

# A Method for Reducing Space Complexity of Reliability based Heuristic Search Maximum Likelihood Decoding Algorithms

Hideki YAGI \*      Toshiyasu MATSUSHIMA \*      Shigeichi HIRASAWA \*

**Abstract**— In this paper, reliability-based heuristic search methods for maximum likelihood decoding of block codes are considered. Based on the decoding algorithm by Battail and Fang (and its improved technique by Valembois and Fossorier), we deduce a method of reducing the space complexity of the heuristic search maximum likelihood decoding algorithm. The proposed method is applicable to the heuristic search method with a certain class of evaluation functions. Simulation results show the efficiency of the decoding algorithm adopting the proposed method.

**Keywords**— maximum likelihood decoding, binary block codes, heuristic search, most reliable basis, reliability

## 1 Introduction

Maximum likelihood decoding (MLD) of block codes minimizes the probability of decoding error when we assume that all codewords have the equal probability to be transmitted. Since the complexity of searching the maximum likelihood (ML) codeword among all codewords is significantly large, many researcher have devoted to develop efficient algorithms of MLD. One of the most efficient MLD algorithms is the reliability based decoding algorithm that uses the column permuted generator matrix in increasing order of reliability.

In this paper, we consider heuristic search MLD algorithms where candidate codewords are generated in increasing value of the evaluation function. G. Battail and J. Fang have proposed a priority-first search method for MLD where the most simple (and primitive) evaluation functions are employed [1] (we will call this method the BF decoding algorithm). Recently, A. Valembois and M. Fossorier have proposed a technique for reducing the space complexity in the BF decoding algorithm with the same class of evaluation functions [4]. Since this kind of evaluation functions employed by both decoding algorithms are the most simple ones, we need to generate more candidate codewords than in the decoding algorithm with more sophisticated ones [2, 3, 5]. We can modify the BM decoding algorithm to perform priority-first MLD even when we use sophisticated evaluation functions [6], however, we cannot adopt the improved technique by Valembois et al. in this case.

In this paper, we propose a method for reducing the space complexity of the priority-first MLD algorithm with effective evaluation functions, which are presented in [3, 5]. Consequently, we show, by computer simulations, that space complexity of the decoding algorithm employing the proposed method is significantly reduced.

## 2 Reliability based MLD Algorithm

Let  $\mathcal{C}$  be a binary linear  $(n, k, d_H)$  block code of the code length  $n$ , the number of information symbols  $k$  and the minimum distance  $d_H$ . We denote a generator matrix of  $\mathcal{C}$  by  $G$  and the weight profile of  $\mathcal{C}$  by  $W_H(\mathcal{C})$ . We assume any codewords  $\mathbf{c} = (c_1, c_2, \dots, c_n) \in \{0, 1\}^n$  of  $\mathcal{C}$  are transmitted over Additive White Gaussian Noise (AWGN) channel. A receiver demodulates a received sequence  $\mathbf{r} = (r_1, r_2, \dots, r_n)$

into a sequence  $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$ ,  $\theta_j = \ln \frac{P(r_j|c_j=0)}{P(r_j|c_j=1)}$ , where  $P(r_j|c_j)$  represents the likelihood of symbol  $c_j$ , and inputs it into a soft-decision decoder<sup>1</sup>. Furthermore, a hard-decision sequence  $\mathbf{z} = (z_1, z_2, \dots, z_n) \in \{0, 1\}^n$  is obtained by setting  $z_j = 0$  if  $\theta_j \geq 0$  and  $z_j = 1$  otherwise. The soft-decision decoder estimates the transmitted codeword from  $\boldsymbol{\theta}$  and  $\mathbf{z}$ , and output the estimated codeword at the end of decoding.

In reliability-based decoding algorithms, we first find *the most reliable and linearly independent* (MRI)  $k$  positions. Then, we permute columns of a generator matrix over MRI positions in increasing value of reliability. The rest of columns are also reordered in increasing value of reliability. We perform the standard row operation with respect to the reordered matrix to make the leftmost  $k$  columns the identity matrix. We denote the resultant matrix by  $\tilde{G}$ .

