Relations between the Local Weight Distributions of a Linear Block Code, Its Extended Code, and Its Even Weight Subcode

Kenji Yasunaga* and Toru Fujiwara†
Graduate School of Information Science and Technology
Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: {k-yasunaga*, fujiwara†}@ist.osaka-u.ac.jp

***Abstract*— Relations between the local weight distributions of a binary linear code, its extended code, and its even weight subcode are presented. In particular, for a code of which the extended code is transitive invariant and contains only codewords with weight multiples of four, the local weight distribution can be obtained from that of the extended code. Using the relations, the local weight distributions of the $(127, k)$ primitive BCH codes for $k \leq 50$, the $(127, 64)$ punctured third-order Reed-Muller , and their even weight subcodes are obtained from the local weight distribution of the $(128, k)$ extended primitive BCH codes for $k \leq 50$ and the $(128, 64)$ third-order Reed-Muller code. We also show an approach to improve an algorithm for computing the local weight distribution proposed before.**

I. Introduction

In a binary linear code, a zero neighbor is a codeword whose Voronoi region shares a facet with that of the all-zero codeword [1]. The local weight distribution [2], [3] (or local distance profile [1], [4], [5], [6], [7]) of a binary linear code is defined as the weight distribution of zero neighbors in the code. Knowledge of the local weight distribution of a code is valuable for the error performance analysis of the code. For example, the local weight distribution could give a tighter upper bound on error probability for soft decision decoding over AWGN channel than the usual union bound [7].

Formulas for local weight distribution are only known for certain classes of codes, Hamming codes and second-order Reed-Muller codes. Although an efficient method to examine zero neighborship of codeword is presented in [1], the computation for obtaining the local weight distribution is a very time-consuming task. As Agrell noted in [1], the automorphism group of the code can help reduce the complexity. Using the automorphism group of cyclic codes, i.e. cyclic permutations, Mohri et al. devised a computation algorithm using a method for obtaining the representative codewords with respect to the cyclic permutations. The algorithm examines the zero neighborship only for the representative codewords. By using the algorithm, they obtained the local weight distributions of the binary primitive BCH codes of length 63 [5], [6].

When the automorphism group properly contains cyclic permutations, for example the affine group, an efficient way of finding the representative codewords is unknown. We proposed an algorithm for computing the local weight distribution of a code using the automorphism group of the set of the all cosets of a subcode in the code [4]. Using the algorithm, we obtained the local weight distributions of the $(128, k)$ extended primitive BCH codes for $k \leq 50$ and the $(128, 64)$ third-order Reed-Muller code [3], [4]. For extended primitive BCH codes, which is closed under the affine group of permutations, our proposed algorithm has considerably smaller complexity than the algorithm in [5] and [6].

However, for cyclic codes, the complexity is not reduced. Then the local weight distributions of the $(127, k)$ primitive BCH codes for $k \geq 36$ were not obtained although those of the corresponding $(128, k)$ extended primitive BCH codes are known. A method for obtaining the local weight distribution of a code from that of its extended code should be considered.

In this paper, a relation between local weight distributions of a binary linear code and its extended code is given. A more concrete relation is presented for the case that the extended code is transitive invariant and contains only codewords with weight multiples of four. Extended binary primitive BCH codes and Reed-Muller codes are transitive invariant codes. A relation between local weight distributions of a binary linear code and its even weight subcode is also given. By using the relations, the local weight distributions of the $(127, k)$ binary primitive BCH codes for $36 \leq k \leq 50$, the $(127, 64)$ punctured third-order Reed-Muller code, and their even weight subcodes are obtained from the local weight distributions of the $(128, k)$ primitive BCH codes for $36 \leq k \leq 50$ and the $(128, 64)$ third-order Reed-Muller code. Finally, we give an approach to improve the algorithm proposed in [4].

