社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

2元線形符号における記号位置置換不変性を用いた局所距離分布の計算法

安永　憲司† 　藤原　融†

† 大阪大学 大学院情報科学研究科 マルチメディア工学専攻
〒 560–8531　豊中市待兼山町 1–3
E-mail: †{k-yasunaga,fujiwara}@ist.osaka-u.ac.jp

あらまし　2元線形符号の局所距離分布を求めるアルゴリズムを提案する. 符号の記号位置置換不変性を利用して計算量を減らす工夫をしている. 巡回置換に対する局所距離分布の不変性は知られているが, これを一般の記号位置置換に対する不変性に拡張した. この不変性を有効に利用するために, 符号をその線形部分符号の剰余類の集合とみなす. そして, 符号語集合ではなく, この剰余類集合を不変性を用いて同値類に分割する. 同値類の代表元に対してだけ計算を行い, 符号の局所距離分布を求めることにより, 計算量を削減した. 提案計算法を 2 元 $(128, k)$ 拡大原始 BCH 符号に適用し, 局所距離分布を $k \leqq 43$ の場合に求めた.

キーワード　局所距離分布，零隣接語, 記号位置置換不変性，剰余類，拡大原始 BCH 符号

An Algorithm for Computing the Local Distance Profile of Binary Linear Codes Closed under a Group of Permutations

Kenji YASUNAGA† and Toru FUJIWARA†

† Graduate School of Information Science and Technology, Osaka University
1–3 Machikaneyamacho Toyonaka, Osaka 560–8531 Japan
E-mail: †{k-yasunaga,fujiwara}@ist.osaka-u.ac.jp

Abstract　We propose an algorithm for computing the local distance profile of binary linear codes which are closed under a group of permutations. An invariance property is used in the algorithm, which is an extension of the invariance property used in a known algorithm for binary cyclic codes. To use the extended invariance property, the proposed algorithm regards the code as the set of cosets of a subcode. The set of cosets are partitioned into equivalence classes by the invariance property. Only the local distance subprofile for the representative coset in each equivalence class is computed. We apply the algorithm to the $(128, k)$ extended primitive BCH codes, and obtain the local distance profile of codes for $k \leqq 43$.

Key words　Local distance profile, zero neighbor, invariance property, coset, extended primitive BCH code

1. Introduction

For a binary linear code, the local distance profile is defined as the weight distribution of the zero neighbors in the code, where a zero neighbor is a codeword whose Voronoi region shares a facet with that of the all-zero codeword [1]. Knowing the local distance profile of a code is useful for the error performance analysis of the code. For example, the local distance profile gives a tighter upper bound on error probability for soft decision decoding over an AWGN (additive white gaussian noise) channel than the usual union bound.

For binary Hamming codes, the formula of the local distance profile is known [2]. For binary second order Reed-Muller codes, the relation between the local distance profile and the weight distribution is known [2]. For other codes, theoretically, we can compute the local distance profile by examining every codeword whether it is a zero neighbor or not. However, it is infeasible to compute the profile except for the codes with small dimensions.

Recently, an algorithm for computing the local distance profile of binary cyclic codes has been proposed by Mohri, Honda and Morii [3]. We call it the MHM algorithm in this report. The algorithm reduces the computational complex-

ity by using the following invariance property: any cyclic permutation of a codeword is a zero neighbor if and only if the codeword is a zero neighbor. The algorithm generates the representative codeword and the number of the codewords for every equivalence class of codewords concerning the group of cyclic permutations, and checks whether each of the representatives is a zero neighbor or not. It is easy to generalize the above invariance property for the group of cyclic permutations to any group of permutations, say affine group of permutation. However, it is not easy to obtain the representative codewords and the number of the equivalent codewords.

