<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>On the reliability-order-based decoding algorithms for binary linear block codes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Tang, Yuansheng; Ling, San; Fu, Fang-Wei</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2006</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/9849">http://hdl.handle.net/10220/9849</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2006 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<a href="http://dx.doi.org/10.1109/TIT.2005.860451">http://dx.doi.org/10.1109/TIT.2005.860451</a>].</td>
</tr>
</tbody>
</table>
On the Reliability-Order-Based Decoding Algorithms for Binary Linear Block Codes

Yuansheng Tang, Member, IEEE, San Ling, and Fang-Wei Fu

Abstract—In this correspondence, we consider the decoding of binary block codes over the additive white Gaussian noise (AWGN) channel with binary phase-shift keying (BPSK) signaling. By a reliability-order-based decoding algorithm (ROBDA), we mean a soft-decision decoding algorithm which decodes to the best (most likely) codeword of the form that is the sum of the hard-decision tuple and an error pattern in a set determined only by the order of the reliabilities of the hard decisions. Examples of ROBDAs include many well-known decoding algorithms, such as the generalized-minimum-distance (GMD) decoding algorithm, Chase decoding algorithms, and the reliability-based decoding algorithms proposed by Fossorier and Lin. It is known that the squared error-correction-radius of ROBDAs can be computed from the minimal squared Euclidean distances (MSEDs) between the all-one sequence and the polyhedra corresponding to the error patterns. For the computation of such MSEDs, we give a new method which is more compact than the one proposed by Fossorier and Lin. These results are further used to show that any bounded-distance ROBDA is asymptotically optimal: The ratio between the probability of decoding error of a bounded-distance ROBDA and that of the maximum-likelihood (ML) decoding approaches 1 when the signal-to-noise ratio (SNR) approaches infinity, provided that the minimum Hamming distance of the code is greater than 2.

Index Terms—Asymptotic optimality, bounded-distance decoding, error performance, linear block codes, maximum-likelihood (ML) decoding, reliability-order-based decoding.

I. INTRODUCTION

For the decoding of binary linear block codes over the additive white Gaussian noise (AWGN) channel with binary phase-shift keying (BPSK) signaling, many soft-decision decoding algorithms, which provide various tradeoffs between the error performance and the decoding complexity, have been proposed in the literature. For some of them, the decoded codeword is selected as the best (most likely) codeword in a list of candidate codewords. In this correspondence, by a reliability-order-based decoding algorithm (ROBDA) we mean a soft-decision decoding algorithm whose candidate codewords are the codewords of the form that is the sum of the hard-decision tuple and an error pattern in a set determined only by the order of the reliabilities of the hard decisions. Examples of ROBDAs include many well-known decoding algorithms, such as the generalized-minimum-distance (GMD) decoding algorithm [1], Chase decoding algorithms [2], and the reliability-based decoding algorithms [3], [4]. A detailed list of references concerning ROBDAs can be found in [5].

In general, it is very difficult to exactly analyze the error performance of a soft-decision decoding algorithm for all signal-to-noise ratios (SNRs). In the literature, the method of simulation is usually used to analyze the error performance for moderate SNRs, whereas the error exponent, from which an exponentially tight estimation of the probability of decoding error can be obtained, is a commonly employed indicator of the error performance for high SNRs. For some soft-decision decoding algorithms, the error exponents can be computed simply from the squared error-correction radius (SECR) [6]. A soft-decision decoding algorithm is called a bounded-distance decoding if its error-correction radius is maximized, i.e., it achieves the guaranteed error-correction radius of the code. For any ROBDA, it is shown by Fossorier and Lin [5] that the SECR can be computed from the minimal squared Euclidean distances (MSEDs) between the all-one sequence and some polyhedra corresponding to the error patterns. An iterative method with linear complexity is further proposed in [5] to compute the SECR for a given error pattern. In this correspondence, we propose a new method to computes such MSEDs. Our method is also of linear complexity and more compact than the one given in [5].

To give a tighter estimation of the probability of decoding error for bounded-distance decoding, a quantity named effective error coefficient is also introduced in the literature. For high SNRs, the ratio between the probability of decoding error of a bounded-distance decoding algorithm and that of the maximum-likelihood (ML) decoding is approximately bounded by the ratio between the effective error coefficients (cf. [7]–[9]). However, it is pointed out in [9], [10] that, as an indicator for the error performance, the effective error coefficient is still not as good as expected. If the minimum Hamming distance of the code is greater than 2, it is proved in [11] that the GMD and Chase decoding algorithms are asymptotically optimal, that is, the ratio between the probability of decoding error of the GMD or any one of the Chase decoding algorithms and that of the ML decoding approaches 1 when the SNR approaches infinity, even though the ratio between their effective error coefficients is much larger than 1. The main result of this correspondence is to generalize this result to the case of bounded-distance ROBDAs. We show that any bounded-distance ROBDA is asymptotically optimal provided that the minimum Hamming distance of the code is greater than 2.

This correspondence is organized as follows. In Section II, we give some definitions and notations needed in this correspondence. In Section III, we discuss the computation of the SECRs of ROBDAs. In Section IV, we show that any bounded-distance ROBDA is asymptotically optimal in the sense shown in [11]. In Section V, we end with some concluding remarks.

II. PRELIMINARIES

Suppose that a binary \((N,K,d_{\text{min}})\) linear block code \(C\) is used for error control over the AWGN channel with BPSK signaling. If the transmitted codeword is \(e = (e_1,e_2,\ldots,e_N) \in C\), then the received sequence \(r \in \mathbb{R}^N\) can be written as \(s(r) + w\), where

\[
s(r) \triangleq (-1)^{r_1}, (-1)^{r_2}, \ldots, (-1)^{r_N}
\]

is the bipolar sequence corresponding to \(e\) and the components of \(w\) are independent Gaussian random variables with common density function

\[
g(w) \triangleq \frac{1}{(\pi N_0)^{N/2}} e^{-w^2/N_0}.
\]
Given the transmitted codeword \(e\), the conditional density function of \(r\) is

\[
p(r \mid e) = \frac{1}{(\pi N_0)^{N/2}} e^{-d_E(r, s(e))/N_0}
\]

(1)

where \(d_E(\cdot, \cdot)\) is the squared Euclidean distance defined, for sequences \(x = (x_1, x_2, \ldots, x_N)\) and \(y = (y_1, y_2, \ldots, y_N)\) in \(\mathbb{R}^N\), by

\[
d_E(x, y) \triangleq (x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_N - y_N)^2.
\]

The well-known maximum a posteriori probability (MAP) decoder outputs the codeword \(e\) which maximizes the a posteriori probability \(Pr(e \mid r) \propto p(r \mid e)Pr(e)/p(r)\) for each given \(r \in \mathbb{R}^N\), where \(p(r)\) is the density function of the received sequence \(r\) and \(Pr(e)\) is the a priori probability that the codeword \(e\) is transmitted. If the codewords are equally likely, i.e., \(Pr(e)\) is the same for all codewords \(e\), maximizing \(Pr(e \mid r)\) is equivalent to maximizing \(p(r \mid e)\). A decoder that outputs the codeword which maximizes \(p(r \mid e)\) is called a maximum-likelihood (ML) decoder.

