to create the set of all possible candidate columns of length $M = 2^{k}$ for the codebook matrix of a linear code.

**Lemma 24.** Given a dimension $k$, the candidate columns set $C_{\text{lin}}^{(M)}$ for linear codes is given by the columns of the $M \times (M - 1)$ matrix

$$
\begin{pmatrix}
0 \\
U_{k}
\end{pmatrix}^{T},
$$

(40)

where $0$ denotes an all-zero row vector of length $k$.

Thus, the codebook matrix of any linear code can be represented by

$$
C_{\text{lin}}^{(M,n)} = \begin{pmatrix} 0 \\
U_{k}
\end{pmatrix} G_{k \times n},
$$

(41)

which consists of columns taken only from $C_{\text{lin}}^{(M)}$. Similarly to (32), since in its type all positions corresponding to candidate columns not in $C_{\text{lin}}^{(M)}$ are zero, we can also use a reduced type vector to describe a $k$-dimensional linear code:

$$
t_{\text{lin}} \triangleq \left[ t_{j_{1}}, t_{j_{2}}, \ldots, t_{j_{K}} \right],
$$

(42)

where $\sum_{\ell=1}^{K} t_{j_{\ell}} = n$ with $j_{\ell}$, $\ell = 1, \ldots, K$, representing the numbers of the corresponding candidate columns in $C_{\text{lin}}^{(M)}$.

**Definition 25.** A linear code is called *fair* if its codebook matrix is constructed by an equal number of all possible candidate columns in $C_{\text{lin}}^{(M)}$. Hence the blocklength of a fair linear code $C_{\text{lin, fair}}^{(M,n)}$ is always a multiple of $K = M - 1$.

**Example 26.** Consider the fair linear code with dimension $k = 3$ and blocklength $n = K = 7$:

$$
C_{\text{lin, fair}}^{(8,7)} = \begin{pmatrix} 0 \\
U_{3}
\end{pmatrix} U_{3}^{T} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1
\end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
$$

(43)

with the corresponding type vector

$$
t_{\text{lin}} = [t_{85}, t_{51}, t_{102}, t_{15}, t_{90}, t_{60}, t_{105}] = [1, 1, 1, 1, 1, 1].
$$

(44)

Note that the fair linear code with $k = 3$ and $n = 7$ is an $(8,7,4)$ Hadamard linear code with all pairwise Hamming distances equal to 4.
3.6 Plotkin Bound

Finally, we recall an important bound that holds for any \((M, n)\) code.

**Lemma 27 (Plotkin Bound [10])**. The minimum distance of an \((M, n)\) binary code \(C(M, n)\) always satisfies

\[
d_{\min}(C(M, n)) \leq \begin{cases} \frac{n \cdot M}{M - 1} & \text{M even}, \\ \frac{n \cdot M + 1}{M} & \text{M odd}. \end{cases} \tag{45} \]

Note that from the proof\(^{12}\) of Lemma 27, one can actually find that a necessary condition for a codebook to meet the Plotkin-bound is that the codebook is composed of weak flip candidate columns. Furthermore, Levenshtein [10, Ch. 2] proved that the Plotkin bound can be achieved provided that Hadamard matrices exist for orders divisible by 4.

4 Previous Results

4.1 SGB Bounds on the Average Error Probability

In [19], Shannon, Gallager, and Berlekamp derive upper and lower bounds on the average error probability of a given code used on a DMC. We quickly review their results.

**Definition 28.** For \(0 < s < 1\) we define

\[
\mu_{\alpha,\beta}(s) \triangleq \log \left( \sum_{y \in Y} P_{Y|X}(y|\alpha)^{1-s} P_{Y|X}(y|\beta)^s \right), \tag{46} \]

and

\[
\mu_{\alpha,\beta}(0) \triangleq \lim_{s \downarrow 0} \mu_{\alpha,\beta}(s), \tag{47} \]

\[
\mu_{\alpha,\beta}(1) \triangleq \lim_{s \uparrow 1} \mu_{\alpha,\beta}(s). \tag{48} \]

Then the *discrepancy* \(D^{(DMC)}(m, m')\) between \(x_m\) and \(x_{m'}\) is defined as

\[
D^{(DMC)}(m, m') \triangleq - \min_{0 \leq s \leq 1} \left\{ \sum_{\alpha \in \mathcal{X}} \sum_{\beta \in \mathcal{X}} \frac{d_{H;\alpha,\beta}(x_m, x_{m'})}{n} \mu_{\alpha,\beta}(s) \right\}, \tag{49} \]

where \(d_{H;\alpha,\beta}(x_m, x_{m'})\) denotes the number of positions \(j\) where \(x_{m,j} = \alpha\) and \(x_{m',j} = \beta\).

Note that the discrepancy is a generalization of the Hamming distance. However, it depends strongly on the conditional channel law. We use a superscript “\(^{(DMC)}\)” to indicate the channel to which the discrepancy refers.

---

\(^{12}\)We omit this proof, but instead refer to our generalization of the Plotkin Bound in Theorem 50 in Section 5.3.
Definition 29. The minimum discrepancy $D_{\text{min}}^{(\text{DMC})}(C^{(M,n)})$ for a codebook is the minimum value of $D^{(\text{DMC})}(m,m')$ over all pairs of codewords

$$D_{\text{min}}^{(\text{DMC})}(C^{(M,n)}) \triangleq \min_{m,m': m \neq m'} D^{(\text{DMC})}(m,m'). \quad (50)$$

The maximum minimum discrepancy is the maximum value of the minimum discrepancy over all possible codebooks $C^{(M,n)}$

$$\max_{C^{(M,n)}} D_{\text{min}}^{(\text{DMC})}(C^{(M,n)}) \quad (51)$$

Theorem 30 (SGB Bounds on Average Error Probability [19]). For an arbitrary DMC, the average error probability $P_e(C^{(M,n)})$ of a given code $C^{(M,n)}$ with $M$ codewords and blocklength $n$ is upper- and lower-bounded as follows:

$$\frac{1}{4M} e^{-n \left(D_{\text{min}}^{(\text{DMC})}(C^{(M,n)}) + \sqrt{\frac{2}{n} \log \frac{1}{P_{\text{min}}}}\right)} \leq P_e(C^{(M,n)}) \leq (M - 1) e^{-n D_{\text{min}}^{(\text{DMC})}(C^{(M,n)})} \quad (52)$$

where $P_{\text{min}}$ denotes the smallest nonzero transition probability of the DMC.

Note that these bounds are specific to a given code design (via $D_{\text{min}}^{(\text{DMC})}$). Therefore, the upper bound is a generally valid upper bound on the optimal performance, while the lower bound may not bound the optimal performance from below unless we apply it to the optimal code or to a suboptimal code that achieves the optimal $D_{\text{min}}^{(\text{DMC})}$. The bounds (52) however are tight enough to derive the error exponent of the DMC (for a fixed number $M$ of codewords).

Theorem 31 ([19]). The error exponent of a DMC for a fixed number $M$ of codewords $E_M \triangleq \lim_{n \to \infty} \max_{C^{(M,n)}} \left\{-\frac{1}{n} \log P_e(C^{(M,n)})\right\}$ is given as

$$E_M = \lim_{n \to \infty} \max_{C^{(M,n)}} D_{\text{min}}^{(\text{DMC})}(C^{(M,n)}). \quad (53)$$

Unfortunately, in general the evaluation of the error exponent is very difficult. For the class of so-called pairwise reversible channels, the calculation of the error exponent turns out to be uncomplicated.

Definition 32. A pairwise reversible channel is a DMC that has $\frac{d}{ds} \mu_{\alpha,\beta}(s) \big|_{s=\frac{1}{2}} = 0$ for any inputs $\alpha, \beta$.

Clearly, the BEC is a pairwise reversible channel.

Note that it is easy to compute the pairwise discrepancy of a linear code on a pairwise reversible channel, so linear codes are quite suitable for computing (52).

Theorem 33 ([19]). For pairwise reversible channels with $M > 2$,

$$E_M = \frac{1}{M(M-1)} \max_{\{M_x\} \text{ s.t. } \sum_x M_x = M} \left\{-\sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} M_x M_{x'} \log \left(\sum_{y \in \mathcal{Y}} \sqrt{P_{Y|x}(y|x) P_{Y|x}(y|x')}\right)\right\}, \quad (55)$$

where each $M_x$ denotes a nonnegative integer satisfying $\sum_{x \in \mathcal{X}} M_x = M$. Moreover, $E_M$ is achieved by fair weak flip codes.
We would like to emphasize that while Shannon et al. proved that fair weak flip codes achieve the error exponent, they did not investigate the error performance of fair weak flip codes for finite $n$. As we will see later, fair weak flip codes can be strictly suboptimal for finite $n$ for the BSC (see also [17], [21]).

### 4.2 PPV Bounds for the BEC

In [22], Polyanskiy, Poor, and Verdú present upper and lower bounds on the optimal average error probability for finite blocklength for general DMCs. For some special cases like the BSC or the BEC, these bounds can be expressed explicitly by closed-form formulas. The upper bound is based on random coding.

**Theorem 34 (PPV Upper Bound [22, Th. 36]).** For the BEC with erasure probability $\delta$, if the codebook $C^{(M,n)}$ is created at random based on a uniform distribution, the expected average error probability (averaged over all codewords and all codebooks) satisfies

$$
E \left[ P_e \left( C^{(M,n)} \right) \right] = 1 - \sum_{j=0}^{n} \binom{n}{j} (1 - \delta)^j \delta^{n-j} \sum_{m=0}^{M-1} \frac{1}{m+1} \binom{M-1}{m} (2^{-j})^m (1 - 2^{-j})^{M-1-m}.
$$

Note that there must exist a codebook whose average error probability achieves (56), so Theorem 34 provides a general achievable upper bound on the error probability, although we do not know the concrete code structure.

Polyanskiy, Poor, and Verdú also provide a new general converse for the average error probability, based on which a closed-form formula can be derived for the BEC.

**Theorem 35 (PPV Lower Bound [22, Th. 38]).** For the BEC with erasure probability $\delta$, any codebook $C^{(M,n)}$ satisfies

$$
P_e \left( C^{(M,n)} \right) \geq \sum_{e=\lfloor n - \log_2 M \rfloor + 1}^{n} \left( \begin{array}{c} n \\ e \end{array} \right) \delta^e (1 - \delta)^{n-e} \left( 1 - \frac{2^{n-e}}{M} \right).
$$

Note that (57) was first derived based on an “ad hoc” (i.e., BEC specific) argument in [22]. It is then shown in [23] that the same result can also be obtained using the so-called meta-converse methodology.

### 5 Column-Wise Analysis of Codes

#### 5.1 $r$-Wise Hamming Distance and $r$-Wise Hamming Match

The minimum Hamming distance is a well-known and widely used quality criterion of a code. Unfortunately, a design solely based on the minimum Hamming distance can be strictly suboptimal even for a very symmetric channel like the BSC and even for linear codes [17], [21].

We therefore start by defining a slightly more general and more concise description of a code: the pairwise Hamming distance vector.

---

13 This is in spite of the fact that the error probability performance of a BSC is completely specified by the Hamming distances between codewords and received vectors!
Definition 36. The \textit{pairwise Hamming distance vector} \(d^{(M,n)}\) of a code \(\mathcal{C}^{(M,n)}\) is defined as the length-\((\frac{1}{2}(M-1)M)\) vector containing as components the Hamming distances of all possible codeword pairs:

\[
d^{(M,n)} \triangleq \left( d_{12}^{(n)}, d_{13}^{(n)}, d_{23}^{(n)}, d_{14}^{(n)}, d_{24}^{(n)}, d_{34}^{(n)}, \ldots, d_{1M}^{(n)}, d_{2M}^{(n)}, \ldots, d_{(M-1)M}^{(n)} \right)
\]

with \(d_{mm}^{(n)} \triangleq d_H(x_m, x_{m'})\), \(1 \leq m < m' \leq M\). We remind the reader of our convention to number the codewords according to rows in the codebook matrix, see (18).

The \textit{minimum Hamming distance} \(d_{\text{min}}\) is then the minimum component of the pairwise Hamming distance vector \(d^{(M,n)}\).

Note that for this definition it is completely irrelevant whether the code is linear or not.

The pairwise Hamming distance vector can directly be derived using the column-wise description of our code, i.e., using the type \(t\) of a code.

Lemma 37. The pairwise Hamming distance vector of a general code of type \(t\) for \(M = 3\) or \(M = 4\) is given as follows:

\[
d^{(3,n)} = (d_{12}^{(n)}, d_{13}^{(n)}, d_{23}^{(n)}) = (t_2 + t_3, t_1 + t_3, t_1 + t_2)
\]

(59)

\[
d^{(4,n)} = (d_{12}^{(n)}, d_{13}^{(n)}, d_{14}^{(n)}, d_{23}^{(n)}, d_{24}^{(n)}, d_{34}^{(n)})
\]

(60)

\[
d^{(4,n)} = (t_4 + t_5 + t_6 + t_7, t_2 + t_3 + t_6 + t_7, t_2 + t_3 + t_4 + t_5, t_1 + t_3 + t_5 + t_7, t_1 + t_3 + t_4 + t_6, t_1 + t_2 + t_5 + t_6)
\]

(61)

\[
d^{(4,n)} = (n - (t_1 + t_2 + t_3), n - (t_1 + t_4 + t_5), n - (t_1 + t_6 + t_7), n - (t_2 + t_4 + t_6), n - (t_2 + t_5 + t_7), n - (t_3 + t_4 + t_7)).
\]

(62)

(63)

(64)

While the pairwise Hamming distance vector already contains more information about a particular code than simply the minimum Hamming distance, it is still not sufficient to describe the exact performance of a code. We will therefore next provide an extension of the pairwise Hamming distance: the so-called \(r\)-wise Hamming Distance and \(r\)-Wise Hamming Match. For a given general codebook \(\mathcal{C}^{(M,n)}\) and an arbitrary integer \(2 \leq r \leq M\), we fix some integers \(1 \leq i_1 < i_2 < \cdots < i_r \leq M\) and define the \textit{r-wise Hamming match} \(a_{i_1,i_2,\ldots,i_r}(\mathcal{C}^{(M,n)})\) to be the number of codebook columns \(c\) whose \(i_1\)th, \(i_2\)th, \(i_r\)th coordinates are all identical:

\[
a_{i_1,i_2,\ldots,i_r}(\mathcal{C}^{(M,n)}) \triangleq \left| \left\{ j \in \{1, \ldots, n\} : c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} \right\} \right|,
\]

\[1 \leq i_1 < i_2 < \cdots < i_r \leq M.\] (65)

The \textit{r-wise Hamming distance} \(d_{i_1,i_2,\ldots,i_r}(\mathcal{C}^{(M,n)})\) is accordingly defined as

\[
d_{i_1,i_2,\ldots,i_r}(\mathcal{C}^{(M,n)}) \triangleq n - a_{i_1,i_2,\ldots,i_r}(\mathcal{C}^{(M,n)}), \quad 1 \leq i_1 < i_2 < \cdots < i_r \leq M.\] (66)

Lin, Moser, Chen, version 7, June 17, 2016 18
It is straightforward to verify that the 2-wise Hamming distances according to Definition 38 are identical to the pairwise Hamming distances given in the pairwise Hamming distance vector (58).