In this report, we propose an algorithm for computing the local distance profile of binary linear codes which are closed under any group of permutations. For this, we extend the invariance property of the codewords to that of the cosets. This idea is used in [4] for computing the weight distributions of extended binary primitive BCH codes. We apply the proposed algorithm to the extended binary primitive BCH codes, which are closed under the affine permutations. The complexity of the proposed algorithm is about 1/21 to 1/56 as much as that of the MHM algorithm for the codes of length 64, and about 1/43 or 1/64 for the codes of length 128 except for some cases. The local distance profile for $(128, 43)$ extended binary primitive BCH code is obtained, which has not been obtained. When the proposed algorithm is applied to cyclic codes, the complexity is almost as much as the MHM algorithm.

2. Local Distance Profile

Let C be an (n, k) binary 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 $\mathbf{R}^n$. The zero neighbor is defined as follows:

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

$$d_E(\boldsymbol{m}_0, s(\boldsymbol{c})) = d_E(\boldsymbol{m}_0, s(\mathbf{0})) < d_E(\boldsymbol{m}_0, s(\boldsymbol{c}')), \quad \text{for every } \boldsymbol{c}' \in C \setminus \{\mathbf{0}, \boldsymbol{c}\}, \quad (1)$$

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

The following lemma is useful to check whether a given codeword is a zero neighbor or not.

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

$$\mathrm{Supp}(\boldsymbol{v}) = \{i : v_i \neq 0,\ 1 \leqq i \leqq n\}. \quad (2)$$

The local distance profile is defined as follows:

［Definition 2］（Local distance profile） Let L_w be the number of zero neighbors with weight w in C. Then, the $(n+1)$-tuple, $(L_0, L_1, \ldots, L_n)$, is the local distance profile of C.

On the local distance profile, we have the following lemma.

［Lemma 2］ [2] Let A_w be the number of codewords with weight w in C and d be the minimum distance of C.

$$L_w = \begin{cases} A_w, & w < 2d, \\ 0, & w > n - k + 1. \end{cases} \quad (3)$$

To obtain the local distance profile, if the weight distribution is known, only L_w with $2d \leqq w \leqq n - k + 1$ are need to be obtained.

The MHM algorithm proposed in [3] uses the following invariance property under cyclic permutations.

［Theorem 1］ [3] Let C be a binary cyclic code. A cyclic permuted codeword of $\boldsymbol{c}$ is a zero neighbor if a codeword $\boldsymbol{c}$ $\in C$ is a zero neighbor.

［Corollary 1］ Let C be a binary cyclic code, and $\sigma^i\boldsymbol{c}$ be an i times cyclic permuted codeword of $\boldsymbol{c} \in C$. Consider a set $S = \{\boldsymbol{c}, \sigma\boldsymbol{c}, \sigma^2\boldsymbol{c}, \ldots, \sigma^{p(\sigma,\boldsymbol{c})-1}\boldsymbol{c}\}$, where $p(\sigma, \boldsymbol{c})$ is the period of σ, which is the minimum i such that $\sigma^i\boldsymbol{c} = \boldsymbol{c}$. We call S a cycle set. Then, (1) if $\boldsymbol{c}$ is a zero neighbor, all codewords in the cycle set S are zero neighbors; and (2) otherwise, all codewords in S are not zero neighbors.

In the MHM algorithm, all codewords are partitioned into cycle sets. Only one codeword in each cycle sets is generated and checked whether a zero neighbor or not.

3. Coset Partitioning for Computing the Local Distance Profile of Codes Closed under a Group of Permutations

Theorem 1 is an invariance property for the class of cyclic permutations, which is useful for cyclic codes. For the extended code of a cyclic code, we can modify the MHM algorithm easily to compute the local distance profile of the code. However, the extended codes of primitive BCH codes and Reed-Muller codes are closed under larger groups of the permutations. In this section, we show an invariance property under a group of permutations, and present the coset partitioning technique to use the property effectively.

3.1 An invariance property under permutations

For a permutation π and a set of vectors D, the set of the permuted vectors $\pi[D]$ is defined as

$$\pi[D] = \{\pi\boldsymbol{v} : \boldsymbol{v} \in D\}. \quad (4)$$

The automorphism group of a code C is the set of all permutations by which C is permuted into C, and denoted by $\mathrm{Aut}(C)$. It is defined as follows:

[Definition 3] (Automorphism group of a code)

$$\mathrm{Aut}(C) = \{\pi : \pi[C] = C\}. \tag{5}$$

An invariance property under the automorphism group is given in the following theorem.

