AAECC **9**, 521–546 (1999)

AAECC
Applicable Algebra in Engineering, Communication and Computing

The BCJR Trellis and Trellis Decoders for Some BCH Codes*

Xiao-an Wang[1], Stephen B. Wicker[2]

[1] Lucent Technologies, Allentown, PA 18103, USA
[2] School of Electrical Engineering, Cornell University, Ithaca, NY 14853, USA

Received: February 11, 1998; revised version: March 10, 1999

Abstract. This paper provides an overview of trellis descriptions for block codes. The design and implementation of efficient trellis decoders for the (32, 26) and the (32, 21) BCH codes is then considered in some detail. Minimum edge and vertex counts for the (32, 26) code are derived, and then generalized for arbitrary extended Hamming codes. A subcode analysis technique is used to tighten the lower bound on edge complexity for the (32, 21) code. A trellis is then found that satisfies the optimized lower bound with equality.

Keywords: Algebraic block codes, Soft decision decoding, Trellis decoding

1 Introduction

The word lengths for paging and mobile messaging systems generally dictate the use of short block codes. For example, POCSAG (a ubiquitous paging standard in the 1980's and early 1990's) and Motorola's FLEX protocol (the de-facto standard in high speed paging in North America, South America, and Southeast Asia) both use the (32, 21) BCH code. This same code is used on the forward channel of Motorola's ReFLEX protocol – a FLEX derivative that provides two-way service and extended capacity through frequency reuse and time sharing [5]. The European Radio Messaging System (ERMES) uses a (30, 18) code that is related to the (32, 21) code. The decoders used in these systems reside in small mobile units, and are thus severely constrained by power and weight limitations. Simple algebraic, hard decision decoders (HDD's) are generally used.

* This work was funded by National Science Foundation Grant Number NCR-9216686.

In this paper, we explore the design and implementation of trellis-based soft decision decoders (SDD's) for the (32, 21) and (32, 26) BCH codes. SDD's are particularly useful in a fading environment, as they can incorporate channel attentuation information into the decoding process [25]. We begin with the (32, 26) BCH code, which is a member of the general family of extended Hamming codes. Minimum-edge and vertex counts are derived for all extended Hamming codes. A simple algorithm is then presented for constructing trellises that satisfy the minimum-edge and vertex counts.

A subcode-based analysis technique is used to find tight lower bounds on the dimensions of the past and future subcodes of the (32, 21) BCH code. This technique begins by identifying a class of "critical" (12, 2, 6) subcodes in the (32, 21) code. A mapping is then defined that relates these subcodes to minimum weight nonzero words within the dual of the (32, 21) code. Knowledge of the dual code weight distribution and subcode profile is then used to create restrictions on the past subcode dimension profile for the (32, 21) code. This leads to a lower bound on edge count that is shown to be optimal through the identification of a trellis that satisfies the bound with equality.

2 Trellis Decoders for Block Codes

This section provides a review of the various techniques that have been developed for designing and analyzing trellis decoders for block codes. Section 2.1 illustrates a method for constructing the trellis of a block code given its parity check matrix. Section 2.2 discusses the various complexity measures that have been proposed for evaluating the resulting trellis.

Two codes are *equivalent* if one can be described as a permutation of the other. On memoryless channels, equivalent codes provide the same coding gain. On the other hand, the minimum-edge trellises for equivalent codes can have substantially different complexity. This has spurred much effort in finding the permutation of a code that results in the "best" trellis, i.e. one of minimal complexity with respect to a certain complexity measure. The identification of the best, or simply a good, permutation of a code requires a detailed exploration of the relationship between the code and its dual. Section 2.3 presents several well-known duality identities necessary for that purpose.

Section 2.4 reviews the Mattson-Solomon (MS) polynomial and some of its properties. The MS polynomial is a useful tool for the study of cyclic codes in general, and the BCH codes in particular. We will make use of the MS polynomial when we search for the minimal complexity trellis for the (32, 21) BCH code.

2.1 Trellises of Linear Binary Block Codes

We now review the trellis construction for binary linear block codes that was first described by Bahl *et al.* in 1974 [2]. Let $H = [\mathbf{h}_1, \ldots, \mathbf{h}_n]$ be a parity check matrix for an (n, k) code C, where $\mathbf{h}_i$, $i = 1, \ldots, n$ are the length $(n-k)$ column vectors of H. The vertices of the trellis to be constructed are a subset of a $2^{(n-k)} \times n$ grid. Identify each of the 2^{n-k} grid positions at depth i with an $(n-k)$-tuple $\mathbf{s}$, and let $\mathbf{0}$ at depth 0 and $\mathbf{0}$ at depth n be the source and sink, respectively. A path from the source to the sink is then completely specified by a state (vertex) sequence $(\mathbf{0}, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1}, \mathbf{0})$. Let $\mathbf{c} = (c_1, \ldots, c_n)$ be a code word, and define the state sequence $\mathbf{s}(\mathbf{c}) = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_n)$ as follows:

$$\begin{aligned} \mathbf{s}_0 &= \mathbf{0} \\ \mathbf{s}_{i+1} &= \mathbf{s}_i + c_{i+1}\mathbf{h}_{i+1} \quad i = 1, \ldots, n \ , \end{aligned} \tag{1}$$

where the coordinate addition takes place in $GF(2)$. By the definition of a parity check matrix (H), we have $\mathbf{s}_n = \mathbf{0}$. Each code word can be mapped onto a unique path from the source to the sink by (1). The set of all paths $\{\mathbf{s}(\mathbf{c}) : \mathbf{c} \in C\}$ form a trellis for the code C. Such a trellis for a (7, 4) Hamming code is shown in Figure 1.

The trellis developed above and exemplified by Figure 1 is commonly referred to here as the BCJR trellis [18]. It is possible to construct numerous trellises for a given code C. Clearly trellises with low edge or vertex count for a code are more useful in actual implementations. In an appendix of [6], Forney proposed a trellis for binary codes which Muder later showed to have a minimal number of vertices at each depth [19]. Further, Muder showed that any minimal trellis is isomorphic to the Forney trellis. In [18], McEliece shows that the BCJR trellis also minimizes the vertex count as well as the edge count at each depth, and therefore the two trellises are identical up to an isomorphism. For these reasons, the minimal trellis or the BCJR trellis of a code C will be referred to as "the trellis of the code C".

The vertex set V_i consists of all states or vertices $\mathbf{s}_i$ in (1). All pairs $(\mathbf{s}_i, \mathbf{s}_{i+1})$ form the edge set $E_{i,i+1}$. The vertex set V and the edge set E of the trellis are the unions of V_i and the unions of E_i, respectively. We now give some characterizations of the V_i and $E_{i,i+1}$. We follow the notation of McEliece [18], from which the following three Theorems have been taken.

Theorem 1 *V_i and $E_{i,i+1}$ are vector spaces over $GF(2)$.*

Definition. *The i^{th} past subcode P_i, the i^{th} past projection P^i, the i^{th} future subcode F_i and the i^{th} future projection F^i of C are defined as*

$$\begin{aligned} P_i &= \{\mathbf{c} \in C : c_{i+1} = \cdots = c_n = 0\} \\ P^i &= \{(c_1, \ldots, c_i) : \mathbf{c} \in C\} \\ F_i &= \{\mathbf{c} \in C : c_1 = \cdots = c_i = 0\} \\ F^i &= \{(c_{i+1}, \ldots, c_n) : \mathbf{c} \in C\} \end{aligned} \tag{2}$$

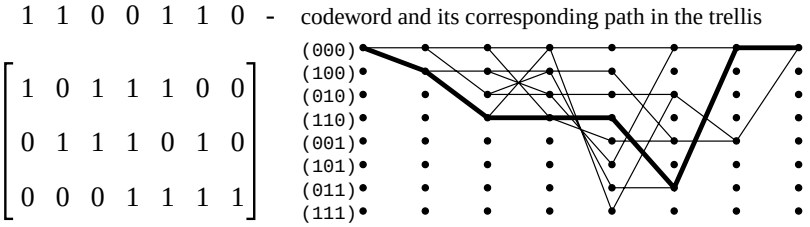

Fig. 1. The parity check matrix H and the trellis for the Hamming (7, 4) code

Their dimensions are denoted by p_i, p^i, f_i and f^i respectively. By convention we take

$$P_n = P^n = F_0 = F^0 = C, \quad P_0 = P^0 = F_n = F^n = (0)$$

and thus

$$p_n = p^n = f_0 = f^0 = k, \quad p_0 = p^0 = f_n = f^n = 0$$

Theorem 2

$$k = p^i + f_i = p_i + f^i \quad . \tag{3}$$

Let $s_i = \dim V_i$ and $b_i = \dim E_{i,i+1}$, we have the following

Theorem 3

$$\begin{aligned} s_i &= k - p_i - f_i \\ b_i &= k - p_i - f_{i+1} \quad . \end{aligned} \tag{4}$$

Given an (n, k) code C, the trellis can be constructed by applying (1) over all code words. This construction is not practical if C has a large dimension. More efficient algorithms for constructing the trellis have been presented in [17, 23].

2.2 Complexity of the Trellis and the Effect of Permutation

The complexity of the trellis of an (n, k) code C can be measured in terms of one or more the following:

- **State complexity:** $s(C) = \max\{s_0, s_1, \ldots, s_n\}$.
- **Branch complexity:** $b(C) = \max\{b_1, b_2, \ldots, b_n\}$.
- **Edge complexity:** $E(C) = |E| = \sum_{i=1}^{n} E_{i-1,i} = \sum_{i=1}^{n} 2^{b_i}$.

A complexity measure similar to the edge complexity was also proposed in [22]. It has been shown that $s(C)$, $b(C)$, and $\log E(C)$ are asymptotically the same in that the ratio of any two of them approaches unity as the code length n increases [10].

