Trellis Decoding Complexity of Linear Block Codes

Aaron B. Kiely, *Member, IEEE,* Samuel J. Dolinar, Jr., *Member, IEEE,* Robert J. McEliece, *Fellow, IEEE,*
Laura L. Ekroot, *Member, IEEE,* and Wei Lin, *Student Member, IEEE*

Abstract— In this partially tutorial paper, we examine minimal trellis representations of linear block codes and analyze several measures of trellis complexity: maximum state and edge dimensions, total span length, and total vertices, edges and mergers. We obtain bounds on these complexities as extensions of well-known dimension/length profile (DLP) bounds. Codes meeting these bounds minimize all the complexity measures simultaneously; conversely, a code attaining the bound for total span length, vertices, or edges, must likewise attain it for all the others. We define a notion of "uniform" optimality that embraces different domains of optimization, such as different permutations of a code or different codes with the same parameters, and we give examples of uniformly optimal codes and permutations. We also give some conditions that identify certain cases when no code or permutation can meet the bounds. In addition to DLP-based bounds, we derive new inequalities relating one complexity measure to another, which can be used in conjunction with known bounds on one measure to imply bounds on the others. As an application, we infer new bounds on maximum state and edge complexity and on total vertices and edges from bounds on span lengths.

Index Terms—Trellis decoding, block codes, decoding complexity, minimal trellises, dimension/length profiles.

I. INTRODUCTION

EVERY linear block code can be represented by a *minimal trellis,* originally introduced by Bahl *et al.* [1], which is a labeled graph that can be used as a template for encoding or decoding. As shown by McEliece [21], the minimal trellis simultaneously minimizes the maximum number of states, and the total numbers of vertices and edges in the trellis.

A code's minimal trellis is unique as long as the ordering of the code's symbols is fixed. However, different permutations of the symbols yield different minimal trellises. An optimum minimal trellis for the code is one which minimizes a suitable measure of trellis complexity over all possible permutations of the code. There are no known efficient algorithms for constructing optimum minimal trellises.

In this paper we examine properties of the minimal trellis representation of a code and its dual for a fixed permutation, and apply these properties to obtain useful relationships among a variety of complexity measures. We extend these results to the problem of finding a trellis that minimizes complexity over all permutations of a code or all codes with the same parameters. We also briefly discuss minimal trellises of maximum complexity.

Many of the basic results reported in this paper have become well known during a flurry of recent research in this area. Our intent is to provide a cohesive tutorial presentation of this material, along with some new results.

II. THE MINIMAL TRELLIS OF A CODE

A. The Minimal Span Generator Matrix

For any linear (n, k) block code C over GF(q) there exists a minimal span generator matrix (MSGM) which can be used to construct a minimal trellis T for the code. The trellis has $n + 1$ levels of vertices and n levels of edges. The vertex levels, called *depths,* are numbered from 0 to n; the edge levels, called *stages,* are numbered from 1 to n. Each stage of edges corresponds to one stage of encoding or decoding using the trellis. Each vertex at depth i represents a possible encoder state after the ith stage of encoding. The ith stage corresponds to the ith column of the MSGM, whereas the ith depth corresponds to the "space between" columns i and $i+1$.

The *edge span* of any row of the MSGM is the smallest set of consecutive stages containing its nonzero positions. The *vertex span* of a row is the set of depths i such that at least one nonzero symbol occurs before and after depth i. We use the term *span length* to refer to the cardinality of a span. Using the MSGM to encode k information symbols in n stages of encoding, the edge span of the jth row represents the interval of stages during which the jth information symbol can affect the encoder output. The vertex span of the jth row is the set of depths at which the jth information symbol can affect the encoder state.

A remarkable result is that the MSGM simultaneously makes all of the spans as short as possible: the edge spans and vertex spans for any other generator matrix representing C always contain the corresponding spans of some row-permuted MSGM [21]. Any generator matrix can be put into minimal-span form using the following greedy algorithm: at each step, perform any row operation that reduces the edge-span length of any row of the matrix. The rows of the MSGM are then "atomic codewords," according to the terminology of Kschischang and Sorokine [15].
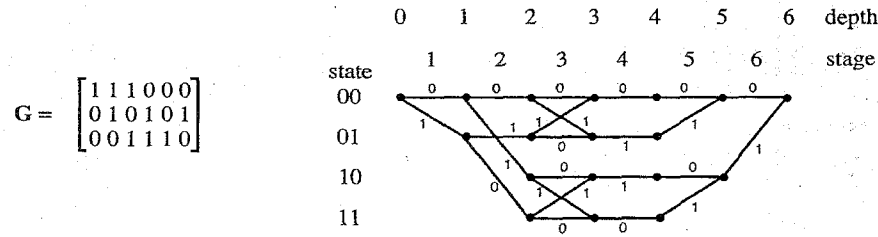
$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Fig. 1. An MSGM and minimal trellis for the $(6, 3, 3)$ shortened Hamming code.

Each vertex or state at a given depth can be uniquely labeled using k or fewer symbols from GF (q), but any state-label symbol can be reused to represent several information symbols, as long as the vertex spans of the corresponding rows of the MSGM do not overlap. This time-sharing of state-label symbols is the key to efficient trellis representations of the code.

For example, Fig. 1 shows a minimal span generator matrix and corresponding minimal trellis for the $(6, 3, 3)$ shortened Hamming code. The edge spans are $\{1, 2, 3\}$, $\{2, 3, 4, 5, 6\}$, and $\{3, 4, 5\}$. The vertex spans are $\{1, 2\}$, $\{2, 3, 4, 5\}$, and $\{3, 4\}$. The trellis needs only 2^2 states, not 2^3, because vertex spans of the first and third rows do not overlap, thus one of the state bits can be time-shared by the first and third information bits.

Following Forney [6], let us define the ith past and future subcodes, denoted $\mathcal{P}_i$ and $\mathcal{F}_i$, to be the linear subcodes consisting of codewords whose edge spans are contained in $[1, i]$ and $[i+1, n]$, respectively. The dimensions of these codes can be easily determined from the MSGM: $f_i \stackrel{\triangle}{=} \dim (\mathcal{F}_i)$ is the number of rows for which the leftmost nonzero entry lies in column $i + 1$ or later, and $p_i \stackrel{\triangle}{=} \dim(\mathcal{P}_i)$ is the number of rows for which the rightmost nonzero entry lies in column i or earlier [21]. This implies that $\{p_i\}_{i=0}^{n}$ increases in steps of 0 or 1 from $p_0 = 0$ to $p_n = k$ and $\{f_i\}_{i=0}^{n}$ decreases in steps of 0 or 1 from $f_0 = k$ to $f_n = 0$.

For each $1 \leq i \leq n$, we define the *left-* and *right-basis indicators*, l_i, r_i, to identify the positions where the future and past subcode dimensions change

$$l_i \stackrel{\triangle}{=} f_{i-1} - f_i \in \{0, 1\}$$
$$r_i \stackrel{\triangle}{=} p_i - p_{i-1} \in \{0, 1\}.$$

For any i, $l_i = 1$ if and only if the ith column of the MSGM is linearly independent of the $i - 1$ columns to the *left*. Similarly, $r_i = 1$ if and only if the ith column is linearly independent of the $n - i$ columns to the *right*. The columns where $l_i = 1$ and the columns where $r_i = 1$ each form a basis for the column space of the generator matrix, called the *left basis* and the *right basis*, respectively. The positions of the left- and right-basis columns can be regarded as information positions when the MSGM is used to encode the information left-to-right or right-to-left, respectively.

Note that $l_i = 0$ if and only if there exists a dual codeword y of the form

$$y = \underbrace{XXX \cdots X}_{i-1} 1 \underbrace{000 \cdots 0}_{n-i}$$

where $XXX \cdots X$ denotes some sequence of symbols from GF (q). Defining $y_1, y_2, \cdots, y_{n-k}$ in this manner for each of the left-dependent columns in the MSGM produces $n - k$ dual codewords of the form

$$y_1 = XXX \cdots X 1000 \cdots \qquad\qquad 0$$
$$y_2 = XXX \cdots \qquad\qquad X 1000 \cdots 0$$
$$\vdots$$
$$y_{n-k} = XXX \cdots \qquad\qquad\qquad X 1.$$

These dual codewords are clearly linearly independent and thus can be used as the rows of the generator matrix for the dual code $\mathcal{C}^\perp$. We see that the positions where $r_i^\perp = 1$ are precisely the positions where $l_i = 0$, and the positions where $l_i^\perp = 1$ are the positions where $r_i = 0$. Here $l_i^\perp$ and $r_i^\perp$ are the left- and right-basis indicators for $\mathcal{C}^\perp$. These observations lead to the following lemma.