In general, a soft-decision decoding algorithm \(A\) divides \(\mathbb{R}^N\) into \(2^N + 1\) disjoint regions: \(D_A(e)\), for each \(e \in C\) and \(F_A\). If the received sequence \(r\) belongs to \(D_A(e)\) for any codeword \(e \in C\), the soft-decision decoding algorithm \(A\) declares that \(e\) is the transmitted codeword. If the received sequence \(r\) belongs to \(F_A\), the soft-decision decoding algorithm \(A\) makes no decision on the transmitted codeword, i.e., a decoding failure occurs in this case. \(D_A(e)\) is called the decision region associated with \(e\). For any ML decoder, the inner part of \(D_{ML}(e)\) is the Voronoi region \(V(e)\)

\[
\{ \mathbf{x} \in \mathbb{R}^N : d_E(\mathbf{x}, s(e)) < d_E(\mathbf{x}, s(e')) \}, \text{ for all } e' \in C \setminus \{e\}.
\]

The error performance analysis of the soft-decision decoding algorithm \(A\) is then to estimate the probability that \(A\) fails to make the correct decision

\[
Pr(e_A) \triangleq \sum_{e \in C} Pr(r \in D_A(e) \mid e)Pr(e).
\]

(2)

where \(D_A(e) \triangleq \mathbb{R}^N \setminus D_A(e)\). The MAP decoder is optimal since it minimizes the probability \(Pr(e_A)\). In the meanwhile, if the codewords are not equally likely, an ML decoder is not necessarily optimal. However, in many cases, the a priori probabilities are not available at the receiver and then an ML decoder is the best feasible one since the optimal decoding is impossible for such cases.

Let \(V^N\) denote the set of binary \(N\)-tuples. For a received sequence \(r = (r_1, r_2, \ldots, r_N) \in \mathbb{R}^N\), let \(z_r = (z_1, z_2, \ldots, z_N) \in V^N\) denote the hard-decision sequence defined by \(z_i = 0\) for \(r_i > 0\) and \(z_i = 1\) for \(r_i \leq 0\). A tuple \(v \in V^N\) is said to be better under the ML decoding criterion (or simply better) than another tuple \(v' \in V^N\) if \(d_E(\mathbf{r}, s(v)) < d_E(\mathbf{r}, s(v'))\). Thus, \(z_r\) is the best tuple in \(V^N\). We note that maximizing \(p(r \mid e)\) is equivalent to minimizing \(d_E(\mathbf{r}, s(e))\). Hence, an ML decoder outputs the best codeword for each received sequence.

Let \(\Lambda_n\) be the set of permutations of \(\{1, 2, \ldots, N\}\). A permutation \(\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N) \in \Lambda_n\) permutes any \(N\)-tuple \(v = (v_1, v_2, \ldots, v_N)\) into the \(N\)-tuple \(\lambda(v) = (v_{\lambda_1}, v_{\lambda_2}, \ldots, v_{\lambda_N})\).

For any \(x = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^N\), let \(\lambda_x = (\lambda_1, \lambda_2, \ldots, \lambda_N) \in \Lambda_n\) be the permutation such that the absolute values of the entries of \(\lambda(x)\) are in descending order

\[
x_1 \geq |x_{\lambda_2}| \geq \cdots \geq |x_{\lambda_N}|
\]

(3)

\[
\lambda_i < \lambda_{i+1}, \text{ if } |x_{\lambda_i}| = |x_{\lambda_{i+1}}|
\]

(4)

For any \(\lambda \in \Lambda_n\), let \(F_{\lambda} \subseteq \mathbb{R}^N\) be a subset of \(V^N\). For \(\mathcal{E} \triangleq \{ \lambda : \lambda \in \Lambda_n \}\), let \(\mathcal{A}(\mathcal{E})\) denote the reliability-order-based decoding algorithm (ROBDA) which decodes the received sequence \(r \in \mathbb{R}^N\) to the best codeword in the search region \(S_{\mathcal{E}}(r) \triangleq \{ \mathbf{z} + e : e \in F_{\lambda} \}\) if it contains some codewords and declares decoding failure otherwise. In general, the subsets \(F_{\lambda}\) are big enough such that \(S_{\mathcal{E}}(r)\) contains at least one codeword for almost all received sequences \(r\). The tuples in \(F_{\lambda}\) are called error patterns. The codewords in the search region \(S_{\mathcal{E}}(r)\) are also called candidate codewords generated by \(\mathcal{A}(\mathcal{E})\). In practical algorithms, the candidate codewords are usually generated one by one.

An ML decoder can be seen as an ROBDA with \(F_{\lambda} = V^N\) for all \(\lambda \in \Lambda_n\). We note that, for many known ROBDAs, such as the GMD decoding algorithm [1] and the Chase decoding algorithms [2], the set \(F_{\lambda}\) is independent of \(\lambda\). Therefore, after permuting the bits with \(\lambda\), such ROBDAs correct fixed error patterns. Clearly, it is more convenient to describe such ROBDAs after permuting the bits with \(\lambda\). However, for the specific ROBDAs proposed in [3], [4], the set \(F_{\lambda}\) is dependent on the most reliable basis which may vary for different permutations \(\lambda\). Hence, we do not assume that the bits are pre-ordered according to (3) and (4).

Suppose that \(\Lambda = \mathcal{A}(\mathcal{E})\) is an arbitrary ROBDA. For each codeword \(e\), let \(G_A(e)\), called the generation region associated with \(e\), denote the set of sequences \(x \in \mathbb{R}^N\) for which \(e\) is contained in \(S_{\mathcal{E}}(x)\)

\[
G_A(e) \triangleq \{ x \in \mathbb{R}^N : e \in S_{\mathcal{E}}(x) \}.
\]

Clearly, a codeword \(e\) is generated as a candidate codeword if and only if the received sequence \(r\) falls in \(G_A(e)\). Let \(Pr(e)\) be the probability that the transmitted codeword is not generated as a candidate codeword

\[
Pr(e) \triangleq \sum_{e \in C} Pr(r \in G_A(e) \mid e)Pr(e).
\]

(5)

The probability \(Pr(e)\) defined by (2) can be bounded by (cf. [5], [11])

\[
\max \left\{ Pr(e) : Pr(e) \leq Pr(e) \right\} \leq Pr(e) \leq Pr(e) + Pr(ecM).
\]

(6)

For two sequences \(x = (x_1, x_2, \ldots, x_N)\) and \(x' = (x'_1, x'_2, \ldots, x'_N)\) in \(\mathbb{R}^N\), write

\[
x \circ x' \triangleq (x_1x'_1, x_2x'_2, \ldots, x_Nx'_N).
\]

For \(x \in \mathbb{R}^N\) and \(\lambda \subseteq \mathbb{R}^N\), write \(x \circ \lambda \triangleq \{ x \circ x' : x' \in \lambda \}\). Let \(H_N\) denote the set of sequences \((x_1, x_2, \ldots, x_N) \in \mathbb{R}^N\) with

\[
x_1 \geq x_2 \geq \cdots \geq x_N \geq 0.
\]

(7)

From the definition of ROBDAs, it is not difficult to show that

\[
Pr(e) = \sum_{\lambda \in \Lambda_n \setminus e \in \mathbb{R}^N \setminus \lambda} \Pr(r \in s(\lambda(e)) \circ H_N)\left| 10 \right| (8)
\]

III. SQUARED ERROR-CORRECTION RADIUS OF ROBDAS

The squared error-correction radius (SECR) of a decoding algorithm \(A\) for a binary linear block code is defined as the largest number, denoted as \(\rho(A)\), such that \(A\) decodes correctly whenever the received sequence is within squared Euclidean distance \(\rho(A)\) of the bipolar sequence corresponding to the transmitted codeword.

