The Capacity of Binary Channels that Use Linear Codes and Decoders

Samuel J. MacMullan, Member, IEEE, and Oliver M. Collins, Member, IEEE

Abstract—This paper analyzes the performance of concatenated coding systems operating over the binary-symmetric channel (BSC) by examining the loss of capacity resulting from each of the processing steps. The techniques described in this paper allow the separate evaluation of codes and decoders and thus the identification of where loss of capacity occurs. They are, moreover, very useful for the overall design of a communications system, e.g., for evaluating the benefits of inner decoders that produce side information. The first two sections of this paper provide a general technique (based on the cost weight distribution of a binary linear code) for calculating the composite capacity of the code and a BSC in isolation. The later sections examine the composite capacities of binary linear codes, the BSC, and various decoders. The composite capacities of the (8,4) extended Hamming, (24,12) extended Golay, and (48,24) quadratic residue codes appear as examples throughout the paper. The calculations in these examples show that, in a concatenated coding system, having an inner decoder provide more information than the maximum-likelihood (ML) estimate to an outer decoder is not a computationally efficient technique, unless generalized minimum-distance decoding of an outer code is extremely easy. Specifically, for the (8,4) extended Hamming and (24,12) extended Golay inner codes, the gains from using any inner decoder providing side information, instead of a strictly ML inner decoder, are shown to be no greater than 0.77 and 0.34 dB, respectively, for a BSC crossover probability of 0.1 or less. However, if computationally efficient generalized minimum-distance decoders for powerful outer codes, e.g., Reed–Solomon codes, become available, they will allow the use of simple inner codes, since both simple and complex inner codes have very similar capacity losses.

Index Terms—Channel capacity, concatenated codes, generalized minimum-distance decoding, mutual information.

I. INTRODUCTION

When a code and a decoding algorithm are used in isolation then the error probability is usually a sufficient measure of performance. However, when a code is used as an inner code in a concatenated coding system, more sophisticated measures of performance are necessary to evaluate it, since the decoder in a concatenated system may provide many types of information beyond its best estimate of the transmitted codeword to an outer decoder. 

The development of this paper is restricted to linear codes operating over the binary-symmetric channel (BSC); however, the methods have wide application and can be used to assess arbitrary codes on other channels such as the Gaussian channel.

The next three sections of this paper together set bounds on the gains provided through the use of side information. Section II provides a quick definition of composite capacity. Section III investigates the capacity of a “composite channel” consisting of an encoder and BSC in isolation. Each of these composite channels has a capacity less than the underlying BSC but greater than the composite of the encoder, BSC, and any nontrivial decoder. Thus this capacity can be used to bound the performance of the inner code, the channel, and any decoder. Section IV investigates the composite capacity of an inner code, BSC, and a maximum-likelihood (ML) inner decoder. The difference between the composite capacities calculated in Sections III and IV for a given code limits the gains that can be achieved by employing an inner decoder that provides side information beyond the ML estimate and thus limits the potential benefits of a generalized minimum-distance outer decoder. Section V calculates the composite capacity of an encoder, BSC, and various popular decoders which provide more information than merely the ML estimate. Section VI then uses all the techniques developed in the paper to determine the capacity loss that results when an (8,4) extended Hamming or (24,12) extended Golay code and a particular decoder are applied to the BSC.

Other papers have employed the concept of composite capacity to assess communication systems.1 Modulation systems are often analyzed using this technique, e.g., [15]–[17] evaluate the capacity of a bandlimited phase-only modulated system on a Gaussian channel. The work here is most closely related to bound the performance of the inner code, the channel, and any decoder.

1 The authors are grateful to one of the anonymous reviewers for pointing out that Theorem 1 and (5) also occur in [14] which was under review when this paper was submitted.
to [18] in which the composite capacity of a random code and BSC was investigated and [19] in which the composite capacity was used to assess orthogonal and bi-orthogonal codes with ML detection on a Gaussian channel. The composite capacity of the “wire-tap channel” and “magnetic recording channel” is used to characterize these channels in [20]–[22]; [23] and [24] discuss the composite capacity of a discrete memoryless channel with a d-decoder and provide a lower bound for this capacity.

II. COMPOSITE CAPACITY

Consider the communications system shown in Fig. 1. In this system, the output $U = (U_1, U_2, U_3, \ldots, U_k)$ of a binary source is encoded by an $(n, k)$ linear block code with generator matrix $G$ to yield a codeword $X = GU$. $X$ is then transmitted on a BSC with crossover probability $p$. At the output of the BSC, $Y$ is received and decoded. The output of the decoder may simply be the ML estimate $\hat{X}$ of $X$. Alternatively, the output of the decoder may be both $\hat{X}$ and other data such as an estimate of the probability of a correct decision given $Y$ or the second most likely codeword given $Y$.

The observation of the output $W$ of a certain stage in Fig. 1 provides on average $I(W; U)$ bits of information about the input, $U$. $I(W; U)$ is given by

$$I(W; U) = \sum_{w, u} p(w, u) \log \left( \frac{p(w | u)}{p(w)} \right)$$

where the summation is extended over all vectors $w$ in the range of $W$ and $u$ in the range of $U$. The capacity of the channel is the maximum amount of information that can be transmitted reliably over the channel per observation of $W$ and is given by

$$C_{WU} = \max_u I(W; U)$$

where the maximum is taken over all test sources $u$. The entropy of $W$ is $H(W)$ and the conditional entropy of $W$ given $U$, is $H(W | U)$. If $U$ is the capacity achieving source distribution

$$C_{WU} = H(W) - H(W | U) = H(U) - H(U | W).$$

Since $X$ is an invertible function of $U$, $C_{WU} = C_{WX}$ for a particular code.

The capacity of a BSC is given by $C_{\text{BSC}} = 1 - H(p)$ where $H(p)$ is the binary entropy function; $nC_{\text{BSC}}$ is the capacity of a BSC used $n$ times. Therefore, $C_{YU}/(nC_{\text{BSC}})$ represents the fraction of the BSC capacity remaining after the application of a code. If $Z$ is the output of a decoder, the fraction of the BSC capacity remaining after the application of a code and the decoder is $C_{ZU}/(nC_{\text{BSC}})$. The next sections of the paper calculate these two quantities.