Lemma 1: The left- and right-basis indicators for a code and its dual are related by

$$l_i + r_i^\perp = l_i^\perp + r_i = 1, \quad 1 \leq i \leq n$$

and the dimensions p_i, f_i, of the past and future subcodes of a code are given in terms of those of the dual code $p_i^\perp$, $f_i^\perp$ as follows:

$$p_i = k - n + i + f_i^\perp, \quad 0 \leq i \leq n$$
$$f_i = k - i + p_i^\perp, \qquad 0 \leq i \leq n.$$

This result has been derived using different approaches in [7], [15].

B. Primitive Structures of a Minimal Trellis

There are four basic building blocks (see also [15], [18]) that can be used to construct the minimal trellis for any code. At any given stage i, all primitive structures are of the same type, which is determined by the values of l_i and r_i. The primitive structures are:

1) Extension $(-)$: This primitive structure appears at stage i when $l_i = 0, r_i = 0$, e.g., stage 4 in Fig. 1. There is a single edge out of each vertex at depth $i - 1$ and a single edge into each vertex at depth i, hence the number of vertices remains constant.

2) Simple expansion $(<)$: corresponds to $l_i = 1, r_i = 0$, e.g., stages 1 and 2 in Fig. 1. There are q edges out of each vertex at depth $i - 1$, and a single edge into each vertex at depth i, hence multiplying by q the number of states from one vertex depth to the next.

TABLE I
DUAL PRIMITIVE STRUCTURES

Code Structure	Dual Structure
Extension $(-)$ $l_i = 0, r_i = 0$	Butterfly $(\mathtt{x}$ or $\circ$ $)$ $l_i^\perp = 1, r_i^\perp = 1$
Simple Expansion $(<)$ $l_i = 1, r_i = 0$	Simple Expansion $(<)$ $l_i^\perp = 1, r_i^\perp = 0$
Simple Merger $(>)$ $l_i = 0, r_i = 1$	Simple Merger $(>)$ $l_i^\perp = 0, r_i^\perp = 1$
Butterfly $(\mathtt{x}$ or $\circ$ $)$ $l_i = 1, r_i = 1$	Extension $(-)$ $l_i^\perp = 0, r_i^\perp = 0$

3) Simple merger $(>)$: corresponds to $l_i = 0, r_i = 1$, e.g., stages 5 and 6 in Fig. 1. A simple merger is a time-reversed simple expansion, reducing the number of states by a factor of q.

4) Butterfly ($\mathtt{x}$ or $\circlearrowleft$): corresponds to $l_i = 1, r_i = 1$, e.g., stage 3 in Fig. 1. There are q edges out of each vertex at depth $i - 1$ and q edges into each vertex at depth i, hence the number of states is constant.

The butterfly structure is degenerate at the ith stage when the edge span of some row of the MSGM is $\{i\}$. This implies that the corresponding information bit is uncoded and the code's minimum distance is $d = 1$. A nondegenerate butterfly ($\mathtt{x}$) has q^2 edges, while a degenerate butterfly ($\circ$) has only q edges.

The total numbers of primitive structures in the trellis are denoted by $N_-, N_<, N_>, N_{\mathtt{x}}$, and $N_\circ$. For example, the trellis in Fig. 1 has $N_< = 3 = N_>, N_{\mathtt{x}} = 2, N_- = 4, N_\circ = 0$. Because the trellis has exactly one initial node and one terminal node, the total number of simple expansions must equal the total number of simple mergers: $N_< = N_>$.

Much information about the trellis for the dual code can be inferred from the trellis structure of the code. For example, if the code has a simple expansion at the ith stage, then $l_i = 1, r_i = 0$, and Lemma 1 implies $l_i^\perp = 1, r_i^\perp = 0$, hence the trellis of the dual code also has a simple expansion structure at this stage. Repeating this procedure we find the "dual" of each primitive structure, shown in Table I. The dual of a nondegenerate butterfly is a set of q extensions, while the dual of a degenerate butterfly is a single extension. The dual relationship for primitive structures implies that $N_<^\perp = N_< = N_> = N_>^\perp$ and $N_- = qN_{\mathtt{x}}^\perp + N_\circ^\perp$.

III. COMPLEXITY MEASURES FOR MINIMAL TRELLISES

A. Definitions and Basic Relationships

For a minimal trellis, the total number of vertices at depth i is q^{v_i} and the total number of edges at stage i is q^{e_i}, where v_i and e_i are the vertex and edge space dimensions [7]

$$v_i = k - f_i - p_i, \qquad i = 0, 1, \cdots, n \tag{1}$$

$$e_i = k - f_i - p_{i-1}, \qquad i = 1, \cdots, n. \tag{2}$$

The most commonly used measure of Viterbi decoding complexity for a minimal trellis, cited as one of the essential characteristics of any code [22], is the maximum dimension of its state space

$$s_{\max} \overset{\triangle}{=} \max_i v_i. \tag{3}$$

Similarly, the maximum edge space dimension is

$$e_{\max} \overset{\triangle}{=} \max_i e_i. \tag{4}$$

Forney argues that this is a more relevant complexity measure because, unlike $s_{\max}$, this quantity cannot be reduced by combining adjacent stages of a trellis [7].

A different metric, used in the derivation of the MSGM [21], is the total length of all the edge spans of the rows of the MSGM

$$\varepsilon \overset{\triangle}{=} \sum_{j=1}^{k} \varepsilon_j$$

where ε_j denotes the length of the edge span of the jth row of the MSGM. A similar span length metric is the total length of all the vertex spans

$$\nu \overset{\triangle}{=} \sum_{j=1}^{k} \nu_j$$

where $\nu_j = \varepsilon_j - 1$ is the length of the vertex span of the jth row of the MSGM. These two metrics are equivalent to the sums of all the edge dimensions or vertex dimensions (summed over stages or depths, respectively)

$$\nu + k = \varepsilon = \sum_{i=1}^{n} e_i = k + \sum_{i=0}^{n} v_i. \tag{5}$$

It is argued in [21] that more meaningful measures of Viterbi decoding complexity are the total number of edges E, vertices V, and mergers M, rather than simply the vertex or edge dimensionality

$$E = \sum_{i=1}^{n} q^{e_i} \tag{6}$$

$$V = \sum_{i=0}^{n} q^{v_i} \tag{7}$$

$$M = \sum_{i=1}^{n} r_i q^{v_i}. \tag{8}$$

E is equal to the number of binary additions required to compute path metrics, and M is the number of q-ary comparisons required to merge trellis paths. The computational complexity of Viterbi decoding is linear in E and M [21].

By counting the edges associated with each primitive trellis structure, we find

$$E = N_- + qN_< + qN_> + q^2 N_{\mathtt{x}} + qN_\circ. \tag{9}$$

Similarly, counting the simple mergers and the mergers included in butterflies yields

$$M = N_> + qN_{\mathtt{x}} + N_\circ. \tag{10}$$

Counting the vertices on the right side of each primitive structure, we account for every vertex in the trellis except the initial node, hence

$$V - 1 = N_- + qN_< + N_> + qN_{\boxtimes} + N_{\circ} . \qquad (11)$$

Combining (9)–(11) we find

$$M = \frac{E - V + 1}{q - 1}.$$

This is the generalization of the binary version of this result found in [21].

Lemma 1 and (9)–(11) lead to the following duality relationships for the trellis dimensions and the complexity measures:

Lemma 2: For each $1 \le i \le n$, the vertex and edge space dimensions of a code and its dual are related as

$$v_i^{\perp} = v_i.$$
$$e_i^{\perp} = e_i + (1 - l_i - r_i).$$

The equivalence of vertex spaces for a code and its dual was first noted by Forney [6].