Wolf [26] gave an upper bound for the state complexity $s(C)$: $s(C) \leq \min\{k, n-k\}$. For most of the codes in their original forms, this upper bound is

often an equality. Since the complexity of the trellis depends on the ordering of the code coordinates [6, 16], tighter bounds can be obtained by permutating the coordinates of the code. By doing so we identify the code C with all its equivalent codes. Improved bounds on $s(C)$ for some BCH codes are given in [8, 22]. In some cases, the lower bounds and the upper bounds are identical, indicating exact values for $s(C)$. Lower bounds on $s(C)$ for some general (n, k, d) codes with certain d are given in [22], and lower bounds or exact values for $s(C)$ for some codes with certain weight distributions are given in [27].

Let $k(i; C)$ be the maximum dimension of any length-i subcode of C. The sequence $\{k(0; C),\ k(1; C), \ldots, k(n; C)\}$ is called the dimension-length profile (DLP) of the code C [7]. The DLP often leads to good lower bounds on $s(C)$ [7, 10]. However, the computation of the DLP is difficult for most codes. Recently, Lafourcade-Jumenbo and Vardy developed a trellis partitioning technique that further improves the lower bounds for a large number of codes [10]. Their method also applies to nonlinear codes.

On the practical side, the edge complexity is a more accurate measure of the trellis complexity with respect to the number of computations. For a given code C, among all the trellises of its equivalent codes, the trellis with the minimum number of edges is referred to as the *minimum-edge* trellis. Similarly, the trellis with the minimum number of vertices is referred to as the *minimum-vertex* trellis. It can be easily established that

$$|E| \leq |V| - 1 \leq 2|E| \ ,$$

which shows that $|V|$ is as good a measure of complexity as $|E|$. A minimum-edge trellis is not necessarily a minimum-vertex trellis and vice versa. Sharper lower bounds on $E(C)$ for many binary codes are derived in [10] through the use of trellis partitioning and nonlinear integer programming. A heuristic algorithm for constructing a trellis with low edge complexity was proposed in [4].

It was shown by Lafourcade-Jumenbo and Vardy [10, 11] that asymptotically good codes have infinite trellis complexity. This means that even if the trellis complexity can be reduced by permutation, it eventually grows at an exponential rate as the code length increases. However, for many practical codes, trellises are still an efficient means for decoding. Luna, Fontaine, and Wicker have developed an iterative technique that provides ML and near-ML trellis decoding with far less complexity than in the straightforward application of Viterbi decoding to the code trellis [12]. Aguado and Farrell developed a hybrid stack decoding algorithm for block codes in [1], which can handle much higher trellis complexity than the Viterbi algorithm (VA) does.

The choice of a complexity measure depends in part on what one hopes to do with the trellis decoder under consideration. McEliece pointed out in [18] that the VA requires $O(|E|)$ arithmetic operations when applied to the trellis. It follows that a software-based decoder will benefit from a minimization of edge complexity. Komura, Oka, Fujiwara, Onoye, Kasami, and Lin, however, have shown that the VA is not an efficient approach for IC-based trellis decoders

[9]. Instead the regularity of the trellis structure was employed, allowing for a straightforward application of pipelining and parallel processing. Recursive trellis structures have been developed for Reed-Muller hybrid-ARQ protocols by Martin, Honary, Markarian, and Wicker in [15] (see also [14]). Komura *et al*. used the structure of a (64, 35) subcode of the (64, 42) Reed-Muller code to develop an extremely fast inner codec for a concatenated system intended for near-earth satellites [9]. Komura *et al*'s codec is intended to run at several hundred megabits per second – a speed that cannot be achieved by a hardware implementation that uses Viterbi decoding and a minimum-edge trellis.

Having said the above, we will pursue minimum-edge complexity in this paper. The minimum-edge complexity of a code provides a convenient gauge of the complexity of SDD's for a block code, and is certainly more concise than the admittedly fuzzy idea of "trellis regularity".

2.3 Duality Properties of Linear Codes

In this section we present a review of various relationships that can be drawn between a linear block code and its dual code. Let C be an (n, k) linear block code, and let $C^\perp$ denote its dual. The past and future subcodes of $C^\perp$ are denoted by $P_i^\perp$ and $F_i^\perp$, and the past and future projections are denoted by $P^{\perp i}$ and $F^{\perp i}$. The results in this section are from Forney [7].

Since $C^\perp$ is an $(n, n-k)$ code, applying (3) in Theorem 2 of Section 2.1, we obtain

$$n - k = p^{\perp i} + f_i^\perp = p_i^\perp + f^{\perp i} \quad . \tag{5}$$

Theorem 4 *P_i and $P^{\perp i}$ are dual codes, and P^i and $P_i^\perp$ are dual codes.*

Note that in Theorem 4 we treat the subcodes P_i and $P_i^\perp$ as of length i, ignoring their last $n - i$ bits that are all zeros.

Proof. Let $\mathbf{c}_i \in P_i$. By the definitions of P_i and $P^{\perp i}$, we have $\mathbf{c}_i \mathbf{d}_i^T = 0$ for all $\mathbf{d}_i \in P^{\perp i}$. On the other hand, if for some $\mathbf{c}_i'$, $\mathbf{c}_i' \mathbf{d}_i^T = 0$ for all $\mathbf{d}_i \in P^{\perp i}$, then the vector $\mathbf{c}'$ defined by

$$\mathbf{c}' = (\mathbf{c}_i', \overbrace{0, \ldots, 0}^{n-i})$$

satisfies $\mathbf{c}'\mathbf{d}^T = 0$ for all $\mathbf{d} \in C^\perp$. Thus $\mathbf{c}' \in C$ and hence $\mathbf{c}_i' \in P_i$. Now we have proved that P_i consists of all the i-tuples orthogonal to $P^{\perp i}$, i.e., P_i and $P^{\perp i}$ are dual codes. The second statement of the theorem can be proved along the same lines. Q.E.D.

By Theorem 4 the dimensions of P_i and $P^{\perp i}$ sum to their length i, as do the dimensions of $P_i^\perp$ and P^i. Thus we have

Corollary 5

$$i = p_i + p^{\perp i} = p_i^{\perp} + p^i \quad . \tag{6}$$

Similarly we can prove the following:

Theorem 6 *F_i and $F^{\perp i}$ are dual codes, and F^i and $F_i^{\perp}$ are dual codes.*

Corollary 7

$$n - i = f_i + f^{\perp i} = f_i^{\perp} + f^i \quad . \tag{7}$$

Other duality identities can be derived from the identities developed in this section and Section 2.1. For example, combining (5), (6) and (7) we get

$$k - p_i - f_i = n - k - p_i^{\perp} - f_i^{\perp} \quad . \tag{8}$$

Let $s_i^{\perp}$ denote the dimension of the vertex set $V_i^{\perp}$ in the trellis of $C^{\perp}$, By (4) and (8) we have

$$s_i = s_i^{\perp} \quad .$$

The above duality identities will be used extensively in the search for minimal complexity trellises.

2.4 The Mattson-Solomon Polynomial

A brief review of the Mattson-Solomon polynomial is presented in this section. All results in this section are from [13].

Let $F = GF(q)$, where q is the power of a prime p. Let n be an integer relatively prime to p. A vector of length n in F, $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ can be represented by a polynomial in $F[x]$, $a(x) = a_0 + a_1 x + \cdots a_{n-1} x^{n-1}$. Let m be the smallest number such that $n|(q^m - 1)$. Let $\mathscr{F} = GF(q^m)$ and let $\alpha \in \mathscr{F}$ be a primitive nth root of unity.

Definition. *The Mattson-Solomon (MS) polynomial* associated with $a(x)$ is the following polynomial in $\mathscr{F}[z]$:

$$A(z) = \sum_{j=1}^{n} A_j z^{n-j} \quad , \tag{9}$$

where

$$A_j = a(\alpha^j) = \sum_{i=0}^{n-1} a_i \alpha^{ij}$$

It is straightforward to show that the MS polynomial has the linearity property:

Theorem 8 *If* $c(x) = a(x) + b(x)$, *then* $C(z) = A(z) + B(z)$.

Lemma 9

$$\sum_{i=0}^{n-1} \alpha^i = 0 \tag{10}$$

Proof. $\alpha \neq 1$ by definition. Thus

$$\sum_{i=0}^{n-1} \alpha^i = (1 - \alpha^n)/(1 - \alpha) = 0$$

Q.E.D.

Let $g(y)$ be any polynomial, and denote by $[g(y)]_n$ the remainder when $g(y)$ is divided by $y^n - 1$. Define the *componentwise product* $g(y) * h(y)$ of two polynomials

$$g(y) = \sum_{i=0}^{n-1} g_i y^i \quad \text{and} \quad h(y) = \sum_{i=0}^{n-1} h_i y^i$$

to be

$$g(y) * h(y) = \sum_{i=0}^{n-1} g_i h_i y^i$$

The following two theorems can be proved by direct substitution and use of (10).

Theorem 10 (Inversion formula)

$$a(x) = \frac{1}{n} \sum_{i=0}^{n-1} A(\alpha^i) x^i \quad . \tag{11}$$

Theorem 11 (i) $c(z) = [a(x)b(x)]_n$ *if and only if* $C(z) = A(z) * B(z)$. (ii) $c(x) = a(x) * b(x)$ *if and only if* $C(z) = \frac{1}{n}[A(z)B(z)]_n$.

Note that the denominator n in Theorems 10 and 11 should be reduced *modulo* p. For the most popular case of $p = 2$, $1/n = 1$ *modulo* p (n must be odd if it is to be relatively prime to $p = 2$).