III. COMPOSITE CAPACITY OF A LINEAR BLOCK ENCODER AND BSC

This section presents a way to calculate $C_{YU}/(nC_{\text{BSC}})$ for a linear code. First, some definitions are necessary. An $(n, k)$ linear block code $C = (x_1, x_2, \ldots, x_{2^n})$ is a $k$-dimensional subspace of the vector space $V_n = (y_1, y_2, \ldots, y_{2^n})$ of all binary $n$-tuples. Given a transmitted codeword $x$, the output $y$ of the BSC can be any one of the $2^n$ $n$-tuples in $V_n$. The standard array associated with $C$ has $2^{n-k}$ rows, called cosets, each consisting of $2^k$ distinct $n$-tuples. Each $n$-tuple occurs once in the standard array. The first row consists of the codewords. The first $n$-tuple of each coset is the coset leader and is a vector of minimum Hamming weight in the coset. For the $j$th coset, the coset leader is represented by $e_j$. An entry in a coset is equal to its coset leader modulo-2 added to the codeword at the top of the standard array column above the entry.

The following lemma is useful in subsequent derivations in this paper:

**Lemma 1:** Given any $x_i, x_j$, and $y$

$$p(Y = y | X = x_i) = p(Y = y + x_j | X = x_i + x_j).$$

**Proof:**

$$p(Y = y | X = x_i) = p^{(n)}(y + x_j)(1 - p)^{n-1}$$

$$= p^{(n)}(y + x_j + x_i + x_j)$$

$$= p(Y = y + x_j | X = x_i + x_j)$$

where $^H(x)$ is the Hamming weight of $x$.

Because of the symmetry of linear block codes, the BSC, and the decoders considered in this paper, $C_{YX}$ is achieved by a binary-symmetric source (BSS), i.e.,

$$p(X = x) = p(U = u) = 1/2^k.$$  

Lemma 2 then simplifies the computation of $H(Y)$ for a particular code by showing that all members of a coset with coset leader $e$ have probability $p(Y = e)$ at the output of the BSC.

**Lemma 2:** Assuming a BSS, $p(Y = y_i) = p(Y = e)$, where $e$ is the coset leader of the coset containing $y_i$ (i.e., $y_i = e + x_i$ for some $x_i$).

**Proof:** Using the law of total probability, $p(X = x) = 1/2^k$, and Lemma 1, $p(Y = y_i)$ can be expressed

$$p(Y = y_i) = \frac{1}{2^k} \sum_x p(Y = y_i | X = x)$$

$$= \frac{1}{2^k} \sum_x p(Y = e | X = x)$$

$$= p(Y = e).$$

Theorem 1 then uses Lemma 2 to determine $C_{YX}$:

**Theorem 1:** $C_{YX} = k + H(R) - nH(p)$, where $R$ is the random variable specifying the row containing $y$. 

---

Fig. 1. Assumed model for communications system.
Proof: The set of rows (or cosets) in the standard array is \( \{ r_1, r_2, \cdots, r_{2^m-1} \} \) where \( r_i \) designates the \( i \)th row. Lemma 2 shows that

\[
H(Y) = \sum_{y} p(Y = y) \log \left( \frac{1}{p(Y = y)} \right)
= k + \sum_{R=r} p(R = r) \log \left( \frac{1}{p(R = r)} \right)
= k + H(R),
\]

(1)

Since \( H(Y|X) = nH(p) \), the theorem follows from \( C_{\gamma} = H(Y) - H(Y|X) \).

An interpretation of (1) is that \( y \) is uniquely specified by its address (i.e., row and column) in the standard array. The entropy of the RV specifying the column containing \( y \) is \( k \) bits. Equation (1) follows since the row and column addresses are independent.

The coset weight distribution of a code fixes \( H(R) \). The weight distribution of a particular coset specifies the number of \( n \)-tuples of each Hamming weight in that coset. The coset weight distribution is used to determine \( p(R = r) \) for all \( r \). From Lemma 2, \( p(R = r) = 2^k p(Y = c_i) \), where \( c_i \) is the coset leader of \( r_i \). \( p(Y = c_i) \) can be written

\[
p(Y = c_i) = \sum_{x} p(Y = c_i | X = x) p(X = x)
= 2^{-k} \sum_{y \in r_i} p^{n}(y)(1 - p)^{n - w_n(y)}.
\]

Therefore, \( H(R) \) is given by

\[
H(R) = \sum_{i=1}^{2^{n-1}} p(R = r_i) \log \left( \frac{1}{p(R = r_i)} \right)
= \sum_{i=1}^{2^{n-1}} \sum_{y \in r_i} p^{n}(y)(1 - p)^{n - w_n(y)} \log \left( \frac{1}{\sum_{y \in r_i} p^{n}(y)(1 - p)^{n - w_n(y)}} \right).
\]

The outer summation in the above expression extends over all rows and the inner summation extends over all \( n \)-tuples in that row. The only property of one of these \( n \)-tuples that affects \( H(R) \) is its Hamming weight. Thus the inner summation can be rewritten in terms of the weight distribution of the row. Defining \( u(i, j) \) as the number of \( n \)-tuples of Hamming weight \( j \) in the \( i \)th row of the standard array of \( C \) yields

\[
H(R) = \sum_{i=1}^{2^{n-1}} \sum_{j=0}^{n} u(i, j) p^j (1 - p)^{n-j} \log \left( \frac{1}{\sum_{j=0}^{n} u(i, j) p^j (1 - p)^{n-j}} \right).
\]

Many cosets will have the same distribution of Hamming weights, and, since the outer summation extends over all rows, all that is needed to calculate \( H(R) \) is the number of rows in the standard array with every possible weight distribution. \( N_d \) in Table I indicates the number of rows with the \( d \)th weight distribution, \( d \in [1, 2, \cdots, D_R] \), for the extended Hamming code, \( H_8 \). Tables II and III present the coset weight distribution for the \((24, 12)\) extended Golay code, \( G_{24} \). \( u^*(d, j) \) is the number of \( n \)-tuples of Hamming weight \( j \) in the \( d \)th weight distribution. \( H(R) \) can be calculated from the data in the tables using

\[
H(R) = \sum_{d=1}^{D_R} N_d \sum_{j=0}^{n} u^*(d, j) p^j (1 - p)^{n-j} \log \left( \frac{1}{\sum_{j=0}^{n} u^*(d, j) p^j (1 - p)^{n-j}} \right).
\]

Determining the coset weight distribution of a code is in general an intractable problem [25]. However, the coset weight distribution is known for several codes including \((2^n-1, 2^n-m-1)\) Hamming codes [26], double-error correcting binary BCH codes and their extensions [27], (23, 12) and (24, 12) Golay codes [26], and \((2m, m)\) quadratic-residue (QR) codes with \( m = 4, 8, 16, \) or 24 (e.g., [28]). Table I is derived from [26, Appendix I]; Tables II and III appear explicitly in [26]. [28] provides a method for determining the coset weight distribution of the \((48, 24)\) quadratic residue code.