Theorem 1: The trellis complexity measures for a code and its dual are related as

$$s_{\max} = s_{\max}^{\perp}$$

$$e_{\max} - 1 \le e_{\max}^{\perp} \le e_{\max} + 1$$

$$\varepsilon^{\perp} - (n - k) = \nu^{\perp} = \nu = \varepsilon - k.$$

$$V^{\perp} = V$$

$$E^{\perp} = E + (q - 1)(N_- - N_{\circ} - qN_{\boxtimes})$$

$$M^{\perp} = M + N_- - N_{\circ} - qN_{\boxtimes}$$

For self-dual codes, Lemma 1 yields stronger results because for any such code $l_i = l_i^{\perp}$ and $r_i = r_i^{\perp}$ for all i. Consequently, as noted in [15]:

Lemma 3: For any self-dual code $\mathcal{C}$, for each $i = 1, 2, \cdots, n$, either a) $l_i = 1$ and $r_i = 0$, or b) $l_i = 0$ and $r_i = 1$. In other words, every stage corresponds to an information symbol when encoding from one direction and a parity symbol when encoding from the other direction. The only primitive trellis structures in $\mathcal{T}(\mathcal{C})$ are simple expansions and simple mergers.

The converse of Lemma 3 does not hold: a code whose minimal trellis contains only simple expansions and simple mergers need not be self-dual.

There are convenient relationships among the complexity measures for self-dual codes:

Theorem 2: For any self-dual code

$$s_{\max} = e_{\max}$$

$$V = \frac{q + 1}{2q} E + 1$$

$$M = \frac{1}{2q} E.$$

Proof: The fact that $\mathcal{T}(\mathcal{C})$ contains no butterfly structures proves the first equality. The others follow as a consequence of Lemma 3 and (9)–(11). $\qquad \square$

We can also restrict the types of structures that can appear in the minimal trellis for certain *divisible codes* [25]. A divisible code is one whose codeword weights are all multiples of some integer greater than one. Examples of divisible codes include the $(31, 10, 12)$ cyclic codes and doubly-even self-dual codes such as the extended Golay code.

Theorem 3: If $\mathcal{C}$ is such that all codeword weights are divisible by some integer $m > 2$, then $\mathcal{C}$ cannot have rate greater than $\frac{1}{2}$, and there does not exist a position i such that $l_i = r_i = 1$, i.e., $\mathcal{T}(\mathcal{C})$ contains no butterfly structures. The complexity measures are related by

$$s_{\max} = e_{\max} \qquad (12)$$

$$V \ge \frac{q + 1}{2q} E + 1 \qquad (13)$$

$$M \le \frac{1}{2q} E \qquad (14)$$

$$V^{\perp} \le \frac{q + 1}{2q} E^{\perp} + 1 \qquad (15)$$

$$M^{\perp} \ge \frac{1}{2q} E^{\perp} \qquad (16)$$

Proof: If $l_i = r_i = 1$ then the ith column begins and ends spans in the MSGM. This implies the existence of codewords of the form $x = XXX \cdots X 10^{n-i}$ and $y = 0^{i-1}(-1)XXX \cdots X$, where (-1) denotes the additive inverse of 1 in GF(q) and $XXX \cdots X$ denotes some string of symbols in GF(q). Then $x + y$ is a codeword of weight $|x| + |y| - 2$ which cannot be divisible by m. This implies $l_i + r_i \le 1$ for all i, so

$$2k = \sum_{i=1}^{n} (l_i + r_i) \le n$$

which establishes the rate constraint. The fact that $\mathcal{T}(\mathcal{C})$ can have no butterfly structures proves (12). Equations (13) and (14) then follow from (9)–(11). Since $l_i + r_i \le 1$, Lemma 1 implies $l_i^{\perp} + r_i^{\perp} \ge 1$, which gives (15) and (16). $\qquad \square$

The converse of Theorem 16 does not hold: a code is not necessarily divisible when $l_i + r_i \le 1$ for all i. If a code and its dual satisfy the conditions of Theorem 3 then the code resembles a self-dual code: the code must have rate $\frac{1}{2}$ and its trellis contains only simple expansions and simple mergers.

B. Bounds Interrelating the Complexity Measures

In this section we present some new inequalities linking one complexity measure to another. These results allow bounds on one complexity measure to immediately imply bounds on other measures.

The following lemma arises from the definitions of $s_{\max}$ and $e_{\max}$, and the fact that the vertex and edge dimensions change by no more than one unit from one index to the next.
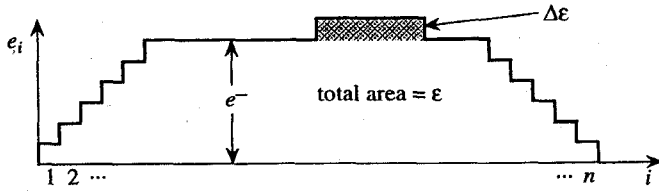
Fig. 2. An edge dimension profile that minimizes E subject to a constraint on total edge span ε.

Lemma 4: The vertex and edge dimensions are upper bounded by

$$v_i \leq \min\{i, n - i, s_{\max}\}, \qquad 0 \leq i \leq n$$
$$e_i \leq \min\{i, n + 1 - i, e_{\max}\}, \quad 1 \leq i \leq n.$$

Summing the inequalities in Lemma 4 according to (5)–(7) leads to

Theorem 4: The total complexity measures ν, ε, V, E are upper-bounded in terms of the maximum complexity measures $s_{\max}$, $e_{\max}$ by

$$\nu \leq s_{\max}(n - s_{\max})$$
$$\varepsilon \leq e_{\max}(n + 1 - e_{\max})$$
$$V \leq \left[n + \frac{q+1}{q-1} - 2s_{\max} \right] q^{s_{\max}} - \frac{2}{q-1}$$
$$E \leq \left[n + \frac{2q}{q-1} - 2e_{\max} \right] q^{e_{\max}} - \frac{2q}{q-1}.$$

Since the average edge dimension over all stages is ε/n and the average vertex dimension over the last n depths is ν/n, loose lower bounds on E and V can be obtained from (6) and (7) using Jensen's inequality and the convexity of q^{v_i} and q^{e_i} as functions of v_i and e_i.

Theorem 5: The complexity measures V, E are lower-bounded in terms of the span length complexity measures ν, ε by

$$V \geq 1 + nq^{\nu/n}$$
$$E \geq nq^{\varepsilon/n}.$$

The bounds in Theorem 5 can be tightened by using Lemma 4 to constrain how fast the vertex and edge dimension profiles can change near the ends of the trellis. For given ν or ε, a vertex or edge dimension profile such as the one in Fig. 2 minimizes V or E. This leads to the following bounds on V and E:

Theorem 6: Given a total span length ν or ε, let

$$\Delta\varepsilon = \varepsilon - e^-(n + 1 - e^-)$$

and

$$\Delta\nu = \nu - s^-(n - s^-)$$

where e^- and s^- are the largest integers such that $\Delta\varepsilon \geq 0$, $\Delta\nu \geq 0$, $e^- \leq (n+1)/2$, and $s^- \leq n/2$. Then

$$V \geq \left[n + \frac{q+1}{q-1} - 2s^- \right] q^{s^-} - \frac{2}{q-1} + (q-1)q^{s^-}\Delta\nu$$
$$E \geq \left[n + \frac{2q}{q-1} - 2e^- \right] q^{e^-} - \frac{2q}{q-1} + (q-1)q^{e^-}\Delta\varepsilon.$$

Notice the similarity of these lower bounds in terms of s^- and e^- with the corresponding upper bounds of Theorem 4 in terms of $s_{\max}$ and $e_{\max}$.

C. Bounds from MSGM Span Length

In this section we present some lower bounds on complexity that are derived from MSGM span length. We refer to an (n, k, d) code with dual distance $d^\perp$ as an $(n, k, d, d^\perp)$ code.

Every row of an MSGM for an $(n, k, d, d^\perp)$ code must have edge-span length $\varepsilon_i \geq d$ and vertex-span length $\nu_i \geq d - 1$. Applying this simple bound to both the code and its dual and using Theorem 1 leads to the following lower bounds on ν and ε.