For an ROBDA \(A = \mathcal{A}(\mathcal{E})\), it is known that the SECR is given by (cf. [5])

\[
\rho(A) = \min \{ \min_{\lambda \in \Lambda_n \setminus e \in \mathbb{R}^N \setminus \lambda} |s(\lambda(e))| \}
\]

(9)
where, for \( e \in V^N \), \( \sigma(e) \) denotes the minimal squared Euclidean distance (MSED) between the sequence \( s(e) \) and the sequences in \( \mathcal{H}_N \) which satisfy (7), i.e.,
\[
\sigma(e) \triangleq d_{E}(s(e), \mathcal{H}_N).
\] (10)

For the computation of the MSEDs \( \sigma(e) \), an iterative method has been proposed in [5]. Below, we present a new method with some ideas from [6]. Our method is more compact than the one given in [5].

A tuple \( e = (v_1, v_2, \ldots, v_m) \in V^m \) is called singular if
\[
\sum_{j=1}^{n} t(v_j) - (-1)^{n_j} \geq 0, \quad \text{for all } 1 \leq i \leq m
\] (11)
where \( t(v) \) is defined by
\[
t(v) \triangleq \max \left\{ 0, \frac{1}{m} \sum_{i=1}^{m} (-1)^{n_j} \right\}.
\] (12)

Clearly, for any singular tuple, its leftmost entry is equal to 0 if and only if the tuple itself is the all-zero tuple \( 0 \).

**Lemma 1:** If \( e \) is a singular tuple in \( V^m \), then \( t(e) \mathcal{H}_m \) is the unique sequence in \( \mathcal{H}_N \) such that
\[
d_E(e, \mathcal{H}_m) = \sigma(e)
\] (13)
where \( \mathcal{H}_m \) is the all-one sequence of length \( m \).

**Proof:** The proof is given in Appendix I. \( \square \)

From Lemma 1, one can deduce the following corollary easily.

**Corollary 1:** If \( e \in V^m \) is a singular tuple with \( l \) nonzero entries, then
\[
\sigma(e) = \begin{cases} 4l(m-l)/m, & \text{if } 2l < m, \\ m, & \text{otherwise}. \end{cases}
\]

For \( 0 \leq j < j' \leq m \) and a tuple \( u = (u_1, u_2, \ldots, u_m) \), let \( \gamma_{j,j'}(u) \) denote the subtuple \((u_{j+1}, u_{j+2}, \ldots, u_{j'})\) of \( u \). Then we have the following theorem.

**Theorem 1:** For \( e = (e_1, e_2, \ldots, e_N) \in V^N \), let \( b(e) = (b_1, b_1, \ldots, b_{p+1}) \) be the sequence of integers such that \( b_0 = 0 < b_1 < \cdots < b_{p+1} = N \) and, for \( 0 \leq k \leq p \), \( b_{k+1} \) is the largest integer satisfying
\[
t(\gamma_{b_k, b_{k+1}}(e)) = \max_{b_k \leq a_{b_k} \leq N} t(\gamma_{b_k, a_{b_k}}(e)).
\] (14)

Then we have the following.

i) The tuples \( \gamma_{0, b_1}(e), \gamma_{b_1, b_2}(e), \ldots, \gamma_{b_p, b_{p+1}}(e) \) are singular and
\[
1 \geq t(\gamma_{0, b_1}(e)) > t(\gamma_{b_1, b_2}(e)) > \cdots > t(\gamma_{b_p, b_{p+1}}(e)) \geq 0.
\] (15)

ii) If \( e_1 = 0 \), then \( e_{b_1+1} \) is the first nonzero entry of \( e \). For each \( k \) with \( 1 \leq k \leq p \), \( e_{b_k} = 0 \) and \( e_{b_k+1} = 1 \).

iii) The sequence \( y \in \mathbb{R}^N \) with
\[
\gamma_{b_k, b_{k+1}}(y) = t(\gamma_{b_k, b_{k+1}}(e)) y_{b_k+1-b_k},
\]
for all \( 0 \leq k \leq p \) is the unique sequence in \( \mathcal{H}_N \) such that
\[
d_E(s(e), y) = \sigma(e).
\]

**Proof:** The proof is given in Appendix II. \( \square \)

According to Theorem 1, the following Algorithm 1 computes the MSED \( \sigma(e) \) for any \( e \in V^N \).

**Algorithm 1**

**Input** A tuple \( e \in V^N \) whose nonzero entries are at the positions \( n_1 < n_2 < \cdots < n_l \).

**Output** The MSED \( \sigma(e) \).

1. **Step 1.** If \( e \) has no nonzero entry, output 0 and end. Otherwise, set \( I = 1, T = 0 \) and \( n_{l+1} = N + 1 \).
2. **Step 2.** Find the largest index \( J > I \) such that \( \max\{(n_j - n_{l+1}) - (2J - I)/((n_j - n_{l+1}) - 0)\} \) is maximized. If the maximum is positive, add \( J-I \) \((n_j - n_{l+1} - J + I)/((n_j - n_{l+1}) - 0)\) to \( T \) and goto Step 3. Otherwise, output \( T + n_{l+1} - n_j \) and end.
3. **Step 3.** If \( J \leq I \), set \( I = J \) and goto Step 2. Otherwise, output \( T \) and end.

We note that the indices \( n_i \) and \( n_j \) in Algorithm 1 correspond to \( b_{i+1} \) and \( b_{i+1} + 1 \), respectively. To efficiently compute the SEC of a ROBDA for long codes, we need further the following lemma (cf. [5, p. 694]).

**Lemma 2:** Let \( e \) and \( e' \) be two binary \( N \)-tuples such that the Hamming weight of \( \gamma_{0, i}(e) \) is not smaller than that of \( \gamma_{0, i}(e') \) for all \( 1 \leq i \leq N \). Then, \( \sigma(e) \geq \sigma(e') \).

**Proof:** For the two tuples \( e \) and \( e' \) in Lemma 2, we write \( e \succeq e' \). The following two examples show that, to compute the SECRs for many ROBDAs, only very few MSEDs need to be computed even if the set \( V^N \setminus \mathbb{A} \) in (9) is very big.