Let  $\tilde{\boldsymbol{\theta}} = (\tilde{\theta}_1, \tilde{\theta}_2, \dots, \tilde{\theta}_n)$  and  $\tilde{\mathbf{z}} = (\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_n)$  be permuted sequences of  $\boldsymbol{\theta}$  and  $\mathbf{z}$ , respectively, in the same ordering of columns of  $\tilde{G}$ . We denote the equivalent code to  $\mathcal{C}$  by  $\tilde{\mathcal{C}}$ , whose codewords are generated by  $\tilde{G}$ . Let  $\mathbf{u} = (u_1, u_2, \dots, u_k) \in \{0, 1\}^k$  be the leftmost  $k$  symbols of  $\tilde{\mathbf{z}}$ , i.e.,  $u_j = \tilde{z}_j, 1 \leq j \leq k$ . The decoder first encodes  $\mathbf{u}$  by  $\tilde{G}$  to obtain the initial codeword  $\tilde{\mathbf{c}}_0 (= \mathbf{u}\tilde{G})$ . Afterwards,  $k$  dimensional vectors, called test error patterns, are iteratively generated and encoded by  $\tilde{G}$ . For a location set  $J \subseteq [1, k]$ , the test error pattern  $\mathbf{t}_J = (t_{J,1}, t_{J,2}, \dots, t_{J,k}) \in \{0, 1\}^k$  is determined by setting  $t_{J,j} = 1$  if  $j \in J$  and  $t_{J,j} = 0$  otherwise ( $J$  is called the support of  $\mathbf{t}_J$ ). Then,  $\tilde{\mathbf{c}}_J = \tilde{\mathbf{c}}_0 \oplus \mathbf{t}_J \tilde{G}$  is a candidate codeword and this procedure is repeated until a sufficient condition for the ML codeword is satisfied<sup>2</sup>.

For a binary vector  $\mathbf{v} = (v_1, v_2, \dots, v_n) \in \{0, 1\}^n$ , we define the correlation discrepancy [4] of  $\mathbf{v}$  as

$$L(\mathbf{v}) = \sum_{j|v_j \neq \tilde{z}_j} |\tilde{\theta}_j|. \quad (1)$$

Then  $\tilde{\mathbf{c}}_{\text{best}}$  is the ML codeword if and only if  $L(\tilde{\mathbf{c}}_{\text{best}}) = \min_{\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}} L(\tilde{\mathbf{c}})$ .

## 3 Battail-Fang Decoding Algorithm

Battail et al. have presented a method for generating test error patterns in increasing value of the evaluation function defined as

$$\Delta(\mathbf{t}_J) = \sum_{j \in J} |\tilde{\theta}_j|. \quad (2)$$

Let  $F$  be arbitrary evaluation function. If  $F$  satisfies the following two conditions, the BF decoding algorithm performs priority-first search of test error patterns [4].

- (C1)  $F(\mathbf{t}_J) \leq F(\mathbf{t}_{J \cup \{j_m\}})$ , for  $j_m \notin J$ .
- (C2)  $F(\mathbf{t}_J) \leq F(\mathbf{t}_{J'}) \Rightarrow F(\mathbf{t}_{J \cup \{j_m\}}) \leq F(\mathbf{t}_{J' \cup \{j_m\}})$ ,  
for  $j_m \notin J$  and  $j_m \notin J'$ .

The function  $\Delta$  actually satisfies (C1) and (C2).

\* Dept. of Industrial and Management Systems Engineering, Waseda University, 3-4-1 Ohkubo Shinjuku-ku, Tokyo, 169-8555 Japan, E-mail: yagi@hirasa.mgmt.waseda.ac.jp

<sup>1</sup> Since the probability of decision error of  $z_j$  becomes smaller as the value of  $|\theta_j|$  is larger,  $|\theta_j|$  is called reliability.

<sup>2</sup>  $\oplus$  represents Exclusive OR operation.

Consider  $k$  lists of test error patterns  $M_1, M_2, \dots, M_k$ . The test error pattern  $\mathbf{t}_J$  with a support  $J$  such that  $J \subseteq \{1, 2, \dots, j_m\}$  and  $j_m \in J$  is supposed to be in  $M_{j_m}$ <sup>3</sup>. Then for any test error pattern  $\mathbf{t}_J$  such that  $J \neq \emptyset$ , the list for storing it is uniquely determined. In a list  $M_j, 1 \leq \forall j \leq k$ , test error patterns are ordered in increasing value of an evaluation function  $F$ .

By the condition (C1), the test error pattern with the minimum value of  $F$  in  $M_j, 1 \leq \forall j \leq k$ , is  $\mathbf{t}_{\{j\}}$  with the Hamming weight one. For  $j = 1, 2, \dots, k$ , we set  $M_j = \{\mathbf{t}_{\{j\}}\}$  after the initial codeword  $\tilde{\mathbf{c}}_\emptyset$  is obtained. Then, the decoder searches the test error pattern with the minimum value of  $F$  (we will call this pattern the *best pattern*) among the set of ones which have not been found. In the decoding algorithm described below, for any candidate codeword  $\tilde{\mathbf{c}}_J (= \tilde{\mathbf{c}}_\emptyset \oplus \mathbf{t}_J \tilde{G})$  given by  $\mathbf{t}_J$ , we assume

$$F(\mathbf{t}_J) \leq L(\tilde{\mathbf{c}}_J). \quad (3)$$

Furthermore, for a test error pattern  $\mathbf{t}_J \in M_{j_m}$ , we call  $\mathbf{t}_{J \cup \{j\}}, \forall j > j_m$ , the *extended pattern* of  $\mathbf{t}_J$ .