$A(z)$ is also referred to as the *discrete Fourier transform* of $a(x)$. $a(x)$ and $A(z)$ are transform pairs, as one can be recovered from the other by (9) and (11).

For any polynomial $g(y)$, its *effective degree* $d_{\text{eff}}(g)$ is defined to be $\deg(g) - i$, where i is the smallest number such that the coefficient g_i of y^i is nonzero. Obviously $d_{\text{eff}}(g) \leq \deg(g)$. Consider a transform pair $\{a(x), A(z)\}$, where both are of degree $n - 1$. Let $s = d_{\text{eff}}(A)$. Then $A(z)$ has s nonzero roots in

$\mathcal{F}$, which means $A(z)$ has no more than s roots in $\{1, \alpha, \ldots, \alpha^{n-1}\}$. From the inversion formula (11) we see that $a(x)$ has at most s zero coefficients, or it has at least $n - s$ nonzero coefficients. We summarize the above discussion in the following theorem.

Theorem 12 *If* **a** *has MS polynomial* $A(z)$*, then the weight of* **a** *satisfies*

$$w(\mathbf{a}) \geq n - d_{\text{eff}}(A) \quad . \tag{12}$$

(12) is a slightly improved version of Corollary 27 in Chapter 8 of [13].

3 Trellises for the (32, 26) and (32, 21) BCH Codes

This section focuses on the construction and implementation of trellis decoders for the $(32, 26, 4)$ and the $(32, 21, 6)$ extended BCH codes. In Section 3.1 we construct the minimum-edge trellis for the (32, 26) code. We note that this code can be viewed within the context of the extended Hamming codes. Bounds on the number of edges and vertices for trellises for the entire class of extended Hamming codes are derived. A simple construction algorithm for minimum-edge and minimum-vertex trellises is then presented. Section 3.2 begins with the construction of a tight lower bound on the edge complexity of the minimum-edge trellis for the $(32, 21, 6)$ code. A subcode analysis technique is used to substantially improve the previous bound. A trellis is then described that satisfies the bound with equality.

3.1 Minimum Edge Trellises for Hamming Codes and Extended Hamming Codes

We start with the (31, 26, 3) Hamming code. A parity check matrix can be constructed by selecting as columns all 31 distinct, nonzero binary 5-tuples. The extended Hamming code is obtained by adding a parity bit to each code word. The following parity check matrix is a typical result. Since we are not distinguishing among equivalent codes, we will refer to this as "the" parity check matrix for the (32, 26, 4) extended Hamming code.

$$H = H_{(32,26)} = \begin{bmatrix} 00000000000000001111111111111111 \\ 00000000111111110000000011111111 \\ 00001111000011110000111100001111 \\ 00110011001100110011001100110011 \\ 01010101010101010101010101010101 \\ 11111111111111111111111111111111 \end{bmatrix} \tag{13}$$

One objective in the search for a minimum-edge trellis for an (n, k) code is the minimization of the number of independent rows of H_i and $\bar{H}_{n-i}$ for each

$i = 1, \ldots, n/2$, where H_i and $\bar{H}_{n-i}$ are the matrices consisting of the first i columns and the last $n-i$ columns of the parity check matrix H, respectively. The rational is as follows. The dimension s_i of the vertex space V_i is

$$s_i = \text{rank}(H_i G_i^T) \leq \min\{\text{rank}(G_i), \text{rank}(H_i)\} \leq \text{rank}(H_i) = p^{\perp i} \quad , \qquad (14)$$

where G_i is the matrix formed by the first i columns of the generator matrix G. The first equality in (14) follows from the fact that each vertex in V_i is a linear combination of the columns in $H_i G_i^T$, as indicated in (1). If $p^{\perp i}$ is made small, then s_i will be small, thus reducing the number of vertices and edges. A similar argument can be made for $f^{\perp i} = \text{rank}(\bar{H}_{n-i})$. For the extended codes we have the following lemma.

Lemma 12 *Let H be a parity check matrix for an extended Hamming code, where one row of H is the all 1's vector. Then for $i = 0, 1, \ldots, n-1$,*

$$p^{\perp i} \geq \lceil \log_2 i \rceil + 1 \qquad (15)$$

$$f^{\perp i} \geq \lceil \log_2 (n-i) \rceil + 1 \quad . \qquad (16)$$

Proof. Let $\mathbf{u}_1, \ldots, \mathbf{u}_{p^{\perp i}}$ be a basis for $P^{\perp i}$ in which $\mathbf{u}_1$ is the all 1's vector. Since the basis consists of linearly independent rows, one of which is constant-valued, any column in H_i is completely determined by its $p^{\perp i} - 1$ bits in rows $\mathbf{u}_2, \ldots, \mathbf{u}_{p^{\perp i}}$. Since all of the columns in H_i are distinct, the number of bits determining column i has to be no less than $\lceil \log_2 i \rceil$, or $p^{\perp i} - 1 \geq \lceil \log_2 i \rceil$. Equation (16) can be proved similarly. Q.E.D.

An examination of (13) reveals that the equality is satisfied in (15) for the left half of H. For the right half of H, we observe that it is identical to the left half except for the first row (by adding the last row to the first row, we can exchange the left and the right halves). The permutation (16, 31), (17, 30), ..., (23, 24) then mirrors the left and the right halves. Adding the first row to the last row, we get the permuted H in its minimal-span form (for the properties of the minimal-span form, see [18]):

$$H^P_{(32,26)} = \begin{bmatrix} 00000000000000001111111111111111 \\ 00000000111111111111111100000000 \\ 00001111000011111111000011110000 \\ 00110011001100111100110011001100 \\ 01010101010101011010101010101010 \\ 11111111111111110000000000000000 \end{bmatrix}$$

It can be seen that both equations (15) and (16) are satisfied, and hence $H^P_{(32,26)}$ minimizes $p^{\perp i}$ and $f^{\perp i}$ for every i. The dimensions p_i and f_i can be read off directly from the corresponding generator matrix $G^P_{(32,26)}$ in its minimal-span

form. We list half of the dimension profiles along with $s_i = 26 - p_i - f_i$ and $b_i = 26 - p_i - f_{i+1}$. The other half follows by symmetry.

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_i :	0	0	0	0	1	1	2	3	4	4	5	6	7	8	9	10	11
f_i :	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	11
s_i :	0	1	2	3	3	4	4	4	4	5	5	5	5	5	5	5	4
b_i :	1	2	3	4	4	5	5	5	5	6	6	6	6	6	6	5	5

The total number of edges is $|E| = 2\sum_{i=0}^{15} 2^{b_i} = 1180$. This equals the lower bound on the number of edges for any $(32, 26)$ code, as established in [4]. The trellis obtained here is a minimum-edge trellis. Dolinar *et al.* found a minimum-edge trellis for the extended $(32, 26, 4)$ BCH code using a heuristic algorithm [4]. The minimum-edge trellis of any extended Hamming code can be constructed in the manner used above for the $(32, 26, 4)$ code. First, a lower bound is derived for s_i.

Lemma 13 *For the* $(2^m, 2^m - m - 1, 4)$ *extended Hamming code,*

$$s_i = \begin{cases} p^{\perp i}, & i = 0, 1, \ldots, 2^{m-1} - 1 \\ f^{\perp i}, & i = 2^{m-1} + 1, \ldots, 2^m \end{cases} . \tag{17}$$

Proof. We have $s_i = s_i^{\perp} = k - f_i^{\perp} - p_i^{\perp} = p^{\perp i} - p_i^{\perp}$. Since the weight distribution of the dual code is $\{B_0 = B_{2^m} = 1, B_{2^{m-1}} = 2^{m+1} - 2\}$, it follows that $p_i^{\perp} = 0$ for $i = 0, 1, \ldots, 2^{m-1} - 1$. The second result can be proved in the same way. Q.E.D.

Corollary 14

$$s_i \geq \lceil \log_2 i \rceil + 1 \quad i = 1, \ldots, 2^{m-1} - 1 \tag{18}$$

$$s_i \geq \lceil \log_2 (2^m - i) \rceil + 1 \quad i = 2^{m-1} + 1, \ldots, 2^m - 1 \tag{19}$$

$$s_{2^{m-1}} \geq m - 1 \ . \tag{20}$$

Proof. Equations (18) and (19) are direct consequences of Lemmas 12 and 13. Equation (20) is obtained by observing that $s_{2^{m-1}} = p^{\perp 2^{m-1}} - p_{2^{m-1}}^{\perp} \geq m - 1$. Q.E.D.

The parity check matrix of the $(2^m - 1, 2^m - m - 1, 3)$ Hamming code has as columns the set of all distinct, nonzero, binary m-tuples. By introducing an m-bit zero column and adding an all-one row, we get the parity check matrix H_m for the extended code. The following algorithm gives the permuted parity check matrix H_m^P:

1. **Sorting:** Identify each column of H_m with an m-bit integer. Arrange the columns in ascending order.

2. **Column Exchange:** Exchange columns $2^{m-1}+i$ and 2^m-1-i of the rearranged matrix for $i=0,1,\ldots,2^{m-1}-1$.
3. **Minimal-Span Form** (optional): Add the top row to the bottom row.

For $m=5$, the above algorithm gives $H_5^P = H_{(32,26)}^P$. By reading off $s_i^\perp = s_i$ from H_m^P, it can be seen that all the equalities are satisfied in equations (18), (19) and (20). Thus the H_m^P has a minimum-vertex trellis. The next result shows that the trellis of H_m^P is also minimum-edge.

Theorem 15 *The trellis $T=(V,E)$ of the H_m^P is both minimum-vertex and minimum-edge. Further, the number of vertices and the number of edges are given by*

$$|V| = (2^{2m+1} - 9\cdot 2^{m-1} + 10)/3$$
$$|E| = (2^{2m+2} - 9\cdot 2^{m+1} + 20)/3$$

respectively.

