# Computational Complexity for Computing Sufficient Conditions on the Optimality of a Decoded Codeword

Tadao Kasami† and Takuya Koumoto‡

†Faculty of Information Science, Department of Computer Science,
Hiroshima City University

‡Graduate School of Information Science,
Nara Institute of Science and Technology

## Abstract

Sufficient conditions on the optimality of a candidate codeword, which is generated in an iterative soft-decision decoding algorithm for block codes, have been derived based on (1) partial knowledge of the distance profile of the code and (2) a number, denoted, $h$, of previously generated candidate codewords. This report presents upperbounds on the computational complexities of the sufficient conditions with $h = 2$ and 3.

# 1 Sufficient Conditions on the Optimality of a Decoded Codeword

Suppose a binary block code $C$ of length $N$ with distance (or weight) profile $W_C \triangleq \{0, w_1 = d_{\min}, w_2, \ldots\}$ is used for error control over the AWGN channel using BPSK signaling, where $w_1 = d_{\min}$ is the minimum Hamming distance of the code. A codeword $\boldsymbol{c}$ is mapped into a bipolar sequence $\boldsymbol{x}$. Suppose $\boldsymbol{x}$ is transmitted and $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ is the received sequence at the output of a matched filter in the receiver. Let $\boldsymbol{z} = (z_1, z_2, \ldots, z_N)$ be the binary hard-decision sequence.

Any soft-decision decoding scheme is devised based on $\boldsymbol{r}$ or reliability information provided by $\boldsymbol{r}$. For the AWGN channel and BPSK transmission, the reliability of a received symbol $r_j$ is generally measured by its magnitude $|r_j|$ since this value is proportional to the log-likelihood ratio associated with symbol hard-decision.

Let $V^N$ denote the vector space of all binary $N$-tuples. For an $N$-tuple $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in V^N$, define the following :

$$D_1(\boldsymbol{u}) \triangleq \{j : u_j \neq z_j, \text{ and } 1 \leq j \leq N\}, \tag{1.1}$$

$$D_0(\boldsymbol{u}) \triangleq \{1, 2, \ldots, N\} \backslash D_1(\boldsymbol{u}), \tag{1.2}$$

$$n(\boldsymbol{u}) \triangleq |D_1(\boldsymbol{u})|, \tag{1.3}$$

$$L(\boldsymbol{u}) \triangleq \sum_{j \in D_1(\boldsymbol{u})} |r_j|. \tag{1.4}$$

$L(\boldsymbol{u})$ is called the **correlation discrepancy** of $\boldsymbol{u}$ with respect to $\boldsymbol{z}$.

For a subset $U$ of $V^N$, let $\underline{L}[U]$ be defined as

$$\underline{L}[U] \triangleq \min_{\boldsymbol{u} \in U} L(\boldsymbol{u}). \tag{1.5}$$

If $U$ is empty, then $\underline{L}[U]$ is defined as $\infty$ (infinity). The maximum likelihood decoding (MLD) of a code $C$ can be stated in terms of the correlation discrepancy as follows: The decoder decodes the received sequence $\boldsymbol{r}$ into the codeword $\boldsymbol{c}_{\text{opt}}$ for which

$$L(\boldsymbol{c}_{\text{opt}}) = \underline{L}[C]. \tag{1.6}$$

We call $\boldsymbol{c}_{\text{opt}}$ the **optimal codeword**. If $\boldsymbol{z}$ is a codeword, then $\boldsymbol{z}$ is the optimal codeword. Let $d_H(\boldsymbol{u}, \boldsymbol{v})$ denote the Hamming distance between two $N$-tuples, $\boldsymbol{u}$ and $\boldsymbol{v}$. For $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, …,

$\boldsymbol{u}_h \in V^N$ and positive integers $d_1, d_2, \ldots, d_h$, let $V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)$ (or $V^N_{d_1,d_2,\ldots,d_h}$ for simplicity) be defined as the following set:

$$V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h) \triangleq \{\boldsymbol{u} \in V^N : d_H(\boldsymbol{u}, \boldsymbol{u}_i) \geq d_i \text{ for } 1 \leq i \leq h\}, \tag{1.7}$$