Fig. 2 shows how \( C_{\gamma} / (nC_{BSC}) \) varies as a function of the crossover probability for \( H_8 \), \( C_{24} \), and the \((48, 24)\) QR code. The figure demonstrates that there is little variation in the capacity loss for these codes. For example, \( C_{\gamma} / (nC_{BSC}) \) for \( H_8 \) and the \((48, 24)\) QR code differs by less than 0.1 for all crossover probabilities. Simple and complex inner codes appear to work equally well on the BSC if the inner decoder...
Fig. 2. Capacity loss for different codes applied to the BSC.

TABLE II

<table>
<thead>
<tr>
<th>$d = 1$</th>
<th>$d = 2$</th>
<th>$d = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_d$</td>
<td>1</td>
<td>276</td>
</tr>
<tr>
<td>$w^*(d, 0)$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 2)$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$w^*(d, 4)$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 6)$</td>
<td>0</td>
<td>77</td>
</tr>
<tr>
<td>$w^*(d, 8)$</td>
<td>759</td>
<td>352</td>
</tr>
<tr>
<td>$w^*(d, 10)$</td>
<td>0</td>
<td>946</td>
</tr>
<tr>
<td>$w^*(d, 12)$</td>
<td>2576</td>
<td>1344</td>
</tr>
<tr>
<td>$w^*(d, 14)$</td>
<td>0</td>
<td>946</td>
</tr>
<tr>
<td>$w^*(d, 16)$</td>
<td>759</td>
<td>352</td>
</tr>
<tr>
<td>$w^*(d, 18)$</td>
<td>0</td>
<td>77</td>
</tr>
<tr>
<td>$w^*(d, 20)$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 22)$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$w^*(d, 24)$</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE III

<table>
<thead>
<tr>
<th>$d = 4$</th>
<th>$d = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_d$</td>
<td>24</td>
</tr>
<tr>
<td>$w^*(d, 1)$</td>
<td>1</td>
</tr>
<tr>
<td>$w^*(d, 3)$</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 5)$</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 7)$</td>
<td>253</td>
</tr>
<tr>
<td>$w^*(d, 9)$</td>
<td>506</td>
</tr>
<tr>
<td>$w^*(d, 11)$</td>
<td>1288</td>
</tr>
<tr>
<td>$w^*(d, 13)$</td>
<td>1288</td>
</tr>
<tr>
<td>$w^*(d, 15)$</td>
<td>506</td>
</tr>
<tr>
<td>$w^*(d, 17)$</td>
<td>253</td>
</tr>
<tr>
<td>$w^*(d, 19)$</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 21)$</td>
<td>0</td>
</tr>
<tr>
<td>$w^*(d, 23)$</td>
<td>1</td>
</tr>
</tbody>
</table>

Theorem 2:

$$\lim_{p\to1/2} \frac{C_{YU}}{nC_{\text{BSC}}} = 1.$$  

Proof: Using a Taylor series expansion around $p = 1/2$, $C_{\text{BSC}}$ can be expressed

$$C_{\text{BSC}}|_{p=1/2} + \frac{\partial}{\partial p} C_{\text{BSC}}|_{p=1/2} \epsilon + \frac{\partial^2}{\partial^2 p} C_{\text{BSC}}|_{p=1/2} \frac{\epsilon^2}{2} + O(\epsilon^3)$$  

(2)
where \( \epsilon = 1/2 - p \). Since \( \frac{\partial}{\partial p} C_{\text{BSC}} |_{p=1/2} = 0 \) and \( \frac{\partial^2}{\partial p^2} C_{\text{BSC}} |_{p=1/2} = 4/\ln 2 \), (2) shows that

\[
C_{\text{BSC}} = \frac{2^2}{\ln 2} + O(\epsilon^3).
\]

An expression with form similar to (2) can be written for \( C_{\text{UY}} \).

Straightforward algebra provides \( H(R) |_{p=1/2} = n - k \),

\[
\frac{\partial}{\partial p} H(R) |_{p=1/2} = \frac{n-k-1}{2^n-1} \left( \sum_{i=1}^{2^{n-k}} \sum_{y \in \mathcal{C}_i} 2^{|y|} (2^{2^k} - 2^n n) \right),
\]

and

\[
\frac{\partial^2}{\partial p^2} H(R) |_{p=1/2} = \frac{2^{2-k} - 2^k}{2^n} \left( \sum_{i=1}^{2^{n-k}} \left( \sum_{y \in \mathcal{C}_i} 2^{|y|} (2^{2^k} - 2^n n) \right)^2 \right).
\]

Because

\[
\sum_{y \in \mathcal{C}_i} w_{\mathcal{C}_i}(y) = 2^{i-1} n, \quad 1 \leq i \leq 2^{n-k},
\]

\( \frac{\partial}{\partial p} H(R) |_{p=1/2} \) and \( \frac{\partial^2}{\partial p^2} H(R) |_{p=1/2} \) are zero. Therefore,

\[
C_{\text{UY}} = (k + H(R) - n H(p)) |_{p=1/2} \epsilon
\]

\[
+ \frac{2^2 - 2^k}{\ln 2} \left( \sum_{i=1}^{2^{n-k}} \left( \sum_{y \in \mathcal{C}_i} 2^{|y|} (2^{2^k} - 2^n n) \right)^2 \right) + O(\epsilon^3),
\]

and, as a result,

\[
\lim_{p \rightarrow 1/2} C_{\text{UY}} = \lim_{p \rightarrow 1/2} n C_{\text{BSC}}.
\]

The next two sections derive the reduction in capacity when a decoder is included in the composite channel.