Proof. We have seen that T is minimum-vertex. To see that T is also minimum-edge, we show that b_i is minimized for each i. For $0 \le i \le 2^{m-1}-1$, $b_i = k - p_i - f_{i+1} = s_i + 1 - (p_{i+1}^\perp - p_i^\perp) = s_i + 1 - p_{i+1}^\perp$. The only possible position for a nonzero $p_{i+1}^\perp$ is at $i = 2^{m-1}-1$, and indeed $p_{2^{m-1}} = 1$. Thus $p_{i+1}^\perp$ is maximized, which means that s_i is minimized, and so is b_i. For $2^{m-1} \le i \le 2^m - 1$, note that $b_i = s_{i+1} + 1 - (f_i^\perp - f_{i+1}^\perp)$ and apply the same argument for f_i.

To compute the number of vertices, note that the trellis T is symmetric, so one need only consider half of the trellis. Using Equations (18) and (20) with equalities, we have

$$\begin{aligned}|V| &= 2^{s_{2^{m-1}}} + 2\sum_{i=0}^{2^{m-1}-1} 2^{s_i} \\ &= 2^{m-1} + 2(1 + 2 + \sum_{l=2}^{2^{m-1}-1} 2^{l-2}2^l + (2^{m-2}-1)2^m \\ &= (2^{2m+1} - 9\cdot 2^{m-1} + 10)/3 \ .\end{aligned}$$

The expression for $|E|$ can be obtained by noting that $b_i = s_i + 1$ for $i = 0,1,\ldots,2^{2^{m-1}} - 2$, $b_{2^{m-1}-1} = s_{2^{m-1}-1}$, and thus $|E| = 2|V| - 2^m - 2^{m+1}$. Q.E.D.

3.2 The Minimum-edge Trellis for the (32, 21, 6) BCH Code

Let C' be the (31, 21, 5) BCH code, and let C be the extension of C' obtained by adding a parity bit. Their dual codes are denoted by $C'^\perp$ and $C^\perp$ respectively. Given $\mathbf{a} \in C$ and $\mathbf{b} \in C^\perp$, let $\mathbf{a}'$ and $\mathbf{b}'$ be the restrictions of $\mathbf{a}$ and $\mathbf{b}$ on C' and $C'^\perp$ respectively. Note that for $\mathbf{a} \in C$, either $\mathbf{a} = (\mathbf{a}', 0)$ or $\mathbf{a} = (\mathbf{a}', 1)$. For $\mathbf{b} \in C^\perp$, however, either $\mathbf{b} = (\mathbf{b}', 0)$ or $\mathbf{b} = \mathbf{1} + (\mathbf{c}', 0)$ for some $\mathbf{c}' \in C'^\perp$,

where **1** is the all-one code word of $C^{\perp}$. Let $\mathbf{1}'$ be the all-one vector obtained by removing the parity check bit from **1**.

The polynomial representations are used exclusively here for code words in C' and $C'^{\perp}$. For example, $a(x)$ is the polynomial of $\mathbf{a}' \in C'$, not $\mathbf{a} \in C$. When we speak of the zeros (nonzeros) of $\mathbf{a} \in C$, we actually mean the zeros (nonzeros) of $a(x)$. We use $m_i(x)$ to denote the minimal polynomial whose roots lie in conjugate class i, and $\mathscr{C}_i$ to denote the index set of conjugate class i (i.e. the associated cyclotomic coset). In our case, we have

$$\begin{aligned} m_0(x) &= x + 1 \\ m_1(x) &= x^5 + x^2 + 1 = x^5 m_{15}(1/x) \\ m_3(x) &= x^5 + x^4 + x^3 + x^2 + 1 = x^5 m_7(1/x) \\ m_5(x) &= x^5 + x^4 + x^3 + x + 1 = x^5 m_{11}(1/x) \ , \end{aligned}$$

and

$$\begin{aligned} \mathscr{C}_0 &= \{0\} \\ \mathscr{C}_1 &= \{1, 2, 4, 8, 16\} \qquad \mathscr{C}_3 = \{3, 6, 12, 24, 17\} \\ \mathscr{C}_5 &= \{5, 10, 20, 9, 18\} \quad \mathscr{C}_7 = \{7, 14, 28, 25, 19\} \\ \mathscr{C}_{11} &= \{11, 22, 13, 26, 21\} \ \mathscr{C}_{15} = \{15, 30, 29, 27, 23\} \ . \end{aligned}$$

We also use the notation such that $\mathscr{C}_j = \mathscr{C}_i$ if $j = (2^k i \bmod 31)$ for some k, e.g., $\mathscr{C}_{-7} = \mathscr{C}_{24} = \mathscr{C}_3$.

The generator polynomial used to represent C' is $g(x) = m_1(x)m_3(x) = x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1$. The only other polynomial that will generate a BCH code C'' with the same parameters is $m_7(x)m_{15}(x)$. However C' and C'' are equivalent; in fact, the code polynomials of one are the reciprocals of the code polynomials of the other. This justifies allusions to the (31, 21, 5) BCH code.

The weight distributions of C' and $C'^{\perp}$ have been found to be [13]

i :	0	5	6	7	8	9
W'_i :	1	186	806	2635	7905	18190
i :	10	11	12	13	14	15
W'_i :	44392	85560	142600	195300	251100	301971

and

i :	0	12	16	20
$W'^{\perp}_i$:	1	310	527	186

respectively, where $W'_{31-i} = W'_i$ for $i = 0, 1, \ldots, 15$. The weight distributions of C and $C^{\perp}$ follow immediately:

i :	0	6	8	10	12	14	16
W_i :	1	992	10540	60152	228160	446400	603942

and

i :	0	12	16	20	32
$W^{\perp}_i$:	1	496	1054	496	1

where $W_{32-i} = W_i$ for $i = 0, 1, \ldots, 15$.

To find a minimum-edge trellis for C, start with a lower bound developed in [4] that states that for an (n, k) code,

$$p_i \le p_i^* = \min\{K(i, d), i - (n-k) + K(n-i, d^\perp)\} \tag{21}$$

$$f_i \le f_i^* = \min\{K(n-i, d), k - i + K(i, d^\perp)\} \tag{22}$$

for any permutations of the code. In (21) and (22), d and $d^\perp$ are the minimum distances of the code in consideration and its dual respectively, and $K(i, d)$ is the largest possible dimension for a binary linear code of length i and minimum distance d. The exact value of $K(i, d)$ or a bound can be found in the tables of [3] for $n \le 127$. The bounds in (21) and (22) can be derived from

$$\begin{array}{l} p_i = i - (n-k) + f_i^\perp \\ f_i = k - i + p_i^\perp \end{array} , \tag{23}$$

which can be obtained through (3), and (5)–(7).

In the case at hand, $(n, k, d, d^\perp) = (32, 21, 6, 12)$. Since the bounds are symmetric, we list half profiles of the bounds along with the derived lower bounds s_i^* and b_i^* on s_i and b_i:

$$\begin{array}{rcccccccccccccccc} i: & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ p_i^*: & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 5 & 5 & 6 \\ f_i^*: & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 10 & 9 & 8 & 7 & 6 \\ s_i^*: & 0 & 1 & 2 & 3 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 8 & 8 & 8 & 8 & 9 & 9 \\ b_i^*: & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 7 & 8 & 8 & 9 & 8 & 9 & 9 & 9 & 10 & 10 \end{array} \tag{24}$$

The lower bound on the edge count is then given by

$$E^* = \sum_{i=0}^{15} 2^{b_i^*} = 8316 \tag{25}$$

The heuristic algorithm in [4] finds a trellis of 17,340 edges. We now show that the lower bound profile in (24) is not achievable, as is the lower bound on the edge count in (25). We begin by noting that

Lemma 16 *$C'^\perp \subset C'$ and hence $C^\perp \subset C$.*

Proof. The generator polynomial $h(x)$ of $C'^\perp$ has $g(x)$ as a factor: with $n = 31$ and $k = 21$ we have

$$\begin{aligned} h(x) &= x^k(x^{-n} + 1)/g(x^{-1}) \\ &= x^{21} m_5(x^{-1}) m_7(x^{-1}) m_{11}(x^{-1}) m_{15}(x^{-1}) m_0(x^{-1}) \\ &= m_{11}(x) m_3(x) m_5(x) m_1(x) m_0(x) \\ &= m_{11}(x) m_5(x) m_0(x) g(x) \end{aligned}$$

Q.E.D.

Theorem 17 *The profile bound in* (24) *is not achievable.*

Proof. Suppose that we have a permutation that achieves the profile in (24). Since $f_{12} = 10$, using $p_i^{\perp} = f_i - (k - i)$, we have $p_{12}^{\perp} = 10 - (21 - 12) = 1$, and $p_i^{\perp} = 0$, for $0 \le i \le 11$. Thus in $C^{\perp}$ (possibly permuted), there is a code word $\mathbf{c}_1$ of weight 12 with all of its 1's packed in the first 12 bits. By Lemma 1, $\mathbf{c}_1$ is also in C. But this would make $p_{12} = p_{11} + 1$, which contradicts $p_{11} = p_{12}$ in the assumption. Q.E.D.

An alternative proof of Theorem 17 uses the fact that $C^{\perp}$ contains the all-ones word $\mathbf{1}$. If the bound in (24) is achieved, then by symmetry, $C^{\perp}$ contains $\mathbf{c}_1$ as well as $\mathbf{c}_2$ of weight 12, with 1's packed in the last 12 bits. But then the word $\mathbf{1} - \mathbf{c}_1 - \mathbf{c}_2$ lies in $C^{\perp}$. This word has weight $32 - 12 - 12 = 8$, a contradiction of $d^{\perp} = 12$.