#### [The BF decoding algorithm]

- S1) Set  $\tilde{\mathbf{c}}_{\text{best}} := \tilde{\mathbf{c}}_\emptyset$  and  $\underline{L} := L(\tilde{\mathbf{c}}_\emptyset)$ .
- S2) Choose the best pattern  $\mathbf{t}_J \in M_{j_m}$  among the topmost test error patterns in non-empty lists  $M_j, 1 \leq j \leq k$ . If  $F(\mathbf{t}_J) \geq \underline{L}$ , then output  $\tilde{\mathbf{c}}_{\text{best}}$  and halt the algorithm.
- S3) Generate the next candidate codeword by  $\tilde{\mathbf{c}}_J := \tilde{\mathbf{c}}_\emptyset \oplus \mathbf{t}_J \tilde{G}$ . If  $L(\tilde{\mathbf{c}}_J) < \underline{L}$ , then set  $\underline{L} := L(\tilde{\mathbf{c}}_J)$  and  $\tilde{\mathbf{c}}_{\text{best}} := \tilde{\mathbf{c}}_J$ .
- S4) At the end of all lists  $M_j, \forall j > j_m$ , store the extended pattern  $\mathbf{t}_{J \cup \{j\}}$ . Delete  $\mathbf{t}_J$  from  $M_{j_m}$ .
- S5) If  $M_j = \emptyset$  for all  $j = 1, 2, \dots, k$ , then output  $\tilde{\mathbf{c}}_{\text{best}}$  and halt the algorithm. Otherwise, go to S2).  $\square$

Valembois et al. have proposed an improved method for choosing the best pattern at S2) where we cost at most  $\lceil \log_2 k \rceil$  comparison operations in [4]<sup>4</sup>. First, we prepare a binary tree with  $k$  leaves where each leaf is allocated to one list  $M_j$ . Let the evaluation value of a leaf be that of the topmost test error pattern in  $M_j$ . Each node represents one of its successor nodes with the smaller evaluation value. By adopting this relation from leaves to the root node recursively, the root node represents the list whose topmost test error pattern is best. We need  $O(k)$  comparison operations to construct this initial tree, however, at most  $\lceil \log_2 k \rceil$  comparison operations are only needed in step S2). In the following, the BF decoding algorithm indicates the algorithm with this improved technique.

By (3), the inequality at S2),  $F(\mathbf{t}_J) \geq \underline{L}$ , represents a sufficient condition that  $\tilde{\mathbf{c}}_{\text{best}}$  is the ML codeword. If a tighter sufficient condition for optimality is employed, we only need to generate less candidate codewords to perform MLD. The evaluation function in [3, 5] is more effective than  $\Delta$  in the sense that it can be a tighter sufficient condition.

For  $\forall \tilde{\mathbf{c}} \in \tilde{\mathcal{C}}$ , let  $\tilde{\mathbf{c}}^L$  and  $\tilde{\mathbf{c}}^R$  be the leftmost  $k$  symbols and the rightmost  $n - k$  symbols of  $\tilde{\mathbf{c}}$ , respectively, i.e.,  $\tilde{\mathbf{c}} = \tilde{\mathbf{c}}^L \circ \tilde{\mathbf{c}}^R$ <sup>5</sup>. For some  $\tilde{\mathbf{c}}_{\text{ref}} (= \tilde{\mathbf{c}}_{\text{ref}}^L \circ \tilde{\mathbf{c}}_{\text{ref}}^R) \in \tilde{\mathcal{C}}$ , we define

$$T(\mathbf{t}_J, \tilde{\mathbf{c}}_{\text{ref}}) = \left\{ (\mathbf{t}_J \oplus \mathbf{u}) \circ \mathbf{v} \mid \mathbf{v} \in \{0, 1\}^{n-k}, \right. \\ \left. \text{and } w_{\text{H}}(\mathbf{t}_J) + d_{\text{H}}(\tilde{\mathbf{c}}_{\text{ref}}^R, \mathbf{v}) \in W_{\text{H}}(\tilde{\mathcal{C}}) \right\}, \quad (4)$$

where  $w_{\text{H}}(\cdot), d_{\text{H}}(\cdot, \cdot)$  are the Hamming weight and the Hamming distance. Then the evaluation function in [3, 5] is de-

<sup>3</sup> i.e.,  $j_m = \max J$  if  $\mathbf{t}_J \in M_{j_m}$ .