[Theorem 2] For $\pi \in \mathrm{Aut}(C)$ and $\boldsymbol{c} \in C$, $\pi\boldsymbol{c}$ is a zero neighbor if $\boldsymbol{c}$ is a zero neighbor.

(Proof) Suppose that $\boldsymbol{c}$ is a zero neighbor and $\pi\boldsymbol{c}$ is not a zero neighbor. There exists a nonzero codeword $\boldsymbol{c}' \in C$ such that $\mathrm{Supp}(\pi\boldsymbol{c}) \supsetneqq \mathrm{Supp}(\boldsymbol{c}')$ from lemma 1. Since $\mathrm{Aut}(C)$ is a group, there exists $\boldsymbol{c}'' \in C$ such that $\boldsymbol{c}' = \pi\boldsymbol{c}''$. Thus $\mathrm{Supp}(\pi\boldsymbol{c}) \supsetneqq \mathrm{Supp}(\pi\boldsymbol{c}'')$, and $\mathrm{Supp}(\boldsymbol{c}) \supsetneqq \mathrm{Supp}(\boldsymbol{c}'')$. This contradicts being the zero neighbor of $\boldsymbol{c}$, from Lemma 1. □

This theorem extends Corollary 1 as follows:

[Corollary 2] For $\boldsymbol{c} \in C$, consider a set $S = \{\pi\boldsymbol{c} : \forall\pi \in \mathrm{Aut}(C)\}$. Then, (1) if $\boldsymbol{c}$ is a zero neighbor, all codewords in S are zero neighbors; and (2) otherwise, all codewords in S are not zero neighbors.

It is straightforward to devise a similar algorithm as the MHM algorithm. However, for almost all group of permutations, no efficient way is known for generating the representative codewords. To use this invariance property, we will apply the invariance property to the set of cosets of a subcode rather than the set of codewords.

3.2 Local distance subprofile for a coset

For a binary linear code C and a linear subcode C' of C, let C/C' denote the set of cosets of C' in C, that is,

$$C/C' = \{\boldsymbol{c} + C' : \boldsymbol{c} \in C\}. \tag{6}$$

For an (n, k) code C and its subcode C' with dimension k',

$$|C/C'| = 2^{k-k'}, \quad \text{and} \quad C = \bigcup_{D \in C/C'} D. \tag{7}$$

[Definition 4] (Local distance subprofile for a coset) The local distance subprofile for a coset $D \in C/C'$ is the weight distribution of zero neighbors of C in D. The local distance subprofile for D is $(|Z_0(D)|, |Z_1(D)|, \ldots, |Z_n(D)|)$, where

$$Z_w(D) = \{\boldsymbol{c} \in D : \mathrm{Supp}(\boldsymbol{c}') \nsubseteqq \mathrm{Supp}(\boldsymbol{c}),\ \forall\boldsymbol{c}' \in C \setminus \{\boldsymbol{0}, \boldsymbol{c}\}, \text{ and the Hamming weight of } \boldsymbol{c} \text{ is } w\}, \tag{8}$$

with $0 \leqq w \leqq n$.

Then, from (7), the local distance profile of C is given as the sum of the local distance subprofiles for the cosets in C/C', that is,

$$L_w = \sum_{D \in C/C'} |Z_w(D)|. \tag{9}$$

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

[Theorem 3] For $D_1, D_2 \in C/C'$, the local distance subprofile for D_1 and for D_2 are the same if there is a permutation π such that $\pi[D_1] = D_2$.

(Proof) For any codewords $\boldsymbol{c} \in D_1$, $\pi\boldsymbol{c} \in D_2$, from Theorem 2, $\pi\boldsymbol{c}$ is a zero neighbor if and only if $\boldsymbol{c}$ is a zero neighbor. Therefore the local distance subprofile for D_1 and that for D_2 are the same. □

Next, we give a condition for being same local distance subprofile for each cosets.