A tighter lower bound on the edge count than (25) will now be established. A trellis will then be described that satisfies the improved bound with equality.

The following is a subcode analysis that identifies a "critical" subcode, and then uses the containment of this subcode in past subcodes to derive bounds on the dimension of the past subcodes. These bounds on dimension are used to improve the profile bound in Equation (24).

Definition. Two code words $\mathbf{a}$ and $\mathbf{b}$ are said to be *nonoverlapping* if $w(\mathbf{a}+\mathbf{b}) = w(\mathbf{a}) + w(\mathbf{b})$.

Lemma 18 *Let* $\mathbf{a} = (a_0, a_1, \ldots, a_{31})$ *be a code word in* $C^{\perp}$ *of weight* 12. *Define* $C(\mathbf{a}) = \{(c_0, c_1, \ldots, c_{31}) \in C : (a_0c_0, a_1c_1, \ldots, a_{31}c_{31}) \in C\}$ ($C(\mathbf{a})$ *is said to be* generated *by* $\mathbf{a}$). *Then* $C(\mathbf{a})$ *is a subcode of* C *and contains two nonoverlapping code words in* C *of weight* 6.

Proof. Obviously C_1 is a subcode of C. Let I be the index set of the nonzero components of $\mathbf{a}$. By Lemma 16, $\mathbf{a}$ is also in C, then

$$\sum_{i \in I} \mathbf{h}_i = \mathbf{0}$$

Since the rank of the parity check matrix $H_{(32,21)}$ is 11 and $|I| = 12$, there must be an $I_1 \subset I$ such that

$$\sum_{i \in I_1} \mathbf{h}_i = \sum_{i \in I \setminus I_1} \mathbf{h}_i = \mathbf{0}$$

Hence, both binary vectors with I_1 and $I \setminus I_1$ as their index sets of the nonzero components are code words of C, and must have weight 6. Q.E.D.

Lemma 18 says that a subcode generated by $\mathbf{a}$ has at least dimension 2. It will now be shown that such a subcode has exactly dimension 2.

Consider the set Ω of all code words of weight 12 in $C^{\perp}$. For a code word $\mathbf{a} \in \Omega$, either $\mathbf{a} = (\mathbf{a}', 0)$ for some $\mathbf{a}' \in C'^{\perp}$, or $\mathbf{a} = \mathbf{1} + (\mathbf{b}', 0)$ for some

$\mathbf{b} \in C'^{\perp}$ with weight 20. Based on this observation, we partition Ω into Ω_1 and Ω_2, where Ω_1 consists of all $\mathbf{a} \in \Omega$ of the form $(\mathbf{a}', 0)$, and $\Omega_2 = \Omega \backslash \Omega_1$. Let Ω' be the set Ω restricted on $C'^{\perp}$; that is, the set of all code words in Ω with the parity bit removed. Ω'_1 and Ω'_2 are similarly defined. Thus all code words (of $C'^{\perp}$) in Ω'_1 have weight 12 and all code words in Ω'_2 have weight 11.

Lemma 19 *For* $\mathbf{a}', \mathbf{b}' \in \Omega'$, *if* $\mathbf{a}' + \mathbf{b}' \notin \Omega'$, *then* $w(\mathbf{a}' + \mathbf{b}') \geq 15$.

Proof. The three possible weights for code words in $C'^{\perp}$ are 12, 16 and 20. Either $\mathbf{a}' + \mathbf{b}' \in C'^{\perp}$ or $\mathbf{a}' + \mathbf{b}' = \mathbf{1}' + \mathbf{c}'$ for some $\mathbf{c}' \in C'^{\perp}$ by the definition of Ω', and $w(\mathbf{c}') < 20$. In the former case we must have $w(\mathbf{a}' + \mathbf{b}') \geq 16$, and in the latter case, $w(\mathbf{a}' + \mathbf{b}') \geq 15$. Q.E.D.

Lemma 20 *Let* $\mathbf{a} \in \Omega_1$. *Then the nonzeros of* $\mathbf{a}$ *are either the zeros of* $m_7(x)$ *or the zeros of* $m_7(x)m_{15}(x)$.

Proof. From the proof of Lemma 16, we have

$$x^{31} + 1 = h(x)m_7(x)m_{15}(x) \quad .$$

Since $\mathbf{a}' \in C'^{\perp}$, it follows that $h(x)$ divides $a(x)$, and there are only three possibilities for the nonzeros of $\mathbf{a}$: (i) the zeros of $m_7(x)$, (ii) the zeros of $m_7(x)m_{15}(x)$ and (iii) the zeros of $m_{15}(x)$. In the last case, the nonzero-coefficient terms of the MS polynomial $A(z)$ of $\mathbf{a}$ are $\{z^{31-i}, i \in \mathscr{C}_{15}\} = \{z, z^2, z^4, z^8, z^{16}\}$. Thus $d_{\text{eff}}[A(z)]$ is 15. By Theorem 12 in Section 2.4, $w(\mathbf{a}) \geq 31 - 15 = 16$, contradicting $\mathbf{a} \in \Omega_1$. Q.E.D.

Corollary 21 *Let* $\mathbf{a} \in \Omega_1$. *Then the nonzero-coefficient terms of its MS polynomial* $A(z)$ *are either* $\{z^i, i \in \mathscr{C}_{-7} \cup \mathscr{C}_{-15} = \mathscr{C}_3 \cup \mathscr{C}_1\}$ *or* $\{z^i, i \in \mathscr{C}_{-7} = \mathscr{C}_3\}$.

Lemma 22 *Let* $\mathbf{a} \in \Omega_2$. *Then the nonzeros of* $\mathbf{a}$ *are the zeros of* $m_0(x)m_7(x)$ $m_{15}(x)$ *or the zeros of* $m_0(x)m_7(x)$.

Proof. Since $\mathbf{a} = \mathbf{1} + (\mathbf{b}', 0)$ for some $\mathbf{b}' \in C'^{\perp}$,

$$a(x) = \sum_{i=0}^{30} x^i + b(x) = \frac{x^{31} + 1}{x + 1} + b(x) \quad .$$

Thus the nonzeros of $\mathbf{a}$ are the nonzeros of $b(x)$ *and* $1 \in GF(2^5)$. The conclusion follows from the argument in the proof of Lemma 20. Q.E.D.

Lemma 23 *Let* $\mathbf{a} \in \Omega_2$. *Then the nonzero-coefficient terms of its MS polynomial* $A(z)$ *are either* $\{z^i, i \in \mathscr{C}_0 \cup \mathscr{C}_3 \cup \mathscr{C}_1\}$ *or* $\{z^i, i \in \mathscr{C}_0 \cup \mathscr{C}_1\}$.

Lemma 24 *Let* $0 \neq \alpha \in GF(2^5)$. *Then the equation* $x^3 + \alpha^3 = 0$ *has only one root* $x = \alpha$ *in* $GF(2^5)$.

Proof. $x^3 + \alpha^3 = (x + \alpha)(x^2 + \alpha x + \alpha^2)$. Suppose $\beta \in GF(2^5)$ is a root of $x^2 + \alpha x + \alpha^2$. Certainly $\beta \neq 0$ and $\beta \neq \alpha$. Since $\beta^2 + \alpha\beta + \alpha^2 = 0$, or $(\beta/\alpha)^2 + (\beta/\alpha) + 1 = 0$, it follows that β/α is in both $GF(2^2)$ and $GF(2^5)$. However, $GF(2^2)$ does not have common elements with $GF(2^5)$ other than 0 and 1, a contradiction. Q.E.D.

Let $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ and $\mathbf{b} = (b_0, b_1, \ldots, b_{n-1})$. Define the *componentwise product* of $\mathbf{a}$ and $\mathbf{b}$ to be

$$\mathbf{ab} = (a_0 b_0, a_1 b_1, \ldots, a_{n-1} b_{n-1})$$

If $A(z)$ and $B(z)$ are the MS polynomials of $\mathbf{a}$ and $\mathbf{b}$ respectively, then by Theorem 11 of Section 2.4, the MS polynomial of $\mathbf{ab}$ is given by

$$AB(z) = [A(z)B(z)]_n$$

Lemma 25 *For any* $\mathbf{a}', \mathbf{b}' \in \Omega'$, $\mathbf{a}'\mathbf{b}' \notin C'$.

Proof. We need to consider three cases: (i) $\mathbf{a}', \mathbf{b}' \in \Omega'_1$; (ii) $\mathbf{a}' \in \Omega'_1$, $\mathbf{b}' \in \Omega'_2$ and (iii) $\mathbf{a}', \mathbf{b}' \in \Omega'_2$. Case (i). By Corollary 21, the nonzero-coefficient terms of $A(z)$ or $B(z)$ are either $\{z, z^2, z^3, z^4, z^6, z^8, z^{12}, z^{16}, z^{17}, z^{24}\}$ or $\{z^3, z^6, z^{12}, z^{17}, z^{24}\}$. It is straightforward to verify that the coefficient of z^{27} in $A(z)B(z)$ is always $(AB)_{31-27} = (AB)_4 = A_7 B_{28} + A_{28} B_7$. Since $A_{28} = a(\alpha^{28}) = a((\alpha^7)^4) = (a(\alpha^7))^4 = A_7^4$ and similarly $B_{28} = B_7^4$, we have $(AB)_4 = A_7 B_7 (A_7^3 + B_7^3)$. Either $(AB)_4$ is zero or not. If $(AB)_4$ is not zero, then α^4 is not a root of $\mathbf{a}'\mathbf{b}'$, thus $\mathbf{a}'\mathbf{b}' \notin C'$. If $(AB)_4 = 0$, then $(A_7^3 + B_7^3) = 0$. By Lemma 24, this only happens when $A_7 = B_7$, or $(A + B)_7 = A_7 + B_7 = 0$. By Corollary 21, $\mathbf{a}' + \mathbf{b}'$ is not in Ω', which implies, in the light of Lemma 19, that $w(\mathbf{a}' + \mathbf{b}') \geq 15$. Hence $w(\mathbf{a}'\mathbf{b}') = [w(\mathbf{a}') + w(\mathbf{b}') - w(\mathbf{a}' + \mathbf{b}')]/2 \leq 4$. Again, $\mathbf{a}'\mathbf{b}'$ cannot be in C'. Cases (ii) and (iii) can be proved using similar arguments, along with Corollaries 21 and 23. Q.E.D.