II. Local Weight Distribution

Let C be a binary (n, k) linear code. Define a mapping s from $\{0, 1\}$ to $\mathbf{R}$ as $s(0) = 1$ and $s(1) = -1$. The mapping s is naturally extended to one from $\{0, 1\}^n$ to $\mathbf{R}^n$. A zero neighbor of C is defined [1] as follows:

Definition 1 (Zero neighbor): For $\boldsymbol{v} \in C$, define $\boldsymbol{m}_0 \in \mathbf{R}^n$ as $\boldsymbol{m}_0 = \frac{1}{2}(s(\mathbf{0}) + s(\boldsymbol{v}))$ where $\mathbf{0} = (0, 0, \ldots, 0)$. The codeword $\boldsymbol{v}$ is a zero neighbor if and only if

$$d_E(\boldsymbol{m}_0, s(\boldsymbol{v})) = d_E(\boldsymbol{m}_0, s(\boldsymbol{0})) < d_E(\boldsymbol{m}_0, s(\boldsymbol{v}')),$$
$$\text{for any } \boldsymbol{v}' \in C \setminus \{\boldsymbol{0}, \boldsymbol{v}\}, \qquad (1)$$

where $d_E(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbf{R}^n$.

A zero neighbor is also called a minimal codeword in [8]. The following lemma is useful to check whether a given codeword is a zero neighbor or not [1].

Lemma 1: $\boldsymbol{v} \in C$ is a zero neighbor if and only if there is not $\boldsymbol{v}' \in C \setminus \{\boldsymbol{0}\}$ such that $\mathrm{Supp}(\boldsymbol{v}') \subsetneq \mathrm{Supp}(\boldsymbol{v})$. Note that $\mathrm{Supp}(\boldsymbol{v})$ is the set of support of $\boldsymbol{v}$, which is the set of positions of nonzero elements in $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$.

The local weight distribution is defined as follows:

Definition 2 (Local weight distribution): Let $L_w(C)$ be the number of zero neighbors with weight w in C. The local weight distribution of C is defined as the $(n+1)$-tuple $(L_0(C), L_1(C), \ldots, L_n(C))$.

On the local weight distribution, we have the following lemma [2], [8].

Lemma 2: Let $A_w(C)$ be the number of codewords with weight w in C and d be the minimum distance of C.

$$L_w(C) = \begin{cases} A_w(C), & w < 2d, \\ 0, & w > n-k+1. \end{cases} \qquad (2)$$

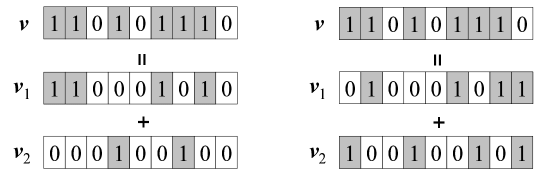
When all the weights w in a code are confined in $w < 2d$ and $w > n-k+1$, the local weight distribution can be obtained from the weight distribution straightforwardly. For example, the local weight distribution of the (n, k) primitive BCH code of length 63 for $k \leq 18$, of length 127 for $k \leq 29$, and of length 255 for $k \leq 45$ can be obtained from their weight distributions.

III. Relations of Local Weight Distributions

A. General relation

Consider a binary linear code C of length n, its extended code C_{ex}, and its even weight subcode C_{even}. For a codeword $\boldsymbol{v} \in C$, let $\mathrm{wt}(\boldsymbol{v})$ be the Hamming weight of $\boldsymbol{v}$ and $\boldsymbol{v}^{(\mathrm{ex})}$ be the corresponding codeword in C_{ex}, that is, $\boldsymbol{v}^{(\mathrm{ex})}$ is obtained from $\boldsymbol{v}$ by adding the over-all parity bit. We define a *decomposable* codeword.

Definition 3 (Decomposable codeword): $\boldsymbol{v} \in C$ is called *decomposable* if $\boldsymbol{v}$ can be represented as $\boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2$ where $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ and $\mathrm{Supp}(\boldsymbol{v}_1) \cap \mathrm{Supp}(\boldsymbol{v}_2) = \emptyset$ (see Figure 1).



(a) decomposable (b) indecomposable

Fig. 1. Examples of a decomposable codeword and an indecomposable codeword.

From Lemma 1, $\boldsymbol{v}$ is not a zero neighbor if and only if $\boldsymbol{v}$ is decomposable. For even weight codewords, we introduce an *only-odd-decomposable* codeword and an *even-decomposable* codeword.