The $r$-wise Hamming distances can be written elegantly with the help of the type vector:

$$d_{i_1 i_2 \ldots i_r}^{(M,n)}(\mathcal{C}) = n - \sum_{j \in \mathcal{J}} t_j, \quad 1 \leq i_1 < i_2 < \cdots < i_r \leq M. \quad (67)$$

Here $t_j$ denotes the $j$th component of the type vector $\mathbf{t}$ of length $\mathcal{J} = 2^{M-1} - 1$, and $c_{j,i_r}$ is the $i_r$th component of the $j$th candidate column $\mathbf{c}_j^{(M)}$ as given in Definition 10, and $\mathcal{J} \triangleq \{1, \ldots, 2^{M-1} - 1\} = \{1, \ldots, \}$ was defined in (22).

When the considered type-$\mathbf{t}$ code is unambiguous from the context, we will usually omit the explicit specification of the code and abbreviate (65) and (66) as $d_{i_1 i_2 \ldots i_r}^{(M,n)}$ or, even shorter, as $d_{\mathcal{I}}^{(M,n)}$ and $d_{\mathcal{I}}^{(M,n)}$ for some given $\mathcal{I} = \{i_1, i_2, \ldots, i_r\}$. Note that there are $\binom{M}{r}$ different choices of parameters $1 \leq i_1 < i_2 < \cdots < i_r \leq M$, i.e., there are $\binom{M}{r}$ different $r$-wise Hamming distances per code.

**Example 39.** For $M = 4$ and $r = 3$, there are $\binom{M}{r} = \binom{4}{3} = 4$ different 3-wise Hamming distances:

$$d_{123}^{(4,n)} = n - t_1, \quad d_{124}^{(4,n)} = n - t_2, \quad d_{134}^{(4,n)} = n - t_4, \quad d_{234}^{(4,n)} = n - t_7, \quad (68)$$

and there is only one 4-wise Hamming distance: $d_{1234}^{(4,n)} = n$. \hfill \lozenge

The definition of the $r$-wise Hamming distances leads to a natural extension of the minimum Hamming distance.

**Definition 40** (Minimum $r$-Wise Hamming Distance). For a given $r \in \{2, \ldots, M\}$, the minimum $r$-wise Hamming distance $d_{\text{min};r}$ of a code $\mathcal{C}^{(M,n)}$ is defined as the minimum of all possible $r$-wise Hamming distances of this $(M,n)$ code:

$$d_{\text{min};r}(\mathcal{C}^{(M,n)}) \triangleq \min_{\mathcal{I} \subseteq \{1, \ldots, M\}, |\mathcal{I}| = r} d_{\mathcal{I}}(\mathcal{C}^{(M,n)}), \quad (69)$$

where the minimization is over all size-$r$ subsets $\mathcal{I} \subseteq \{1, \ldots, M\}$.

Correspondingly, the maximum $r$-wise Hamming match $a_{\text{max};r}$ is defined as the maximum of all possible $r$-wise Hamming matches $a_{\mathcal{I}}(\mathcal{C}^{(M,n)})$ and is given by

$$a_{\text{max};r}(\mathcal{C}^{(M,n)}) = n - d_{\text{min};r}(\mathcal{C}^{(M,n)}). \quad (70)$$

Recall that in traditional coding theory it is customary to specify a code with three parameters $(M, n, d_{H,\text{min}})$, where the third parameter specifies the minimum pairwise Hamming distance. We follow this tradition but replace the minimum pairwise Hamming distance by a vector containing all minimum $r$-wise Hamming distances for $r = 2, \ldots, \ell$:

$$d_{\text{min}} \triangleq (d_{\text{min};2}, d_{\text{min};3}, \ldots, d_{\text{min};\ell}). \quad (71)$$

The reason why we restrict ourselves to $r \leq \ell$ lies in the fact that for weak flip codes the minimum $r$-wise Hamming distance is only relevant for $2 \leq r \leq \ell$; see the remark after Theorem 50 below.
Example 41. We continue with Example 26. The fair linear code with \( k = 3 \) and \( n = 7 \) given in (43) is an \((8, 7, d_{\text{min}})\) Hadamard linear code with \( d_{\text{min}} = (4, 6, 6) \). Similarly, the fair linear code with \( k = 3 \) and \( n = 35 \) that is created by concatenating the codebook matrix (43) five times is an \((8, 35, (20, 30, 30))\) Hadamard linear code.

Both codes are obviously not fair weak flip codes for \( M = 8 \). Later in Theorem 52 we will show that the fair weak flip code with \( M = 8 \) codewords is actually an \((8, 35, (20, 30, 34))\) code. \( \diamondsuit \)

In [11], Wei defines the \( s \)th generalized Hamming weight of a \( k \)-dimensional linear code as the minimum support of any \( s \)-dimensional linear subcode, where the \emph{support} is the number of codebit positions at which not all codewords are zero. Obviously, this definition is strongly restricted because firstly it is only defined for a linear code, and because secondly in general an arbitrarily picked subset of codewords of a linear code is not a linear subcode, i.e., Wei only considers a very much limited number of subsets of codewords taken from the given linear code. Nevertheless, it can be shown that if we pick \( 2^s \) codewords \((s \leq k)\) from a \( k \)-dimensional linear code in such a way that these \( 2^s \) codewords form a linear subcode, then the \( s \)th generalized Hamming weight is equal to the smallest \( r \)-wise Hamming distance among all \( r \) satisfying \( 2^{s-1} < r \leq 2^s \) [24], [25].

Following the classical definition of an \emph{equidistant code} being a code whose pairwise Hamming distance between all codewords is the same, we extend this definition to the \( r \)-wise Hamming distance and define \emph{\( r \)-wise equidistant codes}.

Definition 42 \((r\text{-Wise Equidistant Codes})\). For a given integer \( 2 \leq r \leq M \), an \((M, n)\) code \( C^{(M,n)} \) is called \emph{\( r \)-wise equidistant} if all \( r \)-wise Hamming distances are equal, i.e., if for all choices of integers \( 1 \leq i_1 < i_2 < \cdots < i_r \leq M \)

\[
d_{i_1 \cdots i_r} (C^{(M,n)}) = \text{constant.} \tag{72}\]

We end this section with a relation between the \( r \)-wise Hamming distance and the type vector of a code. To that goal, we first state a property regarding the number of candidate columns with \( r \) equal components.

Lemma 43. For any integer \( 2 \leq r \leq M \) and any choice \( 1 \leq i_1 < i_2 < \cdots < i_r \leq M \), the cardinality of the index set\(^{14}\)

\[
J_{i_1 \cdots i_r} \triangleq \{ j \in J : c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} \} \tag{73}\]

is equal to \( 2^{M-r} - 1 \).

\textbf{Proof:} First, consider the case when \( i_1 = 1 \). Since the first position of each candidate column is always equal to zero, we only need to consider those \( j \in J \) such that \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 0 \). There are in total \( 2^{M-r} \) such columns, but we need to subtract 1 because we exclude the all-zero column.

Second, consider the case when \( i_1 > 1 \). Since the first position is fixed to zero, we ignore it. There are \( 2^{M-1-r} \) columns with \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 0 \) and the same amount with \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 1 \). Once again excluding the all-zero column, we have in total \( 2 \cdot 2^{M-1-r} - 1 \) possible columns. \( \square \)

\(^{14}\)Here again, \( c_{j,i} \) denotes the \( i \)th component of the \( j \)th candidate column \( c_j^{(M)} \).
Corollary 44. The $r$-wise Hamming distance $d_{12...r}(c^{(t,n)}_{t})$ of the first $r$ codewords is given by

$$d_{12...r}^{(M,n)} = \sum_{j=2^{M-r}}^{J} t_{j}. \quad (74)$$

If every candidate column in $C^{(M)}$ is used exactly once in $c^{(t,n)}_{t}$, i.e., $t_{j} = 1$ for $1 \leq j \leq J$, then all $r$-wise Hamming distances $d_{i_{1}...i_{r}}^{(M,n)}$ have an identical value:

$$d_{i_{1}...i_{r}}^{(M,n)} = 2^{M-1} - 2^{M-r}, \quad 1 \leq i_{1} < \cdots < i_{r} \leq M. \quad (75)$$

**Proof:** By the numbering system in Definition 10, together with Definition 38 and Lemma 43, we have

$$d_{12...r}^{(M,n)} = n - \sum_{j \in J_{1}...r} t_{j} = n - \sum_{j=1}^{2^{M-r}-1} t_{j} = \sum_{j=2^{M-r}}^{J} t_{j}. \quad (76)$$

If $t_{1} = t_{2} = \cdots = t_{J} = 1$ (see (21)), we obtain again by Lemma 43 for arbitrary $1 \leq i_{1} < \cdots < i_{r} \leq M$,

$$d_{i_{1}...i_{r}}^{(M,n)} = 2^{M-1} - 1 - \sum_{j=1}^{2^{M-r}-1} 1 = 2^{M-1} - 2^{M-r} = d_{12...r}^{(M,n)}. \quad (77)$$

5.2 Characteristics of Weak Flip Codes

In this section, we concentrate on the analysis of the family of weak flip codes.

First, we pose the question which of the many powerful algebraic properties of linear codes are retained in weak flip codes.

**Theorem 45.** Consider a weak flip code $c^{(M,n)}_{weak}$ and fix some codeword $x_{m} \in c^{(M,n)}_{weak}$. If we add this codeword to all codewords in $c^{(M,n)}_{weak}$, then the resulting code

$$\tilde{C}^{(M,n)} \triangleq \left\{ x \oplus x_{m} : x \in c^{(M,n)}_{weak} \right\} \quad (78)$$

is still a weak flip code; however, it is not necessarily the same one.

**Proof:** Let $c^{(M,n)}_{weak}$ be a weak flip code according to Definition 15. We have to prove that

$$\begin{pmatrix} x_{1} \\ x_{2} \\ \vdots \\ x_{M} \end{pmatrix} \oplus \begin{pmatrix} x_{m} \\ x_{m} \\ \vdots \\ x_{m} \end{pmatrix} = \begin{pmatrix} x_{1} \oplus x_{m} \\ \vdots \\ x_{m} \oplus x_{m} = 0 \\ \vdots \\ x_{M} \oplus x_{m} \end{pmatrix} \triangleq \tilde{c}^{(M,n)} \quad (79)$$

is a weak flip code. Let $c_{j}$, $1 \leq j \leq n$, denote the $j$th column vector of the code matrix of $c^{(M,n)}_{weak}$. Then $\tilde{c}^{(M,n)}$ has the column vectors

$$\tilde{c}_{j} = \begin{cases} c_{j} & \text{if } x_{m,j} = 0, \\ \tilde{c}_{j} & \text{if } x_{m,j} = 1. \end{cases} \quad (80)$$
Since \( c_j \) is a weak flip column, either \( w_H(c_j) = \left\lceil \frac{M}{2} \right\rceil \) or \( w_H(c_j) = \left\lfloor \frac{M}{2} \right\rfloor \), which implies that either \( w_H(\bar{c}_j) = \left\lceil \frac{M}{2} \right\rceil \) or \( w_H(\bar{c}_j) = \left\lfloor \frac{M}{2} \right\rfloor \). Now it only remains to interchange the first codeword of \( \mathcal{C}(M,n) \) and the all-zero codeword in the \( m \)th row in \( \mathcal{C}(M,n) \) (which is always possible, see Remark 8). As a result, \( \mathcal{C}(M,n) \) is also a weak flip code.

Theorem 45 is a beautiful property of weak flip codes; however, it still represents a considerable weakening of the powerful property of linear codes given in Proposition 22. This can be fixed by considering the subfamily of fair weak flip codes.

**Theorem 46 (Quasi-Linear Codes).** Let \( \mathcal{C}_{\text{fair}}(M,n) \) be a fair weak flip code and let \( x_m \in \mathcal{C}_{\text{fair}}(M,n) \) be given. Then the code

\[
\mathcal{C}(M,n) \triangleq \left\{ x \oplus x_m : x \in \mathcal{C}_{\text{fair}}(M,n) \right\}
\]

is equivalent to \( \mathcal{C}_{\text{fair}}(M,n) \).

**Proof:** We divide the weak flip candidate columns in \( \mathcal{C}_{\text{weak}}(M) \) into two subfamilies: one subfamily consists of the columns with the \( m \)th component being zero, and the columns in the other subfamily have their \( m \)th component equal to one. Next we add the \( m \)th codeword to the codewords in \( \mathcal{C}_{\text{fair}}(M,n) \) and then interchange the first and \( m \)th components of each column in the code matrix of \( \mathcal{C}_{\text{fair}}(M,n) \) to form a new code \( \mathcal{C}(M,n) \). It is apparent that the columns in the first subfamily are unchanged by such code-addition-and-interchanging manipulation. However, when \( M \) is odd, the weights of columns in the second subfamily change either from \( \ell \) to \( \bar{\ell} \), or from \( \ell \) to \( \bar{\ell} \), while these weights stay the same when \( M \) is even. As a result, after such code-addition-and-interchanging manipulation, the columns belonging to the second subfamily remain distinct weak flip columns and are still contained in the second subfamily (since their \( m \)th components are still equal to one). Thus, all the weak flip columns remain to be used equally in \( \mathcal{C}(M,n) \), showing that \( \mathcal{C}(M,n) \) is fair.

Comparing Theorem 46 with Proposition 22 and recalling Proposition 23 and the discussion after it, we realize that the family of fair weak flip codes is a considerable enlargement of the family of linear codes.

The following corollary is a direct consequence of Lemma 43.

**Corollary 47.** For any integer \( 2 \leq r \leq M \), the \( r \)-wise Hamming distances \( d^{(M,n)}_{i_1 \ldots i_r} \) of a fair weak flip code \( \mathcal{C}_{\text{fair}}(M,n) \) for any choice \( 1 \leq i_1 < i_2 < \cdots < i_r \leq M \), are all identical and are given by

\[
d^{(M,n)}_{i_1 \ldots i_r} = \frac{n}{L} d^{(M,L)}_{i_1 \ldots i_r} = \frac{n}{L} \left[ L - \left( \frac{2\bar{\ell} - r}{\ell} \right) \right].
\]

**Proof:** By definition of a fair weak flip code, we observe that the \( r \)-wise Hamming distance of arbitrary \( r \) codewords is a fixed integer multiple (i.e., \( n/L \)) of the \( r \)-wise Hamming distance \( d^{(M,L)}_{i_1 \ldots i_r} \) of a fair weak flip code of blocklength \( n = L \).