IV. COMPOSITE CAPACITY OF ENCODER, BSC, AND ML DECODER

The most commonly used inner decoder in a concatenated coding system is the ML decoder. The output, \( \hat{X} \), of the ML decoder is the most likely codeword given \( Y \) (Fig. 3). This section calculates the composite capacity \( C_{\text{XX}} \) when a particular encoder and an ML decoder are applied to the BSC.

The following lemma will be useful in the derivation of \( C_{\text{XX}} \).

Lemma 3: \( p(\hat{X} = x_i | X = x_j) = p(\hat{X} = x_i | X = 0) \)

where \( x_i + x_j = x_t \).

Proof: Defining the set of columns in the standard array as \( \mathcal{C} = (c_1, c_2, \ldots, c_{2^k}) \), where \( c_i \) designates the \( i \)-th column, we can write

\[
p(\hat{X} = x_i | X = x_j) = \sum_{y \in \mathcal{C}_i} p(Y = y | X = x_j) = \sum_{y \in \mathcal{C}_i} p(Y = e + x_i + x_j | X = 0)
\]

\[
= \sum_{y \in \mathcal{C}_i} p(Y = e | X = 0).
\]

Substituting (3) into (4) yields

\[
C_{\text{XX}} = k \cdot \log \left( \frac{1}{\sum_{y \in \mathcal{C}_i} p(Y = y | X = 0)} \right).
\]

5

Therefore, \( C_{\text{XX}} \) becomes

\[
C_{\text{XX}} = k \cdot \log \left( \frac{1}{\sum_{y \in \mathcal{C}_i} p(Y = y | X = 0)} \right).
\]

where the second line in the preceding expression results from Lemma 1. Letting \( x_i + x_j = x_t \),

\[
p(\hat{X} = x_i | X = x_j) = p(Y = e | X = 0) = \sum_{y \in \mathcal{C}_i} p(Y = y | X = 0).
\]

The lemma follows from the observation that

\[
p(\hat{X} = x_i | X = 0) = \sum_{y \in \mathcal{C}_i} p(Y = y | X = 0).
\]

Since \( p(\hat{X} = x) = 1/2^k \), the entropy of the ML estimate is \( k \) bits. Also, because of symmetry,

\[
H(\hat{X} | X) = H(\hat{X} | X = 0).
\]

Therefore,

\[
C_{\text{XX}} = H(\hat{X} | X) - H(\hat{X} | X = 0) = k - H(\hat{X} | X = 0).
\]

By definition

\[
H(X | X = 0) = \sum_{x_i} \sum_{y \in \mathcal{C}_i} p(\hat{X} = x_i | X = 0) = \frac{2^k}{\sum_{x_i} p(\hat{X} = x_i | X = 0)}.
\]

Substituting (3) into (4) yields

\[
C_{\text{XX}} = k \cdot \log \left( \frac{1}{\sum_{y \in \mathcal{C}_i} p(Y = y | X = 0)} \right).
\]

Fig. 3. Decoder—maximum likelihood.
The outer summation in the second line of (5) extends over all the columns in the standard array and the inner summation extends over all \( n \)-tuples in the column. The only property of one of these \( n \)-tuples that affects \( C_{XX} \) is its Hamming weight. Thus the inner summation can be rewritten in terms of the weight distribution of the column. In analogy with the weight distribution of a coset, the weight distribution of a column in the standard array specifies the number of \( n \)-tuples of each Hamming weight in that column. The column weight distribution of a code gives the number of columns of each different weight distribution. Define \( v(i, j) \) as the number of \( n \)-tuples of Hamming weight \( j \) in the \( d \)th column of the standard array. Then (5) can be rewritten

\[
C_{XX} = k - \sum_{i=1}^{2^k} \sum_{j=0}^{n} v(i, j) p^j (1 - p)^{(n-j)} \cdot \log \left( \frac{1}{n} \sum_{j=0}^{n} v(i, j) p^j (1 - p)^{(n-j)} \right) \tag{6}
\]

Just as many rows have the same weight distribution, many columns have the same weight distribution. If \( v^d(d, j) \) is the number of \( n \)-tuples of Hamming weight \( j \) in the \( d \)th weight distribution, \( d \in \{1, 2, \cdots, D_n \} \), for a standard array column and \( N_d \) is the number of columns with the \( d \)th weight distribution, then (6) becomes

\[
C_{XX} = k - \sum_{d=1}^{D_n} N_d \sum_{j=0}^{n} v^d(d, j) p^j (1 - p)^{(n-j)} \cdot \log \left( \frac{1}{n} \sum_{j=0}^{n} v^d(d, j) p^j (1 - p)^{(n-j)} \right) \tag{7}
\]

Table IX in Appendix I gives \( v^d(d, j) \) and \( N_d \) for \( H_8 \) and Tables XII and XIII provide this information for \( G_{24} \). Appendix I shows how the tables were derived. The derivation assumes that the minimum-weight coset leaders are used in the standard array decoding scheme. However, if instead non-minimum-weight coset leaders were used, (7) still applies and can be computed with knowledge of \( v^d(d, j) \) and \( N_d \) for the non-minimum-weight coset leader array. Tables IX, XII, and XIII provide more information than needed to compute \( C_{XX} \). This information will be necessary in Section V to evaluate the capacity loss for decoders other than an ML decoder. The tables show not only \( v^d(d, j) \) but also several different terms that add up to \( v^d(d, j) \). Each of these terms represents the number of \( n \)-tuples of weight \( j \) that result from the modulo-2 addition of a codeword at the top of a standard array column with the \( d \)th weight distribution with each weight coset leader. For example, in Table IX, there are a total of five weight four codewords in the column under a weight four codeword (i.e., \( v^4(2, 4) = v^4(3, 4) = 5 \)). Contributing to this total are four vectors resulting from the addition of a weight two coset leader and the codeword (this is denoted \( 2_2 \)) and one vector resulting from the addition of the all-zero coset leader and the codeword (this is denoted \( 1_0 \)). Note, for a particular standard array column, the weight distribution for \( H_8 \) and \( G_{24} \) depends on the weight of the codeword at the top of the column and, in some cases, depends on whether there is a one in the last coordinate (i.e., the \( n \)th bit) of the codeword. For example, there are 165 weight eight \( n \)-tuples in a column under a weight 12 codeword with a one in the last coordinate. However, there are no weight eight \( n \)-tuples in a column under a weight 12 codeword with a zero in the last coordinate. Appendix I describes why this occurs.