<sup>4</sup> In [4], a method for reducing the space complexity of the BF decoding algorithm with evaluation functions ( $\Delta$ ) satisfying condition (C1) and (C2) has been proposed, however we do not describe it in details here.

fined as,

$$f(\mathbf{t}_J, \tilde{\mathbf{c}}_{\text{ref}}) = \min_{\mathbf{v} \in T(\mathbf{t}_J, \tilde{\mathbf{c}}_{\text{ref}})} \left\{ L(\mathbf{v}) \right\}. \quad (5)$$

Since  $L(\mathbf{v}) = \Delta(\mathbf{t}_J) + \sum_{j=k+1}^n (\tilde{z}_j \oplus v_j) |\tilde{\theta}_j| \geq \Delta(\mathbf{t}_J)$  for  $\forall \mathbf{v} \in T(\mathbf{t}_J, \tilde{\mathbf{c}}_{\text{ref}})$ ,  $f$  can give a tighter sufficient condition for optimality than  $\Delta$ . Remark that  $f$  satisfies (C1) but does not necessarily satisfy (C2) [6]. Therefore, when we store an extended pattern into a list at S4), we need to insert it at the position such that the list remains in increasing value of the evaluation function. In this case, we modify S4) such as

- S'4) For all lists  $M_j$  such that  $j > j_m$ , insert the extended patterns  $\mathbf{t}_{J \cup \{j\}}$  at the position such that the list remains increasing order. Delete  $\mathbf{t}_J$  from  $M_{j_m}$ .

By this modification, the priority-first search of the BF decoding algorithm is maintained [4].

We here describe the complexity of the BF decoding algorithm. In a decoding procedure of a received sequence  $\mathbf{r}$ , the space complexity is  $O(k \times M(\mathbf{r}))$  where  $M(\mathbf{r})$  represents the maximum number of test error patterns stored in lists. As for the time complexity, the number of generating test error patterns is dominant as well as the number of encoding them.

## 4 Proposed Decoding Algorithm

In this section, we propose a method for reducing the space complexity of the BF decoding algorithm with evaluation functions that do not satisfy the condition (C2).

We here define a condition of an evaluation function  $F$ .

**Definition (C3)** For  $\forall J \subseteq [1, k]$ , if  $j_1, j_2 \notin J$  and  $1 \leq j_2 < j_1 \leq k$ , then a function  $F$  satisfies

$$F(\mathbf{t}_{J \cup \{j_2\}}) \geq F(\mathbf{t}_{J \cup \{j_1\}}). \quad (6)$$

$\square$

For the function  $f$ , we show the following lemma.

**Lemma 1** The evaluation function  $f$  satisfies the condition (C1) and (C3).

**(Proof)** We first show that the function  $\Delta$  satisfies (C3). By (2), for  $\mathbf{t}_J$  such that  $j_1 \notin J$ ,  $\Delta(\mathbf{t}_{J \cup \{j_1\}}) = \sum_{j \in J \cup \{j_1\}} |\tilde{\theta}_j| = \Delta(\mathbf{t}_J) + |\tilde{\theta}_{j_1}|$ . Since  $|\tilde{\theta}_j| \geq |\tilde{\theta}_{j+1}|$  for  $1 \leq j \leq k-1$ , if  $1 \leq j_2 < j_1$  and  $j_2 \notin J$ , then

$$\Delta(\mathbf{t}_{J \cup \{j_2\}}) = \sum_{j \in J \cup \{j_2\}} |\tilde{\theta}_j| + |\tilde{\theta}_{j_1}| - |\tilde{\theta}_{j_1}|$$

$$= \Delta(\mathbf{t}_{J \cup \{j_1\}}) + |\tilde{\theta}_{j_2}| - |\tilde{\theta}_{j_1}| \geq \Delta(\mathbf{t}_{J \cup \{j_1\}}).$$

Therefore, the function  $\Delta$  satisfies (C3).