We apply the proof idea of Lemma 43. When \( M = 2\ell - 1 \) is odd, first consider the case of \( i_1 = 1 \). Since the first position of each weak flip column is always equal to zero, the number of weak flip columns with weight \( \ell \) such that \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 0 \) equals \( \left( \begin{array}{c} M - r \\ \ell - 1 \end{array} \right) \), and the number of weak flip columns with weight \( \ell \) is \( \left( \begin{array}{c} M - r \\ \ell \end{array} \right) \). In total, we have the \( r \)-wise Hamming match

\[
d^{(M,n)}_{1 \ldots i_2 \ldots i_r} = \frac{n}{L} \left( \left( \begin{array}{c} M - r \\ \ell - 1 \end{array} \right) + \left( \begin{array}{c} M - r \\ \ell \end{array} \right) \right) = \frac{n}{L} \left( \frac{2\bar{\ell} - r}{\ell} \right),
\]
where we take \( M = 2\ell - 1 \) in the last equality.

Second, consider the case when \( i_1 > 1 \). Since the first position is fixed to zero, we ignore it. There are \( ^{M-r-1}_{\ell-1} \) columns with weight \( \ell - 1 \) such that \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 0 \), and there are \( ^{M-r-1}_{\ell} \) columns with weight \( \ell \) such that \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 0 \). Similarly, there are \( ^{M-r-1}_{\ell-1} \) columns with weight \( \ell - 1 \) such that \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 1 \), and there are \( ^{M-r-1}_{\ell} \) columns with weight \( \ell \) such that \( c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} = 1 \). In total we have the \( r \)-wise Hamming match

\[
e^{(M,n)}_{i_1 \ldots i_r} = \frac{n}{\ell} \left[ \binom{M-r-1}{\ell-1} + \binom{M-r-1}{\ell} + \binom{M-r-1}{\ell-1} + \binom{M-r-1}{\ell-r} \right]
\]

where in the last equality we use \( M = 2\ell - 1 \).

In a similar way, given \( M = 2\ell \) is even, we obtain

\[
a^{(M,n)}_{i_1 \ldots i_r} = \begin{cases} 
\frac{n}{\ell} \binom{M-r}{\ell} & \text{if } i_1 = 1, \\
\frac{n}{\ell} \left[ \binom{M-r-1}{\ell-1} + \binom{M-r-1}{\ell-r} \right] & \text{if } i_1 > 1.
\end{cases}
\]

The proof is completed by combining all possible cases using that \( d^{(M,n)}_{i_1 \ldots i_r} = n - a^{(M,n)}_{i_1 \ldots i_r} \).

Recall that for a given choice of \( r \) column positions \( 1 \leq i_1 < i_2 < \cdots < i_r \leq M \), the \( r \)-wise Hamming match counts how many columns exist in the codebook matrix that have identical entries in these \( r \) positions. Now we would like to look at this the other way around: for a fixed candidate column, we would like to count how many different choices of \( r \) positions \( 1 \leq i_1 < i_2 < \cdots < i_r \leq M \) exist such that all these positions have identical entries.

Since a candidate column with Hamming weight equal to \( h \) has \( h \) components of value 1 and \( M-h \) components of value 0, it is easy to see that the following lemma always holds.

**Lemma 48.** For given an integer \( 2 \leq r \leq \ell \) and an arbitrary candidate column \( c_j \), \( j = 1, \ldots, j \), the cardinality of the set

\[
\{ (i_1, i_2, \ldots, i_r) : 1 \leq i_1 < i_2 < \cdots < i_r \leq M, \ c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_r} \}
\]

is equal to \( \binom{h}{r} + \binom{M-h}{r} \), where \( h = \omega_{H}(c_j) \).

Finally, we simplify Lemma 37 for the situation of a weak flip code:

**Corollary 49.** For \( M = 4 \), the pairwise Hamming distance vector of a weak flip code of type \( t_{\text{weak}} \) can be simplified from (64) as follows:

\[
d^{(4,n)} = (t_5 + t_6, t_3 + t_6, t_3 + t_5, t_3 + t_6, t_3 + t_5, t_3 + t_6)
\]

i.e., each \( t_{jw}, w = 1, 2, 3 \), shows up exactly twice.

Note that for \( M = 3 \), the expression for the pairwise Hamming distance vector of a weak flip code of type \( t_{\text{weak}} \) is as given in (61) and cannot be simplified further.
5.3 Generalized Plotkin Bound for the $r$-wise Hamming Distance

The $r$-wise Hamming distance (together with the type vector $t$) plays an important role in the closed-form expression of the average error probability for an arbitrary code $C_{t}^{(M,n)}$ over a BEC. It is therefore interesting to find some bounds on the $r$-wise Hamming distance. We start with a generalization of the Plotkin bound for the minimum pairwise Hamming distance to the situation of the minimum $r$-wise Hamming distance.

**Theorem 50 (Plotkin Bound for the Minimum $r$-wise Hamming Distance).**
The minimum $r$-wise Hamming distance with $2 \leq r \leq M$ of an $(M,n)$ binary code satisfies

$$d_{\min;r}(C^{(M,n)}) \leq \begin{cases} \frac{n \left( 1 - \frac{(\ell - 1)}{(2^{r-1})} \right)}{r} & \text{if } 2 \leq r \leq \ell, \\ n & \text{if } \ell < r \leq M. \end{cases}$$  \hspace{1cm} (91)

*Proof:* The bound for $r > \ell$ is trivial and therefore needs no proof. We focus on $2 \leq r \leq \ell$. Note that because there are $M(M-1)\cdots(M-r+1)$ different choices for $1 \leq i_1 < \cdots < i_r \leq M$, we have

$$\sum_{\|I\|=r: I \subseteq \{1,\ldots,M\}} a_{I}(C^{(M,n)}) \leq M(M-1)\cdots(M-r+1) \cdot a_{\max;r}(C^{(M,n)}). \hspace{1cm} (92)$$

On the other hand, if we look at the codebook matrix $C^{(M,n)}$ from a column-wise point of view and define $h_j$ to be the number of zeros in the $j$th column (and hence $M - h_j$ to be the number of ones in the $j$th column), we see that the $j$th column contributes $h_j(h_j - 1)\cdots(h_j - r + 1)$ possible choices of picking $r$ different components that are zero and $(M - h_j)(M - h_j - 1)\cdots(M - h_j - r + 1)$ choices of picking $r$ different
components that all are one. Hence,\textsuperscript{15}  

\[
\sum_{I \subseteq \{1, \ldots, M\} : |I| = r} a_{I}(\mathcal{G}(M,n)) = \sum_{j=1}^{n} \left[ h_{j}(h_{j} - 1) \cdots (h_{j} - r + 1) 
+ \left( M - h_{j} \right) \left( M - h_{j} - 1 \right) \cdots (M - h_{j} - r + 1) \right] \geq n \left[ \ell(\ell - 1) \cdots (\ell - r + 1) + \ell(\ell - 1) \cdots (\ell - r + 1) \right],
\]

(95)

where the lower bound is achieved if \( h_{j} = \bar{\ell} \) or \( \ell \) for all \( j = 1, \ldots, n \), i.e., if the columns are weak flip columns. Note that when \( r = \bar{\ell} \) and \( M \) is odd, the first term in the bracket in (96) is zero because \( (\ell - r + 1) = (\ell - \ell + 1) = 0 \).

Combining (92) and (96) (and separately calculating the cases where \( M \) is even or odd), we obtain

\[
a_{\max,r}(\mathcal{G}(M,n)) \geq \begin{cases} 
\frac{n}{2} \ell(\ell-1)(\ell-2)\cdots(\ell-r+1) & \text{if } M = 2\bar{\ell}, \\
\frac{n}{2(2\ell-1)}(\ell-1)(\ell-2)\cdots(\ell-r+1)(\ell-r+1)(\ell-r+2)(\ell-r) & \text{if } M = 2\bar{\ell} - 1 \\
\frac{n}{2\ell-1} & \text{if } M = 2\bar{\ell} - 1
\end{cases}
\]

(97)

(98)

The above theorem only provides absorbing bounds to the \( r \)-wise Hamming distance for \( 2 \leq r \leq \bar{\ell} \), while further increasing the parameter \( r \) only renders trivially \( d_{\min,r} \leq n \). Since the minimum \( r \)-wise Hamming distance of a weak flip code for \( r > \bar{\ell} \) is always equal to this trivial bound \( n \) and therefore is irrelevant for the exact error performance, the vector (71) contains the minimum \( r \)-wise Hamming distances for \( 2 \leq r \leq \bar{\ell} \) only.

It is well-known that Hadamard codes achieve the Plotkin bound (Lemma 27) with equality, i.e., they achieve the largest minimum pairwise Hamming distance (or

\textsuperscript{15} Under \( r \leq h_{j} \leq M - h_{j} \), (95) can be lower bounded as follows:

\[
h_{j}(h_{j} - 1) \cdots (h_{j} - r + 1) + (M - h_{j})(M - h_{j} - 1) \cdots (M - h_{j} - r + 1) 
= r! \left[ \left( \frac{h_{j}}{r} \right) + \left( \frac{M - h_{j}}{r} \right) \right] \geq r! \left[ \left( \frac{h_{j} + 1}{r} \right) + \left( \frac{M - h_{j} - 1}{r} \right) \right] \geq \cdots \geq r! \left[ \left( \frac{\ell}{r} \right) + \left( \frac{\ell}{r} \right) \right],
\]

(93)

where the first inequality holds as long as \( M - h_{j} - 1 \geq h_{j} \) because

\[
\left[ \left( \frac{h_{j}}{r} \right) + \left( \frac{M - h_{j}}{r} \right) \right] - \left[ \left( \frac{h_{j} + 1}{r} \right) + \left( \frac{M - h_{j} - 1}{r} \right) \right] = \left( \frac{M - h_{j} - 1}{r - 1} \right) - \left( \frac{h_{j}}{r - 1} \right) \geq 0
\]

(94)

and we can continue the process of adding one to the top number in the first binomial coefficient and meanwhile subtracting one from the top number in the second binomial coefficient until the last inequality in (93) is reached. The same argument can be used to validate (96) under \( r \leq M - h_{j} \leq h_{j} \). In the special case that \( h_{j} < r \leq M - h_{j} \) (or \( M - h_{j} < r \leq h_{j} \)), which occurs definitely when \( r = \bar{\ell} \) and \( M \) odd, (95) should be refined to

\[
\max\{h_{j}(h_{j} - 1) \cdots (h_{j} - r + 1), 0\} + \max\{(M - h_{j})(M - h_{j} - 1) \cdots (M - h_{j} - r + 1), 0\} 
\geq \max\{(h_{j} + 1)(h_{j} - 1) \cdots (h_{j} - r + 2), 0\} + \max\{(M - h_{j} - 1)(M - h_{j} - 2) \cdots (M - h_{j} - r), 0\} 
\geq \max\{(h_{j} + 2)(h_{j} + 1) \cdots (h_{j} - r + 3), 0\} + \max\{(M - h_{j} - 2)(M - h_{j} - 3) \cdots (M - h_{j} - r - 1), 0\} 
\geq \cdots
\]

for which the process can be repeated \( (r - h_{j}) \) times to reach the case considered in (93); hence (96) still holds.
equivalently, the largest minimum 2-wise Hamming distance) [10, Ch. 2]. Moreover, Hadamard codes are also (pairwise) equidistant. In the following we will investigate generalizations of these two properties for weak flip codes. We will show the following:

1. If a weak flip code (of a certain blocklength $n$) is $r$-wise equidistant, then it is also $s$-wise equidistant for all $s = 2, \ldots, r - 1$.
2. If in addition to be $r$-wise equidistant, it also achieves the $r$-wise Plotkin bound (Theorem 50), then it also achieves the $s$-wise Plotkin bound for all $s = 2, \ldots, r - 1$.
3. Fair weak flip codes are $r$-wise equidistant and achieve the $r$-wise Plotkin bound for all $2 \leq r \leq M$.

The proof will make use of $s$-designs [26] from combinatorial design theory:

**Definition 51** ([26, Ch. 9]). Let $v, \kappa, \lambda_s$, and $s$ be positive integers such that $v > \kappa \geq s$.

An $s$-$(v, \kappa, \lambda_s)$ design or simply $s$-design is a pair $(\mathcal{X}, \mathcal{B})$, where $\mathcal{X}$ is a set of size $v$ and $\mathcal{B}$ is a collection of subsets of $\mathcal{X}$ (called blocks), such that the following properties are satisfied:

1. each block $B \in \mathcal{B}$ contains exactly $\kappa$ points, and
2. every set of $s$ distinct points is contained in exactly $\lambda_s$ blocks.

We now claim that some specific weak flip codes (for an arbitrary $M$ and for certain blocklengths) can be seen as $r$-designs with $2 \leq r \leq \bar{\ell}$ and achieve the generalized Plotkin upper bound (91) with equality (again, it is trivial to see that their $d_{\min;r}$ for $r > \bar{\ell}$ are equal to $n$).

**Theorem 52.** Fix some $M$, a blocklength $n$ with $n \mod L = 0$, and some $2 \leq r \leq \bar{\ell}$.

Then if a weak flip code is $r$-wise equidistant, then it is also $s$-wise equidistant for all $2 \leq s < r$. Moreover, if this $r$-wise equidistant weak flip code $C_{\text{equidist}}^{(M,n)}$ also achieves the generalized Plotkin bound (and hence achieves the largest minimum $r$-wise Hamming distance), i.e., it satisfies

$$d_{\min;r}(C_{\text{equidist}}^{(M,n)}) = n \left(1 - \frac{(\bar{\ell} - 1)(r-1)}{(2r-1)}\right),$$

then $C_{\text{equidist}}^{(M,n)}$ must also achieve the largest minimum $s$-wise Hamming distances for all $2 \leq s < r$.