The next corollary expands the result of Lemma 25 to the extended code.

Corollary 26 *For any* $\mathbf{a}, \mathbf{b} \in \Omega$, $\mathbf{ab} \notin C$.

Proof. Consider two cases: (i) either $\mathbf{a} = (\mathbf{a}', 0)$ or $\mathbf{b} = (\mathbf{b}', 0)$. In this case, $\mathbf{ab}$ is $\mathbf{a}'\mathbf{b}'$ with the addition of a zero parity bit; (ii) $\mathbf{a} \neq (\mathbf{a}', 0)$ and $\mathbf{b} \neq (\mathbf{b}', 0)$. In this case, $\mathbf{ab}$ is $\mathbf{a}'\mathbf{b}'$ with the one parity bit. In either case, since $\mathbf{a}'\mathbf{b}' \notin C'$ by Lemma 25, $\mathbf{ab} \notin C$. Q.E.D.

Let $\mathscr{S}^{12}$ be the set of all subcodes of C generated by code words of weight 12 in $C^\perp$. The following theorem characterizes the structure of such subcodes.

Theorem 27 *Any subcode $S \in \mathscr{S}^{12}$ is a* (12, 2, 6) *code.*

Proof. Let **a** and **b** be in Ω. **b** does not belong to the the subcodes generated by **a**, otherwise we would have $\mathbf{b} = \mathbf{ac}$ for some $\mathbf{c} \in C$, so $\mathbf{ab} = \mathbf{aac} = \mathbf{ac} = \mathbf{b}$, contradicting Corollary 26. Thus there are 496 subcodes in $\mathscr{S}^{12}$, generated by 496 code words in Ω. Since $w(\mathbf{ab}) \leq 6$, no code word in C of weight 6 belongs to both subcodes generated by **a** and **b** by Corollary 26. Since a subcode in $\mathscr{S}^{12}$ contains at least two weight-6 code words of C by Lemma 18, all 496 subcodes contain at least $496 \times 2 = 992$ *different* weight-6 code words of C. But there are exactly 992 weight-6 code words in C, therefore each subcode has exactly 2 (nonoverlapping) weight-6 code words of C. Consequently there are only four code words in the subcode, with a weight distribution $W_0 = W_{12} = 1$, $W_6 = 2$, making it a (12, 2, 6) code. Q.E.D.

Lemma 28 *An* $(n, 3)$ *subcode of C which contains a subcode of $S \in \mathscr{S}^{12}$ must have $n \geq 14$.*

Proof. Let S_3 be such a subcode and $S_2 \in \mathscr{S}^{12}$ be contained in S_3. Let **a** be the weight-12 code word in S_2 and let **b** be a code word in S_3 but not in S_2. Clearly $n = w(\mathbf{a}) + w(\mathbf{b}) - w(\mathbf{ab})$. We prove the result by showing that $w(\mathbf{ab})$, and hence n, is even.

Case (i): both $\mathbf{a}'$ and $\mathbf{b}'$ have odd weights. By Corollary 23, the (possible) nonzero-coefficient terms of $A(z)$ are $\{z^i, i \in \mathscr{C}_0 \cup \mathscr{C}_1 \cup \mathscr{C}_3\}$. Since the generator polynomial of C' is $m_1(x)m_3(x)$, the (possible) nonzero-coefficient terms of $B(z)$ are $\{z^i, i \in \mathscr{C}_0 \cup \mathscr{C}_1 \cup \mathscr{C}_3 \cup \mathscr{C}_5 \cup \mathscr{C}_{11}\}$. Note that $A_0 = B_0 = 1$. Then it can be verified by expanding $A(z)B(z)$ that the constant term (*modulo* $(z^{31} - 1)$) in $A(z)B(z)$ is 1, meaning that $\mathbf{a}'\mathbf{b}'$ has an odd weight. Since the parity bits of both **a** and **b** are 1, we have that $w(\mathbf{ab}) = w(\mathbf{a}'\mathbf{b}') + 1$ is even. Case (ii): at least one of the $\mathbf{a}'$ and $\mathbf{b}'$ has an even weight. The possible nonzero-coefficient terms of $\mathbf{a}'$ and $\mathbf{b}'$ are exactly the same as in case (i) except that at least one of the $A(z)$ and $B(z)$ does not have the constant term. Similarly we can verify that there is no constant term (*modulo* $(z^{31} - 1)$) in $A(z)B(z)$, and it follows that $w(\mathbf{ab}) = w(\mathbf{a}'\mathbf{b}')$ is even. Q.E.D.

A subcode S_n of length n, which contains a subcode S_{n-1}, has a dimension $k(S_n)$ at most $k(S_{n-1}) + 1$. This proves the following corollary.

Corollary 29 *An* $(n, 4)$ *subcode of C containing a subcode of $S \in \mathscr{S}^{12}$ must have $n \geq 15$. An* $(n, 5)$ *subcode of C containing a subcode of $S \in \mathscr{S}^{12}$ must have $n \geq 16$.*

Let i_1 be such that $p_{i_1}^{\perp} = 1$ and $p_i^{\perp} = 0$ for $i = 0, 1, \ldots, i_1 - 1$. Obviously $i_1 \geq 12$. Combining the constraints posed by (23) and (24), Theorem 27, Lemma 28 and Corollary 29, bounds on the half profiles of the (32, 21) code for $i_1 = 12, 13, 14$ and 15 can be written as follows.

$i_1 = 12$:

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_i :	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	4	5
f_i :	21	20	19	18	17	16	15	14	13	12	11	10	10	9	8	7	6
s_i :	0	1	2	3	4	5	5	6	7	8	9	10	9	10	10	10	10
b_i :	1	2	3	4	5	6	6	7	8	9	10	10	10	11	11	11	

(26)

$i_1 = 13$:

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_i :	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	4	5
f_i :	21	20	19	18	17	16	15	14	13	12	11	10	9	9	8	7	6
s_i :	0	1	2	3	4	5	5	6	7	8	9	10	10	10	10	10	10
b_i :	1	2	3	4	5	6	6	7	8	9	10	11	10	11	11	11	

(27)

$i_1 = 14$:

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_i :	0	0	0	0	0	0	1	1	1	2	2	2	2	2	3	4	5
f_i :	21	20	19	18	17	16	15	14	13	12	11	10	9	8	8	7	6
s_i :	0	1	2	3	4	5	5	6	7	7	8	9	10	11	10	10	10
b_i :	1	2	3	4	5	6	6	7	8	8	9	10	11	11	11	11	

(28)

$i_1 = 15$:

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_i :	0	0	0	0	0	0	1	1	1	2	2	3	3	3	3	4	5
f_i :	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	7	6
s_i :	0	1	2	3	4	5	5	6	7	7	8	8	9	10	11	10	10
b_i :	1	2	3	4	5	6	6	7	8	8	9	9	10	11	11	11	

(29)

In the case of $i_1 = 15$, the existing constraints allow that $p_{14} \leq 4$. Suppose that $p_{14} = 4$. The fact that $i_1 = 15$ means that we have a code word in $C^{\perp}$ hence in C with all of its nonzero bits in the first 15 bits, which makes $p_{15} = p_{14}+1 = 5$. This contradicts Corollary 29. We then have $p_{14} \leq 3$ as shown in (29).

For $i_1 = 16$, we have a code word $\mathbf{a}$ in $C^{\perp}$ hence in C with all of its nonzero bits in the first 16 bits, but now there are two possibilities: (i) $w(\mathbf{a}) = 12$ and (ii) $w(\mathbf{a}) = 16$. For case (i), we can write the half profile bound as follows:

$i_1 = 16, \quad w(\mathbf{a}) = 12$:

i :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p_i :	0	0	0	0	0	0	1	1	1	2	2	3	3	3	4	4	5
f_i :	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	6
s_i :	0	1	2	3	4	5	5	6	7	7	8	8	9	10	11	11	10
b_i :	1	2	3	4	5	6	6	7	8	8	9	9	10	11	11	11	

(30)