3.3 Partitioning the set of cosets with the same local distance subprofile

First, the following lemma gives the set of all permutations by which every coset in C/C' is permuted into one in C/C'.

[Lemma 3] For a linear code C and its subcode C',

$$\{\pi : \pi[D] \in C/C', \forall D \in C/C'\} = \mathrm{Aut}(C) \cap \mathrm{Aut}(C'). \tag{10}$$

Usually, $\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$ (or even $\mathrm{Aut}(C)$) is not known. Only subgroups of $\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$ are known. Therefore, we use a subgroup.

[Definition 5] Let Π be a subset of $\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$. For $D_1, D_2 \in C/C'$, we denote $D_1 \sim_\Pi D_2$ if and only if there is $\pi \in \Pi$ such that $\pi[D_1] = D_2$.

[Lemma 4] The relation "$\sim_\Pi$" is an equivalence relation on C/C', if Π forms a group.

When the set of cosets are partitioned into the equivalence classes by the relation "$\sim_\Pi$", the local distance subprofiles for cosets which belong to the same equivalence class are the same.

Finally, we give a useful theorem for partitioning the set of cosets into equivalence classes by the relation "$\sim_\Pi$."

[Theorem 4] Let $\Pi \subseteqq \mathrm{Aut}(C) \cap \mathrm{Aut}(C')$. For $D_1, D_2 \in C/C'$ and $\pi \in \Pi$, $D_1 \sim_\Pi D_2$ if $\pi\boldsymbol{v}_1 \in D_2$ for any $\boldsymbol{v}_1 \in D_1$.

(Proof) Let $\pi\boldsymbol{v}_1 = \boldsymbol{v}_2 \in D_2$. Any codeword in D_1 is represented by $\boldsymbol{v}_1 + \boldsymbol{c} (\boldsymbol{c} \in C')$. Then,

$$\begin{aligned} \pi(\boldsymbol{v}_1 + \boldsymbol{c}) &= \pi\boldsymbol{v}_1 + \pi\boldsymbol{c} \\ &= \boldsymbol{v}_2 + \pi\boldsymbol{c}. \end{aligned} \tag{11}$$

Since $\pi \in \mathrm{Aut}(C')$, $\pi\boldsymbol{c}$ is in C'. Thus $\pi[D_1] = D_2$. □

From Theorem 4, to partition the set of cosets into equivalence classes, we only need to check permuted codeword in a coset is included in another cosets. After partitioning into equivalence classes, the local distance subprofile for only one coset in each equivalence class needs to be computed. Thereby the computational complexity is reduced.

4. An Algorithm for Computing the Local Distance Profile

4.1 Outline of the algorithm

Based on partitioning of the set of cosets in the previ-

ous section, we propose an algorithm to compute the local distance profile as follows:

(1) Choose a subcode C', and a subgroup Π of permutations of $\mathrm{Aut}(C) \cap \mathrm{Aut}(C')$.

(2) Partition C/C' into equivalence classes with permutations in Π.

(3) Compute the local distance subprofiles for the representative cosets in each equivalence classes.

(4) Sum up all the local distance subprofiles for the cosets, and obtain the local distance profile of C.

4.2 Partitioning the set of cosets into equivalence classes

Step (2) of the algorithm can be implemented based on Theorem 4 and Lemma 4. For the efficient partitioning, we can also use the following one-to-one corresponding between the set of the cosets and $\{0,1\}^{k-k'}$: We choose a parity check matrix H' of C' with

$$H' = \begin{pmatrix} H_0 \\ H \end{pmatrix}, \tag{12}$$

where H is a parity check matrix of C, and H_0 is an $n \times (k-k')$ matrix. Then, the one-to-one corresponding is

$$\boldsymbol{v} + C' \leftrightarrow v H_0^T,$$

where H_0^T means the transpose of H_0.