Definition 4: Let $\boldsymbol{v} \in C$ be a decomposable codeword with even $\mathrm{wt}(\boldsymbol{v})$. That is, $\boldsymbol{v}$ is not a zero neighbor in C. $\boldsymbol{v}$ is said to be *only-odd-decomposable*, if all the decomposition of $\boldsymbol{v}$ is of the form $\boldsymbol{v}_1 + \boldsymbol{v}_2$ with the odd weight codewords $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$. Otherwise, $\boldsymbol{v}$ is said to be *even-decomposable*.

When $\boldsymbol{v}$ is even-decomposable, there is a decomposition of $\boldsymbol{v}$, $\boldsymbol{v}_1 + \boldsymbol{v}_2$ such that both $\mathrm{wt}(\boldsymbol{v}_1)$ and $\mathrm{wt}(\boldsymbol{v}_2)$ are even.

The relation between C and C_{ex} with respect to zero neighborship is given in the following theorem, which is also summarized in Table I.

Theorem 1:

1) For a zero neighbor $\boldsymbol{v}$ in C, $\boldsymbol{v}^{(\mathrm{ex})}$ is a zero neighbor in C_{ex}.
2) For a codeword $\boldsymbol{v}$ which is not a zero neighbor in C, the following a) and b) hold.
   a) When $\mathrm{wt}(\boldsymbol{v})$ is odd, $\boldsymbol{v}^{(\mathrm{ex})}$ is not a zero neighbor in C_{ex}.
   b) When $\mathrm{wt}(\boldsymbol{v})$ is even, $\boldsymbol{v}^{(\mathrm{ex})}$ is a zero neighbor in C_{ex} if and only if $\boldsymbol{v}$ is only-odd-decomposable in C.

Proof: 1) Suppose that $\boldsymbol{v}^{(\mathrm{ex})}$ is not a zero neighbor in C_{ex}. Then $\boldsymbol{v}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{v}_1^{(\mathrm{ex})} + \boldsymbol{v}_2^{(\mathrm{ex})}$, and hence $\boldsymbol{v}$ is decomposable into $\boldsymbol{v}_1 + \boldsymbol{v}_2$. This contradicts the indecomposability of $\boldsymbol{v}$.

2) Suppose that $\boldsymbol{v}$ is decomposed into $\boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2$. a) Since $\mathrm{wt}(\boldsymbol{v})$ is odd, the sum of the parity bits in $\boldsymbol{v}_1^{(\mathrm{ex})}$ and $\boldsymbol{v}_2^{(\mathrm{ex})}$ is one. Also, the parity bit in $\boldsymbol{v}^{(\mathrm{ex})}$ is one. Then $\boldsymbol{v}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{v}_1^{(\mathrm{ex})} + \boldsymbol{v}_2^{(\mathrm{ex})}$, and $\boldsymbol{v}^{(\mathrm{ex})}$ is not a zero neighbor in C_{ex}. b) Since $\mathrm{wt}(\boldsymbol{v})$ is even, the parity bit in $\boldsymbol{v}^{(\mathrm{ex})}$ is zero. (If part) Suppose that $\boldsymbol{v}^{(\mathrm{ex})}$ is not a zero neighbor in C_{ex}. Then there exists a decomposition $\boldsymbol{v}^{(\mathrm{ex})} = \boldsymbol{v}_1^{(\mathrm{ex})} + \boldsymbol{v}_2^{(\mathrm{ex})}$. Because the parity bit in $\boldsymbol{v}^{(\mathrm{ex})}$ is zero, the parity bits in $\boldsymbol{v}_1^{(\mathrm{ex})}$ and $\boldsymbol{v}_2^{(\mathrm{ex})}$ must be zero. This implies that $\boldsymbol{v}$ is even-decomposable into $\boldsymbol{v}_1 + \boldsymbol{v}_2$, and contradicts the assumption that $\boldsymbol{v}$ is only-odd-decomposable. (Only if part) Suppose that $\boldsymbol{v}$ is even-decomposable. Then there is a decomposition such that the parity bits in both $\boldsymbol{v}_1^{(\mathrm{ex})}$ and $\boldsymbol{v}_2^{(\mathrm{ex})}$ are zero. For such the decomposition, $\boldsymbol{v}^{(\mathrm{ex})}$ is decomposable into $\boldsymbol{v}_1^{(\mathrm{ex})} + \boldsymbol{v}_2^{(\mathrm{ex})}$, and $\boldsymbol{v}^{(\mathrm{ex})}$ is not a zero neighbor in C_{ex} (see Figure 2). □

From 2)-b) of Theorem 1, there may be codewords that are not zero neighbors in C although their extended codewords are zero neighbors in C_{ex}. Such codewords are the only-odd decomposable codewords. For investigating relations of local weight distributions between a code and its extended code, only-odd decomposable codewords are important.

