

# Dimension/Length Profiles and Trellis Complexity of Linear Block Codes

G. David Forney, Jr., *Fellow, IEEE*

**Abstract**—This semi-tutorial paper discusses the connections between the dimension/length profile (DLP) of a linear code, which is essentially the same as its “generalized Hamming weight hierarchy” [1], and the complexity of its minimal trellis diagram. These connections are close and deep. DLP duality is closely related to trellis duality. The DLP of a code gives tight bounds on its state and branch complexity profiles under any coordinate ordering; these bounds can often be met. A maximum distance separable (MDS) code is characterized by a certain extremal DLP, from which the main properties of MDS codes are easily derived. The simplicity and generality of these interrelationships are emphasized.

**Index Terms**—Dimension/length profiles, generalized Hamming weights, support weights, trellis diagrams, state complexity, linear codes.

## I. INTRODUCTION

THE dimension/length profile (DLP) of an  $(n, k)$  linear block code  $C$  is the sequence  $k(C)$  whose component  $k_i(C)$  is the maximum dimension of any subcode (shortened code) of  $C$  whose effective length (support size) is less than or equal to  $i$ ,  $0 \leq i \leq n$ . The length/dimension profile (LDP) is the sequence  $m(C)$  whose component  $m_j(C)$  is the minimum effective length of any subcode of  $C$  whose dimension is  $j$ ,  $0 \leq j \leq k$ . Either of these two profiles contains equivalent information about  $C$ .

Considerable interest in LDP's has been stimulated by a paper by Wei [1]. Wei calls the LDP the “generalized Hamming weight (GHW) hierarchy” of  $C$ , since  $m_1(C)$  is the minimum Hamming distance of  $C$ . In fact, we shall see that the LDP has more to do with length and dimension than with distance, which accounts for our terminology. Also, for us the DLP is somewhat more useful than the LDP.

The DLP idea is actually much older, going back at least to Helleseth *et al.* [2], where GHW's appear as the minimum nonzero elements of “support weight distributions” [3]. Simonis coined the term “effective length” [4]. It has found diverse uses; Wei's application was to the Type II wire-tap channel [1].

Manuscript received October 29, 1993. This paper was presented in part at the IEEE International Symposium on Information Theory, Trondheim, Norway, June 1994.

The author is with Motorola, Inc., 20 Cabot Boulevard, Mansfield, MA 02048.

IEEE Log Number 9406119.

Quite independently, a sizable literature has developed on trellis diagrams of linear codes. A linear code  $C$  has a well-defined minimal trellis diagram, given a definite ordering of its coordinates [5], [6]. Theoretically, the “complexity” of a linear code may be defined algebraically by the complexity of its minimal trellis diagram. Practically, trellis diagrams often lead to efficient trellis-based decoding algorithms.

Kasami *et al.* [7] connected these two topics by using Wei's results to prove that the standard coordinate ordering of a Reed-Muller code is, in fact, the ordering that minimizes the size of each state space in its minimal trellis diagram. Subsequently, Vardy and Be'ery [8] used GHW results to develop lower bounds on the state complexity of BCH codes. Ytrehus [9] has sharpened some of these bounds.

The main purpose of this paper is to show that the connections between the DLP concept and trellis complexity are close and deep.

Given a coordinate ordering, the state and branch complexity profiles of a linear code  $C$  are simple functions of the corresponding ordered DLP. Consequently, bounds valid for any ordering can be derived from the unordered DLP. An efficient ordering is one for which these bounds are met with equality. For many codes, efficient orderings are known.

There is a close relation between DLP's and MDS (maximum distance separable) codes, as has been recognized by previous authors. MDS codes may be characterized as those codes whose DLP meets a certain outer bound.

Wei [1] proved a striking duality theorem, relating the LDP of a linear code to the LDP of its dual code. We give a number of such duality relations. Probably the most notable is a simple proof, using DLP duality, that the state complexity profile of a linear code and that of its dual are identical [5]. Also, an ordering is efficient for a code if and only if it is efficient for its dual [7].

This paper is semi-tutorial. Most of its results are not new. However, we believe that the simplicity and generality of these ideas are noteworthy and important, and that this is not sufficiently apparent in the prior literature. Therefore we have made an effort to present these ideas as simply as possible, and to make clear their interrelationships. We feel that none of this material would be out of place in a first course in coding theory.

Section II develops basic duality relationships between the subcodes (shortened codes) and projections (punctured codes) of a linear code and its dual.

Section III introduces the DLP, and develops its basic properties. Elementary proofs are given of the fundamental results of Wei [1], including his duality theorem. A simple bound is given on the DLP of a linear code  $C$ , which is met if and only if  $C$  is MDS. From this result follow the main properties of MDS codes.

Section IV introduces the DLP corresponding to a definite coordinate ordering. It is shown that the state and branch complexity profiles of  $C$  are simple functions of this ordered DLP, which leads to DLP bounds on state and branch complexity. These bounds can be computed using trivial graphical manipulations. In turn, known distance bounds on linear codes yield DLP bounds. A number of examples show that these bounds are often met with equality. This leads to such results as: the state complexity profile of the Golay code with its standard coordinate ordering [5] is optimum *componentwise*. A code is MDS if and only if its state complexity profile is invariant under all coordinate permutations.

In Section V, we discuss how sectionalization can reduce apparent state complexity, and show that such a reduction does not occur with branch complexity. We propose that branch complexity ought to be regarded as more significant than state complexity. A sectionalization is called "efficient" if the maximum branch complexity is as small as possible; examples of efficient sectionalizations are given. Section V-C discusses an alternative definition of branch complexity.

Section VI is a brief conclusion, with suggestions for further research.

## II. PRELIMINARIES: PROJECTIONS, SUBCODES, AND DUALITY

An  $(n, k)$  linear code  $C$  over a field  $F$  is a  $k$ -dimensional subspace of the  $n$ -dimensional vector space  $F^n$ . The parameters  $n$  and  $k$  are the *length* and *dimension* of  $C$ . The difference  $r = n - k$  is the *redundancy* of  $C$ . We sometimes denote the length, dimension, and redundancy of  $C$  by  $n(C)$ ,  $k(C)$ , and  $r(C)$ , respectively.

Both  $k$  and  $r$  lie in the same range, namely  $[0, n]$ . The unique  $(n, n)$  code over  $F$  is  $F^n$ . The unique  $(n, 0)$  code is  $\{0\}$ .

Let  $I$  be an *index set* for  $F^n$ . Then an element of  $F^n$  is an  $n$ -tuple  $f = \{f_i, i \in I\}$ . Let  $J \subseteq I$  be any subset of  $I$ ; the complementary subset will be denoted by  $I - J$ .

Projections and subcodes defined on subsets  $J \subseteq I$  will be our fundamental tools. We shall see that, in many respects, they are dual concepts.

The *projection*  $P_J(C)$  of  $C$  onto  $J$  is the image of  $C$  under the projection operator  $P_J$ , namely, the map that zeroes out components outside of  $J$ :

$$P_J(f) \triangleq \begin{cases} f_i, & i \in J, \\ 0, & i \notin J, \end{cases}$$

$$P_J(C) \triangleq \{P_J(c) : c \in C\}.$$

The restriction of a projection  $P_J(C)$  to  $J$  is sometimes called a "punctured code" of  $C$ .

The *effective length* of a code  $C$  is the size of its *support* [4], defined as

$$\text{supp}(C) \triangleq \{i \in I : P_{\{i\}}(C) \neq \{0\}\}.$$

We denote the effective length of  $C$  by  $m(C)$ . Clearly,  $m(C) \leq n(C)$ .

A projection  $P_J(C)$  is a linear code with length  $n[P_J(C)] = n(C) = |I|$ , effective length  $m[P_J(C)] \leq |J|$ , and dimension  $k[P_J(C)] \leq k$ .

The *subcode*  $C_J$  of  $C$  is defined as the set of all code sequences whose components are all zero outside of  $J$ :

$$C_J \triangleq \{c \in C : c_i = 0, i \notin J\}.$$

The restriction of a subcode  $C_J$  to  $J$  is sometimes called a "shortened code" of  $C$ .

Alternatively,  $C_J$  may be defined as the intersection of  $C$  and  $P_J(C)$ . Thus  $C_J$  is a subcode not only of  $C$ , but also of  $P_J(C)$ .

A subcode  $C_J$  is a linear code with length  $n(C_J) = n(C) = |I|$ , effective length  $m(C_J) \leq m[P_J(C)] \leq |J|$ , and dimension  $k(C_J) \leq k[P_J(C)] \leq k$ .