For case (ii), we consider the subcode S of length 16 generated by $\mathbf{a}$. Let $k(S)$ be the dimension of the S. The dual code $S^{\perp}$ of S is then $\{\mathbf{ab} : \mathbf{b} \in C^{\perp}\}$. Since

$$w(\mathbf{a}+\mathbf{b}) = w(\mathbf{a}) + w(\mathbf{b}) - 2w(\mathbf{ab}) = 16 + w(\mathbf{b}) - 2w(\mathbf{ab}) \leq 20 \ ,$$

we have

$$w(\mathbf{ab}) \geq [w(\mathbf{b}) - 4]/2 \geq (12 - 4)/2 = 4 \ .$$

The minimum distance $d(S^{\perp})$ of $S^{\perp}$ is at least 4 (in fact, it is exactly 4). This implies that for $S^{\perp}$, $f_i^{\perp}(S) = 0$, $i = 13, 14, 15$ and 16, from which we get

$$p_i = p_i(S) = k(S) + f_i^{\perp} + i - 16 = k(S) + i - 16 \ ,$$

for $i = 13, 14, 15$ and 16. We can also get a bound for p_{11} from $d(S^{\perp}) \geq 4$. In this case we must have $f_{11}^{\perp} \leq 1$, and

$$p_{11} = k(S) + f_{11}^{\perp} + 11 - 16 \leq k(S) - 4 \ .$$

Since $p_{16} = k(S) \leq 6$ by (24), we have obtained the following profile bound:

$$\begin{array}{l} i_1 = 16, \quad w(\mathbf{a}) = 16 : \\ \begin{array}{rcccccccccccccccc} i: & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ p_i: & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 5 & 6 \\ f_i: & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 6 \\ s_i: & 0 & 1 & 2 & 3 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 9 & 10 & 10 & 10 & 10 & 9 \\ b_i: & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 7 & 8 & 8 & 9 & 10 & 10 & 11 & 11 & 10 & \end{array} \end{array} \tag{31}$$

For $i_1 > 16$, f_i is strictly decreasing with i for $i \leq 16$, giving us the following profile bound.

$$\begin{array}{l} i_1 > 16 : \\ \begin{array}{rcccccccccccccccc} i: & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ p_i: & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 5 & 5 & 6 \\ f_i: & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 \\ s_i: & 0 & 1 & 2 & 3 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 8 & 9 & 9 & 9 & 10 & 10 \\ b_i: & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 7 & 8 & 8 & 9 & 9 & 10 & 10 & 10 & 11 & \end{array} \end{array} \tag{32}$$

We are now able to compute the lower bounds $E_{i_1}^H$ on the edge count for the half profiles for various possible i_1. From Equations (26)–(29) and (32), we have

$$\begin{aligned} &E_{12}^H = 10{,}302, \quad E_{13}^H = 12{,}350, \quad E_{14}^H = 10{,}558, \\ &E_{15}^H = 8{,}510, \quad E_{(i_1>16)}^H = 6{,}974 \end{aligned} \tag{33}$$

E_{16}^H can be obtained by considering both (30) and (31). (30) gives an edge count of 7,998 and (31) gives 8,510. Therefore

$$E_{16}^H = 7{,}998 \tag{34}$$

A lower bound on the edge count over all possible permutations can be obtained immediately from (33) and (34):

$$E \geq 2 \min\{E^H_{12}, E^H_{13}, E^H_{14}, E^H_{15}, E^H_{16}, E^H_{(i_1>16)}\} = 13{,}948 \tag{35}$$

Note that the half profile which gives this lower bound is (32). This new bound is a significant improvement over the bound given by (25) on one hand, but on the other hand it limits further reductions in trellis complexity. Better yet (or worse yet), the bound given in (35) can be further tightened.

A profile with its left half as (32) cannot be symmetric (i.e., $p_i = f_{32-i}$, $f_i = p_{32-i}$). Therefore we cannot have a full profile in which both halves are identical to (32). Note that if we decrease $p_{16} = 6$ to $p_{16} = 5$ in (32), the edge count remains unchanged. It is possible to construct a symmetric profile based on this modified half profile. This can be done by using the symmetric mapping

$$p_i = f_{32-i}, \quad f_i = p_{32-i} \tag{36}$$

to map the modified left half profile to the right half, and concatenating the two halves. The complete profile is given by (37). The profile gives an edge count of $2E^H_{(i_1>16)} = 13,948$. We will show, however, that this bound is not attainable. Some preliminary results are necessary for the proof.

$$\begin{array}{rcccccccccccccccc}
i: & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
p_i: & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 5 & 5 & 5 \\
f_i: & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 \\
s_i: & 0 & 1 & 2 & 3 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 8 & 9 & 9 & 9 & 10 & 10 \\
b_i: & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 7 & 8 & 8 & 9 & 9 & 10 & 10 & 10 & 11 & 11
\end{array}$$

$$\begin{array}{rcccccccccccccccc}
i: & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\
p_i: & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\
f_i: & 5 & 5 & 5 & 4 & 3 & 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_i: & 11 & 10 & 9 & 9 & 9 & 8 & 8 & 7 & 7 & 6 & 5 & 5 & 4 & 3 & 2 & 1 & 0 \\
b_i: & 11 & 10 & 10 & 10 & 9 & 9 & 8 & 8 & 7 & 6 & 6 & 5 & 4 & 3 & 2 & 1 &
\end{array} \tag{37}$$

Lemma 30 *If* $p_{14} = 5$, *then* $f_{14} = 7$.

Proof. $p_{14} = 5$ implies that $i_1 > 14$, since otherwise we would have $p_{14} = 3$ by Corollary 29. Thus f_i is strictly decreasing: $f_i = f_{i-1} - 1$ for $i < 14$, which gives $f_{14} = 7$. Q.E.D.

Lemma 30 shows that the future code F_{14} is an $(18, 7, 6)$ code if the past code P_{14} is a $(14, 5, 6)$ code. We now consider the dual code of F_{14}, which is the future projection subcode $F^{\perp 14}$ of $F^{\perp}$.

Lemma 31 *If $p_{14} = 5$, then the minimum distance of $F^{\perp 14}$ is at least* 3.

Proof. Let $\mathbf{a} \in C^{\perp}$. Let $\mathbf{a}^P \in F^{\perp 14}$ be the projection of $\mathbf{a}$ on $F^{\perp 14}$. Case (i): $w(\mathbf{a}) = 12$. We must have $w(\mathbf{a}^P) \geq 3$ since otherwise we would have a subcode of dimension 6 and length not larger than 16 consisting of P_{14} and $\mathbf{a}$, a contradiction to Corollary 29. Case (ii): $w(\mathbf{a}) = 16$. Again, $w(\mathbf{a}^P) \geq 3$, since the only other possibility is that $w(\mathbf{a}^P) = 2$, in which case there would be a $(16, 6, 6)$ subcode generated by $\mathbf{a}$. This subcode contains P_{14}, which violates (31). Finally, case (iii): $w(\mathbf{a}) = 20$. In this case we have $w(\mathbf{a}^P) \geq 6$. We conclude that $w(\mathbf{a}^P) \geq 3$ for all $\mathbf{a}^P \in F^{\perp 14}$. Q.E.D.

Theorem 32 *If $p_{14} = 5$, then there is no* $(14, 5, 6)$ *subcode in F^{14}.*

Proof. Since the minimum distance of $F^{\perp 14}$ is at least 3 by Lemma 31, any subcode of length 4 of $F^{\perp 14}$ has a dimension of at most 1. This gives $f_{14}^{\perp}(F_{14}) = f_{14}(F^{\perp 14}) < 2$. Applying the duality identities to F_{14} and $F^{\perp 14}$, we have

$$p_{14}(F_{14}) = f_{14}^{\perp}(F_{14}) + 14 + k(F_{14}) - n(F_{14}) < 5$$

where $n(F_{14}) = 18$ and $k(F_{14}) = 7$ by Lemma 30. This proves the theorem. Q.E.D.

Theorem 33 *There are no nonoverlapping* $(14, 5, 6)$ *subcodes.*

Proof. This is a direct consequence of Theorem 32. Q.E.D.

Theorem 34 *The edge count $E > 13{,}948$.*

Proof. Previous lemmas, theorems and corollaries have shown that (37) is the only possible profile to achieve (35). Note that in (37), $p_{14} = f_{18} = 5$, so we have two $(14, 5, 6)$ subcodes, one on the first 14 bits and the other one on the last 14 bits. This contradicts Theorem 33. Q.E.D.

Having precluded the profile in (37), a new lower bound on the edge count can be obtained by considering the following two cases. (i) Both half profiles have $i_1 > 16$ when considered as left half profiles. The best left half profiles with respect to the edge count are given in (32) and (37). The "second best" half profile in this case will have to have $p_{14} = 4$ under the constraint that the other half profile already has $p_{14} = 5$. The resulting full profile will be the same as (37), but with either p_{14} or f_{18} being changed to 4. Such a profile gives an edge count of 14,972. (ii) One half profile has $i_1 \leq 16$. An examination of (33) and (34) shows that the combination of the profiles in (30) and (32) gives the lowest edge count in this case, which is again 14,972. Thus we have established the following

Theorem 35 *The edge count* $E \geq 14{,}972$.

A computer search was devised that used several of the above theorems, lemmas, and corollaries to limit the search space. A permutation which gives a trellis with 14,972 edges was found. Theorem 35 can thus be reexpressed as an exact value for the minimal edge complexity for the extended BCH (32, 21, 6) code.

Consider the following parity check matrix $H_{(32,21)}$ in a standard form,

$$H_{(32,21)} = \begin{bmatrix} 10010100100111101010110000000000 \\ 11011110110100011111101000000000 \\ 11111011111101100101000100000000 \\ 01111101111110110010100010000000 \\ 10101010011000110011100001000000 \\ 11000001101011110011000000100000 \\ 01100000110101111001100000010000 \\ 10100100111101010110000000001000 \\ 01010010011110101011000000000100 \\ 00101001001111010101100000000010 \\ 11100111000101001100100000000001 \end{bmatrix} \tag{38}$$

The permutation

$$(0, 26, 16, 10, 13, 4, 7, 14, 3, 19, 5, 1, 25, 30, 29, 27, 21, 6, 17)$$
$$(2, 23, 15, 28, 22, 8, 11, 24, 18)(9, 20)(12)(31)$$

yields a matrix with an associated trellis that satisfies the bound in Theorem 35 with equality. The permuted parity check matrix is given by

$$H'_{(32,21)} = \begin{bmatrix} 00001101010010100101011110000000 \\ 00010101111110100110001101101000 \\ 00000000011101110110010111111100 \\ 00000001110111010111101110110000 \\ 00000001101101110010001010001010 \\ 00000001110000110001010010111001 \\ 00100100110101000000011110101000 \\ 00000001110111111000010101000000 \\ 01000101001100010101000110101000 \\ 00000010100001010011011101011000 \\ 10000100001011100000011101110000 \end{bmatrix}$$

The profile of the resulting trellis is the same as (37) but with p_{14} being changed to 4.