Using a table with size $2^{k-k'}$, we need to compute the syndromes of length $k-k'$ for all the cosets to partition them into equivalence classes. The computational complexity of partitioning into equivalence classes is $O(n(k-k')2^{k-k'}|\Pi|)$. Note that the actual complexity is much smaller than $O(n(k-k')2^{k-k'}|\Pi|)$, and would be $O(n(k-k')2^{k-k'})$

4.3 Computational Complexity

Let C be an (n,k) binary linear code and C' be an (n,k') binary linear subcode of C. The number of cosets in C/C' is $2^{k-k'}$. The number of codewords in each cosets is $2^{k'}$. When the algorithm presented in [1], the idea of which is based on Lemma 1, is used to check a codeword whether a zero neighbor or not, the computational complexity to check one codeword is $O(n^2k)$. When cosets are partitioned into e classes, the computational complexity of computing the local distance profile is $O(n^2k \cdot e2^{k'})$. The number of equivalence classes affects the complexity of the algorithm.

The complexity of partitioning into equivalence classes is $O(n(k-k')2^{k-k'}|\Pi|)$. Therefore, the computational complexity of the algorithm is $O(n^2k \cdot e2^{k'} + n(k-k')2^{k-k'}|\Pi|)$. If $k' > k/2$, then $2^{k'} > 2^{k-k'}$, the complexity of partitioning into equivalence classes is much smaller than of computing the local distance subprofiles for cosets. We should partition into equivalence classes with some subcodes, and compute the local distance profile with the subcode which has the smallest number of equivalence classes. If the group of permutations which is used to partition into equivalence classes is same in any case of subcodes, the subcode with the smaller dimension should be chosen as long as the complexity of partitioning into equivalence classes is much small. Thereby, the number of equivalence classes is the smaller.

5. Computing the Local Distance Profile of Extended Primitive BCH Codes

Extended primitive BCH codes are closed under the affine permutations. To choose Π as large as possible, we choose an extended primitive BCH subcode as C'.

We apply the proposed algorithm to $(128,k)$ extended primitive BCH codes. We obtained the local distance profile of the codes for $k = 43, 36, 29, 22, 15$, and 8. The obtained local distance profiles are shown in Table 1. It takes about sixteen hours (CPU time) to compute the local distance profile of the $(128,43)$ extended primitive BCH code with 600MHz Alpha 21264 processor. Note that for the $(128,43)$ code, its $(128,22)$ subcode is chosen as C'.

The MHM algorithm can also be used to compute the local distance profile of extended primitive BCH codes. The computational complexity for the extended BCH codes of length n is as much as that for BCH codes, which is about $1/n$ as much as that of the brute force method.

The number of the affine permutations in (n,k) extended primitive BCH codes is $n(n-1)$. When the proposed algorithm is applied to the codes, the complexity is at most about $1/n^2$ as much as that of the brute force method. In fact, the complexity, compared with the MHM algorithm, is about 1/21 to 1/56 in case of $n = 64$, and about 1/43 or 1/64 in case of $n = 128$, except for some cases.

6. Local Distance Profiles for a Code and Its Extended Code

Consider a linear block code C' of length $n-1$ and its extended code C. Let A_i and A'_i be the number of codewords of weight i in C and C', respectively. A trivial relation between them is

$$A'_{2i-1} + A'_{2i} = A_{2i}, \qquad \text{for } 0 \leqq i \leqq \lfloor n/2 \rfloor. \tag{13}$$

[Lemma 5] For a codeword $\boldsymbol{v}' \in C'$, let $\mathrm{ex}(\boldsymbol{v}')$ be the corresponding codeword in C, that is, $\mathrm{ex}(\boldsymbol{v}')$ is obtained from $\boldsymbol{v}'$ by adding the over-all parity bit. Then,
(1) if $\boldsymbol{v}'$ is a zero neighbor in C', so is $\mathrm{ex}(\boldsymbol{v}')$ in C; and
(2) if $\boldsymbol{v}'$ is not a zero neighbor in C' and its weight is odd, so is $\mathrm{ex}(\boldsymbol{v}')$ in C.