The following theorem is a direct consequence of Theorem 1.

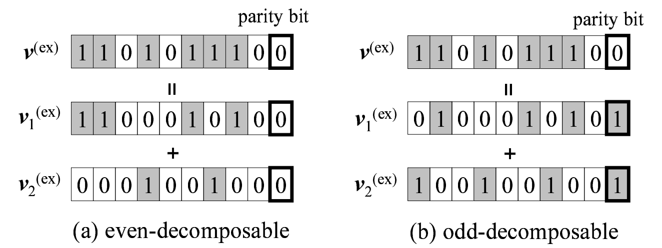


Fig. 2. Examples of an even-decomposable codeword and an odd-decomposable codeword mentioned in the proof of Theorem 1-2)-b).

Theorem 2: For a code C of length n,

$$L_{2i}(C_{\mathrm{ex}}) = L_{2i-1}(C)+L_{2i}(C)+N_{2i}(C),\ 0 \le i \le n/2, \quad (3)$$

where $N_j(C)$ is the number of only-odd decomposable codewords with weight j in C.

From Theorem 2, if there is no only-odd decomposable codeword in C, the local weight distributions of C_{ex} are obtained from that of C. Next, we give a useful sufficient condition under which no only-odd-decomposable codeword exists.

Theorem 3: If all the weights of codewords in C_{ex} are multiples of four, no only-odd-decomposable codeword exists in C.

Proof: If $\boldsymbol{v} \in C$ is an only-odd-decomposable codeword and decomposed into $\boldsymbol{v}_1 + \boldsymbol{v}_2$, the weights of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ can be represented as $\mathrm{wt}(\boldsymbol{v}_1) = 4i - 1$ and $\mathrm{wt}(\boldsymbol{v}_2) = 4j - 1$ where i and j are integers. Then $\mathrm{wt}(\boldsymbol{v}) = \mathrm{wt}(\boldsymbol{v}_1 + \boldsymbol{v}_2) = \mathrm{wt}(\boldsymbol{v}_1) + \mathrm{wt}(\boldsymbol{v}_2) = (4i-1) + (4j-1) = 4i + 4j - 2$. This contradicts the fact that $\mathrm{wt}(\boldsymbol{v})$ is a multiple of four. □

For example, all the weights of codewords in the $(128, k)$ extended primitive BCH code with $k \le 57$ are multiples of four. The parameters of Reed-Muller codes with which all the weights of codewords are multiples of four are given by Corollary 13 of Chapter 15 in [9]. From the corollary, the third-order Reed-Muller codes of length 128, 256, and 512 have only codewords whose weights are multiples of four.

Although the local weight distribution of C_{ex} for these codes can be obtained from that of C by using Theorem 2, what we need is a method for obtaining the local weight distribution of C from that of C_{ex}. We need to know the number of zero neighbors with parity bit one. In Section III-B, we will show a method to obtain the number of zero neighbors with parity bit one for a class of transitive invariant codes.

A similar relation to that between C and C_{ex} holds between C and C_{even}. This relation is given in Theorem 4 without proof (see Table I).

Theorem 4:
1) For an even weight zero neighbor $\boldsymbol{v}$ in C, $\boldsymbol{v}$ is a zero neighbor in C_{even}.
2) For an even weight codeword $\boldsymbol{v}$ which is not a zero neighbor in C, $\boldsymbol{v}$ is a zero neighbor in C_{even} if and only if $\boldsymbol{v}$ is only-odd-decomposable in C.

From Theorem 4, we have Theorem 5.

Theorem 5: For a code C of length n,

$$L_{2i}(C_{\mathrm{even}}) \;=\; L_{2i}(C) + N_{2i}(C), \quad 0 \le i \le n/2. \quad (4)$$

B. Relation for transitive invariant extended codes

A transitive invariant code is a code which is invariant under a transitive group of permutations. A group of permutations is said to be transitive if for any two symbols in a codeword there exists a permutation that interchanges them [10]. The extended primitive BCH codes and Reed-Muller codes are transitive invariant codes. For a transitive invariant C_{ex}, a relation between the (global) weight distributions of C and C_{ex} is presented in Theorem 8.15 in [10]. A similar relation holds for local weight distribution.