The sequence, $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h$, is called a sequence of **reference words**. For a codeword $\boldsymbol{u}$ in $C$ and a positive integer $d$, define the following subcode of $C$:

$$\bar{C}_d(\boldsymbol{u}) \triangleq \{\boldsymbol{v} \in C : d_H(\boldsymbol{v}, \boldsymbol{u}) < d\}. \tag{1.8}$$

At a stage of an iterative decoding algorithm, suppose that candidate codewords $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h$ have been generated. Let $\boldsymbol{u}_{\text{best}}$ denote the best of all candidate codewords that have been generated already. For $d_1, d_2, \ldots, d_h \in W_C - \{0\}$, if

$$L(\boldsymbol{u}_{\text{best}}) \leq \underline{L}[V^N_{d_1,d_2,\ldots,d_h}(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h)], \tag{1.9}$$

then $\boldsymbol{c}_{\text{opt}}$ is either $\boldsymbol{u}_{\text{best}}$ or in $\cup_{i=1}^h \bar{C}_{d_i}(\boldsymbol{u}_i)$.

Formulas for $\underline{L}[V^N_{d_1,d_2}]$ and $\underline{L}[V^N_{d_1,d_2,d_3}]$, (where the sequence of reference words is omitted), will be shown as Theorems 1 and 2, respectively. For the proofs, refer to [1]. For simplicity of discussion, we assume that the bit positions $1, 2, \ldots, N$ are ordered according to the following increasing order of reliability of the received symbols,for $1 \leq i < j \leq N$,

$$|r_i| \leq |r_j|. \tag{1.10}$$

For a subset $X$ of $\{1, 2, \ldots, N\}$ and a positive integer $j \leq |X|$, let $X^{(j)}$ denote the set of $j$ smallest integers in $X$. For a non-positive integer $j$, $X^{(j)} \triangleq \phi$ and for $j > |X|$, $X^{(j)} \triangleq X$.

For a positive integer $h$, let $B^h$ be the set of all binary sequences of length $h$. For $1 \leq i \leq h$ and $\alpha \in B^h$, let $\alpha_i$ denote the $i$-th symbol of $\alpha$. For a sequence of reference words $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_h$ in $V^N$ and $\alpha \in B^h$, let $D_\alpha$ and $n_\alpha$ be defined as

$$D_\alpha \triangleq \bigcap_{i=1}^{h} D_{\alpha_i}(\boldsymbol{u}_i), \tag{1.11}$$

$$n_\alpha \triangleq |D_\alpha|, \tag{1.12}$$

where $D_{\alpha_i}(\boldsymbol{u}_i)$ is the index set given by either (1.1) or (1.2).

The following theorem provides a formula for $\underline{L}[V^N_{d_1,d_2}]$.

**Theorem 1:** Without loss of generality, assume that

$$\delta_1 \geq \delta_2, \tag{1.13}$$

where $\delta_i \triangleq d_i - n(\boldsymbol{u}_i)$ for $1 \le i \le 2$. If $\min\{n_{01}, \delta_1, \lfloor(\delta_1-\delta_2)/2\rfloor\} < \delta_1 - n_{00}$, then $\underline{L}[V^N_{d_1,d_2}] = \infty$. Otherwise,

$$\underline{L}[V^N_{d_1,d_2}] = \sum_{i \in (D_{00} \cup D_{01}^{(\lfloor(\delta_1-\delta_2)/2\rfloor)})^{(\delta_1)}} |r_i|. \tag{1.14}$$

The following theorem gives an expression for $\underline{L}[V^N_{d_1,d_2,d_3}]$.