**Proof:** We start by explaining how we connect the $r$-wise Hamming distance with $2 \leq r \leq \bar{\ell}$ of an $r$-wise equidistant weak flip code to the $s$-$(v, \kappa, \lambda_s)$ design. Consider an $r$-wise equidistant weak flip code with a certain blocklength $n$. Let $\mathcal{M} \triangleq \{1, 2, \ldots, M\}$. Denote by $\mathcal{B}$ the collection containing all $\ell$-size subsets $\mathcal{B} \triangleq \{i_1, i_2, \ldots, i_\ell\} \subseteq \mathcal{M}$ such that $c_{j,i_1} = c_{j,i_2} = \cdots = c_{j,i_\ell}, 1 \leq j \leq n$. It can then be verified from the definition of an $r$-wise equidistant weak flip code that this completes the construction of an $r$-(M, $\ell$, $\lambda_r$)

\[\text{Note that the two properties of a code being equidistant and a code achieving the Plotkin bound do not imply each other. There exist Plotkin-bound achieving codes that are not equidistant, and there also exist equidistant codes that do not achieve the Plotkin bound.}\]
design, where \( \lambda_r \) is by definition equal to \( n - d_{\mathcal{I}}^{(M,n)} \), where \( \mathcal{I} \) can be any size-\( r \) subset of \( M \).

Using a fundamental theorem in combinatorial design theory [26, Thm. 9.4], we next infer that an \( r-(M, \ell, \lambda_r) \) design is also an \( s-(M, \ell, \lambda_s) \) design with \( 2 \leq s < r \) and

\[
\lambda_s = \lambda_r \frac{\binom{M-s}{r-s}}{\binom{r-\ell}{r-s}}. \tag{100}
\]

Since an \( s-(M, \ell, \lambda_s) \) design corresponds to an \( s \)-wise equidistant weak flip code, this proves the first statement.

If we additionally assume that the parameter \( \lambda_r \) is equal to the maximum \( r \)-wise Hamming match \( a_{\text{max};r} \) satisfying (99), we then obtain for \( M = 2\ell \):

\[
a_{\text{max};s} = a_{\text{max};r} \frac{\binom{M-s}{r-s}}{\binom{r-\ell}{r-s}} \tag{101}
\]

\[
= n \frac{(\ell-1)}{(r-1)} \frac{(2\ell-s)}{(r-s)} \tag{102}
\]

\[
= n \frac{(\ell-1)!}{(\ell-r)! (r-1)!} \frac{(2\ell-s)!}{(r-s)! (2\ell-r)!} \tag{103}
\]

\[
= n \frac{(\ell-1)!}{(\ell-s)! (s-1)!} \frac{(2\ell-s)!}{(r-s)! (r-\ell)!} \tag{104}
\]

\[
= n \frac{(\ell-1)}{(2\ell-1)} \frac{(s-1)!}{(s-1)!} \tag{105}
\]

We thus confirm that \( \mathcal{C}_{\text{equidist}}^{(M,n)} \) also meets the smallest maximum \( s \)-wise Hamming matches (i.e., the largest minimum \( s \)-wise Hamming distances) for \( 2 \leq s < r \).

In the case of \( M = 2\ell - 1 \), the definition of weak flip codes indicates that all codewords of \( \mathcal{C}_{\text{equidist}}^{(2\ell-1,n)} \) are contained in \( \mathcal{C}_{\text{equidist}}^{(2\ell,n)} \). Hence,

\[
d_{\text{min};r} (\mathcal{C}_{\text{equidist}}^{(2\ell-1,n)}) \geq d_{\text{min};r} (\mathcal{C}_{\text{equidist}}^{(2\ell,n)}) = n - n \frac{(\ell-1)}{(2\ell-1)}, \tag{106}
\]

which again achieves the Plotkin upper bound for \( r \)-wise Hamming distances in Theorem 50.

The following corollary follows directly from Theorem 52 and Corollary 47.

**Corollary 53.** The fair weak flip code \( \mathcal{C}_{\text{fair}}^{(M,n)} \) achieves the largest minimum \( r \)-wise Hamming distance for all \( 2 \leq r \leq \ell \) among all \( (M,n) \) codes.

**Proof:** The proof is completed by observing that the smallest maximum \( \ell \)-wise Hamming matches of (91) is equal to

\[
n \frac{(\ell-1)}{(2\ell-1)} = n \frac{1}{\ell}. \tag{107}
\]
which, according to Corollary 47 with \( r \) there replaced by \( \bar{\ell} \), is achieved by \( \mathcal{C}_{\text{fair}}^{(M,n)} \).

We make the following remark to Corollary 53: The fair linear code always meets the Plotkin bound for the 2-wise Hamming distance; however, in contrast to the fair weak flip code \( \mathcal{C}_{\text{fair}}^{(M,n)} \), it does not necessarily meet the Plotkin bound for \( r \)-wise Hamming distances for \( r > 2 \). This gives rise to our suspicion that a fair linear code may perform strictly worse than the optimal fair weak flip code even if it is the best linear code. Proper evidence for this claim will be given in Section 6.8.

6 Performance Analysis of the BEC

In Section 3.2 we have shown that any codebook can be described by the type vector \( \mathbf{t} \). Therefore the minimization of the average error probability among all possible codebooks turns into an optimization problem on the discrete vector \( \mathbf{t} \), subject to the condition that \( \sum_{j=1}^{J} t_j = n \). Consequently, the \( r \)-wise Hamming distance and the properties of the type vector play an important role in our analysis.

6.1 Exact Average Error Probability of a Code with an Arbitrary Number of Codewords \( M \)

We firstly derive a useful result that gives the exact average error probability as a function of the type vector \( \mathbf{t} \).

Lemma 54 (Inclusion–Exclusion Principle in Probability Theory [27]). Let \( A_1, A_2, \ldots, A_M \) be \( M \) (not necessarily independent) events in a probability space. The inclusion–exclusion principle states that

\[
\Pr\left( \bigcup_{m=1}^{M} A_m \right) = \sum_{r=1}^{M} (-1)^{r-1} \sum_{\mathcal{I} \subseteq \{1,2,\ldots,M\} : |\mathcal{I}| = r} \Pr\left( \bigcap_{i \in \mathcal{I}} A_i \right). \tag{108}
\]

We will next apply the idea of the inclusion–exclusion principle to the closed decoding regions given in Definition 3. To simplify our notation, we define the following shorthands:

\[
\Pr\left( \mathcal{D}_m^{(M,n)} \mid x_m \right) \triangleq \sum_{y \in \mathcal{D}_m^{(M,n)}} P_{Y \mid X}(y \mid x_m), \tag{109}
\]

\[
\Pr\left( \bigcap_{i \in \mathcal{I}} \mathcal{D}_i^{(M,n)} \mid x_{\ell} \right) \triangleq \sum_{y \in \bigcap_{i \in \mathcal{I}} \mathcal{D}_i^{(M,n)}} P_{Y \mid X}(y \mid x_{\ell, \ell \in \mathcal{I}}), \quad \mathcal{I} \subseteq \mathcal{M}, \tag{110}
\]

where for every \( y \) in \( \bigcap_{i \in \mathcal{I} = \{i_1,i_2,\ldots,i_r\}} \mathcal{D}_i^{(M,n)} \), we note according to Definition 3 that

\[
\max_{1 \leq m' \leq M} P_{Y \mid X}(y \mid x_{m'}) = P_{Y \mid X}(y \mid x_{i_1}) = P_{Y \mid X}(y \mid x_{i_2}) = \cdots = P_{Y \mid X}(y \mid x_{i_r}), \tag{111}
\]

and hence the exact choice of \( \ell \) is irrelevant in (110).

Theorem 55. Consider an \((M,n)\) coding scheme with its corresponding closed ML decoding regions \( \mathcal{D}_m \) as given in Definition 3, where we drop the superscript “\((M,n)\)” for notational convenience. Defining

\[
\mathcal{D}_m \triangleq \mathcal{D}_m \setminus \left( \bigcap_{i \in \{1,\ldots,m-1\}} \mathcal{D}_i \right) \tag{112}
\]
we have
\[
\Pr(D_m \mid x_m) = \Pr(\overline{D}_m \mid x_m) - \sum_{r=1}^{m-1} (-1)^{r-1} \sum_{I \subseteq \{1, \ldots, m-1\} : |I| = r} \Pr\left( \bigcap_{i \in I} (\overline{D}_i \cap \overline{D}_m) \ \bigg| \ x_m \right) \quad (113)
\]
and the exact average success probability can be expressed as
\[
P_c(\mathcal{C}^{(M,n)}) = \frac{1}{M} \sum_{m=1}^{M} (-1)^{r-1} \sum_{I \subseteq \{1, \ldots, M\} : |I| = r} \Pr\left( \bigcap_{i \in I} D_i \ \bigg| \ x_m \right). \quad (114)
\]

**Proof:** By Definition 3, a possible choice of ML decoding regions is given as follows:
\[
\begin{align*}
D_1 & \triangleq \overline{D}_1, \quad (115) \\
D_2 & \triangleq \overline{D}_2 \setminus \overline{D}_1 \quad (116) \\
D_3 & \triangleq \overline{D}_3 \setminus (\overline{D}_1 \cup \overline{D}_2) \quad (117) \\
& \vdots
\end{align*}
\]
i.e., we obtain (112). We rewrite
\[
\overline{D}_m \cap \left( \bigcup_{i \in \{1, \ldots, m-1\}} \overline{D}_i \right) = \bigcup_{i \in \{1, \ldots, m-1\}} (\overline{D}_m \cap \overline{D}_i) \quad (120)
\]
and use Lemma 54 to obtain
\[
\begin{align*}
\Pr(D_m \mid x_m) &= \Pr(\overline{D}_m \mid x_m) - \Pr\left( \bigcup_{i \in \{1, \ldots, m-1\}} (\overline{D}_m \cap \overline{D}_i) \ \bigg| \ x_m \right) \\
&= \Pr(\overline{D}_m \mid x_m) - \Pr\left( \bigcup_{i \in \{1, \ldots, m-1\}} (\overline{D}_m \cap \overline{D}_i) \ \bigg| \ x_m \right) \\
&= \Pr(\overline{D}_m \mid x_m) - \sum_{r=1}^{m-1} (-1)^{r-1} \sum_{I \subseteq \{1, \ldots, m-1\} : |I| = r} \Pr\left( \bigcap_{i \in I} (\overline{D}_m \cap \overline{D}_i) \ \bigg| \ x_m \right), \quad (123)
\end{align*}
\]
which proves (113).

The average success probability can now be expressed as follows:
\[
P_c(\mathcal{C}^{(M,n)}) = \frac{1}{M} \sum_{m=1}^{M} \Pr(D_m \mid x_m) \quad (124)
\]
\[
= \frac{1}{M} \sum_{m=1}^{M} \left( \Pr(\overline{D}_m \mid x_m) - \sum_{r=1}^{m-1} (-1)^{r-1} \sum_{I \subseteq \{1, \ldots, m-1\} : |I| = r} \Pr\left( \bigcap_{i \in I} (\overline{D}_m \cap \overline{D}_i) \ \bigg| \ x_m \right) \right) \quad (125)
\]
\[
\sum_{m=1}^{M} \frac{1}{M} \left( \Pr(D_m | x_m) + \sum_{r=1}^{m-1} (-1)^r \sum_{I \subseteq \{1, \ldots, m-1\} : |I| = r} \Pr\left( \bigcap_{i \in I} (D_m \cap D_i) \mid x_{\ell, \ell \in I \cup \{m\}} \right) \right) = 1
\]

(126)

Here, (125) follows from (123); in (126) we allow different choices of the conditioning argument, which does not change the expression because of (111); in (127) we include the empty set into the sum to take care of the first term; and in (128) and (129) we exchange the two outer sums and then combine the resulting two inner sums. This completes the proof. \(\square\)

By the \(r\)-wise Hamming distance and Theorem 55, we are now able to give a closed-form expression for the exact average error probability of an arbitrary code \(C_{t}^{(M,n)}\) used on a BEC.

**Theorem 56 (Average Error Probability on the BEC).** Consider a BEC with arbitrary erasure probability \(0 \leq \delta < 1\) and an arbitrary code \(C_{t}^{(M,n)}\) with \(M \geq 2\). The average ML error probability can be expressed using the type vector \(t\) as follows:

\[
P_e(C_{t}^{(M,n)}) = \frac{1}{M} \sum_{r=2}^{M} (-1)^r \sum_{I \subseteq \{1, \ldots, M\} : |I| = r} \delta^{d_{I}^{(M,n)}},
\]

(131)

where \(d_{I}^{(M,n)}\) denotes the \(r\)-wise Hamming distance as given in Definition 38.

**Proof:** Comparing (114) and (131), we see that the theorem can be proved by showing that

\[
\Pr(D_m | x_m) = 1, \quad \forall m \in M,
\]

(132)

\[
\Pr\left( \bigcap_{i \in I} D_i \mid x_{\ell, \ell \in I} \right) = \delta^{d_{I}^{(M,n)}}, \quad \forall I \subseteq M \text{ with } |I| \geq 2.
\]

(133)
By definition and because the channel is a BEC,
\[ \overline{D}_m = \{ y : d_H(x_m, x_m(0|y), y(0|y)) = d_H(x_m, x_m(1|y), y(1|y)) = 0 \} \]
\[ = \bigcup_{N=0}^{n} \bigcup_{N \subseteq N_n : |N| = N} \{ y : d_H(2_N, y_N) = d_H(x_m, y_N \setminus \mathcal{N}) = 0 \} \]  
(134)

where we abbreviate \( N_n \equiv \{ 1, \ldots, n \} \) and 2 denotes the all-2 vector. Therefore, the conditional success probability of the closed decoding region \( \overline{D}_m \) is
\[ \Pr(\overline{D}_m | x_m) = \sum_{y \in \overline{D}_m} P_{Y|X}(y | x_m) = \sum_{N=0}^{n} \left( \begin{array}{c} n \\ N \end{array} \right) \delta^N (1 - \delta)^{n-N} = 1. \]  
(136)

Similarly,
\[ \bigcap_{i \in \mathcal{I}} \overline{D}_i = \{ y : d_H(x_i, x_i(0|y), y(0|y)) = d_H(x_i, x_i(1|y), y(1|y)) = 0 \ \forall i \in \mathcal{I} \} \]
\[ = \bigcup_{N=d^{(M,n)}_n}^{n} \bigcup_{N \subseteq N_n \setminus 2_N : |2_N| = N} \{ y : d_H(2_N, y_N) = d_H(x_i, N_n \setminus 2_N , y_N \setminus 2_N) = 0 \}, \]
(137)

where for convenience, we set \( \mathcal{I} = \{ i_1, \ldots, i_r \} \) and \( N_\mathcal{I} \equiv \{ j \in N_n : x_{i_1,j} = x_{i_2,j} = \cdots = x_{i_r,j} \} \). This implies
\[ \Pr(\bigcap_{i \in \mathcal{I}} \overline{D}_i | x_{i_1, i_2}) = \sum_{y \in \bigcap_{i \in \mathcal{I}} \overline{D}_i} P_{Y|X}(y | x_{i_1}) \]
\[ = \sum_{N=d^{(M,n)}_n}^{n} \left( \begin{array}{c} n - d^{(M,n)}_I \\ N - d^{(M,n)}_I \end{array} \right) \delta^N (1 - \delta)^{n-N} \]
\[ = \delta^{d^{(M,n)}_I}. \]  
(140)

### 6.2 Optimal Codes with Two Codewords (M = 2)

**Theorem 57.** For a BEC and for any \( n \geq 1 \), an optimal codebook with two codewords \( M = 2 \) is the (unique) weak flip code with type \( t_1 = n \). It has an error probability
\[ P_e(\mathcal{C}_n^{(2,n)}) = \frac{1}{2} \delta^n. \]  
(142)

**Proof:** By Lemma 9 and since we have only one column in the candidate columns set, any codebook consisting of an arbitrary codeword and its flipped version is equivalent to \( \mathcal{C}_n^{(2,n)} \) and hence is optimal. \( \square \)

### 6.3 Optimal Codes with Three or Four Codewords (M = 3, 4)

Even though we know the exact average error probability for a code with an arbitrary number of codewords \( M \) on a BEC, the optimal code structure is not obvious. We are now trying to shed more light on this problem.