Lemma 3: If C_{ex} is a transitive invariant code of length $n+1$, the number of zero neighbors with parity bit one is $\frac{w}{n+1} L_w(C_{\mathrm{ex}})$.

Proof: This lemma can be proved in a similar way as the proof of Theorem 8.15. Arrange all zero neighbors with weight w in a column. Next, interchange the j-th column and the last column, which is the parity bit column, for all these codewords with the permutation. All the resulting codewords have weight w and must be the same as the original set of codewords. Thus, the number of ones in the j-th column and that in the last column are the same. Denote this number l_w, which is the same as the number of zero neighbors of weight w with parity bit one. Then the total ones in the original set of codewords is $(n+1)\, l_w$, or $L_w(C_{\mathrm{ex}})$ times the weight w. Thus, $(n+1)\, l_w = w L_w(C_{\mathrm{ex}})$, and $l_w = \frac{w}{n+1} L_w(C_{\mathrm{ex}})$. □

It is clear that there are $\frac{n+1-w}{n+1} L_w(C_{\mathrm{ex}})$ zero neighbors with weight w whose parity bit is zero from this lemma. The following theorem is obtained from Theorem 1 and Lemma 3.

Theorem 6: If C_{ex} is a transitive invariant code of length $n+1$,

$$L_w(C) \;=\; \frac{w+1}{n+1} L_{w+1}(C_{\mathrm{ex}}), \qquad \text{for odd } w, \quad (5)$$

$$L_w(C) \;=\; \frac{n+1-w}{n} L_w(C_{\mathrm{ex}}) - N_w(C), \quad (6)$$

$$\le \; \frac{n+1-w}{n+1} L_w(C_{\mathrm{ex}}), \quad \text{for even } w. \quad (7)$$

If there is no only-odd-decomposable codeword in a transitive invariant code C, the equality of (7) holds. That is, in this case, we have that

$$L_w(C) \;=\; \frac{n+1-w}{n} L_w(C_{\mathrm{ex}}), \quad \text{for even } w. \quad (8)$$

Therefore, for a transitive invariant code C_{ex} having no only-odd-decomposable codeword in C, the local weight distributions of C can be obtained from that of C_{ex} by using (5) and (8) in Theorem 6. After computing the local weight distribution of C, that of C_{even} can be obtained by using Theorem 5.

TABLE I

ZERO NEIGHBORSHIP OF $\boldsymbol{v}$ IN A LINEAR BLOCK CODE, $\boldsymbol{v}_{\rm ex}$ IN ITS EXTENDED CODE, AND $\boldsymbol{v}$ IN ITS EVEN WEIGHT SUBCODE.

<table>
<tr><th colspan="3">$\boldsymbol{v}$ in C</th><th colspan="2">$\boldsymbol{v}^{(\rm ex)}$ in $C_{\rm ex}$</th><th colspan="2">$\boldsymbol{v}$ in $C_{\rm even}$</th></tr>
<tr><th>Zero neighborship</th><th>Weight</th><th>Decomposability</th><th>Zero neighborship</th><th>Theorem 1</th><th>Zero neighborship</th><th>Theorem 4</th></tr>
<tr><td rowspan="2">Yes</td><td>Odd</td><td rowspan="2">Not decomposable</td><td rowspan="2">Yes</td><td rowspan="2">1)</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Even</td><td>Yes</td><td>1)</td></tr>
<tr><td rowspan="3">No</td><td>Odd</td><td>Decomposable</td><td>No</td><td>2) - a)</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Even</td><td>Only-odd-decomposable</td><td>Yes</td><td rowspan="2">2) - b)</td><td>Yes</td><td rowspan="2">2)</td></tr>
<tr><td>Even</td><td>Even-decomposable</td><td>No</td><td>No</td></tr>
</table>

IV. OBTAINED LOCAL WEIGHT DISTRIBUTIONS

As discussed in the previous section, the local weight distributions of the $(127,k)$ primitive BCH codes for $k \leq 57$, the punctured third-order Reed-Muller codes of length 127, 255, and 511, and their even weight subcodes are obtained from those of the corresponding extended codes by using Theorems 5 and 6. Since the local weight distribution for the $(128,57)$ extended primitive BCH code and the third-order Reed-Muller codes of length 256 and 512 are unknown, only the local weight distributions of the $(127,k)$ primitive BCH codes for $k = 36, 43, 50$, the $(127,64)$ punctured third-order Reed-Muller code, and their even weight subcodes are obtained. These local weight distributions are presented in Table II. The local weight distributions of the corresponding even weight subcodes are obtained straightforwardly from the distributions given in the table by using Theorem 5.

