UC Riverside

UC Riverside Previously Published Works

Title

Recursive error correction for general Reed-Muller codes

Permalink

https://escholarship.org/uc/item/7vg206xr

Journal

Discrete Applied Mathematics, 154(2)

ISSN

0166-218X

Authors

Dumer, I

Shabunov, K

Publication Date

2006-02-01

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Recursive error correction for general Reed-Muller codes ⋆

Ilya Dumer, Kirill Shabunov

College of Engineering, University of California, Riverside, CA 92521, USA

Abstract

Reed-Muller (RM) codes of growing length n and distance d are considered over a binary symmetric channel. A recursive decoding algorithm is designed that has complexity of order $n \log n$ and corrects most error patterns of weight $(d \ln d)/2$. The presented algorithm outperforms other algorithms with nonexponential decoding complexity, which are known for RM codes. We evaluate code performance using a new probabilistic technique that disintegrates decoding into a sequence of recursive steps. This allows us to define the most error-prone information symbols and find the highest transition error probability p, which yields a vanishing output error probability on long codes.

Key words: Recursive decoding, decoding threshold, Plotkin construction, Reed-Muller codes.

1 Introduction

In this paper, we design and analyze new recursive decoding algorithms for RM codes. These codes are fully defined by two integers $0 \le r \le m$, and are called below $\left\{ {m \atop r} \right\}$-codes. Code design of $\left\{ {m \atop r} \right\}$-codes employs the set $\{f_r^m\}$ of all m-variate Boolean polynomials f_r^m of degree r or less. Here all 2^m code positions x form the complete m-dimensional space E_2^m. A codeword $\mathbf{c}_f$ is obtained as the sequence of binary values that a polynomial $f(x)$ takes on positions $x \in E_2^m$. It is easy to prove [5] that $\left\{ {m \atop r} \right\}$-codes have length $n = 2^m$, dimension $k = \sum_{i=0}^{r} \binom{m}{i}$ and distance $d = 2^{m-r}$.

⋆ This research was supported by NSF grant CCR-0097125.

Email addresses: dumer@ee.ucr.edu (Ilya Dumer), shabunov@ee.ucr.edu (Kirill Shabunov).

RM codes have a simple code structure, which in turn enables fast decoding procedures. Majority algorithm was first developed in [1]. The algorithm executes bounded distance decoding with complexity order of nk or less. Subsequently, it was also proven in [6] that majority decoding substantially extends the bounded-distance threshold of $d/2$. Here, given an infinite sequence of codes $A_i(n_i, d_i)$, we say that a decoding algorithm Ψ has a threshold sequence δ_i and a residual sequence $\epsilon_i \to 0$ if for $n_i \to \infty$:

• Ψ corrects all but a vanishing fraction of error patterns of weight $\delta_i(1-\epsilon_i)$ or less;

• Ψ fails to correct a nonvanishing fraction of error patterns of weight δ_i or less.

It is proven in [6] that for RM codes of *fixed order* r, majority decoding achieves the maximum possible threshold $\delta = n/2$ (here and below we omit index i) with a residual

$$\varepsilon_r^{\mathrm{maj}} = (cm2^{r-m})^{1/2^{r+1}}, \quad m \to \infty, \tag{1}$$

where c is a constant that does not depend on m and r. Note that δ exceeds 2^r times the bounded distance threshold of $d/2$. For long RM codes of *fixed rate* R, it is also proven in [6] that majority algorithm achieves a threshold

$$\delta = (d \ln d)/4. \tag{2}$$

Majority decoding can also be extended [11] for soft decision channels. In particular [11], for RM codes of fixed rate R, soft decision majority decoding gives a threshold of Euclidean weight

$$\varrho = (n/m)^{1/2^{r+1}} n^{1/2}.$$

One more efficient algorithm [7] makes use of the symmetry group of RM codes. For long RM codes $\left\{ {m \atop 2} \right\}$ of the second order, the algorithm [7] reduces the residual term $\varepsilon_2^{\mathrm{maj}}$ from (1) to the lower order of $(cm2^{r-m})^{1/4}$, where one can take any $c > \ln 4$. However, the former complexity $O(nm^2)$ of majority decoding is also increased in algorithm [7] to a nearly square order of $O(n^2 m)$. The corresponding thresholds for higher orders $r \geq 3$ are yet unknown.

Another result of [7] is related to maximum-likelihood (ML) decoding. It is shown that ML decoding of RM codes of fixed order r has a substantially lower residual term

$$\varepsilon_r^{\mathrm{ml}} = m^{r/2} n^{-1/2} (c(2^r - 1)/r!)^{1/2}, \quad m \to \infty,$$

where $c > \ln 4$.

The third technique is based on various recursive algorithms introduced in [2], [3], [4], and [10]. All these algorithms use different metrics but rely on

the *Plotkin construction* $(\mathbf{u}, \mathbf{u}+\mathbf{v})$. The construction allows to decompose RM codes $\left\{ {m \atop r} \right\}$ onto shorter codes, by taking subblocks $\mathbf{u}$ and $\mathbf{v}$ from codes $\left\{ {m-1 \atop r} \right\}$ and $\left\{ {m-1 \atop r-1} \right\}$. It is shown that this recursive structure allows to execute both encoding [2] and bounded distance decoding [4], [10] with the lowest complexity order of $n \min(r, m-r)$ known for RM codes of an arbitrary order r.

Recently, recursive algorithms have been analyzed in [12] and [13] in more detail. Some of the results are summarized in the following statement.