For  $\mathbf{t}_J$  and  $\tilde{\mathbf{c}}_{\text{ref}} \in \tilde{\mathcal{C}}$ , let  $\mathbf{v}^* = (v_1^*, v_2^*, \dots, v_n^*)$  be such that  $f(\mathbf{t}_J, \tilde{\mathbf{c}}_{\text{ref}}) = L(\mathbf{v}^*)$ . Then by (1) and (5),

$$f(\mathbf{t}_J, \tilde{\mathbf{c}}_{\text{ref}}) = L(\mathbf{v}^*) = \Delta(\mathbf{t}_J) + \sum_{j=k+1}^n (\tilde{z}_j \oplus v_j^*) |\tilde{\theta}_j|. \quad (7)$$

The summation of the right hand side of (7) depends only on Hamming weight of  $\mathbf{t}_J$ . Therefore, if  $j_1, j_2 \notin J$  and  $1 \leq j_2 < j_1$ , then

$$f(\mathbf{t}_{J \cup \{j_2\}}) - f(\mathbf{t}_{J \cup \{j_1\}}) = \Delta(\mathbf{t}_{J \cup \{j_2\}}) - \Delta(\mathbf{t}_{J \cup \{j_1\}}), \quad (8)$$

and this implies that the function  $f$  satisfies (C3).  $\square$

In the following, we consider evaluation functions that satisfy both (C1) and (C3).

<sup>5</sup>  $\circ$  represents concatenation of vectors.

The strategy of the proposed method is like *lazy evaluation* where any test error patterns are not generated as long as possible. This approach is similar to an improved method in [4]. We first consider  $k$  lists  $M_j$  as in the BF decoding algorithm. By the condition (C1), the best pattern among all test error patterns in a list  $M_j, 1 \leq \forall j \leq k$ , is  $\mathbf{t}_{\{j\}}$ . Furthermore, by the condition (C3), the best pattern among  $k$  test error patterns  $\mathbf{t}_{\{j\}}, 1 \leq \forall j \leq k$ , is  $\mathbf{t}_{\{k\}}$ . Therefore, we construct the initial lists as

$$M_j = \begin{cases} \{\mathbf{t}_{\{j\}}\}, & \text{if } j = k; \\ \emptyset, & \text{otherwise,} \end{cases} \quad (9)$$

Similar to an improved method of [4], if the proposed method uses a binary tree whose leaves correspond to  $k$  lists, no comparison operations are needed to construct the initial tree.

At S2) of the BF decoding algorithm, if  $\mathbf{t}_J \in M_{j_m}$  is chosen as the best pattern,  $k - j_m$  extended pattern of  $\mathbf{t}_J$  will be stored at S4). However, it is enough to store only its extended pattern  $\mathbf{t}_{J \cup \{k\}}$  in the list  $M_k$ , since the condition (C3) guarantees  $F(\mathbf{t}_{J \cup \{k\}}) \geq F(\mathbf{t}_{J \cup \{j\}})$  for  $j < k$ .

Following this modification, we need to determine when other extended patterns  $\mathbf{t}_{J \cup \{j\}}, j < k$ , are inserted into lists. Assume that a test error pattern  $\mathbf{t}_{J \cup \{j_m\}}, j_m \notin J$ , is stored in the list  $M_{j_m}$  during a decoding procedure. Then extended patterns  $\mathbf{t}_{J \cup \{j\}}, j < j_m$ , cannot be the best pattern at S2), since the condition (C3) guarantees  $F(\mathbf{t}_{J \cup \{j\}}) \geq F(\mathbf{t}_{J \cup \{j_m\}})$ . Therefore these extended patterns need to be stored only after  $\mathbf{t}_{J \cup \{j_m\}}$  is chosen as the best pattern at S2).

Assume that  $\mathbf{t}_{J \cup \{j_m\}}$  is chosen as the best pattern at S2). By the condition (C3), if  $j_m - 1 > \max J$ , the extended pattern  $\mathbf{t}_{J \cup \{j_m - 1\}}$  has the smallest value of  $F$  next to  $\mathbf{t}_{J \cup \{j_m\}}$  among extended patterns, i.e.,

$$F(\mathbf{t}_{J \cup \{j_m - 1\}}) = \min_{\mathbf{t}_{J \cup \{j\}}} \left\{ F(\mathbf{t}_{J \cup \{j\}}) \mid j \notin J, j < j_m \right\}. \quad (10)$$

Therefore, after choosing  $\mathbf{t}_{J \cup \{j_m\}}$  as the best pattern at S2),  $\mathbf{t}_{J \cup \{j_m - 1\}}$  is inserted into the list  $M_{j_m - 1}$ . This modification reduces the space complexity significantly. Note that the next candidate pattern  $\mathbf{t}_{J \cup \{j_m - 1\}}$  is easily obtained from the best pattern  $\mathbf{t}_{J \cup \{j_m\}}$ .

We describe a decoding algorithm employing the above method. Before the following algorithm is performed, the decoder constructs the initial binary tree.