V. COMPUTATION ALGORITHM FOR LONGER CODES

As mentioned in Section I, we have computed the local weight distributions of several extended BCH codes and the third-order Reed-Muller code of length 128 by using our computation algorithm in [4]. In the algorithm, the time complexity is reduced by reducing the number of codewords whose zero neighborship should be checked. However, it is still time consuming to compute the local weight distribution of a (n,k) code with larger k and/or larger n. In improving the computation algorithm, the next target codes include the $(256,93)$ Reed-Muller code and the $(128,57)$ extended BCH code.

In this section, we give an approach for improving the algorithm proposed in [4]. First, we briefly review this algorithm. For a binary (n,k) linear code C and its linear subcode with dimension k', let C/C' denote the set of cosets of C' in C, that is, $C/C' = \{\boldsymbol{v}+C' : \boldsymbol{v} \in C \setminus C'\}$. Then,

$$|C/C'| = 2^{k-k'}, \quad \text{and} \quad C = \bigcup_{D\in C/C'} D. \tag{9}$$

Definition 5 (Local weight subdistribution for cosets): Let D be a coset in C/C' and $LS_w(D)$ be the number of zero neighbors of C in D with weight w. The local weight subdistribution for a coset $D \in C/C'$ is the $(n+1)$-tuple $(LS_0(D), LS_1(D), \ldots, LS_n(D))$.

Then

$$L_w(C) = \sum_{D\in C/C'} LS_w(D). \tag{10}$$

The following theorem gives an invariance property under permutations for cosets.

Theorem 7 (Invariance property for cosets): For $D_1, D_2 \in C/C'$, the local weight subdistribution for D_1 and that for D_2 are the same if there exists $\pi \in \mathrm{Aut}(C)$ such that $\pi[D_1] = D_2$, where $\mathrm{Aut}(C)$ is the automorphism group of C and $\pi[D_1] = \{\pi\boldsymbol{v} : \boldsymbol{v} \in D_1\}$.

In the algorithm in [4], given a subgroup P of $\mathrm{Aut}(C)$, cosets in C/C' are partitioned into equivalence classes such that two cosets D_1 and D_2 are in the same class if and only if there is $\pi \in P$ with $\pi[D_1] = D_2$. Then if the local weight subdistributions only for the representative cosets of the classes and the sizes of the classes are obtained, we could obtain the local weight distribution of C.

Next, we show the approach. To reduce the complexity more, we consider using the invariance property for cosets in computing the local weight subdistributions for the representative cosets. This means that we consider a coset $\boldsymbol{v}+C' \in C/C'$ as the set of cosets of C'', where C'' is a linear subcode of C'.

For a coset $\boldsymbol{v}+C' \in C/C'$, let $(\boldsymbol{v}+C')/C''$ denote the set of all cosets of C'' in $\boldsymbol{v}+C'$, that is, $(\boldsymbol{v}+C')/C'' = \{\boldsymbol{v}+\boldsymbol{u}+C'' : \boldsymbol{u} \in C' \setminus C''\}$. We also call the weight distribution of zero neighbors in $E \in (\boldsymbol{v}+C')/C''$ the local weight subdistribution for E. The following theorem gives an invariance property for cosets in $(\boldsymbol{v}+C')/C''$.

Theorem 8: For $E_1, E_2 \in (\boldsymbol{v}+C')/C''$, the local weight subdistribution for E_1 and that for E_2 are the same if there exists $\pi \in \{\rho : \rho\boldsymbol{v} \in \boldsymbol{v}+C', \rho \in \mathrm{Aut}(C)\cap\mathrm{Aut}(C')\}$ such that $\pi[E_1] = E_2$, where $\pi[E] = \{\pi\boldsymbol{v} : \boldsymbol{v} \in E\}$.