Theorem 7: The total lengths ν and ε of the vertex spans and edge spans for any $(n, k, d, d^\perp)$ code are lower bounded by

$$\nu \geq \max\left\{ k(d - 1), (n - k)(d^\perp - 1) \right\}$$
$$\varepsilon \geq k + \max\left\{ k(d - 1), (n - k)(d^\perp - 1) \right\}.$$

Applying the Singleton bound to these inequalities gives the weaker bounds $\nu \geq (d-1)(d^\perp - 1)$ and $\varepsilon \geq k + (d-1)(d^\perp - 1)$.

We say that a code meeting the bounds of Theorem 7 with equality is a *minimal span code*. To construct an $(n, k, d, 2)$ binary minimal span code for any $d > 2$ and $n \geq d + (k - 1)\lceil \frac{d}{2} \rceil$, let the first row of the MSGM be $1^d 0^{n-d}$ and form each successive row by cyclically shifting the previous row at least $\lceil \frac{d}{2} \rceil$ positions but not more than d positions to the right, such that the total of all the shifts is $n - d$. We use the notation $MS^*(k, d)$ to refer to a minimal-span code of minimal length $n = d + (k - 1)\lceil \frac{d}{2} \rceil$, i.e., each shift is exactly $\lceil \frac{d}{2} \rceil$ positions. $MS^*(k, d)$ is a divisible code when d is even.

The dual of a minimal-span code is also a minimal-span code. These codes are not usually good in terms of distance, but they have very low complexity trellises, as we shall see in Section IV-C.

The span length bounds of Theorem 7, combined with the bounds of Theorems 4–6 lead to lower bounds on the complexity measures $s_{\max}$, $e_{\max}$, V, E:

Theorem 8: For any $(n, k, d, d^\perp)$ code

$$s_{\max}(n - s_{\max}) \geq \max\left\{ k(d-1), (n-k)(d^\perp - 1) \right\}$$
$$e_{\max}(n + 1 - e_{\max}) \geq k + \max\left\{ k(d-1), (n-k)(d^\perp - 1) \right\}$$

and V and E are lower-bounded by substituting the right-hand sides of the bounds in Theorem 7 for ν and ε in Theorem 5 or Theorem 6.

This theorem implies connections between complexity and asymptotic coding gain kd/n. For instance, $e_{\max}$, $s_{\max} + k/n$, $\log_q(E/n)$, and $\log_q((V - 1)/n) + k/n$ are all lower-bounded by kd/n. A slightly weaker version of the bound of Theorem 8 on $s_{\max}$ has been proved in [17] for both linear and nonlinear codes.

IV. UNIFORMLY OPTIMAL MINIMAL TRELLISES

The results of Section II assume a fixed code and a fixed coordinate ordering. The trellis structure, and hence complexity, depend on the choice of code and its permutation. In this

section we discuss minimizing complexity over permutations of a given code or all codes with the same parameters $(n, k, d, d^\perp)$.

A. Uniformly Dominating Past/Future Dimension Profiles

To compare the complexity of two minimal trellises, we must first select the relevant complexity measure. However, in some cases one trellis may be simpler than another at every stage and depth with respect to all of the complexity measures simultaneously. We can see from the complexity measures (3)–(8) that a permutation of C that makes f_i and p_i large wherever possible will produce a low-complexity trellis.

For a given (n, k) code C, let $\{p(C), f(C)\}$ denote its *past/future dimension profile (PFDP)*: $p(C) = \{p_i(C)\}_{i=0}^n$ and $f(C) = \{f_i(C)\}_{i=0}^n$. We say that the profile $\{p(C), f(C)\}$ is *uniformly dominated* by the profile $\{p(C^*), f(C^*)\}$, denoted

$$\{p(C), f(C)\} \preceq \{p(C^*), f(C^*)\}$$

if $p_i(C) \leq p_i(C^*)$ and $f_i(C) \leq f_i(C^*)$ for all i. For the purposes of this definition, either profile may be replaced by a pseudo-PFDP $\{p^*, f^*\}$, where $p^* = \{p_i^*\}_{i=0}^n$ and $f^* = \{f_i^*\}_{i=0}^n$ satisfy the constraints that $\{p_i^*\}_{i=0}^n$ must increase in steps of 0 or 1 from $p_0^* = 0$ to $p_n^* = k$ and $\{f_i^*\}_{i=0}^n$ must decrease in steps of 0 or 1 from $f_0^* = k$ to $f_n^* = 0$. If the pseudo-PFDP is symmetric, $p_i^* = f_{n-i}^*$, we may refer to it by specifying only p^*.

If $\{p^*, f^*\}$ is a valid pseudo-PFDP we associate with it a profile of pseudo-trellis dimensions $\{v_i^*\}_{i=0}^n$ and $\{e_i^*\}_{i=1}^n$ defined by substituting p_i^* and f_i^* into (1) and (2). Similarly, we associate with $\{p^*, f^*\}$ a suite of pseudo-trellis complexity measures $s_{max}^*, e_{max}^*, \nu^*, \varepsilon^*, V^*, E^*$, and M^* defined by substituting $\{v_i^*\}_{i=0}^n$, $\{e_i^*\}_{i=1}^n$, and $\{r_i^* = p_i^* - p_{i-1}^*\}_{i=1}^n$ into (3)–(8). Finally, we define a dual pseudo-PFDP $\{p^{*\perp}, f^{*\perp}\}$ by substituting p_i^* and f_i^* into the last two equations of Lemma 1. All of these definitions are consistent with the actual trellis dimensions and complexity measures when $\{p^*, f^*\}$ is the PFDP of some code C^*, so we usually drop the distinction between PFDP's and pseudo-PFDP's.

The following duality relationship is a straightforward result of the definitions:

Theorem 9: Uniform dominance of the PFDP's of codes and their duals are equivalent: $\{p(C), f(C)\} \preceq \{p^*, f^*\}$ if and only if

$$\{p(C^\perp), f(C^\perp)\} \preceq \{p^{*\perp}, f^{*\perp}\}.$$

Uniform dominance can be used to obtain bounds on all the complexity measures:

Theorem 10: If

$$\{p(C), f(C)\} \preceq \{p^*, f^*\}$$

for some valid uniformly dominating PFDP with (pseudo-)trellis complexity measures defined above, then

$$s_{max}(C) \geq s_{max}^*$$
$$e_{max}(C) \geq e_{max}^*$$
$$\nu(C) \geq \nu^*$$

$$\varepsilon(C) \geq \varepsilon^*$$
$$V(C) \geq V^*$$
$$E(C) \geq E^*$$

and

$$M(C) \geq M^*.$$

Proof: Inequalities for $s_{max}, e_{max}, \nu, \varepsilon, V, E$ follow immediately from the definitions. It remains to show that $M(C) \geq M^*$. From (8)

$$M = \sum_{i=1}^n r_i q^{v_i} = \sum_{i=1}^n r_i q^{k - f_i - p_i} = q^k \sum_{i=1}^n r_i q^{-f_i} q^{-p_i}.$$

Now $r_i = 1$ in precisely the k places where p_i is incremented, so the nonzero values of $r_i q^{-p_i}$ are $q^{-1}, q^{-2}, \cdots, q^{-k}$, which gives

$$M = q^k \sum_{j=1}^k q^{-j} q^{-f_{R_j}}$$

where R_j is the position of the jth 1 in $(r_1, r_2, \cdots r_n)$. A similar expression holds for the (pseudo-)trellis merger count M^*, if we define R_j^* to be the position of the jth 1 in $(r_1^*, r_2^*, \cdots r_n^*)$. Uniform dominance implies $p_i^* \geq p_i(C)$, thus $R_j^* \leq R_j(C)$. Uniform dominance and the monotonicity of $\{f_i^*\}_{i=0}^n$ imply

$$f_{R_j^*}^* \geq f_{R_j^*}(C) \geq f_{R_j(C)}(C).$$

Thus

$$M^* = q^k \sum_{j=1}^k q^{-j} q^{-f_{R_j^*}^*} \leq q^k \sum_{j=1}^k q^{-j} q^{-f_{R_j(C)}(C)} = M(C).$$

$\square$

The bounds of Theorem 10 can sometimes be tightened by invoking additional constraints on the trellis dimensions. For example, we can exploit the constraint that v_i and e_i must be nonnegative, or that $e_i \geq 1$ for a code with dual distance $d^\perp > 1$. Recent work by Lafourcade and Vardy [18] incorporates more sophisticated constraints on $\{v_i\}_{i=0}^n$ and $\{e_i\}_{i=1}^n$ as a means of tightening the bounds.