The column weight distribution is usually more difficult to calculate than the coset weight distribution because the column weights typically exhibit much less symmetry than the coset weights. The next two theorems show how to bound \( C_{XX} \) when the complete column weight distribution for a code cannot be calculated.

**Theorem 3:** \( C_{XX} \geq k - H(P_e) - P_e \log(2^k - 1) \) where the probability of error, \( P_e = p(\hat{X} \neq 0 | X = 0) \).

**Proof:** Fano’s inequality [29] applied to the composite channel states \( H(\hat{X} | X) \leq H(P_e) + P_e \log(2^k - 1) \) where the condition for equality is

\[
p_i = p(\hat{X} = x_i | X = 0) = (1 - P_e)/(2^k - 1), \quad 1 < i \leq 2^k.
\]

The Theorem follows from substituting Fano’s inequality into

\[
C_{XX} = H(\hat{X}) - H(\hat{X} | X).
\]

Theorem 3 requires knowledge of \( P_e \) to lower-bound \( C_{XX} \). If \( P_e \) is unknown, a lower bound for \( C_{XX} \) can still be established by upper-bounding \( P_e \) because \( H(\hat{X}) + P_e \log(2^k - 1) \) is increasing with \( P_e \) for \( P_e \leq 1 - 1/2^k \) (i.e., for meaningful \( P_e \)). For example, one upper bound, \( P_e^* \), for \( P_e \) is

\[
P_e \leq P_e^* = \sum_{i=1}^{n} \left( \begin{array}{c} n \\ i \end{array} \right) p^i (1 - p)^{n-i}
\]

where \( t = \lfloor (d_{\min} - 1)/2 \rfloor \), \( d_{\min} \) is the minimum distance of the code, and \( \lfloor x \rfloor \) denotes the greatest integer less than or equal to \( x \). Then

\[
C_{XX} \geq k - H(P_e^*) - P_e^* \log(Q^k - 1).
\]

If \( d_{\min} \) is also unknown, the Gilbert–Varshamov bound can be used to lower-bound \( t \).

**Theorem 4:**

\[
C_{XX} \leq k - \sum_{i} p_i^* \log(1/p_i^*)
\]

where \( p_i^* = p(\hat{X} = x_i | X = 0) \) and it is assumed that the first column of the standard array is filled with the \( 2^{n-k} \) lowest weight \( n \)-tuples from \( V_n \), the second column is filled with the next \( 2^{n-k} \) lowest weight \( n \)-tuples, \( \ldots \), and the last column is filled with the highest weight \( n \)-tuples (note: this array, constructed solely for the purposes of calculating the bound, depends only on \( n \) and \( k \)).
Fig. 4. Comparison of actual capacity loss and bounds on capacity loss when \((24, 12)\) extended Golay code and ML decoder are applied to BSC.

**Proof:** Consider vectors \(\mathbf{y}_i\) and \(\mathbf{y}_j\) from columns \(\mathbf{c}_i\) and \(\mathbf{c}_j\), respectively, of the standard array constructed in Theorem 4. Suppose \(w_H(\mathbf{y}_j) > w_H(\mathbf{y}_i)\). The contribution, \(\tilde{\beta}_{ij}\), to \(C_{XX}\) from \(\mathbf{c}_i\) and \(\mathbf{c}_j\) is \(\tilde{\beta}_{ij} = p_i^* \log(1/p_i^*) + p_j^* \log(1/p_j^*)\). Suppose the array is modified by switching the position of \(\mathbf{y}_i\) and \(\mathbf{y}_j\). The resulting contribution to \(C_{XX}\) from \(\mathbf{c}_i\) and \(\mathbf{c}_j\) would be \(\tilde{\beta}_{ij} = \tilde{p}_i^* \log(1/\tilde{p}_i^*) + \tilde{p}_j^* \log(1/\tilde{p}_j^*)\) where

\[
\tilde{p}_i^* = p_i^* + p_i^* y_i(1-p)^{y_i} - p_i y_i(1-p)^{y_i} - p_i y_i(1-p)^{y_i} y_i,
\]

and

\[
\tilde{p}_j^* = p_j^* + p_j^* y_j(1-p)^{y_j} - p_j y_j(1-p)^{y_j} - p_j y_j(1-p)^{y_j} y_j.
\]

Because of the concavity of the function \(x \log(1/x)\), \(\tilde{\beta}_{ij} < \beta_{ij}\). Therefore, any change to the standard array constructed in Theorem 4 would decrease \(C_{XX}\). Thus \(C_{XX}\) calculated using this structure bounds the composite capacity for an arbitrary \((n, k)\) code.

The standard array developed to calculate the bound in Theorem 4 is optimum in the sense of maximizing \(C_{XX}\) for a given \(n\) and \(k\). The authors have not discovered a code with a standard array exhibiting this optimum structure. Fig. 4 plots the lower bound of Theorem 3 and the upper bound of Theorem 4 for the composite channel consisting of \(G_{24}\), the BSC, and an ML inner decoder and shows how they compare to the actual composite capacity calculated from Tables XII and XIII. The lower bound is computed using the actual \(p_e\) for \(G_{24}\). Fig. 4 shows that the bounds are tight for useful \(p\) (i.e., \(p < 0.1\)); however, they become increasingly loose as \(p\) approaches 1/2. For the codes we have investigated, the bound in Theorem 4 approaches but does not exceed \(C_{YY}\) as \(p\) approaches 1/2.

**V. Composite Capacity of Decoders Providing Side Information Beyond the ML Estimate**

This section evaluates the composite capacity of an encoder, BSC, and one of several common decoders providing side information. The section examines first a decoder that outputs the ML estimate or an erasure, then a decoder that outputs the ML estimate and the probability the estimate is correct, and finally, a decoder that outputs the \(m\) most likely codewords.