#### [The proposed decoding algorithm]

- P1) Set  $\tilde{\mathbf{c}}_{\text{best}} := \tilde{\mathbf{c}}_{\emptyset}$  and  $\underline{L} := L(\tilde{\mathbf{c}}_{\emptyset})$ .
- P2) Choose the best pattern  $\mathbf{t}_J \in M_{j_m}$  from non-empty lists. If  $F(\mathbf{t}_J) \geq \underline{L}$ , then output  $\tilde{\mathbf{c}}_{\text{best}}$  and halt the algorithm.
- P3) Generate the next candidate codeword by  $\tilde{\mathbf{c}}_J := \tilde{\mathbf{c}}_{\emptyset} \oplus \mathbf{t}_J \tilde{G}$ . If  $L(\tilde{\mathbf{c}}_J) < \underline{L}$ , then set  $\underline{L} := L(\tilde{\mathbf{c}}_J)$  and  $\tilde{\mathbf{c}}_{\text{best}} := \tilde{\mathbf{c}}_J$ .
- P4) a) If  $j_m - 1 \notin J$ , then insert  $\mathbf{t}_{J' \cup \{j_m - 1\}}$  into the list  $M_{j_m - 1}$  where  $J' = J \setminus \{j_m\}$ .  
b) If  $j_m \neq k$ , then insert  $\mathbf{t}_{J \cup \{k\}}$  into the list  $M_k$ . Delete  $\mathbf{t}_J$  from  $M_{j_m}$ .
- P5) If  $M_j = \emptyset$  for  $\forall j = 1, 2, \dots, k$ , then output  $\tilde{\mathbf{c}}_{\text{best}}$  and halt the algorithm. Otherwise, go to S2).  $\square$

The step P4) corresponds to the above modification.

We show the validity of the proposed decoding algorithm.

**Theorem 1** Assume that an evaluation function  $F$  satisfies both (C1) and (C3). During a decoding procedure, if  $\mathbf{t}_J$  is the best among the set of all test error patterns which has not been chosen as the best pattern, then such  $\mathbf{t}_J$  has been already generated and stored in the list  $M_{j_m}$  such that  $j_m = \max J$ .

(**Proof**) We first consider the following two cases.

- (1) In the case of  $\mathbf{t}_J \in M_k$ .

By the condition (C1),  $\mathbf{t}_{J'}$  such that  $J' = J \setminus \{k\}$  satisfies  $F(\mathbf{t}_{J'}) \leq F(\mathbf{t}_J)$ . Therefore, when we assume  $F(\mathbf{t}_{J'})$  has been chosen as the best pattern at P2), then  $\mathbf{t}_J$  has been into the list  $M_k$  at P4-b).

- (2) In the case of  $\mathbf{t}_J \in M_{j_m}, j_m \neq k$ .

From the condition (C3),  $\mathbf{t}_{J'}$  such that  $J' = J \setminus \{j_m\} \cup \{j_m + 1\}$  satisfies  $F(\mathbf{t}_{J'}) \leq F(\mathbf{t}_J)$ . When we assume  $F(\mathbf{t}_{J'})$  has been chosen as the best pattern at P2), then  $\mathbf{t}_J$  has been stored in a list  $M_{j_m}$  at P4-a).

Since the first test error pattern  $\mathbf{t}_{\{k\}}$  has been generated when initial lists has been constructed by (9), the assumptions of (1) and (2) are satisfied by mathematical induction.  $\square$

When an extended pattern is inserted in a list at P4), sorting is needed to keep the list in increasing value of  $F$  and this complexity may be large<sup>6</sup>. In the following, if a evaluation function  $F$  satisfies a certain condition, we show that the time complexity is reduced.

**Definition (C4)** For  $J, J' \subseteq [1, k]$  such that  $j_1, j_2 \notin J \cup J'$  and  $1 \leq j_2 < j_1 \leq k$ , a function  $F$  satisfies,

$$\begin{aligned} F(\mathbf{t}_{J \cup \{j_1\}}) &\leq F(\mathbf{t}_{J' \cup \{j_1\}}), \\ &\Rightarrow F(\mathbf{t}_{J \cup \{j_2\}}) \leq F(\mathbf{t}_{J' \cup \{j_2\}}). \end{aligned} \quad (11) \quad \square$$

Assume that a function  $F$  satisfies (C4). When a test error pattern is inserted into  $M_k$  at P4-b), we need sorting to keep a list in increasing value of  $F$ . However, during a decoding procedure, the best pattern  $\mathbf{t}_{J \cup \{j_m\}}$  such that  $j_m \notin J$  is chosen from  $M_{j_m}$ , it is enough to store  $\mathbf{t}_{J \cup \{j_m - 1\}}$  at the end of  $M_{j_m - 1}$  from the condition (C4). In case that the proposed decoding algorithm employs a function that satisfies (C4), P4) can be modified as follows:

- P'4) a) If  $j_m - 1 \notin J$ , then store  $\mathbf{t}_{J' \cup \{j_m - 1\}}$  at the end of  $M_{j_m - 1}$  where  $J' = J \setminus \{j_m\}$ .
- b) If  $j_m \neq k$ , then insert  $\mathbf{t}_{J \cup \{k\}}$  into the list  $M_k$ . Delete  $\mathbf{t}_J$  from  $M_{j_m}$ .