**Example 1:** The modified GMD decoding algorithm [13], [7] is the ROBDA with
\[
V^N \setminus \mathbb{A}(E_A) = \{ e \in V^N \mid e \succeq e' \}, \quad \text{for all } \mathbb{A} \in \Lambda_N
\]
where \( e' \) is the binary \( N \)-tuple whose nonzero entries are at the position \( N - d_{\text{min}} \) and the positions \( N - d_{\text{min}} + 2l + 1 \) for \( 0 < i < t_0 \triangleq [d_{\text{min}}/2] \). Clearly, \( \gamma_{N-d_{\text{min}}-1}(e') = 0 \) and \( N - d_{\text{min}} + 2 \) is the largest value of \( l \) such that \( \gamma_{N-d_{\text{min}}+1}(e') \) is a singular tuple and \( \gamma_{N-d_{\text{min}}+2, N}(e') \) is also a singular tuple. Then, from Theorem 1
\[
b(e') = (0, N - d_{\text{min}} - 1, N - d_{\text{min}} + 2, N).
\]

From Corollary 1
\[
\sigma(\gamma_{N-d_{\text{min}}-1}(e')) = 0
\]
\[
\sigma(\gamma_{N-d_{\text{min}}+1, N-d_{\text{min}}+2}(e')) = 4 \cdot \frac{1}{3} \cdot (3 - 1) = 8/3
\]
\[
\sigma(\gamma_{N-d_{\text{min}}+2, N}(e')) = d_{\text{min}} - 2
\]
and thus, from (16), \( \sigma(e') = d_{\text{min}} + 2/3 \). Hence, from (9) and Lemma 2, the SEC of the modified GMD decoding algorithm is equal to \( d_{\text{min}} \).

**Example 2:** Some multiple-Chase-like decoding algorithms are proposed in [14]. Like in the case of the original Chase algorithms, the multiple-Chase-like decoding algorithms employ the bounded-distance-(\( t_0 - 1 \)) binary decoder, which outputs a codeword within Hamming distance \( t_0 - 1 \) from the input tuple if any, to generate the candidate codewords. For \( 0 \leq \tau \leq N - t_0 \) and binary \( N \)-tuple \( u \), Chase \( (r, u) \) is a \( 2^\tau \)-stage decoding algorithm with the following feature: At each stage, the bounded-distance-(\( t_0 - 1 \)) binary decoder is applied to a binary \( N \)-tuple, which is obtained by adding \( r \) to an error pattern whose nonzero components are confined in the \( \tau \) least reliable positions. For a positive integer \( h \), the \((h, \tau)\)-Chase decoding is a multiple-Chase-like decoding algorithm consisting of successive
Chase \((r, u^{(i)})\) with \(1 \leq i \leq h\), where \(u^{(i)}\), called the \(i\)th search center, is the best binary \(N\)-tuple among those which are not yet searched by the \((i - 1, r)\)-Chase decoding. The first search center is the hard-decision tuple \(z_r\). The second search center is (cf. [15])

\[
\mathbf{u}^{(2)} = z_r + \lambda_r^{-1} \left( (0, \ldots, 0, 1, \ldots, 1, \ldots, 0) \right).
\]

Hence, the \((2, r)\)-Chase decoding is the ROBDA with \(V_N \setminus \lambda(E_\lambda) = \{ e \in V_N : e \supset f \} \) for all \( \lambda \in \Lambda_N \), where

\[
f \triangleq \left( \begin{array}{cccccc} 0, & \ldots, & 0, & 1, & \ldots, & 1, \ldots, 0 \end{array} \right). \]

From (9) and Lemma 2, the SECR of \((2, r)\)-Chase decoding is equal to the smaller of \(d_{\min}\) and \(\sigma(f)\). Since \(\gamma_{0,N-N-[\beta_0/2]}(f) = 0\) and \(\gamma_{N-N-[\beta_0/2]}(f)\) is a singular tuple, it follows from Theorem 1 and Corollary 1 that

\[
\sigma(f) = \sigma(\gamma_{N-N-[\beta_0/2]}(f)) = \begin{cases} \tau + \lceil \frac{|t_0|}{2} \rceil + t_0, & \text{if } \tau \leq \lceil \frac{|t_0|}{2} \rceil \\ \frac{4}{3} \left( \frac{4}{3} \right)^{d_{\min} / 4} t_0, & \text{otherwise} \end{cases}
\]

Hence, the SECR of \((2, r)\)-Chase decoding is equal to \(d_{\min}\) if and only if

\[
\tau \geq \left\lceil \frac{d_{\min}}{4} \right\rceil - \left\lfloor \frac{|t_0|}{2} \right\rfloor.
\]

IV. ASYMPTOTIC OPTIMALITY OF BOUNDED-DISTANCE ROBDAS

A decoding algorithm \(A\) is called a bounded-distance decoding if \(\rho(A) = d_{\min}\). The GMD and Chase decoding algorithms are examples of bounded-distance decoding. In [11], GMD and Chase decoding algorithms are shown to be asymptotically optimal, i.e., the ratio between the probability of decoding error of the GMD or any one of the Chase decoding algorithms and that of the ML decoding approaches approaches 1 when the SNR approaches infinity. In this section, we show that an ROBDA decoding algorithm is also asymptotically optimal if it is a bounded-distance decoding.

For tuple \(v = (v_1, v_2, \ldots, v_m) \in V_m\), let

\[
P(v) \triangleq \frac{1}{(\pi N_0)^{m/2}} \int_{H_m} e^{-\frac{g(v, \epsilon(\lambda))}{N_0}} dx.
\]

where \(g_\epsilon(x_i)\) is defined as

\[
g_\epsilon(x_i) \triangleq \frac{1}{\sqrt{2\pi N_0}} e^{-(x_i - (0) \epsilon)^2 / N_0}.
\]

For any tuple \(z \in V_N\), from (1) and (17)

\[
\Pr(v \in s(z) \circ \mathcal{H}_N | 0) = P(v).
\]

Then, for any ROBDA \(\mathcal{A}(C)\), from (8) and (18)

\[
\Pr(\epsilon^{(i)}_{\mathcal{A}(C)}) = \sum_{\lambda \in \Lambda_N \setminus E_N \cup E_\lambda} P(\lambda(e)).
\]

From now onwards, all the asymptotic estimates are under the condition \(N_0 \to 0\). For the integral \(P(v)\), an asymptotic estimate is given in the following theorem.

**Theorem 2:** For any singular tuple \(v \in V_m\)

\[
P(v) = O(1) N_0^{\sigma(v)/4} e^{-\sigma(\lambda)/N_0}.
\]

where \(\sigma(v)\) is defined by (10).

**Proof:** The proof is given in Appendix III.

The following corollary is deduced from Theorems 1 and 2.

**Corollary 2:** For any \(e \in V_N\)

\[
P(e) = O(1) N_0^{\sigma(e)/4} e^{-\sigma(\lambda)/N_0}.
\]

Then, from (16), (22), and

\[
\mathcal{P}(e) \leq \prod_{p=0}^{\infty} \mathcal{P}(\epsilon^{(i)}_{\mathcal{A}(C)}(e))
\]

the estimate (21) follows.

Now, from the well-known estimate (cf. [11])

\[
\Pr(e_{\text{ML}}) = \left( \frac{A_{d_{\min}}}{2\sqrt{d_{\min}}} + o(1) \right) N_0^{d_{\min}/2} e^{-d_{\min}/N_0}
\]

where \(A_{d_{\min}}\) is the number of codewords of Hamming weight \(d_{\min}\), we are ready to prove the following theorem.