Finally, we have a strong converse theorem: if a uniformly dominating PFDP exists and any one of four measures of total complexity attains the corresponding bound of Theorem 10, then all seven of the complexity measures must attain their bounds. Specifically

Theorem 11: Suppose that $\{p(C), f(C)\} \preceq \{p^*, f^*\}$ for all codes $C \in \mathcal{Q}$, where $\mathcal{Q}$ is an arbitrary class of codes. If for some $C^* \in \mathcal{Q}$ any of the complexity measures $\varepsilon(C^*), \nu(C^*), E(C^*)$, or $V(C^*)$ meets the corresponding bound $\varepsilon^*, \nu^*, E^*$, or V^* of Theorem 10 with equality, then for all codes $C \in \mathcal{Q}$

$$\{p(C), f(C)\} \preceq \{p(C^*), f(C^*)\} = \{p^*, f^*\}.$$

Proof: Suppose, for some $0 < i < n$, either $p_i(\mathcal{C}^*) < p_i^*$ or $f_i(\mathcal{C}^*) < f_i^*$. Then from (1) and (2), and uniform dominance, $v_i(\mathcal{C}^*) > v_i^*$ and either $e_i(\mathcal{C}^*) > e_i^*$ or $e_{i+1}(\mathcal{C}^*) > e_{i+1}^*$. But this would imply from (5)–(7) that $\nu(\mathcal{C}^*) > \nu^*$, $\varepsilon(\mathcal{C}^*) > \varepsilon^*$, $V(\mathcal{C}^*) > V^*$, and $E(\mathcal{C}^*) > E^*$, contradicting the assumption that one of these complexity measures attains its bound. $\square$

If the PFDP of $\mathcal{C}^*$ uniformly dominates that of any other code within a set $\mathcal{Q}$, we say that the code $\mathcal{C}^*$ and its minimal trellis $\mathcal{T}(\mathcal{C}^*)$ are *uniformly optimal* over $\mathcal{Q}$. Our motivation for defining uniform optimality and studying its consequences lies in the correspondingly strong results obtained for the problem of finding a minimal trellis in the first place, i.e., finding the least complex trellis that represents a fixed permutation of a fixed code. As shown in [21], the minimal trellis is uniformly less complex at every stage and depth than any other trellis that represents the code.

The most prominent research on bounding the complexity of minimal trellises has been based on finding a uniformly dominating past/future dimension profile $\{p^*, f^*\}$. This is the premise of the dimension/length profile bounds discussed in the next sections.

B. Uniformly Efficient Permutations of a Code

In this section we consider re-ordering the symbols of a code to reduce the trellis complexity. Massey calls this "the *art* of trellis decoding" [20].

Let $\mathcal{S}_n$ denote the set of all permutations of $\{1, 2, \cdots n\}$, and for any $\pi \in \mathcal{S}_n$, let $\mathcal{C}\pi$ denote the code $\mathcal{C}$ with coordinates re-ordered according to π. For a given code $\mathcal{C}$ the *dimension/length profile (DLP)* [7], [10], [24] is the set of subcode dimensions $\{K_i(\mathcal{C})\}_{i=0}^n$ satisfying

$$K_i(\mathcal{C}) = \max_{\pi \in \mathcal{S}_n} p_i(\mathcal{C}\pi) = \max_{\pi \in \mathcal{S}_n} f_{n-i}(\mathcal{C}\pi).$$

It can be shown that $\{K_i(\mathcal{C})\}_{i=0}^n$ increases in steps of 0 or 1 from $K_0(\mathcal{C}) = 0$ to $K_n(\mathcal{C}) = k$ [7]. Lemma 1 leads to the well-known duality relationship $K_i(\mathcal{C}^\perp) = i - k + K_{n-i}(\mathcal{C})$ [7], [10], [26]. Much research has been devoted to determining DLP's, which contain the same information about a code as the minimum support weights or generalized Hamming weights (GHW) [3], [5], [8], [9], [12], [13], [23], [26], [27].

The DLP has been used to bound the trellis dimensions and complexity [7], [10], [18], [24]. In the language of the previous section, the DLP defines a symmetric uniformly dominating PFDP for all permutations $\mathcal{C}\pi$ of a fixed code $\mathcal{C}$. Thus by Theorem 10, all of the complexity measures for $\mathcal{C}\pi$ are lower-bounded by the corresponding expressions based on the DLP.

For a fixed code $\mathcal{C}$, we say that a permutation π^* and the corresponding minimal trellis $\mathcal{T}(\mathcal{C}\pi^*)$ are *uniformly efficient* if

$$\{p(\mathcal{C}\pi), f(\mathcal{C}\pi)\} \preceq \{p(\mathcal{C}\pi^*), f(\mathcal{C}\pi^*)\}$$

for all $\pi \in \mathcal{S}_n$. Clearly, π^* is uniformly efficient for $\mathcal{C}$ if and only if

$$p_i(\mathcal{C}\pi^*) = K_i(\mathcal{C}) = f_{n-i}(\mathcal{C}\pi^*).$$

Such permutations have been referred to as "efficient" [7] or "strictly optimum" [10] orderings. From Theorem

11, if any of the four total complexity measures $\varepsilon(\mathcal{C}\pi^*)$, $\nu(\mathcal{C}\pi^*)$, $E(\mathcal{C}\pi^*)$, $V(\mathcal{C}\pi^*)$ attains the corresponding DLP bound, then π^* is uniformly efficient. From Theorem 9 and the DLP duality relationship, a permutation π^* is uniformly efficient for $\mathcal{C}$ if and only if π^* is uniformly efficient for $\mathcal{C}^\perp$.

The standard permutation of any Reed–Muller code is uniformly efficient [10]. The next section gives additional examples of codes with uniformly efficient permutations.

In the remainder of this section we include some theoretical results that impose necessary conditions on uniformly efficient permutations. These conditions can be used to identify codes which cannot meet the DLP bounds.

If π' denotes the permutation that maps coordinates into the order

$$(1, 2, \cdots, i, \ n - j + 1, n - j + 2, \cdots, n, \ i + 1, i + 2, \cdots, n - j)$$

then for any permutation π of an (n, k) code $\mathcal{C}$

$$K_{i+j}(\mathcal{C}) \geq p_{i+j}(\mathcal{C}\pi\pi') = p_i(\mathcal{C}\pi) + f_{n-j}(\mathcal{C}\pi).$$

In the special case where π is a uniformly efficient permutation for $\mathcal{C}$, this gives:

Theorem 12: If an (n, k) code $\mathcal{C}$ has a uniformly efficient permutation, then for any i, j such that $i + j \leq n$

$$K_{i+j}(\mathcal{C}) \geq K_i(\mathcal{C}) + K_j(\mathcal{C}).$$

The fact that $K_d(\mathcal{C}) = 1$ and $K_{d^\perp}(\mathcal{C}^\perp) = 1$ for any $(n, k, d, d^\perp)$ code leads to

Theorem 13: Suppose π^* is a uniformly efficient permutation for an $(n, k, d, d^\perp)$ code $\mathcal{C}$. Then $\mathcal{C}\pi^*$ contains codewords of the form $X^d 0^{n-d}$, $0^{n-d} X^d$ and $\mathcal{C}^\perp \pi^*$ contains codewords of the form $X^{d^\perp} 0^{n-d^\perp}$, $0^{n-d^\perp} X^{d^\perp}$, where 0^j denotes j consecutive zeros, and X^j denotes some sequence of j nonzero symbols from GF (q).

Corollary 1: If a binary $(n, k, d, d^\perp)$ code $\mathcal{C}$ has a uniformly efficient permutation π^*, then $\min(d, d^\perp)$ must be even.