The following relation between the dimensions of a projection  $P_J(C)$  and the subcode  $C_{I-J}$  is a first indication of the duality of projections and subcodes.

*Lemma 1 (first duality lemma):* If  $C$  is an  $(n, k)$  linear code and  $J \subseteq I$ , then

$$k[P_J(C)] + k(C_{I-J}) = k.$$

*Proof:* The projection map  $P_J : C \rightarrow P_J(C)$  is a homomorphism with image  $P_J(C)$  and kernel  $C_{I-J}$ , so  $P_J(C) \cong C/C_{I-J}$ .  $\square$

*Corollary (dimension lower bound):* If  $C$  is an  $(n, k)$  linear code and  $J \subseteq I$ , then

$$k(C_J) \geq k - |I - J|.$$

*Proof:* By Lemma 1,  $k(C_J) = k - k[P_{I-J}(C)] \geq k - |I - J|$ , since  $k[P_{I-J}(C)] \leq |I - J|$ .  $\square$

Thus shortening a code in  $|I - J|$  places can reduce its dimension by at most  $|I - J|$ .

The *dual code*  $C^\perp$  to a linear code  $C \subseteq F^n$  is the set of all elements of  $F^n$  that are orthogonal to all elements of  $C$ , under the usual inner product over  $F$ . If  $C$  is an  $(n, k)$  code, then  $C^\perp$  is an  $(n, r)$  linear code, where  $r = n - k$ . The dual to  $C^\perp$  is  $C$ .  $C$  is *self-dual* if  $C = C^\perp$ , which is possible only if  $k = r$ .

The following lemma shows that projections of  $C^\perp$  are dual to subcodes of  $C$ .

*Lemma 2 (second duality lemma):* If  $C$  is an  $(n, k)$  linear code and  $J \subseteq I$ , then, as subspaces of  $P_J(F^n)$ ,  $C_J$  and  $P_J(C^\perp)$  are dual codes. Consequently,

$$k[P_J(C^\perp)] + k(C_J) = |J|.$$

*Proof:*  $C_J$  is the intersection of  $C$  and  $P_J(F^n)$ . Obviously, if  $f \in P_J(F^n)$  and  $g \in F^n$ , then  $f$  and  $g$  are

orthogonal if and only if  $f$  and  $P_j(g)$  are orthogonal. Every element of  $C_j$  is orthogonal to every element of  $C^\perp$  and thus to every element of  $P_j(C^\perp)$ . Conversely, if  $f \in P_j(\mathbb{F}^n)$  is orthogonal to every element of  $P_j(C^\perp)$ , then  $f$  is orthogonal to every element of  $C^\perp$ , so  $f$  is in  $C$  and thus in  $C_j = C \cap P_j(\mathbb{F}^n)$ . Hence the dual to  $P_j(C^\perp)$  (as a subspace of  $P_j(\mathbb{F}^n)$ ) is  $C_j$ .  $\square$

*Corollary:* If  $C$  is self-dual, then  $C_j$  and  $P_j(C)$  are dual codes.

In summary, given an  $(n, k)$  linear code  $C$  with index set  $I$  and a subset  $J \subseteq I$  of size  $|J|$ , the dimensions of  $C_J$ ,  $P_{I-J}(C)$ ,  $P_J(C^\perp)$ , and  $(C^\perp)_{I-J}$  may be determined from any one of them, say  $k(C_J)$ .

*Theorem 1 (duality):* Let  $C$  and  $C^\perp$  be dual  $(n, k)$  and  $(n, r)$  linear codes, with  $r = n - k$ . Let  $J \subseteq I$ , and define  $r(C_J) \triangleq |J| - k(C_J)$ . Then

$$\begin{aligned} \dim C_J &= k(C_J), & \dim P_{I-J}(C) &= k - k(C_J), \\ \dim P_J(C^\perp) &= r(C_J), & \dim (C^\perp)_{I-J} &= r - r(C_J). \end{aligned}$$

Simonis [4], using punctured and shortened codes, proves Lemmas 1 and 2, but does not state Theorem 1.

As an immediate corollary, we may state a generalization of Wei's Theorem 2 [1], even though we have not yet defined the "generalized Hamming weights"  $m_j(C)$ .

*Corollary:* If  $C$  and  $C^\perp$  are dual  $(n, k)$  and  $(n, r)$  linear codes, then

$$\begin{aligned} m_j(C) &\triangleq \min_J \{ |J|: k(C_J) = j \} \\ &= \min_J \{ |J|: k - k[P_{I-J}(C)] = j \} \\ &= \min_J \{ |J|: |J| - k[P_J(C^\perp)] = j \} \\ &= \min_J \{ |J|: k[(C^\perp)_{I-J}] - r + |J| = j \}. \end{aligned}$$

Wei's Theorem 2 is the third of these identities.

### III. DIMENSION / LENGTH PROFILES

The *dimension/length profile* (DLP) of  $C$  will be defined as the sequence

$$\mathbf{k}(C) = \{k_i(C), 0 \leq i \leq n\},$$

whose  $i$ th component  $k_i(C)$  is the maximum dimension of any subcode  $C_J$  with  $|J| = i$ :

$$k_i(C) \triangleq \max_J \{k(C_J): |J| = i\}, \quad 0 \leq i \leq n.$$

Alternatively,  $k_i(C)$  is the maximum dimension of any subcode  $C_J$  whose effective length is not greater than  $i$ :

$$k_i(C) = \max_J \{k(C_J): m(C_J) \leq i\}, \quad 0 \leq i \leq n.$$

Obviously  $k_i(C)$  is nondecreasing with  $i$ , and  $k_0(C) = 0$ ,  $k_n(C) = k$ . Furthermore, by the corollary to Lemma 1, the increments  $k_{i+1}(C) - k_i(C)$  can be at most 1. Therefore,  $\mathbf{k}(C)$  rises from 0 to  $k$  in  $k$  distinct unit steps.

Equivalently, we may define the *length/dimension profile* (LDP) of  $C$ , whose  $j$ th component is the minimum effective length of any subcode whose dimension is  $j$ :

$$\begin{aligned} m(C) &= \{m_j(C), 0 \leq j \leq k\} \\ m_j(C) &\triangleq \min_J \{ |J|: k(C_J) = j \}, \quad 0 \leq j \leq k. \end{aligned}$$

Alternatively,

$$m_j(C) = \min_J \{m(C_J): k(C_J) = j\}, \quad 0 \leq j \leq k.$$

Obviously,  $m_j(C)$  is nondecreasing with  $j$ , with  $m_0(C) = 0$  and  $m_k(C) = m(C) \leq n(C)$ .

The dimension/length profile of  $C$  determines the length/dimension profile, and *vice versa*:  $m_j(C)$  is the least  $i$  such that  $k_i(C) \geq j$ , and  $k_i(C)$  is the greatest  $j$  such that  $m_j(C) \leq i$ . The  $k$  distinct unit steps in the DLP correspond to  $k$  distinct values of  $m_j(C)$  in the LDP; if  $k_{i+1}(C) - k_i(C) = 1$ , then  $m_j(C) = i + 1$  for  $j = k_{i+1}(C)$ .

*Example 1:* The dimension/length profile of the  $(8, 4)$  binary extended Hamming code is  $\{0, 0, 0, 0, 1, 1, 2, 3, 4\}$ . From this, it follows that its length/dimension profile is  $\{0, 4, 6, 7, 8\}$ , as shown in Fig. 1. Equally, the latter determines the former.

Wei [1] defines  $m_j(C)$  as the  $j$ th *generalized Hamming weight* (GHW) of  $C$ . The terminology arises from the observation that if the minimum Hamming weight of a nonzero codeword of  $C$  is  $d$ , then  $m_1(C) = d$  (since  $k_{d-1}(C) = 0$  and  $k_d(C) = 1$ ). Wei refers to the LDP as the *GHW hierarchy* of  $C$ .

#### A. DLP Duality

We define the *inverse DLP* of  $C$  as the sequence  $\tilde{\mathbf{k}}(C)$  with components

$$\tilde{k}_i(C) \triangleq \min_J \{k[P_J(C)]: |J| = i\}, \quad 0 \leq i \leq n.$$

*Theorem 2:* The inverse DLP and DLP of an  $(n, k)$  linear code  $C$  are related by

$$k_i(C) + \tilde{k}_{n-i}(C) = k, \quad 0 \leq i \leq n.$$

*Proof:* From the first duality lemma,  $k(C_J) + k[P_{I-J}(C)] = k$ .  $\square$

It follows that the inverse DLP  $\tilde{\mathbf{k}}(C)$  can be obtained from the DLP  $\mathbf{k}(C)$  by a horizontal reflection about  $i = n/2$  and a vertical reflection about  $j = k/2$ , or equivalently, by a  $180^\circ$  rotation about  $(n/2, k/2)$ , as illustrated in Fig. 2 for the example  $(8, 4)$  code. The inverse DLP  $\tilde{\mathbf{k}}(C)$  thus rises from 0 to  $k$  in  $k$  distinct unit steps, which occur at the mirror images of the steps of  $\mathbf{k}(C)$ .