**Theorem 3:** Assume that the minimum Hamming distance \(d_{\min}\) of the code \(C\) is greater than 2 and \(\mathcal{A}(C)\) is a bounded-distance reliability-order-based decoding algorithm. Then, when the SNR approaches infinity, \(\mathcal{A}(C)\) is asymptotically optimal, i.e.,

\[
\Pr(e_{\text{ML}}) \equiv (1 + o(1)) \Pr(e_{\text{ML}}), \quad \text{when } N_0 \to 0.
\]

**Proof:** For all \(\lambda \in \Lambda_N\) and \(e \in V_N \setminus E_\lambda\), since \(\mathcal{A}(C)\) is a bounded-distance decoding, i.e., \(\rho(\mathcal{A}(C)) = d_{\min}\), from (9), \(\sigma(\lambda(e)) \geq d_{\min}\), and thus from (21)

\[
P(\lambda(e)) = O(1) N_0^{d_{\min}/4} e^{-d_{\min}/N_0}.
\]

Hence, from (19) and (25)

\[
\Pr(\epsilon_{\text{ML}}) = O(1) N_0^{d_{\min}/4} e^{-d_{\min}/N_0}.
\]

Since the minimum Hamming distance of \(C\) is greater than 2, i.e., \(d_{\min} > 2\), it follows from (23) and (26) that

\[
\Pr(\epsilon_{\text{ML}}) = o(1) N_0^{d_{\min}/4} e^{-d_{\min}/N_0} = o(1) \Pr(e_{\text{ML}}).
\]

Hence, the asymptotic estimate (24) follows from (6) and (27).
Remark: We note that not all bounded-distance decoding algorithms are asymptotically optimal. An example is the GMD/TP decoding, which is originally proposed in [12] and defined in [7]. It is shown in [7] that the decision region $D_{GMD/TP}(\mathbf{e})$ associated to any arbitrary codeword $\mathbf{e}$ is the intersection of the $(2^N)$ open half-spaces

$$\{\mathbf{x} \in \mathbb{R}^N : d_E(\mathbf{x}, \mathbf{s}(\mathbf{e})) < d_E(\mathbf{x}, \mathbf{s}(\mathbf{v} + \mathbf{e}))\}$$

for all tuples $\mathbf{v}$ in $V^N$ of Hamming weight $d_{\text{min}}$. It can be shown easily that the GMD/TP decoding is a bounded-distance decoding and

$$\lim_{N \to 0} \frac{\Pr(\mathbf{e} \in \mathcal{C}_{GMD/TP})}{\Pr(\epsilon \in \mathcal{C}_\text{ML})} = \frac{N}{A_{\text{min}}}$$

which is greater than 1 in general.

V. CONCLUSION

In this correspondence, we study the reliability-order-based decoding algorithms (ROBDAs) for binary block codes over AWGN channels. In [5], it is shown that the squared error-correction-radii of ROBDAs can be computed from the minimal squared Euclidean distances (MSESDs) from the all-one tuple to some polyhedra. For the computation of such MSESDs, we give a new method which is more compact than the iterative method given in [5]. Some examples are given to illustrate our new method. These results are further used to show the main result of this correspondence: The ratio between the probability of decoding error for any bounded-distance ROBDA and that for the ML decoding approaches 1 when the SNR approaches infinity, provided that the minimal Hamming distance of the code is greater than 2.

APPENDIX I

PROOF OF LEMMA 1

Proof: We consider to prove, for all $\mathbf{y} \in \mathcal{H}_m$, that

$$d_E(\mathbf{s}(\mathbf{v}), \mathbf{y}) - d_E(\mathbf{s}(\mathbf{v}), \mathbf{t}(\mathbf{v})1_m) \geq \sum_{j=1}^{m} (y_j - t_j)^2 \tag{A1}$$

which implies that $\mathbf{x} = \mathbf{t}(\mathbf{v})1_m$ is the unique sequence in $\mathcal{H}_m$ satisfying (13).

Assume $\mathbf{v} = (v_1, v_2, \ldots, v_m)$. Let $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ be an arbitrary sequence in $\mathcal{H}_m$. Define $y_0 \triangleq y_1 + 1$ and $y_{m+1} \triangleq 0$. Then

$$y_0 > y_1 \geq y_2 \geq \cdots \geq y_m \geq y_{m+1}. \tag{A2}$$

Let $l$ be an integer with $0 \leq l \leq m$ such that

$$y_l \geq t(v) \geq y_{l+1}. \tag{A3}$$

Since $0 \leq t(v) \leq 1$, such $l$ does exist. From the following identity:

$$\sum_{j=1}^{m} a_j b_j = \sum_{j=1}^{m} (a_j - a_{j+1}) \sum_{j=1}^{j} b_j + a_1 \sum_{j=1}^{1} b_j$$

$$+ a_{i+1} \sum_{j=i+1}^{m} b_j + \sum_{j=i+1}^{m} (a_{i+1} - a_i) \sum_{j=i+1}^{m} b_j$$

we see that

$$d_E(\mathbf{s}(\mathbf{v}), \mathbf{y}) - d_E(\mathbf{s}(\mathbf{v}), \mathbf{t}(\mathbf{v})1_m)$$

$$= \sum_{j=1}^{m} (y_j - t_j)^2 + 2(y_j - t_j)(t_j - (-1)^j)$$

$$= \sum_{j=1}^{m} (y_j - t_j)^2 + 2 \sum_{j=1}^{m} (y_j - y_{j+1}) \sum_{j=1}^{j} t_j - (-1)^j$$

$$+ 2(y_j - t_j) \sum_{j=1}^{m} (t_j - (-1)^j)$$

$$+ 2 \sum_{j=1}^{m} (y_j - y_{j+1}) \sum_{j=1}^{j} t_j - (-1)^j$$

$$\geq \sum_{j=1}^{m} (y_j - t_j)^2$$

$$= \sum_{j=1}^{m} (y_j - t_j)^2 \geq 0$$

for all $i \leq k \leq m$. Then, (A1) follows from (11) and (A2)-(A4).

Now we distinguish between two cases.

Case 1: $t(v) = 0$. From (A2) and (A3), $y_i = 0$ for $i > l$ and thus, (A1) follows from (11) and (A2)-(A4).

Case 2: $t(v) > 0$. From (12),

$$t(v) = \frac{1}{N} \sum_{i=1}^{m} (-1)^j \sum_{j=1}^{m} (y_j - t_j)(t_j - (-1)^j)$$

which implies that $\mathbf{x} = \mathbf{t}(\mathbf{v})1_m$ is the unique sequence in $\mathcal{H}_m$ satisfying (13).

Hence, $\mathbf{f}_k$ is a singular tuple.

For $1 \leq k \leq p$, from $b_k < b_{k+1}$, from (12) and (14)

$$\sum_{j=b_k+1}^{b_{k+1}} (-1)^j \geq 0$$

Hence, $\mathbf{f}_k$ is a singular tuple.