**Theorem 2:** Without loss of generality, assume that

$$\delta_1 \ge \delta_2 \ge \delta_3. \tag{1.15}$$

where $\delta_i \triangleq d_i - n(\boldsymbol{u}_i)$ for $1 \le i \le 3$. Consider the parities (even or odd) of $\delta_i$ with $1 \le i \le 3$. If all the parities are the same, then define $\varepsilon_i \triangleq 0$ for $1 \le i \le 3$. Otherwise, there is an index $j$ such that the parity of $\delta_j$ is different from the parities of other two $\delta_i'$s. Define $\varepsilon_j \triangleq 1$ and $\varepsilon_i \triangleq 0$ for $i \ne j$. Define $\delta_{1i}$ with $2 \le i \le 3$, $\delta_i'$ and $\delta^{(i)}$ with $1 \le i \le 3$ as follows:

$$\delta_{1i} \triangleq \min\{\delta_1, \lfloor(\delta_1 - \delta_i)/2\rfloor\}, \tag{1.16}$$

$$\delta_i' = \delta_i + \varepsilon_i, \tag{1.17}$$

$$\delta^{(1)} \triangleq \max\{0, \delta_{12} - n_{010}, \delta_{13} - n_{001}\}, \tag{1.18}$$

$$\delta^{(2)} \triangleq \min\{n_{011}, \delta_{12}, \delta_{13}, n_{000} + \delta_{12} + \delta_{13} - \delta_1\}, \tag{1.19}$$

$$\delta^{(3)} \triangleq \max\{0, (\delta_2' + \delta_3')/2 - n_{000}\}, \tag{1.20}$$

$$\delta^{(4)} \triangleq \min\{n_{100}, n_{010} - (\delta_1' - \delta_2')/2, n_{001} - (\delta_1' - \delta_3')/2, (\delta_2' + \delta_3')/2\}. \tag{1.21}$$

Then, $\underline{L}[V^N_{d_1,d_2,d_3}]$ is given by

$$\underline{L}[V^N_{d_1,d_2,d_3}] = \min\{\underline{L}_1, \underline{L}_2\}, \tag{1.22}$$

where $\underline{L}_1$ and $\underline{L}_2$ are defined as follows:

If $\delta^{(1)} \le \delta^{(2)}$, then

$$\underline{L}_1 = \min_{\delta^{(1)} \le \delta \le \delta^{(2)}} \sum_{i \in (D_{000} \cup D_{001}^{(\delta_{13}-\delta)} \cup D_{010}^{(\delta_{12}-\delta)})^{(\delta_1-\delta)} \cup D_{011}^{(\delta)}} |r_i|. \tag{1.23}$$

Otherwise, $\underline{L}_1 \triangleq \infty$.

If $\delta^{(3)} \le \delta^{(4)}$, then

$$\underline{L}_2 = \min_{\delta^{(2)} \le \delta \le \delta^{(3)}} \sum_{i \in D_{000}^{((\delta_2'+\delta_3')/2-\delta)} \cup D_{010}^{((\delta_1'-\delta_2')/2+\delta)} \cup D_{001}^{((\delta_1'-\delta_3')/2+\delta)} \cup D_{100}^{(\delta)}} |r_i|. \tag{1.24}$$

Otherwise, $\underline{L}_2 \triangleq \infty$.

## 2 An algorithm for Computing $\underline{L}[V^N_{d_1,d_2}(\boldsymbol{u}_1, \boldsymbol{u}_2)]$

Assume that (1.10) "$|r_i| \le |r_j|$ for $1 \le i < j \le N$" holds and that $\delta_1 \ge \delta_2$ and $\delta_1 > 0$, and $\delta_{12} \triangleq \min\{\delta_1, \lfloor(\delta_1 - \delta_2)/2\rfloor\}$. It follows from Theorem 1 that $\underline{L}[V^N_{d_1,d_2}(\boldsymbol{u}_1, \boldsymbol{u}_2)] \ne \infty$, if and only if

$$n_{00} \ge \lceil(\delta_1 + \delta_2)\rceil/2, \tag{2.1}$$

$$n_{00} + n_{01} \ge \delta_1. \tag{2.2}$$

Define $X_2 \triangleq (D_{00} + D_{01}^{(\lfloor(\delta_1-\delta_2)/2\rfloor)})^{(\delta_1)}$ and for $a \in \{0,1\}$, define $\nu_a \triangleq |D_{0a} \cap X_2|$ and for a nonempty subset $X$ of $\{1, 2, \ldots, N\}$, let $i(X)$ denote the largest index in $X$. Define

$$\bar{d} \triangleq \min\{N, \max\{d_1, \lceil(d_H(\boldsymbol{u}_1, \boldsymbol{u}_2) + d_1 + d_2)/2\rceil\}\}. \tag{2.3}$$