Proof: From Theorem 13

$$1^d 0^{n-d} \in \mathcal{C}\pi^*$$

and

$$1^{d^\perp} 0^{n-d^\perp} \in \mathcal{C}^\perp \pi^*$$

but if $\min(d, d^\perp)$ is odd then these sequences cannot be orthogonal. $\square$

By Corollary 1, the $(23, 12, 7, 8)$ Golay code has no uniformly efficient permutation, neither does the $(2^m - 1, 2^m - m - 1, 3, 2^{m-1})$ Hamming code for any $m \geq 3$. Consequently, no nontrivial perfect binary linear code has a uniformly efficient permutation.

If a trellis converges to a single vertex at depth i, i.e., $v_i = 0$, for some $i \neq 0$ or n, then $\mathcal{C}$ is a *direct-sum* code [19, p. 76]. The direct sum of an (n_1, k_1) code $\mathcal{C}_1$ and an (n_2, k_2) code $\mathcal{C}_2$ is an $(n_1 + n_2, k_1 + k_2)$ code, denoted $\mathcal{C}_1 \oplus \mathcal{C}_2$, consisting of all codewords of the form $c_1 | c_2$ (i.e., c_1 followed by c_2) where $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$. The DLP of a direct-sum code can be computed directly from the DLP's of its component codes.

Theorem 14: For any $0 \leq m \leq n_1 + n_2$

$$K_m(\mathcal{C}_1 \oplus \mathcal{C}_2) = \max_{\substack{i+j=m \\ 0 \leq i \leq n_1 \\ 0 \leq j \leq n_2}} [K_i(\mathcal{C}_1) + K_j(\mathcal{C}_2)]$$

Proof: For any permutation π,

$$\mathcal{P}_m[(\mathcal{C}_1 \oplus \mathcal{C}_2)\pi] = (\mathcal{Q}_1 \oplus \mathcal{Q}_2)\pi$$

for some linear subcodes $\mathcal{Q}_1 \subseteq \mathcal{C}_1$, $\mathcal{Q}_2 \subseteq \mathcal{C}_2$. Clearly we maximize

$$\dim[\mathcal{P}_m[(\mathcal{C}_1 \oplus \mathcal{C}_2)\pi]] = \dim\mathcal{Q}_1 + \dim\mathcal{Q}_2$$

when $\dim\mathcal{Q}_1 = K_i(\mathcal{C}_1)$ and $\dim\mathcal{Q}_2 = K_j(\mathcal{C}_2)$ for some $i + j = m$. $\square$

The definition of uniform efficiency has the desirable property that, if $\mathcal{T}(\mathcal{C})$ is uniformly efficient, then $\mathcal{T}(\mathcal{C} \oplus \mathcal{C})$ is also uniformly efficient. However, combining Theorems 12 and 14 we can show that if $\mathcal{T}(\mathcal{C}_1)$ and $\mathcal{T}(\mathcal{C}_2)$ are uniformly efficient minimal trellises for two different codes, it is not true in general that $\mathcal{C}_1 \oplus \mathcal{C}_2$ has a uniformly efficient permutation, even though $\mathcal{T}(\mathcal{C}_1 \oplus \mathcal{C}_2)$ cannot be improved upon according to any of the complexity measures. It can be argued that $\mathcal{T}(\mathcal{C}_1 \oplus \mathcal{C}_2)$ is efficient, though not uniformly so.

Although many codes lack uniformly efficient permutations, for many such codes there exists some permutation that simultaneously minimizes all of the trellis complexity measures, e.g., the $(7, 4)$ Hamming code. For self-dual codes, Theorem 2 tells us that there is always a single permutation that simultaneously minimizes E, V, and M. We do not know whether this is true in general.

C. Uniformly Concise Codes

We now consider the problem of minimizing trellis complexity over all codes with given $(n, k, d, d^{\perp})$. The inclusion of $d^{\perp}$ as a parameter of the optimization domain elucidates symmetries that are hidden by consideration of only n, k, and d.

Let $\mathcal{Q}(n, k, d, d^{\perp})$ be the set of all $(n, k, d, d^{\perp})$ codes over GF(q). For any $(n, k, d, d^{\perp})$, the *upper dimension/length profile* (UDLP) is the set of subcode dimensions $\{\overline{K}_i(n, k, d, d^{\perp})\}_{i=0}^{n}$ satisfying

$$\overline{K}_i(n, k, d, d^{\perp}) = \max_{\mathcal{C} \in \mathcal{Q}(n, k, d, d^{\perp})} K_i(\mathcal{C}).$$

Since each $K_i(\mathcal{C})$ is associated with a linear subcode of $\mathcal{C}$, many authors have used bounds on the best possible linear codes (i.e., codes with the largest possible minimum distance) to upper-bound the UDLP [5], [7], [9], [10], [14], [15], [22], [24]: $\overline{K}_i(n, k, d, d^{\perp}) \leq \widehat{K}_i(n, k, d, d^{\perp})$ where

$$\widehat{K}_i(n, k, d, d^{\perp}) \triangleq \min[k_{\max}(i, d), k - n + i + k_{\max}(n - i, d^{\perp})]$$

and $k_{\max}(m, d)$ is the largest possible dimension for any q-ary linear block code of length m and minimum distance d. The set $\{\widehat{K}_i(n, k, d, d^{\perp})\}_{i=0}^{n}$ is called the *"best codes" dimension/length profile* (BCDLP) for the code parameters $(n, k, d, d^{\perp})$. Both the UDLP and the BCDLP define symmetric uniformly dominating PFDP's for all codes

$\mathcal{C} \in \mathcal{Q}(n, k, d, d^{\perp})$. Hence Theorem 10 implies corresponding UDLP and BCDLP lower bounds for all of the complexity measures.

Bounds based on the BCDLP are important practically, because much data about the best possible codes has been tabulated [2], and some codes achieve these bounds with equality. However, for many combinations of $(n, k, d, d^{\perp})$ it is not possible for a single code and its dual to both have a series of subcodes, all with the maximum code dimensions.

We say that an $(n, k, d, d^{\perp})$ code $\mathcal{C}^*$ and its corresponding minimal trellis $\mathcal{T}(\mathcal{C}^*)$ are *uniformly concise* if

$$\{\boldsymbol{p}(\mathcal{C}), \boldsymbol{f}(\mathcal{C})\} \preceq \{\boldsymbol{p}(\mathcal{C}^*), \boldsymbol{f}(\mathcal{C}^*)\}$$

for all $(n, k, d, d^{\perp})$ codes $\mathcal{C}$. Clearly, $\mathcal{C}^*$ is uniformly concise if and only if

$$p_i(\mathcal{C}^*) = \overline{K}_i(n, k, d, d^{\perp}) = f_{n-i}(\mathcal{C}^*).$$

Also, from Theorem 11, $\mathcal{C}^*$ is uniformly concise if any of the four total complexity measures $\varepsilon(\mathcal{C}^*)$, $\nu(\mathcal{C}^*)$, $E(\mathcal{C}^*)$, or $V(\mathcal{C}^*)$ meet the UDLP or BCDLP bounds with equality. Both the UDLP and BCDLP definitions are consistent with that of the corresponding dual PFDP: the PFDP for parameters $(n, n-k, d^{\perp}, d)$ equals the dual PFDP for $(n, k, d, d^{\perp})$. Hence, from Theorem 9, $\mathcal{C}^*$ is uniformly concise if and only if $\mathcal{C}^{*\perp}$ is uniformly concise.

Uniformly concise codes are optimum in a very strong sense. Not only do they have an efficient permutation, but they are "concise" in the sense that they have the smallest possible trellis compared to all codes with the same parameters.

Table II lists known uniformly concise binary codes. In each case, the complexities listed are the lowest possible values for any code with the same parameters. Uniform conciseness of first-order Reed–Muller codes follows from the explicit GHW derived by Wei [26] together with the result of Kasami *et al.* [10] that the standard permutation is uniformly efficient. Other examples in the table are obtained by discovering a column permutation that meets the BCDLP. In [4] we reported such a permutation for the $(48, 24, 12, 12)$ quadratic residue code. Permutations that meet the BCDLP for the $(24, 12, 8, 8)$ extended Golay code [6], the $(32, 16, 8, 8)$ second-order Reed–Muller code [10], and the $(16, 7, 6, 4)$ lexicode [16] have been reported elsewhere.