For $1 \leq k \leq p$, from $b_k < b_{k+1} = N$ and (14)

$$0 \leq t(\gamma_{b_k-1,1}(\mathbf{e})) \leq t(\mathbf{f}_{k-1})$$

for any $b_k < i \leq N$ and thus, from (12)

$$\sum_{j=b_k+1}^{b_{k+1}} (-1)^j = (b_{k+1} - b_k - 1)(t(\mathbf{f}_k) - t(\mathbf{f}_{k-1})) \geq 0$$

(B2)

If there is an integer $k$ with $1 \leq k \leq p$ such that $t(\mathbf{f}_{k-1}) \leq t(\mathbf{f}_k)$, then $t(\mathbf{f}_k)$ is also positive and thus, from (12)

$$\sum_{j=b_k+1}^{b_{k+1}} (-1)^j \geq 0 \leq (b_{k+1} - b_k - 1)(t(\mathbf{f}_k))$$

(B3)
From (B2) and (B3)
\begin{align*}
t(\gamma_{b_{k-1} : b_{k+1}}(e)) &= \max \left\{ 0, \frac{1}{b_{k+1} - b_{k}} \sum_{i=b_{k-1}+1}^{b_{k+1}} (-1)^{i-1} \right\} \\
&\geq \frac{(b_k - b_{k-1})t(f_{b_{k-1}}) + (b_{k+1} - b_k)t(f_{b_k})}{b_{k+1} - b_{k-1}} \\
&= t(f_{b_{k-1}})
\end{align*}
which contradicts (B1). Hence, (15) is true.

ii) If \( e_1 = 0 \), it is obvious that \( e_{b_{k+1}} \) is the first nonzero entry of \( e \). Let \( k \) be an integer with \( 1 \leq k \leq p \).

If \( e_{b_{k+1}} = 0 \), from (12), (B1) and (B2),
\begin{align*}
t(\gamma_{b_{k-1} : b_{k+1}}(e)) &= \max \left\{ 0, \frac{1}{b_{k+1} - b_{k}} \sum_{i=b_{k-1}+1}^{b_{k+1}} (-1)^{i-1} \right\} \\
&\geq \frac{(b_k - b_{k-1})t(f_{b_{k-1}}) + (b_{k+1} - b_k)t(f_{b_k})}{b_{k+1} - b_{k-1}} \\
&= t(f_{b_{k-1}})
\end{align*}
which contradicts (14). Hence, \( e_{b_{k+1}} = 1 \).

Suppose \( e_{b_k} = 1 \). If \( b_k = b_{k-1} + 1 \), it follows from (B2) that \( t(f_{b_{k-1}}) = -1 \), which contradicts (12). If \( b_k = b_{k-1} \geq 2 \), from (12) and (B2)
\begin{align*}
t(f_{b_{k-1}}) &\leq \frac{(b_k - 1 - b_{k-1})t(\gamma_{b_{k-1} : b_{k+1}}(e)) - 1}{b_k - b_{k-1}} \\
&< t(\gamma_{b_{k-1} : b_{k+1}}(e)),
\end{align*}
which contradicts (14). Hence, \( e_{b_k} = 0 \).

iii) It follows from (15) that the sequence \( y \) belongs to \( \mathcal{H}_N \). Hence,
\begin{align*}
\sigma(e) \leq d_{E_{\mathcal{H}}}(s(e), y) = \sum_{k=0}^{p} d_{E_{\mathcal{H}}}(s(f_k), t(f_k) 1_{b_{k+1} - b_k}).
\end{align*}

Let \( x \) be an arbitrary sequence in \( \mathcal{H}_N \) such that
\begin{align*}
d_{E_{\mathcal{H}}}(s(e), x) = \sigma(e).
\end{align*}
For \( 0 \leq k \leq p \), since \( f_k \) is singular, from Lemma 1 and \( \gamma_{b_{k+1} : b_{k+1}}(x) \in \mathcal{H}_{b_{k+1} - b_k}
\begin{align*}
d_{E_{\mathcal{H}}}(s(f_k), t(f_k) 1_{b_{k+1} - b_k}) = \sigma(f_k) \leq d_{E_{\mathcal{H}}}(s(f_k), \gamma_{b_{k+1} : b_{k+1}}(x))
\end{align*}
Hence, from (B4)–(B6),
\begin{align*}
\sigma(e) \leq \sum_{k=0}^{p} \sigma(f_k) \leq d_{E_{\mathcal{H}}}(s(e), \gamma_{b_{k+1} : b_{k+1}}(x))
\end{align*}
and thus (16) follows. In particular, for all \( 0 \leq k \leq p \), we have
\begin{align*}
\sigma(f_k) = d_{E_{\mathcal{H}}}(s(f_k), \gamma_{b_{k+1} : b_{k+1}}(x))
\end{align*}
and, therefore, from Lemma 1 and that \( f_k \) is singular
\begin{align*}
\gamma_{b_{k+1} : b_{k+1}}(x) = t(f_k) 1_{b_{k+1} - b_k} = \gamma_{b_{k+1} : b_{k+1}}(y).
\end{align*}
Hence, \( x = y \). This proves the uniqueness of \( y \).

\section*{Appendix III: Proof of Theorem 2}

We first define, for a nonnegative number \( t \)
\begin{align*}
S(t, N_0) \triangleq \int_{0}^{\infty} g_1(x) \left( \int_{0}^{x} g_0(y) dy \right)^{t} dx
\end{align*}
\begin{align*}
&= \frac{1}{(\pi N_0)^{1/4}} \int_{0}^{\infty} \left( \int_{0}^{x} e^{-\pi(y-1)^2/N_0} dy \right)^{t} e^{-\pi x^2/N_0} dx
\end{align*}
\begin{align*}
&= \frac{1}{(\pi N_0)^{1/4}} \int_{1}^{\infty} \left( \int_{2-\alpha u}^{2} e^{-\pi y^2/N_0} dy \right)^{t} e^{-\pi \alpha^2 u^2/N_0} du. \tag{C1}
\end{align*}
To prove Theorem 2, we need some asymptotic estimates for \( S(t, N_0) \) when \( N_0 \) approaches 0. The following two asymptotic estimates are shown in [11]:
\begin{align*}
S(0, N_0) &= \left( \frac{1}{2\sqrt{\pi N_0}} + o(1) \right) N_0^{1/2} e^{-1/N_0} \tag{C2}
\end{align*}
\begin{align*}
S(1, N_0) &= \left( \frac{1}{4\sqrt{2\pi}} + o(1) \right) N_0^{1/2} e^{-2/N_0}. \tag{C3}
\end{align*}
To give an asymptotic estimate for \( S(t, N_0) \) with \( t > 1 \), we prove the following lemma first.