**Lemma 1**: Suppose that (2.1) and (2.2) hold. Then $i(X_2) \le \bar{d}$.

**(Proof)**: If $\nu_1 < \lfloor(\delta_1 - \delta_2)/2\rfloor$, then $X_2 = D_0^{(\delta_1)}$. Hence, $i(X_2) \le n_{1*} + \delta_1 = d_1$. If $\nu_1 = \lfloor(\delta_1 - \delta_2)/2\rfloor$, then $\nu_0 = \lceil(\delta_1 + \delta_2)/2\rceil$ and therefore, $i(X_2) \le n_{1*} + n_{01} + \nu_0 = \lceil(n_{10} + n_{01} + d_1 + d_2)/2\rceil = \lceil(d_H(\boldsymbol{u}_1, \boldsymbol{u}_2) + d_1 + d_2)/2\rceil$. △△

The following $V_{h=2}$ is an algorithm for computing $\underline{L}[V^N_{d_1,d_2}(\boldsymbol{u}_1, \boldsymbol{u}_2)]$. In $V_{h=2}$, real number array $R$ and binary arrays $U_1$ and $U_2$ of length $\bar{d}$ are used. For $1 \le j \le \bar{d}$, $|r_j|$ and the $j$-th elements of $\boldsymbol{z} + \boldsymbol{u}_1$ and $\boldsymbol{z} + \boldsymbol{u}_2$ are stored in the $(\bar{d} - j + 1)$-th elements of $R$, $U_1$ and $U_2$, respectively.

**Procedure** $V_{h=2}$;

begin

  step1: (*Initialization*)

    $L \leftarrow 0$; $j \leftarrow \bar{d} + 1$; $m \leftarrow \delta_1$;

    if $\delta_{12} = 0$, then

      go to step 2;

    else

      $m_1 \leftarrow \delta_{12}$; go to step3;

  step2: (*$m_1 = 0$*)

    updatej;

```
        if U_1(j) = 0, then
            if U_2(j) = 0, then
                updateLm;
        go to step2;
    step3: (*m_1 > 0*)
        updatej;
        if U_1(j) = 0, then
            updateLm;
            if U_2(j) = 1, then
                m_1 ← m_1 − 1;
                if m_1 = 0, then
                    go to step2;
        go to step3;
end.

Procedure updateLm;
begin
    L ← L + R(j); m ← m − 1;
    if m = 0, then
        output L; stop;
    else
        return;
end.

Procedure updatej;
begin
    j ← j − 1;
    if j = 0, then
        output ∞; stop;
    else
        return;
end.
```

Clearly, $V_{h=2}$ always terminates. Let $L_t$, $j_t$, $m_t$ and $m_{1,t}$ denote the final values of $L, j, m$ and $m_1$ respectively. For a current value of $j$, define $Y(j) = \{\bar{d} - j' + 1 : j \le j' \le \bar{d}$ and $L \leftarrow L + R(j')$ has been performed in $V_{h=2}\}$. Then, we can see that

(1) $|D_{0*} \cap Y(j)| = \delta_1 - m$, $|D_{01} \cap Y(j)| = \delta_{12} - m_1$ and

$$Y(j) = (D_{00} \cup D_{01}^{(\delta_{12}-m_1)})^{(\delta_1 - m)}, \tag{2.4}$$

and for $1 \le j' < j \le \bar{d}$, $\{1, 2, \ldots, \bar{d} - j + 1\} \backslash Y(j)$ and $Y(j')$ are mutually disjoint.