We can use the results of the previous sections to find examples of code parameters $(n, k, d, d^{\perp})$ for which no uniformly concise trellis can exist. In the remainder of this section we provide some illustrations of the types of arguments that can be used.

If $\mathcal{C}\pi^*$ is uniformly concise then clearly π^* must be uniformly efficient for $\mathcal{C}$. This fact combined with Theorem 12 can be used to show, for example, that no binary $(6, 3, 2, 2)$ code is uniformly concise.

For another example, Theorem 13 combined with the properties of the DLP and BCDLP can be used to show that there is no binary $(18, 9, 6, 6)$ uniformly concise code. So the quadratic residue code with these parameters cannot have a uniformly efficient permutation.

Since minimal-span codes minimize ε, every minimal-span code is either uniformly concise, or no uniformly concise code

TABLE II
SOME UNIFORMLY CONCISE BINARY CODES

(Codes are grouped with their duals, which are also uniformly concise. Complexity measures M, ε, can be computed as $M = E - V + 1, \varepsilon = \nu + k$.
Notes: [1] Complexity expressions for first-order Reed–Muller and extended Hamming codes are valid for $m \geq 3$, except $e_{\max} = 3$ when
$m = 3$. [2] $\mathcal{C} \uplus x$ denotes the code formed by taking all linear combinations of codewords in $\mathcal{C} \cup \{x\}$. [3] The code is
$MS^*(5,8) \uplus (01)^4(0011)^2 0^8 \uplus (1100)^2(10)^4(1100)^2 \uplus 0^8(0011)^2(01)^4$. [4] The code is $MS^*(4,16) \uplus (10)^{20} \uplus (1100)^{10} \uplus (1^4 0^4)^5$.)

parameters	code	E	V	$s_{\max}$	$e_{\max}$	ν
$(d+(k-1)\lceil\frac{d}{2}\rceil, k, d, 2)$ [dual]	$MS^*(k,d)$, $d > 2$, $k \leq 3$	$2kd$ $4k(d-2)+4$	$2+2k(d-1)$	$\min\{2, d-2\}$	2 2	$k(d-1)$
$(2^m, 1+m, 2^{m-1}, 4)$ [dual]	[1] Reed–Muller $\mathcal{R}(1,m)$ [10], [26] [1] extended Hamming	$(2^{2m+1}+2^4)/3$ -4 $(2^{2m+2}+2^5)/3$ $-4-3(2^{m+1})$	$(2^{2m+1}+2^4)/3$ $-3(2^{m-1})-2$	m	m $m+1$	$(m-1)(2^m-1)$
$(24,12,8,8)$	extended Golay $\mathcal{G}_{24}$ [6] self-dual	3580	2686	9	9	124
$(32,16,8,8)$	Reed–Muller $\mathcal{R}(2,6)$ [10] self-dual	6396	4798	9	9	186
$(48,24,12,12)$	quadratic residue [4] self-dual	860156	645118	16	16	478
$(16,7,6,4)$ [dual]	lexicode [16]	236 252	174	5	5 5	47
$(10,5,4,4)$	[2] $MS^*(4,4) \uplus (10)^5$ formally self-dual	60	46	3	3	19
$(12,6,4,4)$	[2] $MS^*(5,4) \uplus (10)^6$ formally self-dual	76	58	3	3	24
$(16,4,8,2)$ [dual]	[2] $MS^*(3,8) \uplus (1100)^4$	88 132	78	3	3 4	32
$(20,6,8,4)$ [dual]	[2] $MS^*(4,8) \uplus (10)^{10} \uplus (1100)^5$	236 348	206	4	4 5	60
$(24,7,8,4)$ [dual]	[2] $MS^*(5,8) \uplus (10)^{12} \uplus (1100)^6$	300 444	262	4	4 5	75
$(24,8,8,4)$ [dual]	[3]	364 476	302	5	5 5	78
$(40,7,16,4)$ [dual]	[4]	940 1628	878	5	5 6	163
$(16,8,4,4)$	$\mathcal{R}(1,3) \oplus \mathcal{R}(1,3)$ self-dual	88	67	3	3	28
$(48,24,8,8)$	$\mathcal{G}_{24} \oplus \mathcal{G}_{24}$ self-dual	7160	5371	9	9	248

exists with the same parameters. $MS^*(4,3)$ is a $(9,4,3,2)$ minimal-span code. If $\mathcal{C}$ is the direct sum of the $(6,3,3,3)$ shortened Hamming code, and the $(3,1,3,2)$ repetition code, then $\mathcal{C}$ is a $(9,4,3,2)$ code with superior DLP but larger edge span length ε, thus there is no $(9,4,3,2)$ uniformly concise code.

D. Full Minimal Trellises and Uniformly Inefficient Permutations

For any (n,k) code, p_i and f_i both reach maximum values of k ($f_0 = k$ and $p_n = k$) and can fall from these values at a maximum rate of one unit per trellis stage. The trellis dimensions are therefore lower-bounded [7] as $p_i \geq \underline{K}_i(n,k)$, $f_i \geq \underline{K}_{n-i}(n,k)$, where

$$\underline{K}_i(n,k) \triangleq \max(0, k - n + i).$$

The set $\{\underline{K}_i(n,k)\}_{i=0}^n$ is called the *lower dimension/length profile (LDLP)* for the code parameters (n,k).

The LDLP defines a valid symmetric PFDP for an easily constructed (n,k) code $\mathcal{C}^*$. (To construct the jth row of the MSGM of one such code, place 1's in the jth and $(n-k+j)$th positions and 0's elsewhere.) The PFDP of $\mathcal{C}^*$ is uniformly dominated by the PFDP of any other (n,k) code $\mathcal{C}$. Thus

we can apply Theorem 10 in reverse to conclude that all of the complexity measures are upper-bounded by the values associated with the LDLP.

The LDLP bound on maximum state complexity is the well-known Wolf bound: $s_{\max}(\mathcal{C}) \leq \min(k, n-k)$ [28]. The corresponding bound on maximum edge complexity is $e_{\max}(\mathcal{C}) \leq \min(k, n+1-k)$. Explicit expressions for the LDLP bounds on $\nu(\mathcal{C})$, $\varepsilon(\mathcal{C})$, $V(\mathcal{C})$, and $E(\mathcal{C})$ are the same as those obtained by substituting these bounds on $s_{\max}(\mathcal{C})$ and $e_{\max}(\mathcal{C})$ into the inequalities of Theorem 4. Thus the bounds of Theorem 4 are tighter than the LDLP bounds, except when the bounds of both theorems are met with equality.

For a fixed code $\mathcal{C}$, we say that a permutation π^* and the corresponding minimal trellis $\mathcal{T}(\mathcal{C}\pi^*)$ are *uniformly inefficient* if

$$\{p(\mathcal{C}\pi^*), f(\mathcal{C}\pi^*)\} \preceq \{p(\mathcal{C}\pi), f(\mathcal{C}\pi)\}$$

for all $\pi \in \mathcal{S}_n$. From Theorem 9, a permutation π^* is uniformly inefficient for $\mathcal{C}$ if and only if π^* is uniformly inefficient for $\mathcal{C}^\perp$. Many codes have uniformly inefficient minimal trellises in their standard permutations, e.g., cyclic, extended cyclic, and shortened cyclic codes [11], [15]. Additional examples are given in the following two theorems.

Theorem 15: A self-dual code always has a uniformly inefficient permutation.

Proof: By Lemma 3, any $(2k, k)$ self-dual code has k stages of simple expansions and k stages of simple mergers. The columns of the MSGM corresponding to the expansion stages form a linearly independent set, as do the columns corresponding to the mergers. Any permutation which groups all k of the expansion columns followed by all k of the merger columns is uniformly inefficient. □

Theorem 16: If and only if a code is maximum-distance-separable (MDS), every permutation π is uniformly inefficient.

This theorem was proved by Forney [7]; it follows from the fact that a code is MDS if and only if every subset of k columns of its generator matrix is linearly independent. A peculiar consequence of Theorem 16 is that every permutation of an MDS code is also uniformly efficient, and the code is uniformly concise. In fact, the lower bounds of Theorem 8, which are generally looser than the BCDLP bounds, are also met with equality for an MDS code (with bounds on V and E derived from Theorem 6, not Theorem 5). However, we prefer to emphasize the inefficiency of MDS trellises, because no other (n, k) code can have a more complex minimal trellis than that of an MDS code.