\textbf{Lemma 3:}

i) For any given positive number \( \alpha \)
\begin{align*}
\int_{0}^{\infty} e^{-\pi y^2/N_0} dy = \frac{1 + o(1)}{2\alpha} N_0 e^{-\pi \alpha^2/N_0}. \tag{C4}
\end{align*}

ii) For real numbers \( t > 1 \) and \( \xi \) with \( 1 + \xi < \frac{2t}{t+1} < 2 - \xi \)
\begin{align*}
\int_{1+\xi}^{2-\xi} \phi_t(u) du = \left( \frac{\pi^{1/2} (t+1)^{2(t-1)/2}}{2^t} + o(1) \right) N_0^{1/2} \tag{C6}
\end{align*}
where
\begin{align*}
\phi_t(u) \triangleq (2-u)^{-t} e^{-(t+1)(u-2t)/(t+1)^2/N_0}. \tag{C7}
\end{align*}

\textbf{Proof:}

i) The asymptotic estimate (C4) follows from (C1), (C2), and
\begin{align*}
\int_{0}^{\infty} e^{-\pi y^2/N_0} dy = \sqrt{\pi}
\end{align*}
\begin{align*}
\int_{-\infty}^{\infty} e^{-\pi y^2} dy = \sqrt{\pi}
\end{align*}
\begin{align*}
\int_{0}^{\infty} e^{-\pi \alpha^2 u^2/N_0} du = \alpha \int_{0}^{\infty} e^{-\pi x^2} dx.
\end{align*}
and we have
\[ K(\tau) = \frac{1 + o(1)}{2\tau} \cdot \frac{N_0}{t+1} e^{-2(t+1)/N_0} = o(1)N_0^{1/2}. \]  
\[ K(-\tau) - K(\tau) = K(-\infty) - 2K(\tau) = \frac{\sqrt{\tau}}{t+1} o(1)N_0^{1/2}. \]  
\[ (C9) \]

On the other hand, it follows from (C5) that
\[ \tau_0 \triangleq \min \left\{ \frac{2t}{t+1} - (1+\xi), (2-\xi) - \frac{2t}{t+1} \right\} > 0. \]

Let \( \tau \) be a given positive number with \( 0 < \tau \leq \tau_0 \). Then, \( 1+\xi < 2t/(t+1) - \tau < 2t/(t+1) + \tau \leq 2-\xi \) and
\[ 0 \leq \int_{2t/(t+1)+\tau}^{2t/(t+1)+\tau} \phi_t(u) \, du \leq \xi^{-1} K(\tau) \]
\[ (C11) \]
\[ 0 \leq \int_{2t/(t+1)-\tau}^{2t/(t+1)-\tau} \phi_t(u) \, du \leq \left( \frac{2t}{t+1} + \tau \right)^{-1} K(\tau) \]
\[ (C12) \]
\[ 0 \leq \int_{2t/(t+1)-\tau}^{2t/(t+1)+\tau} \phi_t(u) \, du \leq \left( \frac{2t}{t+1} - \tau \right)^{-1} (K(-\tau) - K(\tau)) \]
\[ (C13) \]
\[ 0 \leq \int_{2t/(t+1)-\tau}^{2t/(t+1)+\tau} \phi_t(u) \, du \leq \left( \frac{2t}{t+1} - \tau \right)^{-1} (K(-\tau) - K(\tau)) \]
\[ (C14) \]

Thus, (C6) follows from the arbitrariness of \( \tau \) and (C9) to (C14). \( \square \)

An asymptotic estimate for \( S(t, N_0) \) with \( t > 1 \) is given in the following lemma.

**Lemma 4**: For any positive number \( t > 1 \)
\[ S(t, N_0) = \left( (t+1) \frac{1}{2t} \right)^{1/2} o(1)N_0^{1/2} e^{-\frac{2(t+1)}{2t} N_0} \]  
\[ (C15) \]

**Proof**: Since \( t > 1 \), there is a small positive number \( \xi \) such that (C5) holds and
\[ \min \{ 1 + t(1-\xi)^2, (2-\xi)^2 \} > \frac{At}{t+1} \]  
\[ (C16) \]

In the following, we assume that \( \xi \) is such a positive number. From (C4), (C8), and (C16)
\[ \int_{2t/(t+1)-\tau}^{2t/(t+1)+\tau} \frac{t}{2t} e^{-\frac{2t}{2t} N_0} du \]
\[ \leq \int_{2t/(t+1)-\tau}^{2t/(t+1)+\tau} \left( t - (1-\xi)^2 \right)^{1/2} e^{-2(t+1)/N_0} du \]
\[ \leq t - (1-\xi)^2/2 \]
\[ \int_{2t/(t+1)-\tau}^{2t/(t+1)+\tau} \left( 1 + o(1) \right) N_0 e^{-2(t+1)/(t+1) N_0} \]
\[ = o(1)N_0^{(1/2)} e^{-\frac{2(t+1)}{2(t+1)} N_0} \]
\[ (C17) \]

From (C4), for any positive number \( \epsilon \), there is a positive number \( M \)
\[ \int_{1}^{\epsilon} e^{-\epsilon^2/2 N_0} du \leq \frac{1}{2} N_0 e^{-\epsilon^2/2 N_0} \]
and consequently, for any positive number \( a \) and \( 0 < N_0 < a^2 M \)
\[ \int_{a}^{\epsilon} e^{-\epsilon^2/2 N_0} du \leq \frac{1}{2} N_0 e^{-\epsilon^2/2 N_0} \]
\[ (C19) \]

For \( 1 < u < 2 \) and \( N_0 > 0 \), write
\[ f(u, N_0) \triangleq \frac{1}{2(2-u)} N_0 e^{-2(u-2)^2/2 N_0} \]
\[ (C20) \]

Then, for \( 0 < N_0 < \xi^2/2 M \), and \( u \) with \( 1+\xi < u < 2-\xi \), from (C19) and \( \xi < 2-\xi < 1-\xi \)
\[ \int_{2-u}^{1} e^{2(u-2)/N_0} du \leq \int_{2-u}^{\epsilon} e^{2(u-2)/N_0} du \leq (1+\epsilon) f(u, N_0) \]
\[ (C21) \]
\[ \int_{2-u}^{1} e^{2(u-2)/N_0} du = \int_{2-u}^{\epsilon} e^{2(u-2)/N_0} du - \int_{\epsilon}^{\epsilon} e^{2(u-2)/N_0} du \]
\[ \leq \frac{1}{2(2-u)} \frac{1}{2} N_0 e^{-2(2-u)^2/2 N_0} \]
\[ = (1+\epsilon) f(u, N_0) \]
\[ (C22) \]

Since \( (1-\xi)^2 - 1 = -(2-\xi) \xi < 0 \), from (C21) and (C22), there is a positive number \( M' \) such that, for \( 0 < N_0 < M' \) and \( 1+\xi < u < 2-\xi \)
\[ \int_{2-u}^{1} e^{-2(u-2)/N_0} du \leq 2\epsilon f(u, N_0) \]
\[ (C23) \]

Then, from (C6), (C7), (C20), and (C23)
\[ \int_{2-u}^{1} e^{-2(u-2)/N_0} du \]
\[ = (1+o(1)) \int_{2-u}^{1} \left( \frac{2}{2(2-u)} N_0 e^{-2(u-2)^2/2 N_0} \right) du \]
\[ = (1/2 + o(1)) e^{-(u-1)^2/2 N_0} \]
\[ (C24) \]

Hence, (C15) follows from (C1), (C17), (C18), and (C24). \( \square \)

**Proof of Theorem 2**: If \( \sigma = 0 \), then \( \sigma(v) = 0 \) and (20) follows from the fact that \( 0 \leq R(v) \leq 1 \). Now we consider the case that \( \sigma \) has at least one nonzero entry. Assume that the nonzero entries of \( \sigma \) are at the positions \( m_1 = 1 < m_2 < \ldots < m_l \), where \( l \) is the number of the nonzero entries of \( \sigma \). Clearly, for \( 1 \leq j \leq l \), in the entries of \( \sigma \) at the leftmost \( (m_j-1) \) positions, there are \( (j-1) \) ones and \( (m_j-j) \) zeros, respectively. We distinguish between two cases.
in other words, the leftmost positions is not greater than \( \sigma(\mathbf{r}) \).