(2) $V_{h=2}$ stops in step 2, if and only if $m_{1,t} = 0$, i.e., $|D_{01} \cap Y(j_t)| = \delta_{12}$.

(3) $V_{h=2}$ outputs $L_t = \sum_{i \in Y(j_t)} |r_i|$, if and only if $m_t = 0$, i.e., $|D_{0*} \cap Y(j_t)| = \delta_1$. From (1), $|D_{01} \cap Y(j_t)| \le \delta_{12}$. Hence, $n_{0*} \ge \delta_1$ and $n_{00} = n_{0*} - n_{01} \ge \lceil (\delta_1 + \delta_2)/2 \rceil$. From (2.1), (2.2) and (2.4), $Y(j_t) = (D_{00} \cup D_{01}^{(\delta_{12})})^{(\delta_1)} = (D_{00} \cup D_{01}^{(\lfloor (\delta_1 - \delta_2)/2 \rfloor)})^{(\delta_1)}$ and $L_t = \underline{L}[V^N_{d_1,d_2}(\boldsymbol{u}_1, \boldsymbol{u}_2)]$.

(4) $V_{h=2}$ outputs $\infty$ as $L$, if and only if $m_t > 0$ and $j_t = 0$. Then, $|D_{0*} \cap Y(j_t)| < \delta_1$ from (1). If (2.1) and (2.2) hold, then $j(X_2) > \bar{d}$, which contradicts Lemma 1. Hence $\underline{L}[V^N_{d_1,d_2}(\boldsymbol{u}_1, \boldsymbol{u}_2)] = \infty$.

In $V_{h=2}$, the number of access to array $R$ and real number addition is at most $\delta_1$, the number of access to array $U_1$, binary test jump on $U_1(j)$ and the index decrement and zero test jump of $j$ is at most $\bar{d}$, the number of access to array $U_2$ and binary test jump on $U_2(j)$ is at most $\bar{d} - d_1$ and the numbers of index decrements and zero test jumps of $m$ and $m_1$ are at most $\delta_1$ and $\delta_{12}$, respectively.

For the algorithm $V_{h=2}$, it is sufficient to assume that $|r_i| \le |r_j|$ for either $1 \le i < j \le \bar{d}$ or $1 \le i \le \bar{d} < j \le N$.

# 3 Computational Complexity for Computing $\underline{L}[V^N_{d_1,d_2,d_3}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3)]$

(1) Computing $\underline{L}_1$.

For convenience, assume that if $|r_i| = |r_j|$ for $i \in D_{000}$ and $j \notin D_{000}$, then

$$i < j, \tag{3.1}$$

and define

$$\rho_{10} \triangleq \delta_{12}, \text{ and } \rho_{01} \triangleq \delta_{13}. \tag{3.2}$$

$$X(\delta) \triangleq (D_{000} \cup D_{001}^{(\rho_{01}-\delta)} \cup D_{010}^{(\rho_{10}-\delta)})^{(\delta_1-\delta)} \cup D_{011}^{(\delta)}, \text{ for } \delta^{(1)} \leq \delta \leq \delta^{(2)},$$

$$L_1(\delta) \triangleq \sum_{i \in X(\delta)} |r_i|.$$

We first prove the following lemma:

**Lemma 2**: There exists an integer $\tilde{\delta_1}$ such that $\delta^{(1)} \leq \tilde{\delta_1} \leq \delta^{(2)}$, and $L_1(\delta)$ is nonincreasing for $\delta^{(1)} \leq \delta \leq \tilde{\delta_1}$, and is nondecreasing for $\tilde{\delta_1} \leq \delta \leq \delta^{(2)}$.