Moreover, from the second duality lemma, the DLP and inverse DLP of  $C^\perp$  may be determined from the DLP of  $C$  via the following dual relationships.

*Theorem 3 (dual DLP):* If  $C$  is an  $(n, k)$  linear code with dual  $C^\perp$ , then for  $0 \leq i \leq n$ ,

$$k_i(C) + \tilde{k}_i(C^\perp) = i.$$

*Proof:* By Lemma 2,  $k(C_J) + k[P_J(C^\perp)] = |J|$ .  $\square$

In other words, if  $C$  and  $C^\perp$  are dual linear codes of length  $n$ , then the inverse DLP of  $C$  and the DLP of  $C^\perp$  are related by

$$\mathbf{k}(C) + \tilde{\mathbf{k}}(C^\perp) = \mathbf{i},$$

where  $\mathbf{i} \triangleq \{i: 0 \leq i \leq n\}$ .

*Corollary:* If  $C$  is self-dual, then  $\mathbf{k}(C) + \tilde{\mathbf{k}}(C) = \mathbf{i}$ .

It follows that the DLP  $\mathbf{k}(C^\perp)$  is the difference  $\mathbf{i} - \tilde{\mathbf{k}}(C)$ , as illustrated in Fig. 3 for the example  $(8, 4)$  code.

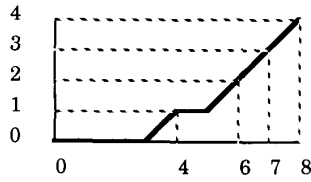


Fig. 1. Dimension/length profile  $k(C)$  of  $(8, 4)$  binary code.

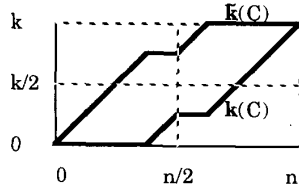


Fig. 2. The inverse DLP of  $C$  is obtained by rotating the DLP  $180^\circ$  about  $(n/2, k/2)$ .

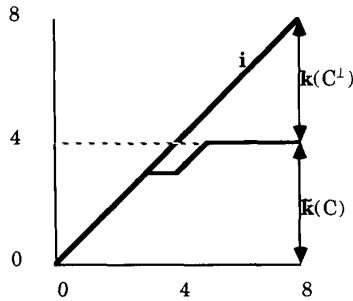


Fig. 3. The DLP of  $C^\perp$  is the difference between  $i$  and the inverse DLP of  $C$ .

Thus  $k(C^\perp)$  rises from 0 to  $r = n - k$  in  $r$  distinct unit steps, which occur whenever there is no step in  $\tilde{k}(C)$ .

*Example 2:* The DLP of the  $(16, 5)$  first-order Reed–Muller (RM) code is given by [1]

$$k(C) = \{0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 4, 5\}.$$

By Theorem 2, its inverse DLP is

$$\tilde{k}(C) = \{0, 1, 2, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5\}.$$

By Theorem 3, the DLP and inverse DLP of the dual  $(16, 11)$  second-order RM code are

$$k(C^\perp) = \{0, 0, 0, 0, 1, 1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10, 11\},$$

$$\tilde{k}(C^\perp) = \{0, 1, 2, 3, 4, 5, 6, 7, 7, 8, 9, 10, 10, 11, 11, 11, 11\}.$$

The unit steps of these profiles occur at the following places, as shown in Fig. 4:

$$k(C): \{8, 12, 14, 15, 16\},$$

$$\tilde{k}(C^\perp): \{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13\},$$

$$\tilde{k}(C): \{1, 2, 3, 5, 9\},$$

$$k(C^\perp): \{4, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16\}.$$

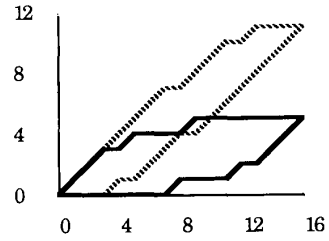


Fig. 4. DLP and inverse DLP of  $(16, 5)$  code (solid) and  $(16, 11)$  code (dashed).

Since the unit steps of  $k(C)$  and  $k(C^\perp)$  occur at the nonzero GHW's of  $C$  and  $C^\perp$ , respectively, Wei's duality result thus follows from Theorems 2 and 3.

*Theorem 4 [1]:* Given a linear code  $C$  with dual code  $C^\perp$ , for  $1 \leq i \leq n$ , either  $i$  is a GHW of  $C^\perp$  or  $n - i + 1$  is a GHW of  $C$ , but not both. In other words,

$$\{m_j(C^\perp), 1 \leq j \leq r\} = [1, n] - \{n - m_j(C) + 1, 1 \leq j \leq k\}.$$

While this is a striking duality result, it does not appear to be directly related to the MacWilliams identities, as was suggested by Wei [1]. However, Kløve [3] and Simonis [4] have been able to prove generalized MacWilliams identities for “support weight distributions,” which determine GHW hierarchies.

**B. DLP Bounds and MDS Codes**

Graphically, the “slope” of the DLP or of the inverse DLP is either  $0^\circ$  or  $45^\circ$ . Therefore, the DLP and inverse DLP are bounded within the regions illustrated in Fig. 5 for high-rate ( $r < k$ ) and low-rate ( $r \geq k$ ) codes.

From these bounds and the fact that  $m_1(C)$  is the minimum Hamming distance  $d$  of  $C$ , we obtain the Singleton bound:  $d \leq r + 1$ . Wei therefore calls the lower bound of Fig. 5 a “generalized Singleton bound” [1].

A maximum distance separable (MDS) code is an  $(n, k)$  linear code  $C$  ( $k \geq 1$ ) that meets the Singleton bound with equality. In this paper, a trivial  $(n, 0)$  code will also be defined as MDS, even though its minimum distance is conventionally defined as  $\infty$ .

The following theorem shows that the bounds of Fig. 5 are met everywhere with equality if and only if  $C$  is MDS. In turn, this result implies some of the most important properties of MDS codes.

*Theorem 5 (MDS bound):* The DLP and inverse DLP of an  $(n, k)$  linear code  $C$  are bounded by

$$k(C) \geq \{0, \dots, 0, 1, 2, \dots, k\},$$

$$\tilde{k}(C) \leq \{0, 1, 2, \dots, k, \dots, k\}.$$

These bounds are met with equality everywhere if and only if  $C$  is MDS.

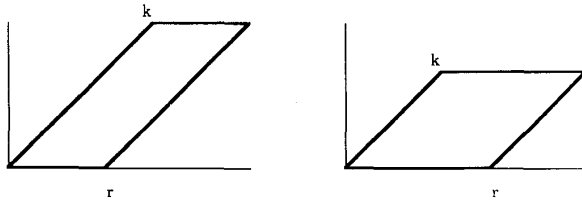


Fig. 5. Bounds on DLP and inverse DLP of high-rate and low-rate codes.

*Proof:* The bounds follow from the elementary bounds  $\dim P_J(C) \leq |J|$ ,  $\dim P_J(C) \leq \dim C = k$ ,  $\dim C_J \geq 0$ , and  $\dim C_J \geq \dim C - |I - J| = |J| - r$ , which are connected by the first duality lemma, and which hold for every subset  $J \subseteq I$ .

If the bounds are met with equality, then  $d(C) = m_1(C) = n - k + 1$ , so  $C$  is MDS. Conversely, if  $C$  is MDS, then  $m_1(C) = n - k + 1$ , which implies  $k_{n-k}(C) = 0$ , so in view of the unit-step constraint, the only possible DLP is the lower bound to  $k(C)$ .  $\square$

*Remark:* In the case of equality, the LDP of  $C$  is  $\{0, n - k + 1, n - k + 2, \dots, n\}$ .

Thus MDS codes are those codes whose DLP's meet the outer bounds of Fig. 5. Their most important characteristics follow directly from this extremal property.