From the implementation perspective it is sometimes desirable to have a symmetric trellis (that is, $p_i = f_{n-i}$, $f_i = p_{n-i}$ for an (n, k) code). It is not difficult to see, from the arguments for Theorem 35, that the number of edges for symmetric trellises is lower bounded by the edge count of the following profile:

$$
\begin{array}{rcccccccccccccccc}
i: & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
p_i: & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 \\
f_i: & 21 & 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 \\
s_i: & 0 & 1 & 2 & 3 & 4 & 5 & 5 & 6 & 7 & 7 & 8 & 8 & 9 & 9 & 9 & 10 & 10 \\
b_i: & 1 & 2 & 3 & 4 & 5 & 6 & 6 & 7 & 8 & 8 & 9 & 9 & 10 & 10 & 10 & 11 & 11
\end{array}
$$

$$
\begin{array}{rcccccccccccccccc}
i: & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\
p_i: & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 \\
f_i: & 5 & 5 & 4 & 4 & 3 & 3 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
s_i: & 11 & 10 & 9 & 9 & 9 & 8 & 8 & 7 & 7 & 6 & 5 & 5 & 4 & 3 & 2 & 1 & 0 \\
b_i: & 11 & 10 & 10 & 10 & 9 & 9 & 8 & 8 & 7 & 6 & 6 & 5 & 4 & 3 & 2 & 1 &
\end{array}
\tag{39}
$$

The only difference between (37) and (39) is that p_{14} and f_{18} are 4 in (39) rather than 5 in (37). The edge count in (39) is 15,996. Again, the profile in (39) is achievable. For example, the following permutation yields a trellis with the desired profile.

$$
\begin{array}{l}
(0, 26, 17, 2, 16, 27, 22, 9, 29, 28, 23, 12, 3, 15) \\
(1, 21, 7, 18, 25, 20, 11, 5, 13, 4, 14) \\
(6, 24, 19, 10)(8)(30)(31)
\end{array}
\tag{40}
$$

The permuted parity check matrix is given by (41).

$$
H^P_{(32,21)} = \begin{bmatrix}
01101101100110010000100101000000 \\
00110001101111110000101011110000 \\
00001100101011110101111010101000 \\
00011011100111100101111101000000 \\
00011001001001010111001011000000 \\
10011101100000110001010110000000 \\
00111100100000101100101011000000 \\
00010101100100010101101000100100 \\
00101001011010100001101110000000 \\
00010100000001000101110111100010 \\
00100100001100110100110001100001
\end{bmatrix}
\tag{41}
$$

4 Conclusions

This paper focused on the construction of minimum-edge trellises for some BCH codes, and in particular, the $(32, 26, 4)$ and $(32, 21, 6)$ extended BCH codes, which have been considered or employed in wireless paging systems.

The minimum-edge trellises provide the most efficient structures for maximum-likelihood decoding using the Viterbi algorithm. The construction of such a trellis for the $(32, 26, 4)$ extended BCH code was presented in the paper and generalized to the entire class of the $(2^m, 2^m - m - 1, 4)$ extended Hamming codes, which is a subclass of the extended BCH codes.

A subcode analysis was developed to show that the minimal edge count for the trellis for the (32, 21, 6) extended BCH code is 14,948. A trellis with this edge complexity was identified. The analysis means developed in the paper can be applied to the construction of minimum-edge trellises for the variants of the (32, 21) BCH code, such as the (30, 18) shortened BCH code used in ERMES, and for other BCH codes as well.

The trellises developed for the (32, 26) and (32, 21) codes are highly irregular, i.e. the constituents of their vertex sets change significantly with trellis depth. This creates a problem with the implementation, for it can require a separate lookup table at each trellis depth i. It has been shown by one of the authors [24] that a mapping technique can be used to pack the vertices at depth i into the set $\{0, 1, \ldots, 2^{s_i} - 1\}$, where s_i is the dimension of the vector space formed by the vertex set at depth i. The map is described by a single matrix, which requires far less storage capacity than n distinct lookup tables.

The primary applications of interest for this work involve small mobile devices that are typically used in a fading environment. It has been shown elsewhere [21] that the performance of the SDD's developed here are superior to that of the standard HDD's over a Rayleigh fading channel. The (32, 21) SDD provides substantial coding gain relative to the (32, 21) HDD, while the (32, 26) SDD is comparable to the (32, 21) HDD. The (32, 21) SDD thus offers the opportunity for improved performance, while the (32, 26) SDD offers an increase in data throughput through a reduction in overhead.

References

1. L. E. Aguado, P. G. Farrell: On hybrid stack decoding algorithms for block codes, IEEE Trans Inf Theory, 44(1), 398–409 (1998)
2. L. R. Bahl, J. Cocke, F. Jelinek, J. Raviv: Optimal decoding of linear codes for minimizing symbol error rate, IEEE Trans Inf Theory 20(2), 284–287 (1974)
3. A. E. Brouwer, T. Verhoeff: An updated table of minimum-distance bounds for binary linear codes, IEEE Trans Inf Theory 39(2), 662–677 (1993)
4. S. Dolinar, L. Ekroot, A. Kiely, W. Lin, R. J. McEliece: Trellis complexity of linear block codes, Proceedings of the 32nd Allerton Conference on Communications, Control and Computing, 1994
5. J. Dorenbosch: Capacity grows with demand in ReFLEX protocol, Wireless Systems Design 3(1), 46–54 (1998)
6. G. D. Forney, Jr.: Coset codes – Part II: Binary lattices and related codes, IEEE Trans Inf Theory 34(5), 1152–1187 (1988)
7. G. D. Forney, Jr.: Dimension/length profiles and trellis complexity of linear block codes, IEEE Trans Inf Theory 40(6), 1741–1752 (1994)
8. T. Kasami, T. Takata, T. Fujiwara, S. Lin: On the optimum bit orders with respect to state complexity of trellis diagrams for linear block codes, IEEE Trans Inf Theory 39(1), 242–243 (1993)
9. T. Komura, M. Oka, T. Fujiwara, T. Onoye, T. Kasami, S. Lin: VLSI Architecture of a Recursive Maximum Likelihood Decoding Algorithm for a (64, 35) Subcode of the (64, 42) Reed-Muller Code, Proceedings of the 1996 IEEE International Symposium on Information Theory and its Applications, Victoria, B. C., Canada, pp. 709–712, September 17–20, 1996

10. A. Lafourcade-Jumembo, A. Vardy: Lower bounds on trellis complexity of block codes, preprint
11. A. Lafourcade-Jumembo, A. Vardy: Asymptotically good codes have infinite trellis complexity, IEEE Trans Inf Theory 41(2), 555–558 (1995)
12. A. Luna, S. B. Wicker: Iterative Maximum Likelihood Trellis Decoding for Block Codes, IEEE Transactions on Communications, March 1999
13. F. J. MacWilliams, N. J. A. Sloane: The Theory of Error-Correcting Codes, Amsterdam: North-Holland, 1977
14. H. H. Manoukian, B. Honary: BCJR trellis construction for binary linear block codes, IEE Proc.-Commun 144(6), 367–371 (1997)
15. I. Martin, B. Honary, G. Markarian, S. B. Wicker: Trellis Based Type-II Hybrid ARQ Protocols for Reed-Muller Codes, Proceedings of the 1997 IEEE International Symposium on Information Theory, Ulm, Germany, June 29–July 4, 1997
16. J. L. Massey: Foundations and methods of channel coding, Proc. Int. Conf. Inform. Theory and Systems, NTG-Fachberichte Vol. 65, pp. 148–157, 1978
17. R. J. McEliece, L. Swanson: Reed-Solomon codes and the exploration of the solar system, In: S. B. Wicker, V. K. Bhargava (eds) Reed-Solomon Codes and Their Applications, IEEE press, 1994
18. R. J. McEliece: On the BCJR trellis for linear block codes, IEEE Trans Inf Theory 42(4), 1072–1092 (1996)
19. D. J. Muder: Minimal trellises for block codes, IEEE Trans Inf Theory 34(5), 1049–1053 (1988)
20. J. G. Proakis: Digital Communications, McGraw-Hill Second Edition, 1989
21. T. L. Tapp, A. A. Luna, X.-A. Wang, S. B. Wicker: Extended Hamming and BCH Soft Decision Decoders for Mobile Data Applications, IEEE Transactions on Communications, March 1999
22. A. Vardy, Y. Be'ery: Maximum-likelihood soft decision decoding of BCH codes, IEEE Trans Inf Theory, 40(2) 546–554 (1994)
23. V. V. Vazirani, H. Saran, B. S. Rajan: An algebraic characterization of minimal trellises for group codes, and an efficient algorithm for construction over rings Z_m, preprint
24. X.-A. Wang: Trellis Based Decoders and Neural Network Implementations, Doctoral Dissertation, Georgia Institute of Technology, March 1996
25. S. Wicker: Error Control Systems for Digital Communication and Storage, Englewood Cliffs: Prentice Hall, 1995
26. J. K. Wolf: Efficient maximum likelihood decoding of linear block codes using a trellis, IEEE Trans Inf Theory IT-24(1), 76–80 (1978)
27. Ø. Ytrehus: On the trellis complexity of certain binary linear codes, IEEE Trans Inf Theory 41(2), 559–560 (1995)