From this lemma, we have that

$$\underline{L}_1 = L_1(\tilde{\delta_1}). \tag{3.3}$$

**(Proof)** Define $\nu_\alpha(\delta) \triangleq |D_{0\alpha} \cap X(\delta)|$, for $\alpha \in B^2$. Then, $\nu_{11}(\delta) = \delta$, and

$$\nu_{00}(\delta) + \nu_{01}(\delta) + \nu_{10}(\delta) = \delta_1 - \delta. \tag{3.4}$$

Define

$$i_\alpha(\delta) \triangleq \text{the largest index of } D_{0\alpha} \cap X(\delta), \text{ for } \alpha \in B^2,$$

$$i(\delta) \triangleq \text{the largest index of } X(\delta) \backslash \{D_{011}\}.$$

Let $\beta(\delta)$ indicate that $i(\delta)$ is in $D_{0\beta(\delta)}$. It follows from (1.19) that for $\delta^{(1)} \leq \delta < \delta^{(2)}$,

$$\rho_\alpha > \delta, \text{ for } \alpha \in \{01, 10\}. \tag{3.5}$$

There are three cases (i), (ii) and (iii), and each case is partitioned into two subcases.

Case (i) $\nu_{01}(\delta) < \rho_{01} - \delta$ and $\nu_{10}(\delta) < \rho_{10} - \delta$:

$$X(\delta+1) = \{i_{11}(\delta+1)\} \cup X(\delta) \backslash \{i(\delta)\}. \tag{3.6}$$

Subcase (i.1) $i_{11}(\delta+1) < i(\delta)$: Then $L_1(\delta+1) \leq L_1(\delta)$.

Subcase (i.2) $i(\delta) < i_{11}(\delta+1)$: Then $L_1(\delta) \leq L_1(\delta+1)$, and

$$i(\delta+1) < i_{11}(\delta+1). \tag{3.7}$$

Case (ii) For $\alpha \in \{01, 10\}$, $\nu_\alpha(\delta) = \rho_\alpha - \delta$ and $\nu_{\bar{\alpha}}(\delta) < \rho_{\bar{\alpha}} - \delta$: Then $\nu_\alpha(\delta) > 0$(from (3.5)) and

$$X(\delta+1) = \{i_{11}(\delta+1)\} \cup X(\delta) \backslash \{i_\alpha(\delta)\}. \tag{3.8}$$

$$\nu_\alpha(\delta+1) = \rho_\alpha - \delta - 1. \tag{3.9}$$

Subcase (ii.1) If $i_{11}(\delta+1) < i_\alpha(\delta)$: Then $L_1(\delta+1) \leq L_1(\delta)$.

Subcase (ii.2) If $i_\alpha(\delta) < i_{11}(\delta+1)$: Then $L_1(\delta) \le L_1(\delta+1)$.

Case (iii) $\nu_{01}(\delta) = \rho_{01} - \delta$ and $\nu_{10}(\delta) = \rho_{10} - \delta$: Then, $\nu_{01}(\delta) > 0$ and $\nu_{10}(\delta) > 0$ (from (3.5)). It follows from (3.4) and (1.19) that

$$\nu_{00}(\delta) < n_{000}. \tag{3.10}$$

Therefore,

$$X(\delta+1) = \{i_{00}(\delta+1), i_{11}(\delta+1)\} \cup X(\delta) \backslash \{i_{01}(\delta), i_{10}(\delta)\}, \tag{3.11}$$

where $i_{00}(\delta+1)$ denotes the $(i_{00}(\delta)+1)$-th index of $D_{000}$. If $\beta(\delta) = 00$, then $i(\delta) < i_{00}(\delta+1)$. Otherwise, if $i(\delta) > i_{00}(\delta+1)$, then from (3.1) and the definition of $X(\delta)$, $i_{00}(\delta+1) \in X(\delta)$, a contradiction. Hence, we have that

$$i_\alpha(\delta) \le i(\delta) < i_{00}(\delta+1), \text{ for } \alpha \in \{01, 10\}. \tag{3.12}$$

Subcase (iii.1) The following inequality holds:

$$|r_{i_{00}(\delta+1)}| + |r_{i_{11}(\delta+1)}| < |r_{i_{01}(\delta)}| + |r_{i_{10}(\delta)}|. \tag{3.13}$$

Then, $L_1(\delta+1) < L_1(\delta)$.