For the function  $f$ , we show the following lemma.

**Lemma 2** The evaluation function  $f$  satisfies (C4).

(**Proof**) By (8), the following equation holds.

$$f(\mathbf{t}_{J \cup \{j_2\}}) - f(\mathbf{t}_{J \cup \{j_1\}}) = f(\mathbf{t}_{J' \cup \{j_2\}}) - f(\mathbf{t}_{J' \cup \{j_1\}}). \quad (12)$$

By transposing (12),

$$f(\mathbf{t}_{J \cup \{j_1\}}) - f(\mathbf{t}_{J' \cup \{j_1\}}) = f(\mathbf{t}_{J \cup \{j_2\}}) - f(\mathbf{t}_{J' \cup \{j_2\}}). \quad (13)$$

Then  $f(\mathbf{t}_{J \cup \{j_2\}}) \leq f(\mathbf{t}_{J' \cup \{j_2\}})$  if and only if  $f(\mathbf{t}_{J \cup \{j_1\}}) \leq f(\mathbf{t}_{J' \cup \{j_1\}})$ .  $\square$

If the function  $f$  is employed by the proposed decoding algorithm, P'4) instead of P4) can be used. This saves the time complexity for sorting.

In terms of the time and space complexity of the proposed decoding algorithm, we show the following theorems.

**Theorem 2** The proposed decoding algorithm achieves MLD. Then, the maximum list size in the proposed decoding algorithm is less than that of the BF decoding algorithm, if both decoding algorithms employ the same evaluation function satisfying (1) and (3).  $\square$

**Theorem 3** The number of generating test error patterns in the proposed decoding algorithm is no more than that in the BF decoding algorithm, if both decoding algorithms employ the same evaluation function satisfying (1) and (3).  $\square$

<sup>6</sup> Since the total list size is reduced compared to that of the BF decoding algorithm, the complexity for sorting is also reduced.

## 5 Simulation Results

In this section, we evaluate the effectiveness of the proposed decoding algorithm by computer simulations.

### 5.1 Conditions for Simulation

For binary (63,30,13) BCH code and binary (104,52,20) quadratic residue (QR) code, we perform MLD by the BF decoding algorithm (we denote with “BF” in tables) and the proposed decoding algorithm (we denote with “Proposed” in tables). At each signal to noise ratio (SNR)  $E_b/S_0$  [dB], both decoding algorithms are carried out 10000 times. In tables, we use the following notations:

$N(\mathbf{r})$  : the number of generating test error patterns in decoding of  $\mathbf{r}$

$M(\mathbf{r})$  : the maximum list size in decoding of  $\mathbf{r}$

ave : the average value among 10000 decoding

max : the maximum value among 10000 decoding

We use the function  $f$  as the evaluation function in both decoding algorithm. Since the function  $f$  does not satisfy (C2), S'4) instead of S4) is used in the BF decoding algorithm. We assume that the weight profiles  $W_H(\mathcal{C})$  of these two codes are unknown and we use their supersets  $W'_H(\mathcal{C}) = \{0, d_H, d_H+1, \dots, n\}$ . Furthermore, we set the reference codeword as  $\tilde{\mathbf{c}}_{\text{ref}} = \tilde{\mathbf{c}}_{\text{best}}$  for the calculation of (5). Each time a new estimated codeword  $\tilde{\mathbf{c}}_{\text{best}}$  is obtained, the reference codeword is updated<sup>7</sup>.

### 5.2 Results and Discussion

By Table 1, the maximum list size  $\max M(\mathbf{r})$  in the proposed decoding algorithm is less than 1/3 of that in the BF decoding algorithm in each SNR. Furthermore, the average value of the maximum list size  $\text{ave } M(\mathbf{r})$  in the proposed decoding algorithm is less than 1/4 of that in the BF decoding algorithm. These results show that the effectiveness of the proposed decoding algorithm. By Table 2, the values  $\max M(\mathbf{r})$  and  $\text{ave } M(\mathbf{r})$  in the proposed decoding algorithm are less than 1/4 and 1/5 of ones in the BF decoding algorithm, respectively. These results indicate the proposed method also works well for a longer code.