*Corollary:* If  $C$  is an  $(n, k)$  MDS code, then

- (a) its dual code  $C^\perp$  is an  $(n, r)$  MDS code;
- (b) for every subset  $J \subseteq I$ , the punctured code  $P_J(C)$  is MDS;
- (c) for every subset  $J \subseteq I$ , the shortened code  $C_J$  is MDS;
- (d) in particular, if  $|J| = k$ , then  $\dim P_J(C) = k$ —i.e., every subset  $J$  of size  $k$  is an information set for  $C$ ;
- (e) in particular, if  $|J| = r + 1 = d$ , then  $\dim C_J = 1$ —i.e., for every subset  $J$  of size  $d$ , there exists a codeword of weight  $d$  with support  $J$ .

*Remark:* From (a) and (e), if  $|J| = k + 1$ , then there exists a codeword of  $C^\perp$  of weight  $k + 1$  with support  $J$ —i.e., every subset  $J$  of size  $k + 1$  is a check set for  $C$ .

*Proof:* (a) If the inverse DLP of  $C$  is  $k(C) = \{0, 1, 2, \dots, k, \dots, k\}$ , then from Theorem 3, the DLP of  $C^\perp$  is  $k(C^\perp) = \{0, \dots, 0, 1, 2, \dots, r\}$ , so by Theorem 5,  $C^\perp$  is MDS.

(b) and (c): If the bounds of Theorem 4 are met with equality, then  $\dim P_J(C) = k_{|J|}(C)$  and  $\dim C_J = k_{|J|}(C)$  for every  $J \subseteq I$ , as noted in the proof of Theorem 5. The inverse DLP of  $P_J(C)$  and the DLP of  $C_J$  are thus equal to those of  $C$  in the range  $0 \leq i \leq |J|$ , so by Theorem 5, both are MDS (when regarded as codes of length  $|J|$ ).

(d) and (e): Special cases of (b) and (c).  $\square$

In fact, if  $C$  is an  $(n, k)$  MDS code over a finite field  $GF(q)$ , then its entire weight distribution is determined as a function of  $n, k$ , and  $q$  by these properties.

### C. Distance Bounds on Profiles

If  $C$  is an  $(n, k)$  linear code and the minimum Hamming distance between codewords is  $d$ , then  $C$  is called an  $(n, k, d)$  code.

If  $C$  is an  $(n, k, d)$  code, then its minimum distance  $d$  imposes bounds on its DLP, or equivalently, on its GHW hierarchy. For any subcode  $C_J$  of  $C$  must have minimum distance  $d(C_J) \geq d(C) = d$ . Let  $k_{\max}(n, d)$  denote the maximum dimension of a linear code with length  $n$  and minimum Hamming distance  $d$ . Then,

$$k_i(C) \leq k_{\max}(i, d).$$

The Singleton bound is the general bound  $k_{\max}(n, d) = n - d + 1$ , which applies over any field  $F$ . Application of this bound results in an alternative derivation of the MDS bound of Theorem 5.

When  $F$  is a finite field, sharper bounds are often known. For binary codes, Brouwer and Verhoeff [10] tabulate results obtained by generations of coding researchers. For example, the bounds for  $d = 4$  and  $d = 8$  are as follows.

*Example 3:* The greatest possible DLP for a binary linear  $(n, k, 4)$  code is

$$\{0, 0, 0, 0, 1, 1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10, 11, 11, 12, \dots\}.$$

Any  $(2^m, 2^m - m - 1, 4)$  extended Hamming code  $C$ ,  $m \geq 2$ , has this profile—i.e., the  $(4, 1, 4)$  code, the  $(8, 4, 4)$  code, the  $(16, 11, 4)$  code, and so forth [1]. Shortened codes of these codes also achieve this profile. The corresponding LDP (GHW hierarchy) is

$$\{0, 4, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 18, \dots\}.$$

*Example 4:* The greatest possible DLP for a binary linear  $(n, k, 8)$  code is

$$\{0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 4, 5, 5, 6, 7, 8, 9, 10, 11, 12, 12, \dots\}.$$

This profile is attained by the  $(8, 1, 8)$  repetition code, the  $(16, 5, 8)$  first-order Reed–Muller code, and the  $(24, 12, 8)$  Golay code [1], but not by any  $(32, 16, 8)$  code. The corresponding LDP is  $\{0, 8, 12, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, \dots\}$ .

### IV. TRELLIS COMPLEXITY AND DLP'S

A linear block code  $C$  may be regarded as (the set of output sequences of) a time-varying linear dynamical system, provided that its index set  $I$  has a definite order. Without loss of generality, we may take  $I$  as the set

$$I = \{0, 1, \dots, n - 1\},$$

which has an implicit natural order. We may then think of  $I$  as the *time axis* of  $C$ , and use temporal language such as “before,” “after,” and so forth.

In particular, the following special notation will be used for the *past*  $i^-$  and the *future*  $i^+$  with respect to time  $i$ :

$$i^- \triangleq \{i' \in I: i' < i\} = \{0, 1, \dots, i-1\};$$

$$i^+ \triangleq \{i' \in I: i' \geq i\} = \{i, i+1, \dots, n-1\}.$$

It is shown in [5], [6] that, given an ordered time axis  $I$ , there exists a well-defined minimal realization of a linear code  $C$ , and a corresponding well-defined minimal trellis diagram for  $C$  based on well-defined state spaces  $\Sigma_i(C)$  of  $C$  at each time  $i$ ,  $0 \leq i \leq n$ . Parameters of minimal trellis diagrams will be given below.

#### A. Ordered Profiles

Assuming that the time axis  $I$  is  $\{0, 1, \dots, n-1\}$ , we define the *ordered dimension/length profile* of  $C$  by

$$\vec{k}_i(C) \triangleq k(C_{i^-}), \quad 0 \leq i \leq n.$$

Clearly,  $\vec{k}(C) \leq k(C)$ , since  $|i^-| = i$ . Like  $k(C)$ ,  $\vec{k}(C)$  rises from  $\vec{k}_0(C) = 0$  to  $\vec{k}_n(C) = k$  in  $k$  unit steps.

The *inverse ordered DLP* of  $C$  is defined by

$$\tilde{\vec{k}}_i(C) \triangleq k[P_i(C)], \quad 0 \leq i \leq n.$$

Clearly,  $\tilde{\vec{k}}(C) \geq \vec{k}(C)$ , and  $\tilde{\vec{k}}(C)$  rises from 0 to  $k$  in  $k$  unit steps.

By Lemma 1,  $\tilde{\vec{k}}_i(C) = k - k(C_{i^+})$ , which is not necessarily equal to  $k - \vec{k}_{n-i}(C)$ , since  $k(C_{n-i^-})$  may not be equal to  $k(C_{i^+})$ . However, by Lemma 2,  $\tilde{\vec{k}}_i(C) + \vec{k}_i(C^\perp) = i$ .

We say that a coordinate ordering is an *efficient ordering* for  $C$  if  $\vec{k}(C) = k(C)$  and  $\tilde{\vec{k}}(C) = \vec{k}(C)$ . We then have the result of Kasami *et al.* [7].

**Lemma 3:** A coordinate ordering is efficient for a linear code  $C$  if and only if it is efficient for its dual code  $C^\perp$ .

*Proof:* By DLP and ordered DLP duality,  $\tilde{\vec{k}}(C^\perp) = i - \vec{k}(C) = i - \tilde{\vec{k}}(C) = k(C^\perp)$ .  $\square$

For an MDS code, the ordered profiles are invariant under coordinate permutations, by the corollary to Theorem 5, and hence all orderings are efficient. In fact, this property, which has nothing explicitly to do with distance, suffices to characterize MDS codes.

**Theorem 6:** The ordered DLP of an  $(n, k)$  linear code  $C$  is invariant under all coordinate permutations if and only if  $C$  is MDS.

*Proof:* By the corollary to Theorem 5, part (c), if  $C$  is MDS, then every ordered DLP of  $C$  is equal to  $k(C) = \{0, \dots, 0, 1, \dots, k\}$ . Conversely, if every ordered DLP is equal, say, to  $m(C) = \{0, d, \dots\}$  (assuming  $k \geq 1$ ; if  $k = 0$ , then  $C$  is trivially MDS), then the minimum distance of  $C$  is  $d$ ; moreover, for every subset  $J$  of size  $d$ , there exists a codeword of weight  $d$  with support  $J$ . Taking  $J = \{(0, d), [1, d+1] \dots [n-d, n]\}$ , we obtain  $n-d+1$  linearly independent codewords, so  $k \geq n-d+1$ . By the Singleton bound,  $k = n-d+1$ , so  $C$  is MDS.  $\square$

We shall see shortly that Theorem 6 implies this corollary:  $C$  is MDS if and only if its state complexity profile is invariant under all coordinate permutations.