Subcase (iii.2) The above inequality does not hold. Then $L_1(\delta+1) \ge L_1(\delta)$.

Suppose that $\delta^{(1)} \le \delta$, $\delta+1 < \delta^{(2)}$ and either subcase (i.2), (ii.2) or (iii.2) holds for $\delta$. We show that neither subcase (i.1), (ii.1) nor (iii.1) occurs afterward.

Subcase (i.2) Then, $i(\delta+1) < i_{11}(\delta+1) < i_{11}(\delta+2)$ from (3.8). Since $i_\alpha(\delta+1) \le i(\delta+1) < i_{11}(\delta+2)$ for $\alpha \in \{01, 10\}$, neither subcase (i.1) nor subcase (ii.2) holds for $\delta+1$. If case (iii) holds for $\delta+1$, then from (3.12)

$$i_\alpha(\delta+1) \le i(\delta+1) < i_{00}(\delta+2) \text{ for } \alpha \in \{01, 10\}.$$

Hence, subcase (iii.1) can not hold for $\delta+1$.

Subcase (ii.2) From (3.10), case (i) can not hold for $\delta+1$. For $\alpha = 01$ or $10$ such that $\nu_\alpha(\delta) = \rho_\alpha - \delta$, $i_\alpha(\delta+1) < i_\alpha(\delta) < i_{11}(\delta+1) < i_{11}(\delta+2)$ and therefore, subcase (ii.1) can not hold for $\delta+1$. If case (iii) holds for $\delta+1$, then $i_\alpha(\delta+1) < i_{00}(\delta+2)$ for $\alpha \in \{01, 10\}$(from (3.12)) and therefore, subcase (iii.1) can not hold for $\delta+1$.

Subcase (iii.2) Since $\delta+1 < \delta^{(2)}$, case (iii) holds for $\delta+1$. Since $i_\alpha(\delta+1) < i_\alpha(\delta)$ for $\alpha \in \{01, 10\}$ and $i_\alpha(\delta+1) < i_\alpha(\delta+2)$ for $\alpha \in \{00, 11\}$, (3.13) can not hold for $\delta+1$.

Consequently, by induction, once one of subcase (i.2), (ii.2) and (iii.2) holds at $\delta$, then neither subcase (i.1), (ii.1) nor (iii.1) can occur afterward. If there is such a $\delta$, let $\tilde{\delta_1}$ denote the smallest one for which subcase (i.2), (ii.2) or (iii.2) holds. Otherwise, $\tilde{\delta_1} \triangleq \delta^{(2)}$. Then, Lemma 2 holds. △△

$\tilde{\delta_1}$ can be found as follows:

(1.1) First, obtain $X(\delta^{(1)})$. This can be done only by non-real number operations, that is, array access, array index operations and binary operations whose total number is $O(N)$.

(1.2) Then, for $\delta^{(1)} \leq \delta < \delta^{(2)}$, we construct $X(\delta+1)$ from $X(\delta)$ step by step by a few non-real number operations until either subcase (i.2), (ii.2) or (iii) occurs or $\delta + 1$ becomes to $\delta^{(2)}$. If subcase (i.2) or (ii.2) occurs first at $\delta$, then $\tilde{\delta_1} = \delta$. For $\alpha \in B^2$, $i_\alpha(\tilde{\delta_1}) = i_\alpha(\delta)$.

Otherwise, let $\delta_{iii}$ denote the smallest $\delta$ for which case (iii) holds with no preceding subcase (i.2) nor (ii.2). For case (iii), we need two real number additions and one real number comparison to decide which of subcases (iii.1) and (iii.2) holds. $\tilde{\delta_1}$ can be found by a binary search. The number of addition equivalent operations for the binary search is at most

$$3\lceil \log_2(\delta^{(2)} - \delta_{iii} + 1)\rceil. \tag{3.14}$$

Then, $\nu_\alpha(\tilde{\delta_1})$ with $\alpha \in B^2$ are also found.