The number of generating test error patterns  $N(\mathbf{r})$  is one of indices to evaluate time complexity in heuristic search MLD algorithms [2, 4]. By Table 1 and 2,  $N(\mathbf{r})$  in the proposed decoding algorithm are less than 2/5 of  $N(\mathbf{r})$  in the BF decoding algorithm even at low SNRs. These results indicate the proposed method reduces the time complexity of the BF decoding algorithm as well as the space complexity.

## 6 Conclusion and Future Works

In this paper, we propose a method for reducing the space complexity of the Battail-Fang decoding algorithm that is a priority-first heuristic search MLD. The proposed method is applicable to search methods with more effective evaluation functions. The decoding algorithm employing the proposed method are guaranteed to perform MLD since the set of generated candidate codewords there is identical to that in the BF decoding algorithm. The proposed decoding algorithm reduces not only the space complexity but the time one in the BF decoding algorithm. The proposed decoding algorithm can be straightforwardly modified to sub-optimal soft-decision decoding by limiting a set of candidate

<sup>7</sup> For a justifiable comparison, we assign the same value of the evaluation function to the same test error pattern in both decoding algorithm even when  $\tilde{\mathbf{c}}_{\text{ref}}$  is updated. i.e., the number of generating candidate codewords is the same in both decoding algorithm.

Table 1: The results of decoding for (63,30,13) BCH code

$E_b/N_0$ [dB]			BF	Proposed
5.0	ave	$N(\mathbf{r})$	33.7	2.42
		$M(\mathbf{r})$	1.20	0.145
	max	$M(\mathbf{r})$	2073	411
4.0	ave	$N(\mathbf{r})$	94.6	20.7
		$M(\mathbf{r})$	11.6	1.86
	max	$M(\mathbf{r})$	9586	2406
3.0	ave	$N(\mathbf{r})$	593	205
		$M(\mathbf{r})$	99.9	20.7
	max	$M(\mathbf{r})$	15191	4435
2.0	ave	$N(\mathbf{r})$	2960	1190
		$M(\mathbf{r})$	553	130
	max	$M(\mathbf{r})$	70519	20552

Table 2: The results of decoding for (104, 52, 20) QR code

$E_b/N_0$ [dB]			BF	Proposed
6.0	ave	$N(\mathbf{r})$	1.53	0.173
		$M(\mathbf{r})$	0.248	0.0228
	max	$M(\mathbf{r})$	187	18
5.0	ave	$N(\mathbf{r})$	33.3	4.91
		$M(\mathbf{r})$	3.66	0.358
	max	$M(\mathbf{r})$	5215	789
4.0	ave	$N(\mathbf{r})$	1210	349
		$M(\mathbf{r})$	170	29.5
	max	$M(\mathbf{r})$	463354	98746
3.0	ave	$N(\mathbf{r})$	41000	14700
		$M(\mathbf{r})$	6860	1410
	max	$M(\mathbf{r})$	11452892	2673788

codewords, and the same effectiveness can be anticipated as presented in this paper.

As future works, we need to develop a method for heuristic search MLD algorithm with powerful evaluation functions such as in [2]. The analytical guarantees of the reduction ratio of the complexity in the proposed method to that in the BF decoding algorithm are also needed.

## Acknowledgement

H. Yagi wishes to express his gratitude to Dr. M. Kobayashi at Shonan Institute of Technology for his valuable comments.

## References

- [1] G. Battail and J. Fang, “Décodage pondéré optimal descodes linéaires en blocs,” *Annales des télé-communications*, vol.41, nos.11–12, pp.580–604, Nov.–Dec. 1986.
- [2] Y.S. Han, C.R.P. Hartmann, and C.C. Chan, “Efficient priority-first search maximum likelihood soft decision decoding of linear block codes,” *IEEE Trans. Inform. Theory*, vol.39, no.5, pp.1514–1523, Sept. 1993.
- [3] D. Gazelle and J. Snyders, “Reliability-based code-search algorithm for maximum-likelihood decoding of block codes,” *IEEE Trans. Inform. Theory*, vol.43, no.1, pp.239–249, Jan. 1997.
- [4] A. Valembois and M. Fossorier, “An improved method to compute lists of binary vectors that optimize a given weight function with application to soft decision decoding,” *IEEE Commun. Lett.*, vol.5, no.3, pp.456–458, Nov. 2001.
- [5] M.P.C. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” *IEEE Trans. Inform. Theory*, vol.41, no.5, pp.1379–1396, Sept. 1995.
- [6] A. Valembois and M. Fossorier, “A comparison between “most-reliable-basis reprocessing” strategies”, *IEICE Trans. fundamentals*, vol.E85-A, no.7, pp.1727–1741, July 2002.