#### B. State Spaces

The *state space* of  $C$  at time  $i$  is induced by the partition of  $I$  into the past and future with respect to time  $i$ :  $I = \{i^-, i^+\}$ . It may be defined as a quotient space in several equivalent ways [5], [6]:

$$\Sigma_i(C) \triangleq C/(C_{i^-} \oplus C_{i^+}) \simeq P_{i^-}(C)/C_{i^-} \simeq P_{i^+}(C)/C_{i^+}.$$

From any of its definitions, the dimension of  $\Sigma_i(C)$  is

$$\begin{aligned} s_i(C) &\triangleq \dim \Sigma_i(C) = k - k(C_{i^-}) - k(C_{i^+}) \\ &= \tilde{\vec{k}}_i(C) - \vec{k}_i(C). \end{aligned}$$

We call  $s(C)$  the *state complexity profile* of  $C$ . Its maximum component will be denoted by  $s_{\max}(C) \triangleq \max_i \{s_i(C)\}$ . The minimum  $s_{\max}(C)$  over all coordinate orderings is called the *state complexity*  $s(C)$  of  $C$ . Muder [11] has argued that the parameter  $s(C)$  is "a fundamental descriptive characteristic [of  $C$ ], comparable to the quantities  $n$  (length),  $k$  (size), and  $d$  (minimum distance)."

From the first expression above, the state complexity profile of a linear code  $C$  is a simple function of its ordered dimension/length profile.

**Lemma 4:** For a linear code  $C$ , the state complexity profile of  $C$  is the difference between the ordered inverse DLP of  $C$  and the ordered DLP of  $C$ :

$$s(C) = \tilde{\vec{k}}(C) - \vec{k}(C).$$

It is notable that Lemma 4 yields an easy proof of an important duality result.

**Theorem 7 ([5]):** The state complexity profile of the dual  $C^\perp$  to a linear code  $C$  is the same as the state complexity profile of  $C$ .

*Proof:* DLP duality (Theorem 3) applies equally to ordered DLP's. Therefore,

$$\begin{aligned} s(C^\perp) &= \tilde{\vec{k}}(C^\perp) - \vec{k}(C^\perp) = [i - \vec{k}(C)] - [i - \tilde{\vec{k}}(C)] \\ &= s(C). \end{aligned} \quad \square$$

Now, since ordered profiles are bounded by unordered profiles, we have the following.

**Theorem 8 (DLP bound on state complexity):** If  $C$  is a linear code with DLP  $k(C)$  and inverse DLP  $\vec{k}(C)$ , then the state complexity profile  $s(C)$  of  $C$  is bounded by

$$s(C) \geq \vec{k}(C) - k(C),$$

with equality if and only if the coordinate ordering of  $C$  is efficient.

*Proof:*  $\vec{k}(C) \leq k(C)$  and  $\tilde{\vec{k}}(C) \geq \vec{k}(C)$ .  $\square$

In general, to minimize state complexity, the ordered DLP of  $C$  should rise as rapidly as possible. In this sense, an ordered DLP that meets the bound of Fig. 5 is the worst possible profile for an  $(n, k)$  code. Theorems 5 and 6 show that, regardless of its coordinate ordering, an MDS code has this worst possible profile.

Explicitly, in the high-rate case, an MDS code has the state complexity profile

$$s_i(C) = \begin{cases} i, & 0 \leq i \leq r, \\ r, & r \leq i \leq k, \\ n - i, & k \leq i \leq n. \end{cases}$$

The low-rate case is the dual of the high-rate case:

$$s_i(C) = \begin{cases} i, & 0 \leq i \leq k, \\ k, & k \leq i \leq r, \\ n - i, & r \leq i \leq n. \end{cases}$$

Thus we obtain the *Wolf bound* [12]:  $s_i(C) \leq \min\{k, r\}$ . This implies  $s_{\max}(C) \leq \min\{k, r\}$ ; equality holds for MDS codes with any coordinate ordering.

The high-rate and low-rate cases coincide when  $k = r = n/2$ . For any  $(n, k)$  code, we have  $s_i(C) \leq n/2$ , with equality if and only if  $C$  is an  $(n, n/2)$  MDS code and  $i = n/2$ .

*C. Distance / DLP Bounds on State Complexity Profiles*

The distance bounds of Section III-C and the DLP bound of Theorem 8 yield bounds on the entire state complexity profile. (This extends Muder's method of bounding  $s(C)$  [11].) For a number of good codes, these bounds can be achieved.

*Example 1 (cont.):* The  $(8, 4, 4)$  binary code  $C$  has DLP  $k(C) = \{0, 0, 0, 0, 1, 1, 2, 3, 4\}$ , so the DLP bound on its state complexity profile is  $s(C) \geq \{0, 1, 2, 3, 2, 3, 2, 1, 0\}$ , as shown in Fig. 6. Since, by the  $d = 4$  bounds,  $k(C)$  is the best possible DLP for an  $(8, 4, 4)$  code, this is the best possible state complexity profile (componentwise) for any  $(8, 4, 4)$  code. In fact, there exists an efficient coordinate ordering that attains this profile [5].

*Example 4 (cont.):* The  $(24, 12, 8)$  binary Golay code  $C$  has DLP

$$k(C) = \{0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 3, 4, 5, 5, 6, 7, 8, 9, 10, 11, 12\},$$

so the DLP bound on its state complexity profile is

$$s(C) \geq \{0, 1, 2, 3, 4, 5, 6, 7, 6, 7, 8, 9, 8, 7, 6, 7, 6, 5, 4, 3, 2, 1, 0\},$$

as shown in Fig. 7. Since  $k(C)$  is the best possible DLP for a  $(24, 12, 8)$  binary code, this  $s(C)$  is the best possible state complexity profile for a  $(24, 12, 8)$  binary code. This not only confirms Muder's result [11] that the state complexity  $s(C)$  of the Golay code is 9; it also shows that this state complexity profile is optimum *componentwise*. In fact, there exists an efficient coordinate ordering that attains this profile [5].

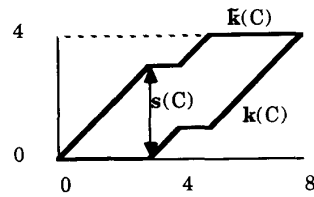


Fig. 6. DLP bound on state complexity profile of  $(8, 4, 4)$  binary code.

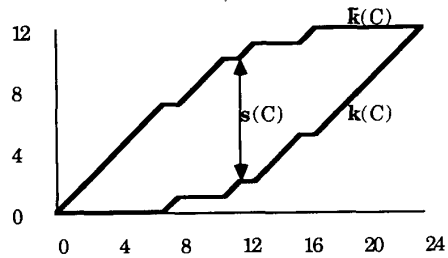


Fig. 7. DLP bound on state complexity profile of  $(24, 12, 8)$  binary Golay code.

*Example 3 (cont.):* For a code that does not meet the distance bound on its state complexity profile, consider a  $(7, 3, 4)$  linear binary code  $C$ . It is known that such a code must have one word of weight 0 and seven words of weight 4. The distance bounds on its DLP, inverse DLP, and state complexity profile are

$$\begin{aligned} k(C) &\leq \{0, 0, 0, 0, 1, 1, 2, 3\} \\ \tilde{k}(C) &\geq \{0, 1, 2, 2, 3, 3, 3\} \\ s(C) &\geq \{0, 1, 2, 2, 2, 2, 1, 0\}. \end{aligned}$$

But an efficient coordinate ordering is impossible, since that would imply that both  $(1111000)$  and  $(0001111)$  are codewords, which would imply that the weight-6 word  $(1110111)$  is a codeword. The best possible ordered profiles for a  $(7, 3, 4)$  code are thus

$$\begin{aligned} \bar{k}(C) &= \{0, 0, 0, 0, 1, 1, 2, 3\}, \\ \tilde{\tilde{k}}(C) &= \{0, 1, 2, 3, 3, 3, 3\}, \\ s(C) &= \{0, 1, 2, 3, 2, 2, 1, 0\}; \end{aligned}$$

the state complexity  $s(C)$  is thus 3, not 2. (These results can also be derived from the more general results of [13] and [14], applied either to  $C$  or to the dual  $(7, 4, 3)$  code.)