We say that the minimal trellis of an (n, k) code C is *full* if $p_i(C) = \underline{K}_i(n, k)$ and $f_i(C) = \underline{K}_{n-i}(n, k)$. Equivalently, $T(C^*)$ is *full* if

$$\{\boldsymbol{p}(C^*), \boldsymbol{f}(C^*)\} \preceq \{\boldsymbol{p}(C), \boldsymbol{f}(C)\}$$

for all (n, k) codes C. A full minimal trellis reaches the maximum possible vertex dimension $\min\{k, n - k\}$ in the minimum number of stages from both directions and stays there if $k < n - k$. From Theorem 9, $T(C^*)$ is full if and only if $T(C^{*\perp})$ is full. The complexity of any full minimal trellis equals the LDLP bounds, i.e., a full minimal trellis has the maximum complexity for any (n, k) code.

In the proofs of the two previous theorems, the uniformly inefficient permutations produced full minimal trellises. This fact generalizes as follows:

Theorem 17: If π^* is a uniformly inefficient permutation of C, then $T(C\pi^*)$ is full.

Proof: If C is k-dimensional, any generator matrix for C has k linearly independent columns. Let π_1 and π_2 be two permutations which place k independent columns in the first and last k positions, respectively. If π^* is uniformly inefficient, then

$$p_i(C\pi^*) \leq p_i(C\pi_2) = \max(0, k - n + i)$$

and

$$f_i(C\pi^*) \leq f_i(C\pi_1) = \max(0, k - i)$$

i.e., π^* must achieve the LDLP bounds on both the past and the future, implying that both the first k columns and last k columns of the corresponding MSGM are linearly independent. The corresponding minimal trellis is full. □

V. CONCLUSION

In this paper we approached the trellis complexity problem by first considering the minimal span generator matrix (MSGM) for a fixed permutation of a code. After briefly describing the connection between a minimal trellis and the MSGM, we identified the primitive structures of a minimal trellis and gave some fundamental duality relationships. The ideas in this section grew from the work of [21] and closely resemble that of [15], to which the reader is referred for more details on constructing minimal trellises or extending these results to nonlinear codes.

McEliece [21] showed that the minimal trellis minimizes not only the maximum state dimension of the trellis but other complexity measures as well. Here we have augmented the list of reasonable complexity measures and have shown many connections among them and with the primitive structures. The duality properties lead to interesting relationships among several of the complexity measures for the special cases of self-dual codes and certain divisible codes.

We gave some simple inequalities bounding one complexity measure in terms of another, which can be used in conjunction with any known bounds on one complexity measure to imply bounds on the others. By applying these results, we were able to infer some simple new bounds on the maximum state or edge complexity and on the total numbers of vertices and edges by bounding instead the total span lengths associated with the MSGM. These bounds imply a simple link between complexity and asymptotic coding gain, similar to that shown in [17], where this connection is discussed in more detail.

The trellis complexity analysis for a fixed code was extended to codes allowed to vary over a domain of optimization. We looked at two useful domains, the set of permutations of a given code and the set of all codes with given parameters. Bounds on maximum state complexity derived by other authors from the dimension/length profile (DLP) of a code (see especially [7]) generalize easily to similar bounds on all the complexity measures over each domain. Conversely, we have shown that if a minimal trellis attains the bounds for certain complexity measures (total span length, total vertices, or total edges), it must necessarily be uniformly optimal, but this is not true for the simpler measures of maximum state or edge dimension usually considered by other authors. This lends credence to the argument in [21] that a measure of total complexity is more useful than a measure of maximum complexity.

We attempted to unify the theory of DLP bounds over different domains of optimization by defining the simple concept of a uniformly dominating past/future dimension profile (PFDP). A code or permutation is uniformly optimal within a given domain if its associated PFDP uniformly dominates that of any other code or permutation within the same domain, a definition that does not require reference to any particular bounds. We demonstrated that this is a useful concept by presenting several examples of uniformly optimal trellises. However, uniform optimality is such a strong requirement that in many cases we were able to prove that no uniformly optimal code or permutation can exist. This shows

that there is room for considerably more research to define what should be meant by an "optimal" code or permutation without the qualifier "uniformly."

Finally, to round out the subject, we briefly examined minimal trellises of maximal complexity. The theory is much simpler here, because one can easily construct a minimal trellis that has the maximum number of vertices and edges for any code of given length and dimension. Maximum-distance-separable codes always have full minimal trellises, as do many other codes if inefficiently permuted. It is much easier to construct full minimal trellises than to find concise ones.

ACKNOWLEDGMENT

REFERENCES

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, no., 2, pp. 284–287, Mar. 1974.

[2] A. E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 662–677, 1993.

[3] H. Chung, "The second generalized Hamming weight of double-error correcting binary BCH codes and their dual codes," *Lecture Notes in Computer Science*, vol. 539, pp. 118–129, 1991.

[4] S. Dolinar, L. Ekroot, A. Kiely, W. Lin, and R. J. McEliece, "The permutation trellis complexity of linear block codes," in *Proc. 32nd Annual Allerton Conf. on Communication, Control, and Computing* (Allerton, IL, Oct. 1994), pp. 60–74.

[5] G. L. Feng, K. K. Tzeng, and V. K. Wei, "On the generalized Hamming weights of several classes of cyclic codes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1125–1130, 1992.

[6] G. D. Forney, "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187 (Appendix A), 1988.

[7] _____, "Dimension/length profiles and trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1741–1752, Nov. 1994.

[8] T. Helleseth, T. Kløve, and J. Mykkeltveit, "The weight distribution of irreducible cyclic codes with block lengths $n_1((q^l - 1)/N))$," *Discr. Math.*, vol. 18, pp. 179–211, 1977.

[9] T. Helleseth, T. Kløve, and Ø. Ytrehus, "Generalized Hamming weights of linear codes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1133–1140, 1992.

[10] T. Kasami, T. Takata, T. Fujiwara, and S. Lin "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 242–245, 1993.

[11] _____, "On complexity of trellis structure of linear block codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1057–1064, 1993.

[12] T. Kløve, "Support weight distribution of linear codes," *Discr. Math.*, vol. 106, pp. 311–316, Sept. 1992.

[13] _____, "Minimum support weights of binary codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 648–654, 1993.

[14] A. D. Kot and C. Leung, "On the construction and dimensionality of linear block code trellises," in *Proc. 1993 IEEE Int. Symp. on Information Theory* (Budapest, Hungary), p. 291.

[15] F. R. Kschischang and V. Sorokine, "On the trellis structure of block codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 6, pp. 1924–1937, Nov. 1995.

[16] F. R. Kschischang and G. B. Horn, "A heuristic for ordering a linear block code to minimize trellis state complexity," in *Proc. 32nd Annual Allerton Conf. on Communication, Control, and Computing* (Allerton, IL, Oct. 1994), pp. 75–84.

[17] A. Lafourcade and A. Vardy, "Asymptotically good codes have infinite trellis complexity," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 555–559, Mar. 1995.

[18] _____, "Lower bounds on trellis complexity of block codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 6, pp. 1938–1954, Nov. 1995.

[19] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[20] J. L. Massey, "Foundations and methods of channel coding," in *Proc. Int. Conf. on Information Theory and Systems*, vol. 65, NTG-Fachberichte, Sept. 1978.

[21] R. J. McEliece, "On the BCJR trellis for linear block codes" *IEEE Trans. Inform. Theory*, vol. 42, no. 4, pp. 1072–1092, July 1996.

[22] D. J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, 1988.

[23] G. van der Geer and M. van der Vlugt, "On generalized Hamming weights of BCH codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 543–546, 1994.

[24] A. Vardy and Y. Be'ery, "Maximum-likelihood soft decision decoding of BCH codes," *IEEE Trans. Inform. Theory* vol. 40, pp. 546–554, 1994.

[25] H. N. Ward, "Divisible codes," *Arch. Math.*, vol. 36, pp. 485–494, 1991.

[26] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1412–1418, 1991.

[27] _____, "On the generalized Hamming weights of product codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1709–1713, 1993.

[28] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, 1978.