**A. ML Estimate or Erasure**

Suppose the decoder output, \(Q\), consists of the ML estimate or an erasure, \(e\), as shown in Fig. 5. The decoder outputs an erasure if \(\mathbf{y} \in \Psi\) where \(\Psi\) is a subset of the rows of the standard array.

The following lemmas simplify the calculation of the capacity \(C_{QX}\) of the composite channel shown in Fig. 5.

**Lemma 4:** \(p(Q = x_i | X = x_j) = p(Q = x_i | X = 0)\) where \(x_i + x_j = x_t\).
Proof:

\[ p(Q = x_i | X = x_j) = \sum_{y \in \mathbb{F}^n} p(Y = y | X = x_j) \]
\[ = \sum_{e \in \mathbb{F}^n} p(Y = e + x_i | X = x_j) \]
\[ = \sum_{e \in \mathbb{F}^n} p^\exists(1 - p)^{n-u_3(e+x_i;x_j)} \]
\[ = p(Q = x_i | X = 0). \]

Lemma 5: \( p(Q = \epsilon | X = x_j) = p(Q = \epsilon) \).

Proof:

\[ p(Q = \epsilon | X = x_j) \]
\[ = \sum_{e \in \Psi} \sum_{i=1}^{2^k} p(Y = e + x_i | X = x_j) \]
\[ = \sum_{e \in \Psi} \sum_{i=1}^{2^k} p^\exists(1 - p)^{n-u_3(e+x_i;x_j)}. \]

Setting \( x_i = x_i + x_j \), the above expression becomes

\[ p(Q = \epsilon | X = x_j) = \sum_{e \in \Psi} \sum_{i=1}^{2^k} p^\exists(1 - p)^{n-u_3(e+x_i)}. \]

The inner summation in the above expression is independent of \( x_j \). The lemma follows.

Theorem 5 then presents a formula to determine \( C_{QX} \).

Theorem 5:

\[ C_{QX} = \sum_{i=1}^{2^k} p(Q = x_i | X = 0) \cdot \log \left( \frac{p(Q = x_i | X = 0)}{p(Q = x_i)} \right). \]  

(8)

Proof: \( C_{QX} \) can be written

\[ C_{QX} = 2^{-k} \sum_{i=1}^{2^k} \left( \sum_{j=1}^{2^k} p(Q = x_i | X = x_j) \cdot \log \left( \frac{p(Q = x_i | X = x_j)}{p(Q = x_i)} \right) + p(Q = \epsilon | X = x_j) \cdot \log \left( \frac{p(Q = \epsilon | X = x_j)}{p(Q = \epsilon)} \right) \right). \]

The proof follows from the application of Lemmas 4 and 5 to the above expression.

Theorem 6:

\[ C_{QX} = C_{X \times X} + H(p(Q = \epsilon)). \]

Proof: In the communications system shown in Fig. 6, the source determines if a certain erasure decoder would output an erasure based on feedback from the output of a memoryless channel. The source then passes this information for each codeword directly to the sink over an error free binary channel \( E \).

Feedback does not increase capacity [30]; therefore, \( C_{X \times X} = C_{X \times X} \). Also, the capacity of \( E \) is no more than \( H(p(Q = \epsilon)) \) bits per channel \( E \) use. (\( E \) is used only once per codeword.) Thus

\[ C_{QX} = C_{X \times X} + H(p(Q = \epsilon)). \]

(9)
where $\Delta$ is given by
\[
\Delta = \sum_{i=1}^{2^k} p(\hat{X} = x_i | X = 0) \log \left( \frac{1}{p(\hat{X} = x_i | X = 0)} \right) - p(Q = x_i | X = 0) \log \left( \frac{1}{p(Q = x_i | X = 0)} \right).
\]

When $k \geq 2$, (10) is bounded by
\[
\Delta \leq P_c \log \left( \frac{1}{P_c} \right) - p(Q = 0 | X = 0) \log \left( \frac{1}{p(Q = 0 | X = 0)} \right) + (2^k - 1)(\gamma \log (\frac{1}{\gamma}) - p(\hat{Q} = \hat{x} | X = 0) \log \left( \frac{1}{p(\hat{Q} = \hat{x} | X = 0)} \right)
\]
where the probability the ML codeword is correct, $P_c = 1 - P_e$, $p(\hat{Q} = \hat{x} | X = 0) \leq p(Q = x_i | X = 0)$ for all $x_i \neq 0$ (e.g., $\hat{x} = 1$ for $H_8$ and $G_{24}$), and
\[
\gamma = \frac{p(Q = \hat{e}) - P_c + p(\hat{X} = 0 | X = 0)}{2^k - 1} + p(Q = \hat{x} | X = 0).
\]

Proof: Equations (9) and (10) follow from (4), Theorem 5, and $p(Q = \hat{e}) = (1 - p(Q = \hat{e}))/2^k$. To derive the bound on $\Delta$, note that
\[
p(Q = \hat{e}) = \sum_{i=1}^{2^k} p(\hat{X} = x_i | X = 0) - p(Q = x_i | X = 0).
\]
The above expression indicates that $p(Q = \hat{e})$ is taken from the probabilities $p(\hat{X} = x_i | X = 0)$. Because of the concavity of $x \log (1/x)$, $\Delta$ can be bounded, for $k \geq 2$, by assuming that $p(Q = x_i | X = 0) = p(Q = \hat{x} | X = 0)$ for $x_i \neq 0$ and that each $p(\hat{X} = x_i | X = 0)$, $x_i \neq 0$, contributes the same amount to $p(Q = \hat{e})$, i.e.,
\[
p(\hat{X} = x_i | X = 0) = \gamma, \quad x_i \neq 0.
\]
The bound on $\Delta$ follows from these assumptions and (10).