*Example 5:* From known results on binary linear  $(n, k, 6)$  codes [10], the distance bound on the DLP and state complexity profile of a  $(16, 7, 6)$  binary linear code are

$$\begin{aligned} k(C) &\leq \{0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 3, 4, 4, 5, 6, 7\}, \\ s(C) &\geq \{0, 1, 2, 3, 3, 4, 4, 4, 5, 4, 4, 4, 3, 3, 2, 1, 0\}, \end{aligned}$$

so  $s(C) \geq 5$ . It can be shown that lexicographic code construction of Conway and Sloane [15] (which necessarily generates codes satisfying the "chain condition" [16], [17])

produces a (16, 7, 6) "lexicode" with a coordinate ordering that meets the DLP bound and achieves  $s_{\max}(C) = 5$ , although the inverse DLP and state complexity profile do not achieve their bounds everywhere:

$$\vec{k}(C) = \{0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 3, 4, 4, 5, 6, 7\},$$

$$\tilde{k}(C) = \{0, 1, 2, 3, 4, 4, 5, 6, 6, 7, 7, 7, 7, 7, 7, 7\},$$

$$s(C) = \{0, 1, 2, 3, 4, 4, 4, 5, 5, 5, 5, 4, 3, 3, 2, 1, 0\}.$$

By contrast, Ytrehus [9] has shown that  $s(C) = 6$  for the (16, 7, 6) BCH code (after Vardy and Be'ery [8] failed to find a coordinate ordering with  $s_{\max}(C) < 6$ ). It is also interesting to note that Hellesteth and Kumar [18] have shown that the (15, 6, 6) Kasami code achieves the same DLP as the (15, 6, 6) "lexicode."

*Example 6:* The following example indicates how these ideas can be applied to more general codes. The Nordstrom-Robinson (NR) code is a nonlinear binary "(16, 8, 4)" code. It has been shown [19], [20] that the NR code is the binary image of a linear self-dual (8, 4) code  $C$  over the ring  $\mathbf{Z}_4$  of integers modulo 4 (the "octacode"), and therefore, as a group code [6], its trellis diagram with any coordinate ordering is well defined. From the Griesmer bound applied to the binary image of  $C$ , its "dimension"  $k_4(C) \triangleq \max_J \{\log_4 |C_J| : |J| = 4\}$  is bounded by  $k_4(C) \leq 0.5$ , which implies that  $s_4(C) \geq 3$ . In fact, a known coordinate ordering [20] achieves

$$\vec{k}(C) = \{0, 0, 0, 0, 0.5, 1, 2, 3, 4\},$$

$$\tilde{k}(C) = \{0, 1, 2, 3, 3.5, 4, 4, 4, 4\},$$

$$s(C) = \{0, 1, 2, 3, 3.0, 3, 2, 1, 0\},$$

where  $s_i(C) \triangleq \log_4 \Sigma_i(C)$ . The same 64-state profile  $s(C)$  is achieved by a "twisted squaring construction" for the NR code [5].

Kasami *et al.* [7], using the GHW hierarchies for RM codes of [1], have shown that the standard RM coordinate ordering [5] is efficient and yields the best possible state complexity profile for all RM codes. The simple bound on the DLP of a  $d = 4$  binary code given in Example 3, along with DLP duality, proves this result for extended Hamming codes, first-order RM codes, and their shortened codes.

Vardy and Be'ery [8], using the GHW idea and simple low-rate Griesmer bounds, have shown that if  $d \geq 2(n + 2)/5$ , then  $s_{\max}(C) \geq k - 1$ , while if  $d \geq (n + 2)/3$ , then  $s_{\max}(C) \geq k - 2$ ; thus if  $d/n$  is large, then the Wolf bound cannot be improved very much. (The codes of Examples 5 and 6 meet these lower bounds.) They have also used DLP duality to compute lower bounds on  $s_{\max}(C)$  for high-rate duals of low-rate codes, and have used known distance bounds to compute lower bounds on  $s_{\max}(C)$  for mid-rate codes, which are generally well below the Wolf bound. They give coordinate orderings for BCH codes that approach these bounds, but in general do not equal them. Ytrehus [9] has sharpened some of these bounds.

## V. BRANCH COMPLEXITY AND DLP'S

For Viterbi decoding, the total number of trellis branches per unit time is usually regarded as a more accurate measure of decoding complexity than the size of the state space. Thus the branch complexity profile may be of more practical importance than the state complexity profile.

We show in this section that the branch complexity profile is easily computed from the ordered DLP, like the state complexity profile. Furthermore, we show that the maximum branch complexity (unlike the maximum state complexity) cannot be reduced by constructing a trellis with sections comprising more than one time unit (sectionalization). We therefore propose that branch complexity should be regarded as more significant than state complexity.

Finally, we briefly discuss an alternative definition of branch complexity, applicable when a set of parallel transitions may be regarded as a single branch.

### A. Branch Spaces

In general [6], a *trellis branch* between states at time  $i$  and time  $j \geq i$  is a triple

$$t_{i,j}(c) \triangleq (\sigma_i(c), P_{[i,j]}(c), \sigma_j(c)), \quad c \in C,$$

where  $\sigma_i(c) \in \Sigma_i(C)$  and  $\sigma_j(c) \in \Sigma_j(C)$  are the states through which the code sequence  $c$  passes at times  $i$  and  $j$ , respectively, and  $P_{[i,j]}(c)$  is the part of  $c$  that occurs during the interval  $[i, j]$ . The set of all such branches is the *branch space* (or *trellis section*)

$$T_{i,j}(C) \triangleq \{t_{i,j}(c) : c \in C\}.$$

The branch space  $T_{i,j}(C)$  is a linear vector space over  $\mathcal{F}$ . We define its dimension as the *branch complexity* between times  $i$  and  $j$ :

$$b_{i,j}(C) \triangleq \dim T_{i,j}(C).$$

It can be shown [6] that the branch space is isomorphic to the quotient space

$$T_{i,j}(C) \simeq C / (C_{i-} \oplus C_{j+}).$$

This shows that the following lemma holds.

*Lemma 5 (branch complexity):* The branch complexity  $b_{i,j}(C)$  of an  $(n, k)$  linear code  $C$  is equal to

$$b_{i,j}(C) = k - k(C_{i-}) - k(C_{j+}) = \vec{k}_j(C) - \vec{k}_i(C).$$

Note that if  $j = i$ , then  $T_{i,i}(C) = \Sigma_i(C)$ ; therefore  $b_{i,i}(C)$  is equal to the state complexity  $s_i(C)$ . For larger intervals  $[i, j]$ ,  $b_{i,j}(C)$  cannot decrease.

*Lemma 6 (monotonicity of branch complexity):* If  $C$  is linear and  $[i, j] \subseteq [i', j']$ , then  $b_{i',j'}(C) \geq b_{i,j}(C)$ , with equality if and only if  $k(C_{i'-}) = k(C_{i-})$  and  $k(C_{j'+}) = k(C_{j+})$ . In particular, if  $j \geq i$ , then  $b_{i,j}(C) \geq s_i(C)$ .

*Proof:* If  $i' \leq i$  and  $j \leq j'$ , then  $b_{i',j'}(C) = k - k(C_{i'-}) - k(C_{j'+}) \geq k - k(C_{i-}) - k(C_{j+}) = b_{i,j}(C)$ , since  $k(C_{i'-}) \leq k(C_{i-})$  and  $k(C_{j'+}) \leq k(C_{j+})$ .  $\square$



For the remainder of this subsection, we consider branch spaces with  $j = i + 1$ . (In the next subsection, we shall consider longer intervals  $[i, j]$ .) For brevity, we write  $b_i(C)$  instead of  $b_{i,i+1}(C)$ ,  $0 \leq i \leq n - 1$ . The parameter  $b_{\max}(C) = \max_i \{b_i(C)\}$  will be called the *branch complexity* of  $C$ .

For  $|j - i| = 1$ , we have the following tight bounds.

**Theorem 9:** If  $C$  is linear, then for any given coordinate ordering, the branch complexity is bounded by

$$s_i(C) \leq b_i(C) \leq s_i(C) + 1,$$

$$s_{i+1}(C) \leq b_i(C) \leq s_{i+1}(C) + 1.$$

*Proof:* The lower bounds follow from Lemma 5. The upper bounds follow from the fact that the increment  $k(C_{i+1}) - k(C_i)$  cannot exceed 1.  $\square$

This result demonstrates a close relationship between state and branch complexity when the trellis sections have length  $|j - i| = 1$ .

Now we develop a DLP bound on branch complexity analogous to Theorem 8. From Lemma 5,

$$b_i(C) = \tilde{k}_{i+1}(C) - \tilde{k}_i(C).$$

In other words, the branch complexity is the difference between the ordered LDP of  $C$ , shifted by one position, and the ordered DLP of  $C$ . Define the *shift operator*  $\sigma$  by

$$\sigma[\tilde{k}_i(C)] = \tilde{k}_{i+1}(C), \quad 0 \leq i \leq n - 1;$$

then

$$b(C) = \sigma[\tilde{k}(C)] - \tilde{k}(C).$$

**Theorem 10 (DLP bound on branch complexity):** If  $C$  is a linear code with DLP  $\tilde{k}(C)$  and inverse DLP  $\tilde{k}(C)$ , then

$$b(C) \geq \sigma[\tilde{k}(C)] - \tilde{k}(C),$$

with equality if and only if the coordinate ordering of  $C$  is efficient.