(1.3) Once $\tilde{\delta_1}$ is found, $X(\tilde{\delta_1}) = D_{000}^{(\nu_{00}(\tilde{\delta_1}))} \cup D_{001}^{(\nu_{01}(\tilde{\delta_1}))} \cup D_{010}^{(\nu_{10}(\tilde{\delta_1}))} \cup D_{011}^{(\tilde{\delta_1})}$ and, since $|X(\delta)| = \delta_1$, $L_1(X(\tilde{\delta_1}))$ can be found by $\delta_1$ real number additions and non-real-number operations of order $N$. Thus, the total number of addition equivalent operations for computing $\underline{L}_1$ is at most

$$\delta_1 + 3\lceil \log_2(\delta^{(2)} + 1)\rceil \leq \delta_1 + 3\lceil \log_2(\delta_1 + 1)\rceil \tag{3.15}$$

(2) Computing $\underline{L}_2$.

For $\alpha \in B^3$ and $1 \leq j \leq n_\alpha$, let $i_\alpha(j)$ be the $j$-th index of $D_\alpha$. For convenience, define

$$X'(\delta) \triangleq D_{000}^{(\rho'_{000}-\delta)} \cup D_{010}^{(\rho'_{010}+\delta)} \cup D_{001}^{(\rho'_{001}+\delta)} \cup D_{100}^{(\delta)},$$

for $\delta^{(3)} \leq \delta \leq \delta^{(4)}$, where $\rho'_{000} \triangleq (\delta'_2 + \delta'_3)/2$, $\rho'_{010} \triangleq (\delta'_1 - \delta'_2)/2$, and $\rho'_{001} \triangleq (\delta'_1 - \delta'_3)/2$, $L_2(\delta) = \sum_{i \in X'(\delta)} |r_i|$. Let $\tilde{\delta_2}$ be the smallest of $\delta$ with $\delta^{(3)} \leq \delta \leq \delta^{(4)}$ such that

$$\underline{L}_2 = L_2(\tilde{\delta_2}).$$

Note that $L_2(\delta) \leq L_2(\delta+1)$, for $\delta^{(3)} \leq \delta < \delta^{(4)}$, if and only if

$$|r_{i_{000}(\rho'_{000}-\delta)}| \leq |r_{i_{010}(\rho'_{010}+\delta+1)}| + |r_{i_{001}(\rho'_{001}+\delta+1)}| + |r_{i_{100}(\delta+1)}|. \tag{3.16}$$

It follows from (1.10) and the definition of $i_\alpha(j)$ that if (3.16) holds for $\delta$, then it holds for all $\delta'$ such that $\delta \le \delta' \le \delta^{(4)}$. Hence, if there is $\delta$ with $\delta^{(3)} \le \delta < \delta^{(4)}$ which satisfies (3.16), then $\tilde{\delta_2}$ is the smallest of such $j$, and otherwise $\tilde{\delta_2} \triangleq \delta^{(4)}$. Consequently, $\tilde{\delta_2}$ can be found by a binary search whose number of addition equivalent operations is $3\lceil \log_2(\delta^{(4)} - \delta^{(3)} + 1)\rceil$. Once $\tilde{\delta_2}$ is found, since $|X'(\tilde{\delta_2})| = \delta_1' + 2\tilde{\delta_2}$, the number of addition equivalent operations for computing $L_2(\tilde{\delta_2})$ is

$$\delta_1' + 2\tilde{\delta_2}.$$

Hence, the number is at most

$$\delta_1' + 2\delta^{(4)} + 3\lceil \log_2(\delta^{(4)} + 1)\rceil \le 3\delta_1' + 3\lceil \log_2(\delta_1' + 1)\rceil.$$

The number of non-real-number operations is of order $N$.

# References


[1] T. Kasami, T. Takata, T. Koumoto, T. Fujiwara, H. Yamamoto and S. Lin, "The Least Stringent Sufficient Condition on Optimality of Suboptimal Decoded Codewords," *Technical Report of IEICE*, IT94-82, The Inst. of Electronics, Information and Communication Engineers, Japan, Jan. 1995. A revised version is in preparation.