We start with the following lemma.
Lemma 58 ([17, Lem. 32]). Fix the number of codewords \( M \) and a DMC. The success probability \( P_c(\mathcal{C}(M,n)) \) for a sequence of codes \( \{\mathcal{C}(M,n)\}_{n \geq 1} \), where each code is generated by appending a column to the code of smaller blocklength, is nondecreasing with respect to the blocklength \( n \).

Proof: See [17, Sec. VIII-B]. \( \square \)

Lemma 58 suggests a recursive code construction that guarantees the largest total success probability increase, i.e., we can find some locally optimal code type.

Theorem 59. For a BEC with arbitrary erasure probability \( 0 \leq \delta < 1 \), an optimal code with three codewords \( M = 3 \) or four codewords \( M = 4 \) and with a blocklength \( n \) is

\[
\mathcal{C}_{BEC}^{(M,2)} = \begin{cases} 
\begin{bmatrix} \mathbf{c}_1^{(M)} \\ \mathbf{c}_2^{(M)} \end{bmatrix} & \text{if } M = 3, \\
\begin{bmatrix} \mathbf{c}_3^{(M)} \\ \mathbf{c}_5^{(M)} \end{bmatrix} & \text{if } M = 4.
\end{cases}
\]

(143)

If we recursively construct a locally optimal codebook with three codewords \( M = 3 \) or four codewords \( M = 4 \) and with a blocklength \( n \geq 3 \) by appending a new column to \( \mathcal{C}_{BEC}^{(M,n-1)} \), where we append a “⋄” to \( (M,n) \) to denote a locally optimal recursive-constructed code of size \( M \) and length \( (n-1) \), the increase in average success probability is maximized by the following choice of appended columns:

\[
\begin{aligned}
&\begin{cases} 
\mathbf{c}_3^{(M)} & \text{if } n \mod 3 = 0, \\
\mathbf{c}_1^{(M)} & \text{if } n \mod 3 = 1, \\
\mathbf{c}_2^{(M)} & \text{if } n \mod 3 = 2,
\end{cases} \quad \text{when } M = 3, \\
&\begin{cases} 
\mathbf{c}_6^{(M)} & \text{if } n \mod 3 = 0, \\
\mathbf{c}_3^{(M)} & \text{if } n \mod 3 = 1, \\
\mathbf{c}_5^{(M)} & \text{if } n \mod 3 = 2,
\end{cases} \quad \text{when } M = 4.
\end{aligned}
\]

(144) (145)

Proof: See Appendix A. \( \square \)

This theorem suggests that for a given fixed code size \( M \), a sequence of good codes can be generated by appending the correct columns to a code of smaller blocklength. For a given DMC and code of blocklength \( n \), we ask the question what is the optimal improvement (i.e., the maximum reduction of error probability) when increasing the blocklength from \( n \) to \( n + 1 \) when \( M = 3 \) or 4. (Note that in general one might achieve better results if we design a sequence of codes that increases from blocklength \( n \) to \( n + \gamma \) with a step-size \( \gamma > 1 \); however, as we will see below, for \( M = 3 \) or \( M = 4 \), \( \gamma = 1 \) turns out to be optimal.) The answer to this question then leads to the recursive construction of (144) and (145).

While Theorem 59 only guarantees local optimality for the given recursive construction, further investigation shows that the given construction is actually globally optimum.

Theorem 60. For a BEC and for any \( n \geq 2 \), an optimal codebook with \( M = 3 \) or \( M = 4 \) codewords is the weak flip code of type \( t_{\text{weak}}^* \) where for \( M = 3 \)

\[
\begin{align*}
t_1^* &= \left\lfloor \frac{n+2}{3} \right\rfloor, \\
t_2^* &= \left\lfloor \frac{n+1}{3} \right\rfloor, \\
t_3^* &= \left\lfloor \frac{n}{3} \right\rfloor.
\end{align*}
\]

(146)

\( ^{17} \)See [17, Def. 33].
and for \( M = 4 \)

\[
t^*_3 = \left\lfloor \frac{n + 2}{3} \right\rfloor, \quad t^*_5 = \left\lfloor \frac{n + 1}{3} \right\rfloor, \quad t^*_6 = \left\lfloor \frac{n}{3} \right\rfloor.
\]

(147)

Note that the recursively constructed code of Theorem 59 is equivalent to the optimal code given here:

\[
C^{(M,n)}_{\text{BEC}} \equiv C^{(M,n)}_{t^*_{\text{weak}}}.
\]

(148)

**Proof:** See Appendix B.

Using the shorthand

\[
k \triangleq \left\lfloor \frac{n}{3} \right\rfloor,
\]

(149)

the code parameters of these optimal codes can be summarized as

\[
t^*_{\text{weak}} = \begin{cases} 
[t^*_1, t^*_2, t^*_3] & \text{for } M = 3, \\
[t^*_5, t^*_6] & \text{for } M = 4 \\
[k, k, k] & \text{if } n \mod 3 = 0, \\
[k + 1, k, k] & \text{if } n \mod 3 = 1, \\
[k + 1, k + 1, k] & \text{if } n \mod 3 = 2.
\end{cases}
\]

(150)

(151)

From (144) and (145), or from (146) and (147), or from (150), we confirm again that \( C^{(M,n)}_{t^*_{\text{weak}}} \) can be obtained by simply removing the last codeword of \( C^{(4,n)}_{t^*_{\text{weak}}} \) (compare with Remark 14).

The corresponding optimal average error probabilities are given as

\[
P_e(C^{(M,n)}_{t^*_{\text{weak}}}) = \begin{cases} 
\frac{1}{3} (\delta^n - t^*_1 + \delta^n - t^*_2 + \delta^n - t^*_3 - \delta^n) & \text{if } M = 3, \\
\frac{1}{4} (2\delta^n - t^*_5 + 2\delta^n - t^*_6 + 2\delta^n - t^*_6 - 3\delta^n) & \text{if } M = 4.
\end{cases}
\]

(152)

6.4 A Brief Comparison between BSC and BEC

In [17], it has been shown that the optimal codes for \( M = 3 \) or \( M = 4 \) for the BSC are weak flip codes with type

\[
t^*_{\text{weak}} = \begin{cases} 
[k + 1, k, k - 1] & \text{if } n \mod 3 = 0, \\
[k + 1, k, k] & \text{if } n \mod 3 = 1, \\
[k + 1, k + 1, k] & \text{if } n \mod 3 = 2.
\end{cases}
\]

(153)

which by (151) immediately gives the following corollary.

**Corollary 61.** For \( M = 3 \) or \( M = 4 \) and for \( n \mod 3 \neq 0 \), the weak flip codes with type \( t^*_{\text{weak}} \) defined in (153) (equivalently, (151)) are optimal for both BSC and BEC.

The corresponding pairwise Hamming distance vectors of the BSC optimal codes (Corollary 49) for \( M = 3 \) and \( M = 4 \) are respectively\(^{18}\)

\[
d^{(3,n)*} = \begin{cases} 
(2k - 1, 2k, 2k + 1) & \text{if } n \mod 3 = 0, \\
(2k, 2k + 1, 2k + 1) & \text{if } n \mod 3 = 1, \\
(2k + 1, 2k + 1, 2k + 2) & \text{if } n \mod 3 = 2,
\end{cases}
\]

(154)

\(^{18}\)For weak flip codes with \( M = 3 \) or \( M = 4 \) codewords, we only need to compare the pairwise Hamming distances because the 3-wise and 4-wise Hamming distances are all equal to \( n \) and hence are identical.
Comparing these to the corresponding pairwise Hamming distance vectors of the BEC optimal codes (Theorem 60),

\[ d^{(3, n)\ast} = \begin{cases} 
(2k, 2k) & \text{if } n \mod 3 = 0, \\
(2k, 2k + 1) & \text{if } n \mod 3 = 1, \\
(2k + 1, 2k + 1) & \text{if } n \mod 3 = 2
\end{cases} \tag{156} \]

and

\[ d^{(4, n)\ast} = \begin{cases} 
(2k, 2k, 2k) & \text{if } n \mod 3 = 0, \\
(2k, 2k + 1, 2k + 1) & \text{if } n \mod 3 = 1, \\
(2k + 1, 2k + 1, 2k + 2) & \text{if } n \mod 3 = 2
\end{cases} \tag{157} \]

we note that when \( n \mod 3 = 0 \), the optimal codes for the BEC are fair and therefore maximize the minimum Hamming distance, while this is not the case for the very symmetric BSC (i.e., on the BSC, an optimal code of length \( n \mod 3 = 0 \) does not maximize the minimum Hamming distance among all code designs of the same size and length!). In fact, for \( M = 3 \) or 4 and for every \( n \), a code maximizes the minimum Hamming distance if, and only if, it is an optimal code for the BEC. However, when \( M > 4 \), numerical evidence can be created to disprove the statement that a code maximizing the minimum Hamming distance is an optimal code for the BEC! As we will see in the case of \( M = 8 \), the pairwise Hamming distance vector (2-wise Hamming distance) is not sufficient for determining global optimality, but the \( r \)-wise Hamming distances with \( r > 2 \) have to be taken into account.

### 6.5 Application to Known Bounds on the Error Probability for a Finite Blocklength (\( M = 3, 4 \))

Since we now know the optimal code structure, we can compare its performance to the known bounds in Section 4.

Note that the optimal error exponents for \( M = 3, 4 \) codewords on the BEC are

\[ E_3 = E_4 = -\frac{2}{3} \log \delta. \tag{158} \]

Moreover, for \( M = 3, 4 \),

\[ D_{\text{min}}^{(\text{BEC})} (\delta^{(M, n)}_{\text{weak}}) = \begin{cases} 
-\frac{2}{3} \log \delta & \text{if } n \mod 3 = 0, \\
-\left[ \frac{j + 1}{n} \right] \log \delta & \text{if } n \mod 3 = 1, \\
-\left[ \frac{j + 1}{n} \right] \log \delta & \text{if } n \mod 3 = 2
\end{cases} \tag{159} \]

Figures 2 and 3 compare the exact optimal performance for \( M = 3 \) and \( M = 4 \), respectively, with the following bounds: the SGB upper and lower bounds based on the optimal code as used by Shannon \textit{et al.} for a blocklength \( n \mod 3 = 0 \) (thereby
Figure 2: Exact value of, and bounds on, the performance of an optimal code with $M = 3$ codewords on the BEC with $\delta = 0.3$ as a function of the blocklength $n$.

confirming that this lower bound is valid generally), the Gallager upper bound, and also the PPV upper and lower bounds.

We can see that the SGB upper bound is closer to the exact optimal performance (and hence tighter) than the PPV upper bound and the Gallager upper bound. Note that the PPV upper bound is not exactly the same as the Gallager upper bound, even though for $M = 3$ their curves look almost identical. Also note that the SGB upper bound does exhibit the correct error exponent. It is shown in [22] that when $n$ goes to infinity under fixed $M$, the PPV upper bound only tends to the suboptimal Gallager exponent [18]; this fact is also confirmed by the two figures.

Regarding the lower bounds we see that the PPV lower bound is much better for finite $n$ than the SGB lower bound. However, the exponential growth rate of the PPV lower bound only approaches that of the sphere-packing bound [23], and does not equal the optimal exponent $E_M$ either [19].

Once more we would like to emphasize that even though for $M = 3, 4$, the fair weak flip codes are optimal for the BEC and achieve the optimal error exponent for both the BEC and the BSC, they are strictly suboptimal for every $n \mod 3 = 0$ for the BSC.

6.6 Optimal Codes with Five or Six Codewords ($M = 5, 6$)

The idea of recursively designing a locally optimal code turned out to be a powerful approach to obtain globally optimal codes for $M = 3, 4$. Unfortunately, for larger values of $M$, we might need a recursion from $n$ to $n + \gamma$ with a step-size $\gamma > 1$, and—according to our numerical examination— this step-size $\gamma$ might be a function of the blocklength.
Figure 3: Exact value of, and bounds on, the performance of an optimal code with $M = 4$ codewords on the BEC with $\delta = 0.3$ as a function of the blocklength $n$. 
Since the exact average error probability expression becomes involved as $M$ grows, we only succeeded in investigating a locally optimal code construction subject to the recursive design approach when the blocklength $n$ is a multiple of $L$. Based on our definition of fair weak flip codes and on Conjecture 62 below, we conjecture\(^{19}\) that the necessary step-size for global optimality satisfies $\gamma \leq L$.

**Conjecture 62.** For a BEC and for any $n$ being a multiple of $L = 10$, an optimal codebook with $M = 5$ or $M = 6$ codewords is the corresponding fair weak flip code.

Note that the restriction on $n$ stems from the fact that fair weak flip codes are only defined for blocklengths satisfying $n \mod L = 0$ (the code uses each weak flip column $\tau$ times, where $\tau = n/L$ is an integer). We can show that if we relax the error minimization problem by allowing noninteger values for the type $t$, the optimal type will be equally distributed among all possible weak flip columns also when $n \mod L \neq 0$. Unfortunately, a block code always must use an integer number of candidate columns, and the globally optimal choice of an integer in the neighborhood of the optimal noninteger value is rather involved. Based on this observation and on our extensive numerical examinations, we give the following conjecture.