*Proof:*  $\tilde{k}(C) \leq k(C)$  and  $\sigma[\tilde{k}(C)] \geq \sigma[k(C)]$ .  $\square$

**Example 1 (cont.):** The DLP bound on the branch complexity profile of the (8, 4, 4) binary code is

$$\{1, 2, 3, 3, 3, 3, 2, 1\},$$

as shown in Fig. 8. In this case,  $b_{\max}(C) = s_{\max}(C) = 3$ . Since this is the best possible DLP for an (8, 4, 4) binary code, this is the best possible branch complexity profile for any (8, 4, 4) code. The efficient coordinate ordering of [5] in fact attains this profile.

**Example 4 (cont.):** The DLP bound on the branch complexity profile of the (24, 12, 8) Golay code is

$$\{0, 1, 2, 3, 4, 5, 6, 7, 7, 7, 8, 9, 9, 9, 9, 8, 7, 7, 7, 6, 5, 4, 3, 2, 1, 0\}$$

as shown in Fig. 9. In this case,  $b_{\max}(C) = s_{\max}(C) = 9$ . Since this is the best possible DLP for a (24, 12, 8) binary code, this is the best possible branch complexity profile

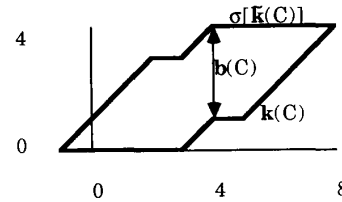


Fig. 8. DLP bound on branch complexity profile of (8, 4, 4) binary code.

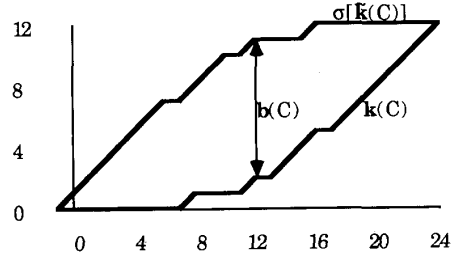


Fig. 9. DLP bound on branch complexity profile of (24, 12, 8) binary Golay code.

for a (24, 12, 8) code. The efficient coordinate ordering of [5] in fact attains this profile.

The results of [1] and [7] show that for all Reed–Muller codes, the standard ordering of [5] is efficient and yields the best possible branch complexity profile.

Again, the worst case occurs for MDS codes in any coordinate ordering. Then, in the high-rate case, we have

$$b(C) = \{1, 2, \dots, r + 1, r + 1, \dots, r + 1, r, \dots, 1\}.$$

In the low-rate case (now not the dual of the high-rate case), we have

$$b(C) = \{1, 2, \dots, k, k, \dots, k, k - 1, \dots, 1\}.$$

The Wolf-type bound for branch complexity is thus  $b_i(C) \leq \min\{k, r + 1\}$ . The two cases now coincide when  $k = (n + 1)/2$ ,  $r = (n - 1)/2$ . This implies that, for any  $(n, k)$  code, we have  $b_i(C) \leq (n + 1)/2$ , with equality if and only if  $C$  is an  $(n, (n + 1)/2)$  MDS code and  $i = (n - 1)/2$ .

### B. Sectionalization

The apparent maximum state complexity can be reduced by *sectionalization*; that is, by partitioning the time axis  $I$  into trellis sections  $T_{i,j}$  of lengths  $|j - i|$  possibly greater than 1. For the most part, we will let all section lengths be equal.

**Example 1 (cont.):** For the (8, 4, 4) binary code, if the time axis is divided into four sections of length 2, then the state complexity profile at the section boundaries (the values of  $s_i(C)$  for  $i$  even) is  $\{0, 2, 2, 2, 0\}$ , so the apparent state space size is only 4 states, rather than 8 [5].

**Example 4 (cont.):** For the (24, 12, 8) Golay code, if the time axis is divided into three sections of length 8, then the state complexity profile at the section boundaries (the values of  $s_i(C)$  for  $i \equiv 0 \pmod{8}$ ) is  $\{0, 6, 6, 0\}$ , so the

apparent state space size is only 64 states, rather than 512 [5].

Practically, sectionalizations that reduce the apparent state complexity in this way often lead to efficient trellis-based decoding algorithms (see, e.g., [5] and later papers). Theoretically, however, this apparent simplification is somewhat dubious. For instance, if the entire time axis  $I$  is collapsed into one section, then the apparent state complexity is reduced to zero [11].

By contrast, the following holds.

**Theorem 11:** The maximum branch complexity of a linear code  $C$  cannot be reduced by sectionalization.

*Proof:* Let  $b_{i,i+1}(C)$  attain its maximum  $b_{\max}(C)$  at  $i^*$ . Then, in any partition of  $I$ , there must be some subinterval  $[i, j)$  that includes  $\{i^*\} = [i^*, i^* + 1)$ . But then, by Lemma 6,  $b_{i,j}(C) \geq b_{i^*,i^*+1}(C) = b_{\max}(C)$ .  $\square$

This theoretical result, in combination with the practical importance of branch complexity, suggests that branch complexity ought to be regarded as more significant than state complexity.

An *efficient sectionalization* will be defined as one in which the maximum branch complexity is equal to  $s_{\max}(C)$ , the maximum state complexity of  $C$ . In all examples so far that admit an efficient coordinate ordering, except for high-rate MDS codes, we have seen that  $b_{\max}(C) = s_{\max}(C)$ ; in other words, the exhaustive partition of  $I$  is an efficient sectionalization.

Our previous results can be extended easily to sectionalized trellises.

**Theorem 12 (DLP bound on section complexity):** If  $C$  is a linear code with DLP  $k(C)$  and inverse DLP  $\bar{k}(C)$ , then for any interval  $[i, j)$ , the section complexity is bounded by

$$b_{i,j}(C) \geq \bar{k}_j(C) - k_i(C) = \sigma^{|j-i|}[\bar{k}_i(C)] - k_i(C),$$

with equality if the coordinate ordering of  $C$  is efficient.

*Proof:* Follows from  $b_{i,j}(C) = \bar{k}_j(C) - \bar{k}_i(C) \geq \bar{k}_j(C) - k_i(C)$ , and the definition of  $\sigma$ .  $\square$

In other words, if all section lengths  $|j - i|$  are equal, then the section complexity profile can be evaluated graphically by moving the inverse DLP to the left by  $|j - i|$  units, and measuring section complexity at the multiples of  $|j - i|$ .

**Example 1 (cont.):** For the (8, 4, 4) code, with four sections of length 2, the DLP bound on the section complexity profile is {2, 3, 3, 2}, as shown in Fig. 10(a). Since  $b_{\max}(C) = s_{\max}(C) = 3$ , this sectionalization is efficient. Indeed, even the partition into two equal sections of length 4 is efficient, since the section complexity profile is then {3, 3}, as shown in Fig. 10(b).

**Example 4 (cont.):** For the (24, 12, 8) code, with six sections of length 4, the DLP bound on the section complexity profile is {4, 7, 9, 9, 7, 4}, as shown in Fig. 11(a). Since  $b_{\max}(C) = s_{\max}(C) = 9$ , this sectionalization is efficient. For this code, however, a partition into three equal sections of length 8 is an inefficient sectionalization, since the section complexity profile is then {7, 10, 7}, as shown in Fig. 11(b).

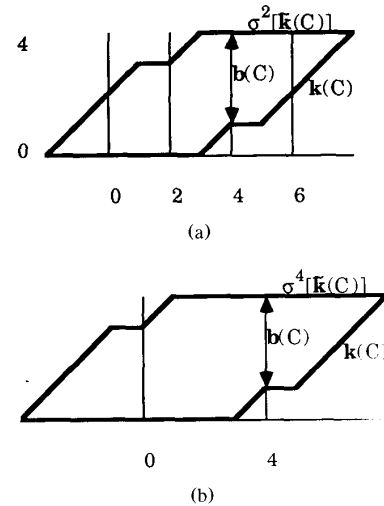


Fig. 10. DLP bound on section complexity profile of (8, 4, 4) binary code: (a) four sections; (b) two sections.