We should note the case where $\boldsymbol{v}'$ is not a zero neighbor in C' and its weight is even. In this case, $\mathrm{ex}(\boldsymbol{v}')$ is a zero neighbor

Table 1 Local distance profiles of some extended primitive BCH codes of length 128.

$(128, 8)$ extended BCH code

w	L_w
64	254

$(128, 29)$ extended BCH code

w	L_w
44	373,888
48	2,546,096
52	16,044,672
56	56,408,320
60	116,750,592
64	152,623,774
68	116,750,592
72	56,408,320
76	16,044,672
80	2,546,096
84	373,888

$(128, 36)$ extended BCH code

w	L_w
32	10,688
36	16,256
40	2,048,256
44	35,551,872
48	353,494,848
52	2,028,114,816
56	7,216,135,936
60	14,981,968,512
64	19,484,132,736
68	14,981,968,512
72	7,216,127,808
76	2,028,114,816
80	348,203,520
84	35,551,872
88	2,048,256

$(128, 43)$ extended BCH code

w	L_w
32	124,460
36	8,810,752
40	263,542,272
44	4,521,151,232
48	44,899,876,672
52	262,118,734,080
56	915,924,097,536
60	1,931,974,003,456
64	2,476,669,858,944
68	1,931,944,645,120
72	915,728,180,224
76	261,375,217,152
80	43,168,588,288
84	2,464,897,280

$(128,15)$extended BCH code

w	L_w
56	8,128
64	16,510
72	8,128

$(128,22)$extended BCH code

w	L_w
48	42,672
56	877,824
64	2,353,310
72	877,824
80	42,672

in C if and only if the weight of any codeword $\boldsymbol{u}' \in C'$ with $\boldsymbol{u}' \neq \boldsymbol{0}$ such that $\mathrm{Supp}(\boldsymbol{u}') \subsetneqq \mathrm{Supp}(\boldsymbol{v}')$ is odd.

Let $(L_0, L_1, \ldots, L_n)$ and $(L'_0, L'_1, \ldots, L'_{n-1})$ be the local distance profiles of C and C', respectively. From (13) and Lemmas and 5, we have that

$$\begin{cases} L'_{2i-1} + L'_{2i} = L_{2i}, & \text{for } 0 \leqq i < d', \\ L'_{2i-1} + L'_{2i} \leqq L_{2i}, & \text{for } d' \leqq i \leqq \lfloor n/2 \rfloor, \end{cases} \tag{14}$$

where d' is the minimum distance of C'. Suppose that the extended code C is doubly transitive. For example, the extended code of a binary primitive BCH code and Reed-Muller code are doubly transitive [5]. Then, a simple relation on the weight distributions of the code and its original code is known as Theorem 8.15 in [5]. We see that a similar but weaker relation holds for the local distance profile, due to (14).

[Theorem 5]

$$L'_{2i-1} = \frac{2i}{n} L_{2i}, \quad \text{for } 0 \leqq i \leqq \lfloor n/2 \rfloor, \tag{15}$$

$$L'_{2i} \begin{cases} = & \frac{n-2i}{n} L_{2i}, \quad \text{for } 0 \leqq i < d', \\ \leqq & \frac{n-2i}{n} L_{2i}, \quad \text{otherwise.} \end{cases} \tag{16}$$

7. Conclusion

In this report, we propose an algorithm for computing the local distance profile of binary linear codes which are closed under a group of permutations. The algorithm uses an invariance property under the automorphism group. This property is applied to the set of cosets of a subcode. The proposed algorithm can be applied to Reed-Muller codes, which are closed under the general affine group.

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

[3] M. Mohri, Y. Honda, and M. Morii,"A method for computing the local distance profile of binary cyclic codes," IEICE Trans. Fundamentals (Japanese Edition), vol.J86-A, no.1, pp.60–74, Jan. 2003.

[4] T. Fujiwara and T. Kasami, "The weight distribution of $(256, k)$ extended binary primitive BCH codes with $k \leqq 63$ and $k \geqq 207$," IEICE Technical Report, IT97-46, Sept. 1997.

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