**Conjecture 63.** Consider the BEC and a blocklength $n \geq 3$ that is not a multiple of $L = 10$ (as the case of $n \mod 10 = 0$ has been taken care in Conjecture 62), and define the shorthand

$$\tau \triangleq \left\lfloor \frac{n}{10} \right\rfloor.$$  (160)

An optimal code that minimizes the average error probability among all code designs with $M = 5$ codewords is a weak flip code of type

$$t_{\text{weak}} = [t_3, t_5, t_6, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}]$$

\[
= \begin{cases}
\{\tau + 1, \tau, \tau, \tau, \tau, \tau, \tau, \tau, \tau, \tau\} & \text{if } n \mod 10 = 1, \\
\{\tau + 1, \tau + 1, \tau + 1, \tau^2 - 1, \tau, \tau, \tau, \tau, \tau\} & \text{if } n \mod 10 = 2, \\
\{\tau + 1, \tau + 1, \tau, \tau + 1, \tau, \tau, \tau, \tau, \tau\} & \text{if } n \mod 10 = 3, \\
\{\tau + 1, \tau + 1, \tau, \tau + 1, \tau, \tau, \tau, \tau, \tau + 1\} & \text{if } n \mod 10 = 4, \\
\{\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau, \tau, \tau\} & \text{if } n \mod 10 = 5, \\
\{\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau, \tau, \tau\} & \text{if } n \mod 10 = 6, \\
\{\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1\} & \text{if } n \mod 10 = 7, \\
\{\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1\} & \text{if } n \mod 10 = 8, \\
\{\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1\} & \text{if } n \mod 10 = 9.
\end{cases}
\]  (161)

Except for $n \mod 10 = 7$, an optimal code that minimizes the average error probability

\(^{19}\)Note that in the following conjectures, despite of Conjecture 63, we actually can prove local optimality of the proposed type vector by verifying the Karush–Kuhn–Tucker (KKT) conditions. However, since the discrete multivariate average error probability function is not convex, we did not succeed in confirming global optimality.
among all code designs with $M = 6$ codewords is a weak flip code of type

$$t_{\text{weak}} = [t_7, t_{11}, t_{13}, t_{14}, t_{21}, t_{22}, t_{25}, t_{26}, t_{28}]$$

$$= \begin{cases} 
\tau + 1, \tau, \tau, \tau, \tau, \tau, \tau, \tau, \tau, \tau & \text{if } n \text{ mod } 10 = 1, \\
\tau + 1, \tau + 1, \tau, \tau, \tau, \tau, \tau, \tau, \tau, \tau & \text{if } n \text{ mod } 10 = 2, \\
\tau + 1, \tau + 1, \tau, \tau, \tau + 1, \tau, \tau, \tau & \text{if } n \text{ mod } 10 = 3, \\
\tau + 1, \tau + 1, \tau, \tau, \tau + 1, \tau + 1, \tau, \tau, \tau & \text{if } n \text{ mod } 10 = 4, \\
\tau + 1, \tau + 1, \tau, \tau + 1, \tau + 1, \tau + 1, \tau, \tau, \tau & \text{if } n \text{ mod } 10 = 5, \\
\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau & \text{if } n \text{ mod } 10 = 6, \\
\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau & \text{if } n \text{ mod } 10 = 8, \\
\tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau + 1, \tau & \text{if } n \text{ mod } 10 = 9. 
\end{cases} \quad (162)$$

For $n \text{ mod } 10 = 7$ and $M = 6$, an optimal code that minimizes the average error probability among all code designs is actually not a weak flip code but a nonweak flip code of type $t$ satisfying

$$\begin{cases} 
t_{14} = t_{22} = t_{26} = t_{28} = \tau, \\
t_7 = t_{11} = t_{13} = t_{19} = t_{21} = t_{25} = \tau + 1, \\
t_{30} = 1, \\
t_j = 0 \text{ for the remaining indices}. \end{cases} \quad (163)$$

Note that $t_{30}$ is the only nonweak flip column in this code.

Surprisingly, the optimal code for $n \text{ mod } 10 = 7$ and $M = 6$ is not a weak flip code. We point out again that the exact average error probability expression for the BEC with $M = 6$ is a function of the discrete multivariate nonnegative integers $t_1, t_2, \ldots, t_{31}$ under the constraint that their sum equals $n$. If we allow noninteger solutions, the minimizers are $t_j = n/L$ for all $t_j$ belonging to weak flip columns. Yet, (163) shows that the nearest integer minimizer might be a only “nearly weak flip” code instead of a weak flip code.

Note that according to Conjecture 63, it is possible to recursively construct optimal codes with $M = 5, 6$ codewords using a step size $\gamma < 10$.

For our quest of understanding the optimal code design for larger $M$, we believe that it will be useful to substantiate these observations further.

### 6.7 Codes with Large $r$-Wise Hamming Distances for Arbitrary $M$

We have already pointed out that a code having a large (or even maximum) pairwise Hamming distance is not necessarily an optimal code. It is crucial to look at all $r$-wise Hamming distances for $2 \leq r \leq \bar{\ell}$.

In the following theorem we will confirm this intuition once again.

**Theorem 64.** Let the number of codewords be $M = 2\bar{\ell}$ or $2\bar{\ell} - 1$ where $\bar{\ell}$ is an arbitrary positive even integer, and let the blocklength $n$ be such that $n \text{ mod } L = 0$. Then
an \((\ell - 1)\)-wise equidistant weak flip code that achieves the largest minimum \((\ell - 1)\)-wise Hamming distance\(^{20}\) but not the largest minimum \(\ell\)-wise Hamming distance has a strictly worse performance on the BEC than the fair weak flip code.

Proof: We will prove the theorem only for the case of \(M = 2\ell - 1\), the case of \(M = 2\ell\) will be similar. So, let \(M = 2\ell - 1\) with \(\ell\) even and let the blocklength be \(n = L\tau\) for some \(\tau \in \mathbb{N}\). Let \(C_{\text{weak}}^{(M,n)}\) be an \((\ell - 1)\)-wise equidistant weak flip code that achieves the largest minimum \((\ell - 1)\)-wise Hamming distance, but does not achieve the largest minimum \(\ell\)-wise Hamming distance, and let \(C_{\text{fair}}^{(M,n)}\) be the fair weak flip code that according to Corollary 53 is \(\ell\)-wise equidistant and achieves the largest minimum \(\ell\)-wise Hamming distance. Therefore, according to Theorem 52 and Theorem 56, we have

\[
P_e(C_{\text{weak}}^{(M,n)}) - P_e(C_{\text{fair}}^{(M,n)})
\]

\[
= \frac{1}{M} \sum_{r=2}^{M} (-1)^r \sum_{|I| = r} \delta d_x(C_{\text{weak}}^{(M,n)}) - \frac{1}{M} \sum_{r=2}^{M} (-1)^r \sum_{|I| = r} \delta d_x(C_{\text{fair}}^{(M,n)})
\]

\[= \frac{(-1)^\ell}{M} \sum_{|I| = \ell} \delta d_x(C_{\text{weak}}^{(M,n)}) - \frac{(-1)^\ell}{M} \sum_{|I| = \ell} \delta d_x(C_{\text{fair}}^{(M,n)})
\]

\[= \frac{1}{M} \sum_{w=1}^{L} \delta^{n-t_{jw}} - \frac{1}{M} \cdot L \cdot \delta^{-\frac{n}{\ell}}
\]

\[= \frac{L}{M} \delta^n \left[ \frac{1}{\ell} \sum_{w=1}^{L} \delta^{-t_{jw}} - \delta^{-\tau} \right]
\]

\[> \frac{L}{M} \delta^n \left[ \left( \frac{1}{\ell} \sum_{w=1}^{L} t_{jw} \right)^{\frac{1}{\ell}} - \delta^{-\tau} \right]
\]

\[= \frac{L}{M} \delta^n \left[ \delta^{-\frac{1}{\ell}} \sum_{w=1}^{L} t_{jw} - \delta^{-\tau} \right]
\]

\[= \frac{L}{M} \delta^n \left[ \delta^{-\frac{1}{\ell}} - \delta^{-\tau} \right]
\]

\[= 0.
\]

Here, the second equality follows because the distance structure of the two codes only differ in the case of \(r = \ell\) (and \(\ell\) must be even in order to make the difference positive); in the subsequent equality we use the type vector to express the \(\ell\)-wise Hamming distances of both codes and also use the fact that \(\ell\) is even; the inequality holds because the arithmetic mean (AM) is strictly larger than the geometric mean (GM); and finally we note that \(\sum_{w=1}^{L} t_{jw} = n\). Note that since we assume that \(C_{\text{weak}}^{(M,n)}\) does not achieve the \(\ell\)-wise Plotkin bound, it follows that there must exist some \(t_{jw} \neq \tau\) and therefore the inequality is strict. \(\Box\)

\(^{20}\)By Theorem 52 such a weak flip code also is \(s\)-wise equidistant and maximizes the \(s\)-wise Hamming distances for all \(2 \leq s \leq \ell - 1\).
6.8 Linear vs. Nonlinear Codes with Eight Codewords \((M = 8)\)

In this work, we are not really interested in linear codes as our focus lies on optimality in the sense of smallest average error probability. To emphasize this point, we will next compare linear with nonlinear codes for the case of \(M = 8\). We will see that best linear codes are often strictly suboptimal.

Unfortunately, even when restricting oneself to linear candidate columns, it is difficult to find the best choice of the type vector \(t_{\text{lin}} = (t_1, \ldots, t_K)^T\) subject to the condition that \(\sum_{j=1}^{K} t_j = n\).

**Conjecture 65.** For a BEC and for any blocklength \(n\) being a multiple of \(K = 7\), a best linear code with dimension \(k = 3\) is the fair linear code.

Next, we show that the fair linear code is strictly suboptimal on the BEC.

**Example 66.** Consider the fair linear code and the (nonlinear) fair weak flip code for \(M = 2^3\) and \(n = 35\). From Theorem 56 we obtain

\[
\begin{align*}
P_e(C_{\text{lin, fair}}^{(8,35)}) &= \frac{1}{8} \left( \binom{8}{2} \delta^{n-15} - \binom{8}{3} \delta^{n-5} + 14\delta^{n-5} + \left( \binom{8}{4} - 14 \right) \delta^{n} 
- \binom{8}{5} \delta^n + \binom{8}{6} \delta^n - \binom{8}{7} \delta^n + \binom{8}{8} \delta^n \right),
\end{align*}
\]

(172)

and from Corollary 47 and also Theorem 56, we get

\[
\begin{align*}
P_e(C_{\text{fair}}^{(8,35)}) &= \frac{1}{8} \left( \binom{8}{2} \delta^{n-15} - \binom{8}{3} \delta^{n-5} + \binom{8}{4} \delta^{n-1} 
- \binom{8}{5} \delta^n + \binom{8}{6} \delta^n - \binom{8}{7} \delta^n + \binom{8}{8} \delta^n \right).
\end{align*}
\]

(173)

Thus,

\[
P_e(C_{\text{lin, fair}}^{(8,35)}) - P_e(C_{\text{fair}}^{(8,35)}) = \frac{14}{8} (\delta^{n-5} + 4\delta^n - 5\delta^{n-1}),
\]

(174)

which can be seen to be strictly positive using an argument similar to the proof of Theorem 64 (AM–GM inequality). Hence, the fair linear code with dimension \(k = 3\) and blocklength \(n = 35\) is not optimal.

Actually, this example can be generalized to any blocklength being a multiple of 7 except \(n = 7\). The derivation is based on elaborately extracting \(n\) columns from the codebook matrix of a fair weak flip code with blocklength larger than \(n\) to form a new \((8, n)\) nonlinear code (that actually is a concatenation of several nonlinear Hadamard codes). The technique fails for \(n = 7\) because taking any seven columns from the code matrix of the \((8, 35)\) fair weak flip code always results in a Hadamard linear code.

**Theorem 67.** For \(n \mod 7 = 0\) apart from \(n = 7\), the fair linear code with \(M = 8\) codewords (as given in Conjecture 65) is strictly suboptimal over the BEC.

**Proof:** See Appendix C. □

It is interesting that for \(M = 8\) and for all blocklengths \(n \mod L = 0\), the fair linear code and the fair weak flip code both are 2-wise and 3-wise equidistant and both achieve...
the 2-wise and the 3-wise Plotkin bounds. However, only the fair weak flip code is also 4-wise equidistant and achieves the 4-wise Plotkin bound. This is in agreement with Theorem 64 and explains why the fair linear code is outperformed on the BEC. Based on these insights, we actually believe that the fair weak flip code is globally optimal for \( M = 8 \).

For blocklengths \( n \mod L \neq 0 \), however, the situation is unclear because the optimal discrete solution to the “fair noninteger” distribution among all weak flip columns might even end up with nonweak flip columns (compare with Conjecture 63).

7 Conclusion

In this paper we have broken away from traditional coding theory that focuses on finding codes with sufficient structure (like linearity) to allow efficient encoding and decoding and that analyzes such codes’ performance for large blocklengths. Instead we have put our emphasis on optimal design in the sense of minimizing the average error probability (under ML decoding) for any finite blocklength. To that goal we have proposed a column-wise approach to the codebook matrix that allows us to define families of codes with interesting properties. Also based on the column-wise analysis of codebooks, we have further proposed an extension to the pairwise Hamming distance, called \( r \)-wise Hamming distance, investigated its properties and proven that it is a key factor to determine the exact error probability of a binary code of arbitrary blocklength \( n \) on a BEC.

We have introduced the weak flip codes, a new class of codes containing both the class of binary nonlinear Hadamard codes and the class of linear codes as special cases. We have shown that weak flip codes have many desirable properties; in particular, we have succeeded in proving that besides retaining many of the good Hamming distance properties of Hadamard codes, they are actually optimal with respect to the minimum error probability over a BEC for certain numbers of codewords \( M \) and many finite blocklengths \( n \).

The family of fair weak flip codes—a subclass of the nonlinear weak flip codes—can be seen as a generalization of linear codes to arbitrary numbers of codewords \( M \). We have shown that fair weak flip codes are optimal with respect to the average error probability for the BEC for \( M \leq 4 \) and a blocklength that is a multiple of \( L \) and we have conjectured that this result continues to hold also for \( M > 5 \). Furthermore, we have also shown that the optimal code performance is really close to the upper bound of Shannon–Gallager–Berlekamp on the BEC for \( M \leq 4 \), while for the BSC this is not the case.

Note that it has been known for quite some time that binary nonlinear Hadamard codes have good Hamming distance properties \([10]\); however, their behavior with respect to error probability for finite blocklength remained uninvestigated. In particular, while fair weak flip codes have been used before (although without being named) in the derivation of results related to error probability \([19]\) and have been shown to be best-error-exponent achieving, their global (among all possible linear or nonlinear codes) optimality with respect to the error probability was not known so far.

In conclusion, this paper tries to build a bridge between coding theory, which usually is concerned with the design of codes with good Hamming distance properties (like, e.g., the binary nonlinear Hadamard codes), and information theory, which deals with error probability and with the existence of codes that have good or optimal error probability.
behavior (often in the asymptotic sense for very large blocklengths). Our results suggest that in order to have good performance in the finite blocklength regime for the BEC, one must find a code design with large minimum $r$-wise Hamming distances for all $2 \leq r \leq \ell$.