If $\hat{x}$ is unknown, a slightly looser bound results by assuming that $p(Q = x_i | X = 0) = 0$ for $x_i \neq 0$ and
\[
p(\hat{X} = x_i | X = 0) = \frac{p(Q = \hat{e}) - P_c + p(\hat{X} = 0 | X = 0)}{2^k - 1}, \quad x_i \neq 0.
\]
With these assumptions
\[
\Delta \leq P_c \log \left( \frac{1}{P_c} \right) - p(Q = 0 | X = 0) \log \left( \frac{1}{p(Q = 0 | X = 0)} \right) + (2^k - 1)(\gamma \log (\frac{1}{\gamma}) - p(\hat{Q} = \hat{x} | X = 0) \log \left( \frac{1}{p(\hat{Q} = \hat{x} | X = 0)} \right)
\]
In practical circumstances, $P_c \geq 1/e$. Assuming this condition and noting that $P_c \geq p(Q = 0 | X = 0)$, we have
\[
P_c \log \left( \frac{1}{P_c} \right) \leq p(Q = 0 | X = 0).
\]
Therefore, from (10),
\[
\Delta \leq p(Q = \hat{e}) \log \left( \frac{2^k - 1}{p(Q = \hat{e})} \right) - k p(Q = \hat{e}) + p(Q = \hat{e}) \log \left( \frac{1}{p(Q = \hat{e})} \right).
\]
Combining (9) and (11) yields the bound in Theorem 6.

### B. ML Estimate and Probability that the Estimate is Correct

Suppose now the decoder output consists of the ML estimate $\hat{X}$ and the probability $\rho$ of a correct decision given $Y$. Such a decoder is the starting point for Hagenauer’s soft-output Viterbi algorithm [8]. The mutual information between this decoder output and the encoder output is given by $I(\hat{X}, \rho; X) = I(\hat{X}, X) + I(\hat{X}; \rho | X)$ [29]. Since $I(\hat{X}, \rho; X) \geq 0$, $I(X, \rho; X) \geq I(\hat{X}; X)$. As a result, this decoder implementation yields a capacity at least as large as the capacity given by an ML decoder.

From Bayes’ Theorem and Lemmas 1 and 2
\[
\rho \geq \frac{p(X = x_i | Y = e + x_i)}{p(Y = e)}.
\]
If $e_i$ and $e_j$ have the same coset weight distribution
\[
\frac{p(Y = e_i | X = 0)}{p(Y = e_i)} - \frac{p(Y = e_j | X = 0)}{p(Y = e_j)}.
\]
Therefore, there are at most $D$ possible values of $\rho$. Equality may also hold in cases where $e_i$ and $e_j$ do not have the same coset weight distribution. Suppose there are $\Gamma$ possible values of $\rho$, i.e., $\rho \in \{\rho_1, \rho_2, \ldots, \rho_{\Gamma}\}$ where $\rho_i$ is the $i$th possible probability. $I(\hat{X}, \rho; X)$ can then be computed using [29]
\[
I(\hat{X}, \rho; X) = \sum_{i=1}^{\Gamma} \sum_{j=1}^{\Gamma} \sum_{i=1}^{\Gamma} \frac{1}{p(X = x_i, \rho = \rho_i, X = x_j)} \log \left( \frac{p(X = x_i | \hat{X} = x_i, \rho = \rho_i)}{p(X = x_j)} \right).
\]
Noting that the BSS is the capacity achieving source distribution
\[
p(X = x_j | \hat{X} = x_i, \rho = \rho_i) = \frac{p(X = x_j | \rho = \rho_i, X = x_i)}{p(X = x_j)}.
\]
and
\[
p(X = x_i, \rho = \rho_i, X = x_j) = \frac{p(X = x_i, \rho = \rho_i, X = x_j)}{2^k}.
\]
can be expressed

\[ C_{X,\rho X} = \sum_{i=1}^{2^k} \sum_{j=1}^{2^k} \sum_{t=1}^{\Gamma} p(\hat{X} = x_{ij}, \rho = \rho_T | X = x_j) \]

\[ \cdot \log \left( \frac{p(\hat{X} = x_{ij}, \rho = \rho_T | X = x_j)}{p(\hat{X} = x_j, \rho = \rho_T)} \right). \] (12)

The following lemma can then be used to simplify (12).

**Lemma 6:**

\[ p(\hat{X} = x_{ij}, \rho = \rho_T | X = x_j) = p(\hat{X} = x_{ij}, \rho = \rho_T | X = 0) \]

where \( x_t = x_i + x_j \).

**Proof:**

\[ p(\hat{X} = x_{ij}, \rho = \rho_T | X = x_j) = \sum_{y \in c_\tau \text{ and distribution yielding } \rho_T} p(Y = y | X = x_j) \]

\[ = \sum_{e \in \text{coset with coset weight distribution yielding } \rho_T} p_{Y|\hat{X}}(e \oplus x_\tau | x_{ij}) \]

\[ \cdot (1 - p)^{n-\text{wt}_1(e \oplus x_\tau)} \]

Substituting \( x_t = x_i + x_j \) gives

\[ p(\hat{X} = x_{ij}, \rho = \rho_T | X = x_j) = \sum_{e \in \text{coset with coset weight distribution yielding } \rho_T} p_{Y|\hat{X}}(e \oplus x_\tau) \]

\[ \cdot (1 - p)^{n-\text{wt}_1(e \oplus x_\tau)} \]

\[ = p(\hat{X} = x_{ij}, \rho = \rho_T | X = 0). \] (13)

Theorem 8 then allows the computation of \( C_{X,\rho X} \).

**Theorem 8:**

\[ C_{X,\rho X} = 2^k \sum_{j=1}^{2^k} \sum_{t=1}^{\Gamma} p(\hat{X} = x_{ij}, \rho = \rho_T | X = 0) \]

\[ \cdot \log \left( \frac{p(\hat{X} = x_{ij}, \rho = \rho_T | X = 0)}{p(\hat{X} = x_j, \rho = \rho_T)} \right). \]

**Proof:** Follows from the use of Lemma 6 to simplify (12).

\[ p(\hat{X} = x_{ij}, \rho = \rho_T | X = 0) \] can be calculated from (13) if the Hamming weight of the \( n \)-tuples in each column \( c_\tau \) and in cosets yielding \( \rho_T \) can be computed. This can be determined using Tables IX, XII, and XIII for \( H_8 \) and \( G_{24} \). For these codes, it is found that each different coset weight distribution yields a different \( \rho_T \). Thus \( \Gamma = D_R \) for these codes.

\[ p(\hat{X} = x_{ij}, \rho = \rho_T) \] is given by

\[ = \sum_{e \in \text{coset with coset weight distribution yielding } \rho_T} p(Y = e) \]

\[ = p(\hat{X} = 0, \rho = \rho_T). \]

Thus \( p(\hat{X} = x_{ij}, \rho = \rho_T) \) is independent of \( x_j \) and can be computed from the weight distribution of the cosets yielding \( \rho_T \). This can also be determined using Tables IX, XII, and XIII for \( H_8 \) and \( G_{24} \).