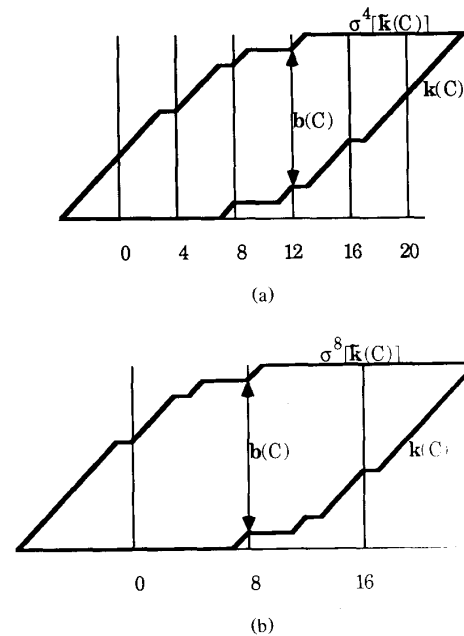


Fig. 11. DLP bound on section complexity profile of (24, 12, 8) Golay code: (a) six sections; (b) three sections.

For low-rate MDS codes ( $k/n \leq \frac{1}{2}$ ), since  $s_{\max}(C) = k$ , any sectionalization is efficient, even the trivial sectionalization with a single section equal to  $I$ . For high-rate MDS codes ( $k/n > \frac{1}{2}$ ), on the other hand, no sectionalization is efficient, since even with the exhaustive partition,  $b_{\max}(C) = r + 1 > s_{\max}(C) = r$ .

In general, by Lemma 6, efficient sectionalization is possible only if, at each  $i$  for which  $s_i(C)$  attains its

maximum, the ordered DLP of  $C$  is horizontal to the left, and/or the inverse ordered DLP is horizontal to the right. Then the longest efficient sections extend from the least  $i'$  such that  $\vec{k}_{i'}(C) = \vec{k}_i(C)$  to the greatest  $i''$  such that  $\vec{k}_{i''}(C) = \vec{k}_i(C)$ . Efficient sectionalizations are thus more likely with low-rate codes.

### C. State-Pair Spaces

If  $C_{[i,j]}$  is nontrivial, then there is a space of parallel branches of dimension  $k[C_{[i,j]}]$  connecting the zero state in  $\Sigma_i(C)$  to the zero state in  $\Sigma_j(C)$ . By linearity, there is a set of parallel transitions (a coset of  $C_{[i,j]}$  in  $P_{[i,j]}(C)$ ) for every state pair  $(\sigma_i, \sigma_j) \in \Sigma_i(C) \times \Sigma_j(C)$  that actually occurs in the trellis section  $T_{i,j}(C)$  [6].

In some applications (e.g., in lattices [21]), it may be appropriate to regard each such set of parallel transitions as a single branch. In other words, we may be concerned only with the number of state pairs in the trellis section  $T_{i,j}(C)$ .

The state-pair space  $\Sigma_{i,j}(C)$  of  $C$  may be obtained by projecting  $T_{i,j}(C)$  onto  $\Sigma_i(C) \times \Sigma_j(C)$ , and is isomorphic to [6]

$$\Sigma_{i,j}(C) \cong C / (C_{i^-} \oplus C_{[i,j]} \oplus C_{j^+}).$$

Its dimension, called the state-pair complexity, is thus equal to

$$\begin{aligned} a_{i,j}(C) &\triangleq \dim \Sigma_{i,j}(C) \\ &= k - k(C_{i^-}) - k[C_{[i,j]}] - k(C_{j^+}) \\ &= b_{i,j}(C) - k[C_{[i,j]}]. \end{aligned}$$

The state-pair complexity  $a_{i,j}(C)$  is obviously not less than either  $s_i(C)$  or  $s_j(C)$ , and not more than the branch complexity  $b_{i,j}(C)$ . The improvement over  $b_{i,j}(C)$  is limited by  $k_{|j-i|}(C)$ , which is the maximum possible value of  $k[C_{[i,j]}]$ . Thus, the following holds.

**Theorem 13 (DLP bound on state-pair complexity):** If  $C$  is a linear code with DLP  $k(C)$  and inverse DLP  $\vec{k}(C)$ , then

$$a_{i,j}(C) \geq \vec{k}_j(C) - k_j(C) - k_{|j-i|}(C). \quad \square$$

**Example 4 (cont.):** The three-section trellis for the (24, 12, 8) Golay code has a state-pair complexity profile of {6, 9, 6}, which meets the DLP bound. Since  $a_{\max}(C) = s_{\max}(C) = 9$ , this sectionalization may be regarded as at least "state-pair efficient."

## VI. CONCLUSION

It is clear that the topics of dimension/length profiles (GHW hierarchies, etc.) of linear codes and of the trellis complexity of such codes are parts of a single subject. Henceforth, any result in one area should be translated to a result in the other.

For example, much is known about optimum state complexity profiles of linear block codes. (See, e.g., Vardy and Be'ery [8] for a good recent summary.) These results can be translated into corresponding results about optimum DLP's.

Other natural topics for research include:

- development of upper bounds for DLP's,
- determination of DLP's for particular codes,
- identification of codes that have efficient coordinate orderings.

A substantial amount of work has already been done on topics (a) and (b).

One obvious extension of this work is to lattices. The theory of trellis complexity of lattices has progressed in parallel with that of linear block codes, due to their common group structure [6]. A sequel [21] develops some analogous results for lattices, which suggest that these ideas will be equally fruitful in that field.

Another extension would be to convolutional and trellis codes. Projections and subcodes of convolutional codes onto finite intervals are block codes. Thus convolutional code bounds can be derived from block code bounds.

In conclusion, a unified approach to a number of topics in coding theory is emerging, opening broad avenues for further research.

## ACKNOWLEDGMENT

This paper was inspired by a tutorial lecture by V. K. Wei [22]. The assistance of V. K. Wei and T. Kløve with the GHW literature is also gratefully acknowledged.

## REFERENCES

- V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1412–1418, Sept. 1991.
- T. Helleseth, T. Kløve, and J. Mykkelveit, "The weight distribution of irreducible cyclic codes with block lengths  $n_i((q^l - 1)/N)^n$ ," *Discr. Math.*, vol. 18, pp. 179–211, 1977.
- T. Kløve, "Support weight distribution of linear codes," *Discr. Math.*, vol. 107, pp. 311–316, 1992.
- J. Simonis, "The effective length of subcodes," preprint.
- G. D. Forney, Jr., "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, Sept. 1988.
- G. D. Forney, Jr. and M. D. Trott, "The dynamics of group codes: State spaces, trellis diagrams and canonical encoders," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1491–1513, Sept. 1993.
- T. Kasami, T. Takata, T. Fujikiwara, 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, Jan. 1993.
- A. Vardy and Y. Be'ery, "Maximum-likelihood soft-decision decoding of BCH codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 546–554, March 1994.
- Ø. Ytrehus, "On the trellis complexity of linear block codes," *IEEE Trans. Inform. Theory*, to appear.
- 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, Mar. 1993.
- D. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.
- J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, Jan. 1978.

- [13] T. Helleseeth, T. Kløve, and Ø. Ytrehus, "Generalized Hamming weights of linear codes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1133–1140, May 1992.
- [14] T. Kløve, "Minimum support weights of binary codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 648–654, Jan. 1993.
- [15] J. H. Conway and N. J. A. Sloane, "Lexicographic codes: Error-correcting codes from game theory," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 337–348, May 1986.
- [16] V. K. Wei and K. Yang, "On the generalized Hamming weights of product codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1709–1713, Sept. 1993.
- [17] S. Encheva and T. Kløve, "Codes satisfying the chain condition," *IEEE Trans. Inform. Theory*, vol. 40, pp. 175–180, Jan. 1994.
- [18] T. Helleseeth and P. V. Kumar, "The weight hierarchy of the Kasami codes," *Discr. Math.*, to appear.
- [19] R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The  $Z_4$ -linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 301–319, Mar. 1994.
- [20] G. D. Forney, Jr., N. J. A. Sloane, and M. D. Trott, "The Nordstrom–Robinson code is the binary image of the octacode," in *Coding and Quantization: DIMACS/IEEE Workshop (Oct. 19–21, 1992)*, R. Calderbank *et al.*, eds., Providence, RI: Am. Math. Soc., 1993, pp. 19–26.
- [21] G. D. Forney, Jr., "Density/length profiles and trellis complexity of lattices," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1753–1773, Nov. 1994.
- [22] V. K. Wei, "A survey of results on generalized Hamming weights for linear codes," in *Proc. 1993 IEEE Inform. Theory Workshop*, (Susono, Japan), June 1993, pp. 45–46.