Then, the estimate (20) follows from (11) and \( t(\mathbf{v}) = 0 \) for \( 2 \leq j \leq l \), from (11) and \( t(\mathbf{v}) = 0 \):

\[
0 \leq \sum_{i=1}^{m_j-1} (t(\mathbf{v}) - (-1)^{\gamma_i}) = (j - 1) - (m_j - j)
\]

which implies that \( \mathbf{r} \) has at most \( (j - 1) \) zero entries at the leftmost \( (m_j - 1) \) positions. Assume that the zero entries of \( \mathbf{r} \) are at the positions \( n_1 < n_2 < \cdots < n_{m_j} \). Then

\[
n_j > m_j, \quad \text{for} \ 1 \leq j \leq m - l.
\]

(C25)

Since

\[
\mathcal{P}(\mathbf{v}) = \int_0^\infty g_1(x_m) dx_{m_1} \times \int_0^{x_{m_1}} g_1(x_{m_2}) dx_{m_2} \times \cdots \times \int_0^{x_{m_{l-1}}} g_1(x_{m_l}) dx_{m_l} \leq \int_0^{x_{m_1}} g_0(x) dx \times \cdots \times \int_0^{x_{m_{l-1}}} g_0(x) dx = O(1) N_0^{(m-l)/2} e^{-m N_0},
\]

(C27)

Then, the estimate (20) follows from \( 2l \geq m = \sigma(\mathbf{r}) \) and (C27).

Case \( 2l < m \). From (12) and Corollary 1, \( t(\mathbf{v}) = (m - 2l)/m \) and \( \sigma(\mathbf{r}) = 4l(m - l)/m \). For any \( 2 \leq j \leq l \), from (11)

\[
\sum_{i=1}^{m_j-1} (t(\mathbf{v}) - (-1)^{\gamma_i}) = (m_j - 1) \cdot \frac{m - 2l}{m} = ((m_j - j) - (j - 1)) \geq 0
\]

which is equivalent to

\[
m_j - j \leq \frac{(j - 1)(m - l)}{l}.
\]

(C28)

The inequality (C28) implies that the number of zero entries of \( \mathbf{r} \) at the leftmost \( (m_j - 1) \) positions is not greater than \((j - 1)(m - l)/l\) or, in other words, the \( [((j - 1)(m - l)/l) + 1] \)th zero entry of \( \mathbf{r} \) must be to the right of the \( j \)th nonzero entry of \( \mathbf{r} \). Hence, from (C15), (C26), and (C28)

\[
\mathcal{P}(\mathbf{v}) \leq \int_0^{x_{m_1}} g_1(x_m) dx_{m_1} \times \int_0^{x_{m_1}} g_1(x_{m_2}) dx_{m_2} \times \cdots \times \int_0^{x_{m_{l-1}}} g_1(x_{m_l}) dx_{m_l} \left[ \frac{m_j - 2l}{m_{l-1}} \right] \left[ \frac{m_j - 2l}{m_{l-2}} \right] \cdots \left[ \frac{m_j - 2l}{m_1} \right] \leq \int_0^{x_{m_1}} g_1(x_m) dx_{m_1} \times \cdots \times \int_0^{x_{m_{l-1}}} g_0(x) dx \left[ \frac{m_j - 2l}{m_{l-1}} \right] \left[ \frac{m_j - 2l}{m_{l-2}} \right] \cdots \left[ \frac{m_j - 2l}{m_1} \right] \leq \int_0^{x_{m_1}} g_1(x_m) dx_{m_1} \times \cdots \times \int_0^{x_{m_{l-1}}} g_0(x) dx \left[ \frac{m_j - 2l}{m_{l-1}} \right] \left[ \frac{m_j - 2l}{m_{l-2}} \right] \cdots \left[ \frac{m_j - 2l}{m_1} \right]
\]

Thus, the estimate (20) follows from \( 2l(m - l)/m = \sigma(\mathbf{r}) \) and (C29).

\[
\Box
\]

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees and the Associate Editor M. Fossorier for their useful comments and suggestions that helped to improve the correspondence.

REFERENCES


A code covering radius of the coordinate values to a given word in a code is denoted by \( r \). A code covering radius is the smallest integer so that any word in the code is covered by a word in the code. A word \( w \) is said to cover another word \( u \) if \( u + w \) is the zero word. A code with covering radius \( r \) is called a \( r \)-ball code. Analogously, if we allow only \( 1 \)-s to become \( 0 \)-s, we get the upward directed ball. Finally the union of \( r \)-ball and \( r \)-ball is the \( (r+r) \)-ball.

Covering codes have been studied extensively—see [1] and its references. The main motivations being football pools and data compression, various types of applications have indeed acted as stimulation for this correspondence. In this correspondence, binary unidirectional covering codes are studied with an emphasis on (bounds on) minimum cardinalities of the unidirectional ball around a code word covering radius. Best known bounds on those cardinalities for \( r \leq 6 \) and \( n \leq 8 \) are tabulated in [1] and [2].

For a prescribed value of \( r \), the set of words obtained by making at most \( r \) changes to a given word is called a \( r \)-ball around \( u \). The \( r \)-ball has radius \( r \) and upper directed ball of \( u \) is the \( r \)-ball with \( u \) at its center. Analogously, if we allow only \( 1 \)-s to become \( 0 \)-s, we get the \( r \)-ball with \( u \) at its center. In Section V, we give a method for determining all values of \( n \) for \( r \leq 27 \).

The main approach used to determine lower bounds on these cardinalities is the integer programming formulation. We utilize for \( r \leq 9 \) and \( n \leq 17 \) and \( 5 \leq r \leq n \leq 19 \) several integer programming formulations. The program glpk [12] was used to solve instances of these integer programming problems. Determination of \( n \) for \( r \leq 9 \) was comparatively easy and the number of variables and constraints is moderate. For \( r \geq 10 \) the number of variables and constraints is huge and integer programming becomes computationally intractable. We were able to determine all values of \( n \) for \( r \leq 12 \) and \( n \leq 13 \). For \( r \geq 14 \) we had to resort to searching for \( n \) using tabu search; this approach is analogous to known constructions for covering codes are discussed in Sections II-A2 and II-A3. The program glpk [12] was used to solve instances of these integer programming problems.

Covering codes, integer programming, tabu search, unidirectional covering codes, unidirectional correcting codes and via analogies discover unexplored types of covering codes. Some constructions for unidirectional covering codes are summarized and tabulated in [1].