**C. \( m \) Most Likely Codewords**

Next consider the decoder that for a given \( Y \) outputs the \( m \) most likely codewords denoted by \( \{\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m\} \).

**Lemma 7:** The composite capacity of the code, \( BSC \), and decoder outputting the \( m \) most likely codewords is nondecreasing as a function of \( m \).

**Proof:** Using the chain rule for mutual information [29], we can write

\[ I(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m; X) = \sum_{i=1}^m I(\hat{X}_i; X | \hat{X}_1, \cdots, \hat{X}_{i-1}) \]

\[ = I(\hat{X}_1; X | \hat{X}_1, \cdots, \hat{X}_{m-1}, X) \]

\[ + I(\hat{X}_m; X | \hat{X}_1, \cdots, \hat{X}_{m-1}, X) \]

\[ \geq I(\hat{X}_1; X | \hat{X}_1, \cdots, \hat{X}_{m-1}, X). \]

For the decoder outputting the \( m \) most likely codewords, the composite capacity between the encoder output and the decoder output is given by (since the BSS is the capacity achieving source distribution)

\[ C_{X_1, X_2, \cdots, X_m} = \sum_{x_1, x_2, \cdots, x_m} p(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m | X) \]

\[ \cdot \log \left( \frac{p(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m | X)}{p(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m)} \right). \] (14)

Equation (14) can be simplified noting that

\[ p(\hat{X}_1 = x_{\hat{1}}, \hat{X}_2 = x_{\hat{2}}, \cdots, \hat{X}_m = x_{\hat{m}} | X = x_j) \]

\[ = p(\hat{X}_1 = x_{\hat{1}}, \hat{X}_2 = x_{\hat{2}}, \cdots, \hat{X}_m = x_{\hat{m}} | X = 0) \]

where \( x_t = x_i + x_j \). Thus (14) becomes

\[ C_{X_1, X_2, \cdots, X_m} = \sum_{x_1, x_2, \cdots, x_m} p(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m | X = 0) \]

\[ \cdot \log \left( \frac{p(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m | X = 0)}{p(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_m)} \right). \]

For \( H_8 \), a simple computer program can be written to evaluate \( C_{X_1, X_2, \cdots, X_m} \) for \( m \) up to four. The authors have found that only the calculation of \( C_{X_1, X_2} \) is computationally tractable for \( G_{24} \).

**VI. EXAMPLES**

This section applies the previous results to specific codes, e.g., \( H_8, G_{24} \), and the repetition code. The first column in Table IV shows, for \( H_8 \), the additional capacity loss in bits
Fig. 7. Capacity loss for (8, 4) extended Hamming code and various decoders applied to BSC.

per codeword produced by an ML decoder, i.e., $C_{YX} - C_{XY}$, for several crossover probabilities. The second column shows how much of this loss can be made up by using an optimum erasure decoder. An exhaustive comparison of $C_{QX} - C_{XY}$ for all potential $\Psi$ indicates that the optimum erasure decoding rule for $H_8$ is $\Psi = \{r_{10}, r_{11}, \ldots, r_{16}\}$ (i.e., $\Psi$ includes all rows with coset leader of weight two or $\tau = 1$). For small $p$, erasure decoding can eliminate nearly all the loss introduced by an ML decoder. For large $p$, erasure decoding is still effective although substantial residual loss remains. In all cases, $C_{QX} - C_{XY}$ is small for $H_8$.

For $G_{24}$, an exhaustive comparison of $C_{QX} - C_{XY}$ for all $\Psi$ is not practical. Appendix II, however, shows that the optimum choice of $\Psi$ is

$$\Psi_{\text{optimum}} = \begin{cases} \{r_{2326}, r_{2327}, \ldots, r_{4006}\}, & p \leq 0.04 \\ \text{subset of } \{r_{2326}, r_{2327}, \ldots, r_{4006}\}, & 0.04 < p < 0.09 \\ \{\}, & 0.09 \leq p \leq 0.24 \\ \{\}, & 0.24 < p \end{cases} \tag{15}$$

where $\{r_{2326}, r_{2327}, \ldots, r_{4006}\}$ is the set of rows with weight four coset leader. Table V shows $C_{YX} - C_{XY}$ and $C_{QX} - C_{XY}$ for $G_{24}$ and $\Psi = \{r_{2326}, r_{2327}, \ldots, r_{4006}\}$. The improvement in capacity over an ML decoder is much less for the bigger code than it was for $H_8$, and, as shown in Appendix II, no gains are possible for $0.09 \leq p \leq 0.24$. Figs. 7 and 8 compare the composite capacity of an encoder and BSC in isolation to the composite capacity of an encoder, BSC, and various decoders for the Hamming and Golay codes. The upper plot in Fig. 7 shows the composite capacity for the channel consisting of the $H_8$ encoder and BSC in isolation and is, therefore, an upper bound to the composite capacity of any system which includes a decoder. The lower curves in Fig. 7 show the composite capacity of the $H_8$ encoder, BSC, and various decoders. A decoder outputting the four most likely codewords provides the greatest composite capacity followed by: a decoder outputting the three most likely codewords, a decoder outputting the ML estimate and the probability that this estimate is correct, and a decoder outputting the two most likely codewords. All these decoders provide a larger composite capacity than is provided by an ML decoder, the bottom curve in Fig. 7. The utility of any of these decoders will depend on the availability of generalized minimum-distance decoding algorithms for an outer code. The relative position of the curves in Fig. 7 and the equivalent curves for other candidate inner codes can also suggest what types of side information are most useful to incorporate in the generalized minimum-distance decoding procedure.

Table VI presents the reduction in capacity loss provided by using one of the decoders generating information beyond the ML estimate. Table VI demonstrates that some of these decoding techniques regain a large fraction of the capacity lost by ML decoding. However, the total gains which can be achieved by generalized minimum-distance decoding are surprisingly