We consider partitioning $(\boldsymbol{v}+C')/C''$ into the equivalence classes. Permutations which are used to partition cosets into equivalence classes are presented in the following lemma.

Lemma 4: For a coset $\boldsymbol{v}+C' \in (\boldsymbol{v}+C')/C''$,

$$\begin{aligned}&\{\pi : \pi[E] \in (\boldsymbol{v}+C')/C'' \text{ for any } E \in (\boldsymbol{v}+C')/C''\}\\ &\quad = \{\rho : \rho\boldsymbol{v} \in \boldsymbol{v}+C',\ \rho \in \mathrm{Aut}(C)\cap\mathrm{Aut}(C')\cap\mathrm{Aut}(C'')\}.\end{aligned}$$

In order to partition cosets into equivalence classes, we should use permutations presented in Lemma 4. Even if $\mathrm{Aut}(C)$, $\mathrm{Aut}(C')$, and $\mathrm{Aut}(C'')$ are known, we should obtain permutations π that satisfy $\pi\boldsymbol{v} \in \boldsymbol{v}+C'$.

TABLE II

THE LOCAL WEIGHT DISTRIBUTIONS OF THE $(127, k)$ PRIMITIVE BCH CODES FOR $k = 36, 43,$ and 50 AND THE PUNCTURED THIRD-ORDER REED-MULLER CODE OF LENGTH 127 .

$(127, 36)$ BCH code

w	L_w
31	2,667
32	8,001
35	4,572
36	11,684
39	640,080
40	1,408,176
43	12,220,956
44	23,330,916
47	132,560,568
48	220,934,280
51	823,921,644
52	1,204,193,172
55	3,157,059,472
56	4,059,076,464
59	7,022,797,740
60	7,959,170,772
63	9,742,066,368
64	9,742,066,368
67	7,959,170,772
68	7,022,797,740
71	4,059,071,892
72	3,157,055,916
75	1,204,193,172
76	823,921,644
79	217,627,200
80	130,576,320
83	23,330,916
84	12,220,956
87	1,408,176
88	640,080

$(127, 43)$ BCH code

w	L_w
31	31,115
32	93,345
35	2,478,024
36	6,332,728
39	82,356,960
40	181,185,312
43	1,554,145,736
44	2,967,005,496
47	16,837,453,752
48	28,062,422,920
51	106,485,735,720
52	155,632,998,360
55	400,716,792,672
56	515,207,304,864
59	905,612,814,120
60	1,026,361,189,336
63	1,238,334,929,472
64	1,238,334,929,472
67	1,026,345,592,720
68	905,599,052,400
71	515,097,101,376
72	400,631,078,848
75	155,191,535,184
76	106,183,681,968
79	26,980,367,680
80	16,188,220,608
83	1,617,588,840
84	847,308,440

$(127, 50)$ BCH code

w	L_w
27	40,894
28	146,050
31	4,853,051
32	14,559,153
35	310,454,802
36	793,384,494
39	10,538,703,840
40	23,185,148,448
43	199,123,183,160
44	380,144,258,760
47	2,154,195,406,104
48	3,590,325,676,840
51	13,633,106,229,288
52	19,925,309,104,344
55	51,285,782,220,204
56	65,938,862,854,548
59	115,927,157,830,260
60	131,384,112,207,628
63	158,486,906,385,472
64	158,486,906,385,472
67	131,258,388,369,668
68	115,816,225,032,060
71	64,917,266,933,304
72	50,491,207,614,792
75	15,345,182,164,032
76	10,499,335,164,864

$(127, 64)$ punc. RM code

w	L_w
15	11,811
16	82,677
23	13,889,736
24	60,188,856
27	684,345,088
28	2,444,089,600
31	77,893,639,488
32	233,680,918,464
35	5,097,898,213,632
36	13,027,962,101,504
39	172,489,249,981,440
40	379,476,349,959,168
43	3,259,718,804,643,840
44	6,223,099,536,138,240
47	35,130,035,853,803,520
48	58,550,059,756,339,200
51	218,602,288,622,075,904
52	319,495,652,601,495,552
55	766,899,891,905,495,040
56	986,014,146,735,636,480
59	1,306,771,964,441,395,200
60	1,481,008,226,366,914,560
63	258,664,522,171,023,360
64	258,664,522,171,023,360