A Proof of Theorem 59

We refer to [17, Def. 33] and define

$$P_c(\mathcal{C}^{(M,n+\gamma)}) = P_c(\mathcal{C}^{(M,n)}) + \frac{1}{M} \sum_{m=1}^{M} \sum_{y^{(n+\gamma)} \text{ s.t. } y^{(n)} \in D^{(M,n)}_m \text{ but } y^{(n+\gamma)} \in D^{(M,n+\gamma)}_m \text{ for some } m' \neq m} \left( P_{Y|X}(y^{(n+\gamma)} | x_{m'}^{(n+\gamma)}) - P_{Y|X}(y^{(n+\gamma)} | x_{m}^{(n+\gamma)}) \right)$$

(175)

$$\triangleq P_c(\mathcal{C}^{(M,n)}) + \Delta \Psi(\mathcal{C}^{(M,n+\gamma)}).$$

(176)

In the proof of Theorem 59, our goal is to maximize the total probability increase $\Delta \Psi(\mathcal{C}^{(M,n+\gamma)})$ among all possible $\mathcal{C}^{(M,\gamma)}$ with $\gamma = 1$ for $M = 3, 4$. Note that the codebook $\mathcal{C}^{(M,n+\gamma)}$ is formed by concatenating $\mathcal{C}^{(M,n)}$ with $\mathcal{C}^{(M,\gamma)}$. The proof is based on induction and follows along the same lines as in the proof for the BSC shown in [17, App. C.A] with some modifications that take into account the details of the decoding rule for the BEC. Similarly to [17, App. C.A], we need a case distinction depending on $n \mod 3$. For space reason, we only outline the case from $n = 1 \equiv 3k - 1$ to $n = 3k$. Moreover, we only consider the more complicated case of $M = 4$. Similar arguments can be applied to $M = 3$.

We start with the code $\mathcal{C}_\text{weak}^{(4,n-1)}$, whose type is as follows:

$$\mathbf{t}_\text{weak} = [t^0_3, t^3_5, t^6_6] = [k, k, k - 1],$$

(177)

and need to pick a candidate columns from $\mathcal{C}^{(4)}$ to append to $\mathcal{C}_\text{weak}^{(4,n-1)}$. We require to show that appending $\mathcal{C}_0^{(4)}$ yields the largest total probability increase among all possible candidate columns in $\mathcal{C}^{(4)}$.

To that goal, we investigate how to extend the decoding regions of $\mathcal{C}_\text{weak}^{(4,n-1)}$. For each codeword, there are three possible extended decoding regions of blocklength $n$:

$$[D_{m,n-1}^{(4,1)}], \quad [D_{m,n-1}^{(4,2)}], \quad [D_{m,n-1}^{(4,3)}], \quad m = 1, \ldots, 4.$$  

(178)

Owing to the fact that for a BEC $P_{Y|X}(0|1) = P_{Y|X}(1|0) = 0$, and using $b \in \{0, 1\}$ to denote the value of the appended bit to the $m$th codeword, $x_{m,n} = b$, we see that the decoding region $D_{m,n}^{(4,1)}$ should include both $[D_{m,n-1}^{(4,1)} b]$ and $[D_{m,n-1}^{(4,2)} b]$, and that all the received vectors in $[D_{m,n-1}^{(4,3)} b]$ will be decoded to one of the other three codewords. Since

$$\psi_m(\mathcal{C}^{(4,n-1)}) = \psi_m(\mathcal{C}^{(4,n-1)}) \cdot (1 - \delta + \delta)$$

(179)

$$= \text{Pr}(D_{m,n-1}^{(4,1)} | x_{m}^{(n-1)}) \left( P_{Y|X}(b | b) + P_{Y|X}(2 | b) \right)$$

(180)

$$= \text{Pr}(D_{m,n-1}^{(4,1)} b | x_{m}^{(n-1)} b) + \text{Pr}(D_{m,n-1}^{(4,2)} 2 | x_{m}^{(n-1)} b),$$

(181)
we obtain that \([D_m^{(4,n-1)} \setminus 1] \cup [D_m^{(4,n-1)} \setminus 2]\) does not create any probability increase, i.e., the total probability increase for each codeword will be determined by how the received vectors in \([D_m^{(4,n-1)} \setminus 1]\) are moved to one of decoding regions of the other three codewords.

We now investigate each possible appended column in a case-by-case fashion.

**Appending \(c_1^{(4)}\):** We build a new length-\(n\) code \(C_t^{(4,n)}\) from the given code \(C_{weak}^{(4,n)}\) by appending \(c_1^{(4)} = (0 0 0 1)^T\). The type becomes

\[
t_1 = [1, 0, k, 0, k, k - 1, 0]. \tag{182}
\]

We now compute the total probability increase in this case. Because \(x_{4,n} = 1\) and \(x_{m,n} = 0\) for \(m = 1, 2, 3\), some\(^2\) of the vectors in the extended decoding regions \([D_{t_1:m}^{(4,n-1)} 1]\) for \(m = 1, 2, 3\) will be moved to \(D_{t_1:4}^{(4,n)}\) (and some of the received vectors in the extended decoding region \([D_{t_1:4}^{(4,n)} 0]\) will be moved to one of \(D_{t_1:m}, m = 1, 2, 3\)). The total probability increase \(\Delta \Psi (C_{t_1}^{(4,n)})\) is

\[
\Delta \Psi (C_{t_1}^{(4,n)}) = \Pr \left( |D_4^{(4,n-1)} 1| \cap \left( |D_1^{(4,n-1)} 1| \cup |D_2^{(4,n-1)} 1| \cup |D_3^{(4,n-1)} 1| \right) \bigg| x_4^{(n-1)} 1 \right)
\]

\[
= \Pr \left( |D_4^{(4,n-1)} 1| \cap \left( \bigcup_{m=1}^{3} |D_m^{(4,n-1)} 1| \right) \bigg| x_4^{(n-1)} \right) (1 - \delta) \tag{184}
\]

\[
= \Pr \left( \bigcup_{m=1}^{3} \left( |D_m^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \right) \bigg| x_4^{(n-1)} \right) (1 - \delta) \tag{185}
\]

\[
= \left( \Pr \left( |D_1^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) + \Pr \left( |D_2^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) + \Pr \left( |D_3^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) - \Pr \left( |D_1^{(4,n-1)} 1| \cap |D_2^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) - \Pr \left( |D_1^{(4,n-1)} 1| \cap |D_3^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) - \Pr \left( |D_2^{(4,n-1)} 1| \cap |D_3^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) + \Pr \left( |D_1^{(4,n-1)} 1| \cap |D_2^{(4,n-1)} 1| \cap |D_3^{(4,n-1)} 1| \cap |D_4^{(4,n-1)} 1| \bigg| x_4^{(n-1)} \right) \right) (1 - \delta) \tag{186}
\]

\[
= (\delta^{n-1} - \delta^{n-1} - \delta^{n-1} + \delta^{n-1} - \delta^{n-1} - \delta^{n-1} - \delta^{n-1} + \delta^{n-1}) (1 - \delta) \tag{187}
\]

\[
= (\delta^{2k-1} + \delta^{2k-1} + \delta^{2k} - 2\delta^{n-1}) (1 - \delta), \tag{188}
\]

where \((183)\) holds because of the definition of the closed decoding regions and because \([D_1^{(4,n-1)} 1] \cap [D_4^{(4,n-1)} 1], m = 1, 2, 3, \) are not empty; \((184)\) is because the BEC is a DMC; \((186)\) follows directly from applying the inclusion–exclusion

\(^2\)The reason why we write “some” instead of “all” is that some vectors in \([D_{t_1:4}^{(4,n)} 0]\) cannot occur and fall out of consideration.
principle; and finally, (187) follows from the same $r$-wise Hamming distances perspective as already used in the derivations of Theorem 56.

**Appending $c_2^{(4)}$:** The derivations here are similar to the first case (or, indeed, also for the cases of appending $c_4^{(4)}$ or $c_2^{(4)}$), so we omit the details and directly state the total probability increase:

\[
\Delta \Psi \left( \Phi_{t_2}^{(4,n)} \right) = \left( \delta^{n-1} - \delta^n - \delta^n - \delta^n + \delta^n - \delta^n - \delta^n + \delta^n - \delta^n - \delta^n + \delta^n - \delta^n \right) (1 - \delta) \quad (189)
\]

(proof continued on the next page.)

**Appending $c_3^{(4)}$:** If we append $c_3^{(4)} = (0 0 1 1)^T$, the new type for blocklength $n$ becomes

\[
t_3 = [0, 0, k + 1, 0, k, k - 1, 0].
\] (191)

Since $x_{1,n} = x_{2,n} = 0$ and $x_{3,n} = x_{4,n} = 1$, again using an argument like in the first case, we find that some received vectors in the extended decoding regions $D_1^{(4,n-1)}$ and $D_2^{(4,n-1)}$ will be moved to either $D_3^{(4,n)}$ or $D_4^{(4,n)}$. We obtain a total probability increase

\[
\Delta \Psi \left( \Phi_{t_3}^{(4,n)} \right) = \left( \Pr \left( \left[ D_1^{(4,n-1)} \right] \cup \left[ D_2^{(4,n-1)} \right] \right) \cap \left[ D_3^{(4,n-1)} \right] \left| x_{3}^{(n-1)} \right] \right) + \Pr \left( \left[ D_1^{(4,n-1)} \right] \cup \left[ D_2^{(4,n-1)} \right] \right) \left[ D_4^{(4,n-1)} \right] \left| x_{4}^{(n-1)} \right] \right) - \Pr \left( \left[ D_1^{(4,n-1)} \right] \cup \left[ D_2^{(4,n-1)} \right] \right) \left[ D_3^{(4,n-1)} \right] \left| x_{3}^{(n-1)} \right] \right) (192)
\]

(proof continued on the next page.)
where in (192) we use the rule of total probability; in (193) we apply the inclusion–exclusion principle; and where (194) again follows from the $r$-wise Hamming distances perspective.

**Appending $c_4^{(4)}$:** Using an argumentation similar to the case of appending $c_1^{(4)}$, we have a total probability increase

$$
\Delta \Psi\left(\psi_{t_4}^{(4,n)}\right) = \left(\delta^{n-1-t_3^2} + \delta^{n-1-t_6^2} - \delta^{n-1} - \delta^{n-1} \right) (1 - \delta) \tag{196}
$$

$$
= \left(\delta^{2k-1} + \delta^{2k} + \delta^{2k-1} - 2\delta^{n-1}\right) (1 - \delta). \tag{197}
$$

**Appending $c_5^{(4)}$:** Using an argumentation similar to the case of appending $c_3^{(4)}$, we have a total probability increase

$$
\Delta \Psi\left(\psi_{t_5}^{(4,n)}\right) = \left(\delta^{n-1-t_3^2} + \delta^{n-1-t_6^2} - \delta^{n-1} + \delta^{n-1-t_6^2} - \delta^{n-1} \right) (1 - \delta) \tag{198}
$$

$$
= \left(\delta^{2k-1} + \delta^{2k} + \delta^{2k-1} - 3\delta^{n-1}\right) (1 - \delta). \tag{199}
$$

**Appending $c_6^{(4)}$:** Using an argumentation similar to the case of appending $c_3^{(4)}$, we have a total probability increase

$$
\Delta \Psi\left(\psi_{t_6}^{(4,n)}\right) = \left(\delta^{n-1-t_3^2} + \delta^{n-1-t_6^2} - \delta^{n-1} + \delta^{n-1-t_6^2} - \delta^{n-1} \right) (1 - \delta) \tag{200}
$$

$$
= \left(\delta^{2k-1} + \delta^{2k} + \delta^{2k-1} - 3\delta^{n-1}\right) (1 - \delta). \tag{201}
$$

**Appending $c_7^{(4)}$:** Using an argumentation similar to the case of appending $c_1^{(4)}$, we have a total probability increase

$$
\Delta \Psi\left(\psi_{t_7}^{(4,n)}\right) = \left(\delta^{n-1-t_3^2} + \delta^{n-1-t_6^2} + \delta^{n-1-t_6^2} - \delta^{n-1} - \delta^{n-1} + \delta^{n-1} \right) (1 - \delta) \tag{202}
$$

$$
= \left(\delta^{2k-1} + \delta^{2k} + \delta^{2k-1} - 2\delta^{n-1}\right) (1 - \delta). \tag{203}
$$

Using the fact that $\delta^d$ is strictly decreasing in $d$ for $0 < \delta < 1$, we can conclude that

$$
\arg\max_{1 \leq j \leq 7} \Delta \Psi\left(\psi_{t_j}^{(4,n)}\right) = 6. \tag{204}
$$

This completes the proof. The proofs for $n \mod 3 = 1$ or 2 are similar and omitted.

## B Proof of Theorem 60

The proof of Theorem 60 is based on the exact average success probability for a BEC as a function of the type vector $t$ with a blocklength $n = \sum_{j=1}^7 t_j$. This problem is then transformed into a discrete multivariate constrained optimization problem.
We define the region of all possible types $\mathbf{t}$ as 
\begin{equation}
\mathcal{T}^{(M)} \triangleq \left\{ \mathbf{t} \in (\mathbb{N} \cup \{0\})^J : \sum_{j=1}^{J} t_j = n \right\}.
\end{equation}

Our goal is to find the globally optimized type $\mathbf{t}^*$ that satisfies
\begin{equation}
\mathbf{t}^* = \arg\min_{\mathbf{t} \in \mathcal{T}^{(M)}} P_e(\mathcal{E}^{(M,n)}_{\mathbf{t}}).
\end{equation}

Applying Theorem 56 for $M = 3$ or $M = 4$, we have
\begin{align}
P_e(\mathcal{E}^{(3,n)}_{\mathbf{t}}) &= \frac{1}{3} (\delta^{n-t_1} + \delta^{n-t_2} + \delta^{n-t_3} - \delta^{n}) ; \\
P_e(\mathcal{E}^{(4,n)}_{\mathbf{t}}) &= \frac{1}{4} (\delta^{n-(t_1+t_2+t_3)} + \delta^{n-(t_1+t_4+t_5)} + \delta^{n-(t_1+t_6+t_7)} + \delta^{n-(t_2+t_3+t_4)} + \delta^{n-(t_2+t_5+t_7)} + \delta^{n-(t_3+t_4+t_7)} + \delta^{n-t_1 + \delta^{n-t_2} - \delta^{n-t_4} - \delta^{n-t_7} + \delta^n}).
\end{align}

Since we consider the optimization problem for any fixed blocklength $n$ and hence $\delta^n$ is a constant, we can reformulate the discrete multivariate constrained minimization problem as follows:
\begin{equation}
\begin{aligned}
\text{minimize} & \quad f^{(M)}(\mathbf{t}) \triangleq \frac{M}{\delta^n} P_e(\mathcal{E}^{(M,n)}_{\mathbf{t}}) + (-1)^{M+1} \\
\text{subject to} & \quad \mathbf{t} \in \mathcal{T}^{(M)}
\end{aligned}
\end{equation}
where the minimization objective functions for $M = 3$ or $M = 4$ are 
\begin{equation}
f^{(3)}(\mathbf{t}) = \delta^{-t_1} + \delta^{-t_2} + \delta^{-t_3}
\end{equation}
and 
\begin{equation}
f^{(4)}(\mathbf{t}) = \delta^{-t_1-t_2-t_3} + \delta^{-t_1-t_4-t_5} + \delta^{-t_1-t_6-t_7} + \delta^{-t_2-t_4-t_6} + \delta^{-t_2-t_5-t_7} + \delta^{-t_3-t_4-t_7} - \delta^{-t_1} - \delta^{-t_2} - \delta^{-t_4} - \delta^{-t_7},
\end{equation}
respectively. Note that we add $(-1)^{M+1}$ in (209) to simplify the expression of $f^{(M)}(\mathbf{t})$.