Let $\mathrm{RM}(r, m)$ denote the r-th order Reed-Muller code of length 2^m. We consider the case of the $(256, 93)$ third-order Reed-Muller code, denoted by $\mathrm{RM}(3, 8)$. The equivalent cosets in $\mathrm{RM}(3, 8)/\mathrm{RM}(2, 8)$ are presented in [11], and there are 32 equivalence classes. We choose $\mathrm{RM}(1, 8)$ as a subcode of $\mathrm{RM}(2, 8)$. Then the general affine group [9] is a subgroup of $\mathrm{Aut}(\mathrm{RM}(3, 8)) \cap \mathrm{Aut}(\mathrm{RM}(2, 8)) \cap \mathrm{Aut}(\mathrm{RM}(1, 8))$. For each cosets in $\mathrm{RM}(3, 8)/\mathrm{RM}(2, 8)$, the estimated time for computing the local weight subdistribution is about 54 days with the algorithm in [4]. The total estimated time is about 1700 days. To compute the local weight distribution of $\mathrm{RM}(3, 8)$ in practical time, we should find the permutations π that satisfy $\pi\boldsymbol{v} \in \boldsymbol{v} + \mathrm{RM}(2, 8)$ for each coset $\boldsymbol{v} + \mathrm{RM}(2, 8)$ in $\mathrm{RM}(3, 8)/\mathrm{RM}(2, 8)$. If we could find more than 50 such permutations for each cosets, the local weight distribution of $\mathrm{RM}(3, 8)$ may be computable.

VI. CONCLUSION

In this paper, some relations between local weight distributions of a binary linear code, its extended code, and its even weight subcode are presented. The local weight distributions of the $(127, k)$ primitive BCH codes for $k = 36, 43, 50$, the $(127, 64)$ punctured third-order Reed-Muller code, and their even weight subcodes are obtained. If the local weight distribution of the $(128, 57)$ extended primitive BCH code and the $(256, 93)$ third-order Reed-Muller code are obtained, we can obtain the local weight distributions of the $(127, 57)$ primitive BCH code, the $(255, 93)$ punctured third-order Reed-Muller code, and their even weight subcodes.

REFERENCES

[1] E. Agrell, “Voronoi regions for binary linear block codes,” *IEEE Trans. Inform. Theory*, vol.42, no.1, pp.310–316, Jan. 1996.

[2] E. Agrell, “On the Voronoi Neighbor Ratio for Binary Linear Block Codes,” *IEEE Trans. Inform. Theory*, vol.44, no.7, pp.3064–3072, Nov. 1998.

[3] K. Yasunaga and T. Fujiwara, “The local weight distributions of the (128,50) extended binary primitive BCH code and (128,64) Reed-Muller code,” *IEICE Technical Report*, IT2004-19, Jul. 2004.

[4] K. Yasunaga and T. Fujiwara, “An algorithm for computing the local weight distribution of binary linear codes closed under a group of permutations,” *Proc. of ISITA2004*, pp.846–851, Oct. 2004.

[5] M. Mohri, and M. Morii, “On computing the local distance profile of binary cyclic codes,” *Proc. of ISITA2002*, pp.415–418, Oct. 2002.

[6] M. Mohri, Y. Honda, and M. Morii,“A method for computing the local weight distribution of binary cyclic codes,” *IEICE Trans.Fundamentals (Japanese Edition)*, vol.J86-A, no.1, pp.60–74, Jan. 2003.

[7] G.D. Forney, Jr., “Geometrically uniform codes,” *IEEE Trans. Inform. Theory*, vol.37, no.5, pp.1241–1260, Sept, 1991.

[8] A. Ashikhmin and A. Barg, “Minimal vectors in linear codes,” *IEEE Trans. Inform. Theory*, vol.44, no.5, pp.2010–2017, Sept. 1998.

[9] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.

[10] W.W. Peterson and E. J. Weldon, Jr., *Error-correcting codes, 2nd Edition*, MIT Press, 1972.

[11] X. Hou, “GL(m,2) acting on R(r,m)/R(r-1,m),” *Discrete Mathematics*, 149, pp.99–122, 1996.