We firstly consider the easier case of $M = 3$. Taking the locally optimal type $\mathbf{t}^*$ from Theorem 59, we will now prove that it is actually globally optimal for (210). Using $t_3 = n - t_1 - t_2$, we have
\begin{align}
f^{(3)}(\mathbf{t}) &= \delta^{-t_1} + \delta^{-t_2} + \delta^{t_1+t_2-n} \\
&\geq 2\sqrt{\delta^{-t_1} \delta^{-t_2} + \delta^{t_1+t_2-n}} \\
&\triangleq 2\delta^{-t} + \delta^{2t-n} \\
&\triangleq h(t),
\end{align}
where (213) holds because the arithmetic mean (AM) is never smaller than the geometric mean (GM), and in (214) we define $t \triangleq (t_1 + t_2)/2$. It can be seen that the function $2\delta^{-t} + \delta^{n-2t}$ is convex in $t$. Hence, its global minimum $3\delta^{-n/3}$ is given for the $t$ satisfying
\begin{equation}
\frac{\partial}{\partial t} (2\delta^{-t} + \delta^{2t-n}) \overset{!}{=} 0,
\end{equation}
where “$=\frac{1}{\delta}$” means “should be equal to,” i.e., the global minimizer of $h(t)$ is $t^* = \frac{n}{3}$.

However, one must be aware that the minimizer of $f^{(3)}(t)$ must be a positive integer. So, if $n = 3k$, taking $t_1^* = t_2^* = t_3^* = t^*$ trivially achieves the global minimum of $h(t)$, i.e., $3^\delta - n/3$. In the following we will investigate the discrete minimizer $t^*$ for $h(t)$ for the case of $n = 3k + 1$. The case $n = 3k + 2$ is similar and omitted.

Since the function $h(t)$ is convex, the minimizer should be equal to $k$ or $k + 1$. Therefore,

\[
\min \{ h(k), h(k + 1) \} = \min \left\{ 2\delta^{-k} + \delta^{-(k+1)}, 2\delta^{-(k+1)} + \delta^{-(k-1)} \right\} \tag{217}
= 2\delta^{-k} + \delta^{-(k+1)} \tag{218}
= h(k). \tag{219}
\]

Here we again use the AM–GM inequality to show that $2\delta^{-k} < \delta^{-(k+1)} + \delta^{-(k-1)}$. Thus the discrete global minimizer for $h(t)$ is $t^* = k$. Finally, since the inequality of (213) is achievable by $[t_1, t_2, t_3] = [k, k, k+1]$, we can conclude that a discrete global minimizer for $f^{(3)}(t)$ is $t^* = [k, k, k+1]$. Note that in Theorem 60, we state that the optimal type is $t^* = [k + 1, k, k]$. It is not difficult to show that the performance of these two codes is equivalent; so the optimal codes are not unique when $n = 3k + 1$.

In the case of $M = 4$ we must first prove that the globally optimal type $t^*$ must satisfy $t_1^* = t_2^* = t_3^* = t_4^* = 0$ for an arbitrary blocklength $n$. This turns out to be quite technical.

We reformulate the optimization problem in (209) as follows: introducing

\[u_j \triangleq \delta^{-t_j}, \quad 1 \leq j \leq J, \tag{220}\]

and noting that $1 \leq u_j \leq \delta^{-n}$ for $0 < \delta < 1$, we rewrite (211) as

\[g^{(4)}(u) \triangleq f^{(4)}(t) \tag{221}\]

and the optimization region (205) as

\[\mathcal{U}^{(4)} \triangleq \left\{ u \in \mathbb{R}^J : u_j \geq 1 \text{ and } \prod_{j=1}^{J} u_j = \delta^{-n} \right\}. \tag{222}\]

Note that while $\mathcal{T}^{(4)}$ is convex, $\mathcal{U}^{(4)}$ is not. We have

\[g^{(4)}(u) = u_1 u_2 u_3 + u_1 u_4 u_5 + u_1 u_6 u_7 + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 - (u_1 + u_2 + u_4 + u_7) \tag{223}\]
\[\geq u_1 (3(u_2 u_3 + u_4 u_5 + u_6 u_7 - 1) + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 - (u_2 + u_4 + u_7)) \tag{224}\]
\[\geq u_1 \left( 3 \left( \frac{\delta^{-n}}{u_1} \right)^{\frac{1}{2}} - 1 \right) + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 - (u_2 + u_4 + u_7) \tag{225}\]
\[= u_1 \left( 3 \left( \frac{\delta^{-n}}{u_1} \right)^{\frac{1}{2}} - 1 \right) + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 - (u_2 + u_4 + u_7) \tag{226}\]
\[= \left( 3 \delta^{-n} u_1^{\frac{2}{3}} - u_1 \right) + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 - (u_2 + u_4 + u_7). \tag{227}\]

Here, (225) follows from the AM–GM inequality, where equality holds if

\[u_2 u_3 = u_4 u_5 = u_6 u_7. \tag{228}\]
In (226), we use the fact that \( \prod_{j=1}^{7} u_j = \delta^{-n} \). The first term in parentheses on the right-hand-side (RHS) of (227) is concave and nondecreasing in \( u_1 \) for \( 1 \leq u_1 \leq \delta^{-n} \), and independent of the other variables \( u_2, \ldots, u_7 \). This implies that if we want to minimize (227), we should have \( u_1^* = 1 \) and the minimization is irrelevant to \( u_2^*, \ldots, u_7^* \).

To achieve equality in (225), we only need to satisfy the condition (228), which means that \( u_1^* = 1 \) is both the discrete global minimizer of the RHS of (227) and \( g^{(4)}(\mathbf{u}) \).

Using the same argument, we can also show that the discrete global optimizer \( \mathbf{u}^* \) must satisfy that \( u_1^* = u_3^* = u_4^* = u_7^* = 1 \), i.e., \( t_1^* = t_2^* = t_4^* = t_7^* = 0 \). So the discrete multivariate constrained optimization problem is reduced to

\[
\min_{t_{\text{weak}} \in \mathcal{T}_{\text{weak}}^{(4)}} f^{(4)}(t_{\text{weak}}) = \min_{t_{\text{weak}} \in \mathcal{T}_{\text{weak}}^{(4)}} \left( 2\delta^{-t_3} + 2\delta^{-t_5} + 2\delta^{-t_6} - 4 \right),
\]

where

\[
\mathcal{T}_{\text{weak}}^{(4)} \triangleq \left\{ t_{\text{weak}} \in (\mathbb{N} \cup \{0\})^7 : t_j \geq 0, j \in \{3, 5, 6\}, \text{ and } \sum_{j \in \{3,5,6\}} t_j = n \right\}.
\]

This problem can be solved in an analogous way as for \( M = 3 \). We obtain

\[
t^* = t_{\text{weak}}^* = [t_3^*, t_5^*, t_6^*] = \left[ \left\lfloor \frac{n+2}{3} \right\rfloor, \left\lfloor \frac{n+1}{3} \right\rfloor, \frac{n}{3} \right].
\]

### C Proof of Theorem 67

The proof is based on the exact average ML error probability formula expressed as a function of the linear type vector \( t_{\text{lin}} \). Applying Lemma 24 and Theorem 56 for the general three-dimensional linear code (whose corresponding \( r \)-wise Hamming distances can be derived from Example 26), we obtain

\[
f^{(8)}(t_{\text{lin}}) \triangleq \frac{8}{\delta^n} P_{\lambda}(\mathcal{F}^{(8,n)}_{t_{\text{lin}}})
= 4(u_1 u_2 u_3 + u_1 u_4 u_5 + u_1 u_6 u_7 + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 + u_3 u_5 u_6)
- 8(u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7)
+ 2(u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7)
+ \left( \frac{8}{4} \right) - 14 - \left( \frac{8}{5} \right) + \left( \frac{8}{6} \right) - \left( \frac{8}{7} \right) + \left( \frac{8}{8} \right)
= 4(u_1 u_2 u_3 + u_1 u_4 u_5 + u_1 u_6 u_7 + u_2 u_4 u_6 + u_2 u_5 u_7 + u_3 u_4 u_7 + u_3 u_5 u_6)
- 6(u_1 + u_2 + u_3 + u_4 + u_5 + u_6 + u_7) + 21,
\]

where for convenience we set

\[
u_\ell \triangleq \delta^{-t_\ell}, \quad 1 \leq \ell \leq K = 7.
\]

For a blocklength \( n = 7\kappa \), we know that the type of the fair linear code is

\[
t_{\text{lin}}^* = t_{\text{lin}}^* = \cdots = t_{\text{lin}}^* = \kappa.
\]

Plugging this into (234), we obtain that a fair linear code with blocklength \( n \) being a multiple of 7 has

\[
f^{(8)}(t_{\text{lin}}^*) = 28\delta^{-3\kappa} - 42\delta^{-\kappa} + 21.
\]
To show that this fair linear code is strictly suboptimal, we start to find a code of identical size and blocklength that has better performance. According to Example 66, such a code can be constructed from the fair weak flip code $C_{\text{fair}}^{(8,n)}$ of blocklength $n \mod L = 0$ (for $M = 8$ we have $L = 35$). By Corollary 47, a fair weak flip code with blocklength $n = 35\tau$ for $\tau \in \mathbb{N}$ and corresponding type $t_{\text{fair}}^{(\tau)}$

$$t_{j_1} = t_{j_2} = \cdots = t_{j_{35}} = \tau$$

satisfies

$$f^{(8)}(t_{\text{fair}}) = \left(\frac{8}{2}\right)\delta^{-15\tau} - \left(\frac{8}{3}\right)\delta^{-5\tau} + \left(\frac{8}{4}\right)\delta^{-\tau} - \left(\frac{8}{5}\right) + \left(\frac{8}{6}\right) - \left(\frac{8}{7}\right)$$

$$= 28\delta^{-15\tau} - 56\delta^{-5\tau} + 70\delta^{-\tau} - 36.$$  

Although no fair weak flip code exist if $n = 7\kappa \neq 35\tau$, we can define a generalized fair weak flip code for every $\kappa = 5\tau + \eta$, $0 \leq \eta \leq 4$, by carefully choosing $n = 7\kappa$ columns from the fair weak flip code with blocklength $35(\tau + 1) > n$ to form a new $(8,n)$ nonlinear weak flip code that is a concatenation of different $(8,7)$ Hadamard codes. I.e., the type vector consists of $7\eta$ components corresponding to weak flip columns that are equal to $\tau + 1$, and $(35 - 7\eta)$ components corresponding to weak flip columns that are equal to $\tau$. We denote this type vector as $t_{\text{weak}}^{(\kappa,\tau)}$.

Note that while there are many different $(8,7)$ Hadamard codes, they are all equivalent, i.e., they are only row- and column-permutations of $(37)$. For each of these $(8,7)$ Hadamard code, all the pairwise and three-wise Hamming matches are equal to 3 and 1, respectively; and there are 14 four-wise Hamming matches equal to 1 and $(35 - 7\eta)$ components corresponding to weak flip columns that are all pairwise Hamming matches equal to $3\kappa$ and that all three-wise Hamming matches equal to $\kappa$. For the four-wise Hamming matches, we select the Hadamard carefully to minimize the resulting four-wise Hamming matches. Indeed, we repetitively append the $(8,7)$ Hadamard code $\eta$ times to the fair weak flip code with $n = 35\tau$ to create an $(8,n = 7\kappa = 35\tau + 7\eta)$ generalized fair weak flip code such that $14\eta$ four-wise Hamming matches equal to $\tau + 1$ and $70 - 14\eta$ four-wise Hamming matches equal to $\tau$.

Hence, we see that

$$f^{(8)}(t_{\text{weak}}^{(\kappa)}) = 28\delta^{-3\kappa} - 56\delta^{-\kappa} + 14\eta\delta^{-(\tau+1)} + (70 - 14\eta)\delta^{-\tau} - 36.$$  

The proof is completed if one can show that except for $\tau = 0$ and $\eta = 1$

$$f^{(8)}(t_{\text{lin}}^{(\tau)}) - f^{(8)}(t_{\text{weak}}^{(\kappa)}) = 14\left[\delta^{-\kappa} - \delta^{-(\tau+1)} + (5 - \eta)\delta^{-\tau}\right] > 0.$$  

To that goal define $u \triangleq \delta^{-1} > 1$, and rewrite the terms in the bracket on the RHS of (242) as

$$p(u) \triangleq u^{5\tau+\eta} + 4 - \eta u^{\tau+1} - (5 - \eta)u^{\tau}.$$  

Observe that $p(1) = 0$ and that for $\tau = 0$,

$$\frac{\partial p(u)}{\partial u} = \eta u^{\tau-1} - \eta > 0, \quad \text{if } \eta \neq 1,$$

Lin, Moser, Chen, version 7, June 17, 2016 49
where the inequality holds because $u > 1$ and for $\tau \geq 1$,

$$
\frac{\partial p(u)}{\partial u} = (5\tau + \eta)u^{5\tau + \eta - 1} - \eta(\tau + 1)u^\tau - (5\tau - \eta\tau)u^{\tau - 1}
$$

(245)

$$
> (5\tau + \eta)u^{5\tau + \eta - 1} - \eta(\tau + 1)u^{\tau - 1} - (5\tau - \eta\tau)u^{\tau - 1}
$$

(246)

$$
= (5\tau + \eta)u^{5\tau + \eta - 1} - (5\tau + \eta)u^{\tau - 1}
$$

(247)

$$
\geq 0
$$

(248)

(where the inequalities again hold because $u > 1$). This implies that $p(u)$ is strictly larger than zero unless $\tau = 0$ and $\eta = 1$.

References