Theorem 1 *Long RM codes* $\left\{ {m \atop r} \right\}$ *of fixed order* r *can be decoded with linear complexity* $O(n)$ *and decoding threshold*

$$\delta = n/2, \quad \varepsilon_r = ((2r \ln m)/d)^{1/2^{r+1}}, \quad m \to \infty, \tag{3}$$

or with quasi-linear complexity $O(n \log n)$ *and decoding threshold*

$$\delta = n/2, \quad \overline{\varepsilon}_r = (cm/d)^{1/2^r}, \quad c > \ln 4, \quad m \to \infty. \tag{4}$$

Note that Theorem 1 increases decoding threshold of the recursive techniques introduced in [2] and [4] from the order of $d/2$ to $n/2$ while keeping linear decoding complexity. It also improves both the complexity and the residual of majority decoding. When compared with the algorithm of [7], this theorem reduces the quadratic complexity $O(n^2 \log n)$ to a quasi-linear complexity $O(n \log n)$ and is also extended to RM codes of any *fixed* order $r \geq 2$.

However, as mentioned in [13], the probabilistic tools utilized there do not allow one to extend the above results for an arbitrary *growing* order r or more specifically, for any nonvanishing code rate R. Therefore below we develop the new tools that allow us to accomplish this task. The main result of this paper is given in the following theorem.

Theorem 2 *Long RM codes* $\left\{ {m \atop r} \right\}$ *can be decoded with complexity order of* $(3n \log_2 n)/2$ *or less, and achieve the following thresholds and residuals*

$$\begin{aligned} &\delta = \frac{n}{2}, \qquad \varepsilon = \left(\frac{4m}{d}\right)^{1/2^r}, \quad \textit{if} \quad \frac{r}{\ln m} \to 0, \\ &\delta = \frac{d \ln d}{2}, \quad \varepsilon' = \frac{\ln(4m)}{\ln d}, \qquad \textit{if} \quad \frac{\min(r, m-r)}{\ln m} \to \infty. \end{aligned} \tag{5}$$

Thus, Theorem 2 increases $\ln d$ times the threshold $d/2$ of bounded distance decoding and also doubles that of majority decoding. Both improvements are

also achieved at a lower complexity. Our proof of Theorem 2 will be done in Sections 4, 5, and 6.

Below in Section 2 we consider recursive structure of RM codes in more detail. Here we mostly follow the description of [13]. Such a description also allows one to derive some properties of RM codes in a relatively simple way. In particular, we show that recursive structure yields generator matrices completely formed by the codewords of minimum weight d.

In Section 3, we proceed with decoding techniques and describe two different recursive algorithms Ψ_r^m and Φ_r^m introduced in [13]. The basic recursive procedure will split RM code $\left\{ {m \atop r} \right\}$ of length n into two RM codes of length $n/2$. Decoding is then relegated further to the shorter codes until we reach basic codes of order $r \leq 1$ or $r = m$. In all intermediate steps, we shall only recalculate the newly defined symbols. Here we first prove that these algorithms guarantee bounded distance decoding.

In Sections 4, 5, and 6, we proceed with more advanced analysis. For each information symbol, we relate its bit error probability to some random variable (rv). Our main goal here is to establish some partial ordering on the information bits relative to their decoding failure. This ordering will allow us to find the information bits least protected from the channel noise. In so doing, we will find the power and central moments of the corresponding rv. Here we will extend this analysis beyond the first two central moments, as opposed to [13], in which the simpler tools turned out to suffice.

2 Recursive structure of RM codes

Consider any m-variate Boolean polynomial $f = f_r^m$ and the corresponding codeword $\mathbf{c}(f)$, with symbols $f(x)$ on positions $x = (x_1, ..., x_m) \in E_2^m$. Let x_1 be the most senior digit, and x_m be the junior digit in the lexicographic order of positions $x \in E_2^m$. We decompose any polynomial f as

$$f_r^m(x_1, ..., x_m) = f_r^{m-1}(x_2, ..., x_m) + x_1 f_{r-1}^{m-1}(x_2, ..., x_m), \tag{6}$$

using the new polynomials f_r^{m-1} and f_{r-1}^{m-1}. Then we obtain the codewords $\mathbf{u} = \mathbf{c}(f_r^{m-1})$ and $\mathbf{v} = \mathbf{c}(f_{r-1}^{m-1})$ from the codes $\left\{ {m-1 \atop r} \right\}$ and $\left\{ {m-1 \atop r-1} \right\}$, respectively. Now any codeword $\mathbf{c}(f) \in \left\{ {m \atop r} \right\}$ is represented in the form $(\mathbf{u}, \mathbf{u} + \mathbf{v})$. This is the well known Plotkin construction.

By continuing this process, we obtain RM codes taken over $m - 2$ variables $x_3, ..., x_m$ and so on. Finally, we arrive at the repetition codes $\left\{ {g \atop 0} \right\}$ for any $g = 1, ..., m - r$ and full spaces $\left\{ {h \atop h} \right\}$ for any $h = 1, ..., r$. Similarly to [13],

this is schematically shown in Fig. 1.1 for RM codes of length 8. In Fig. 1.2, we consider an incomplete decomposition for codes of length 32 terminated at the biorthogonal codes and single-parity check codes.

Below $\mathbf{a}_r^m = \{a_j | j = 1, k\}$ denotes a block of k information bits a_j that encode a vector $(\mathbf{u}, \mathbf{u} + \mathbf{v})$. The above splitting also decomposes $\mathbf{a}_r^m$ into two information subblocks that encode vectors $\mathbf{u}$ and $\mathbf{v}$, respectively. When arriving at some code $\{{}_h^g\}$ in our splitting procedure, we use notation $\mathbf{a}_h^g$ for its information block. In the following steps, information subblocks are split further. Thus, any specific codeword can be encoded from the information strings assigned to the end nodes $\left\{ {}_0^g \right\}$ or $\left\{ {}_h^h \right\}$. Only one information bit is assigned to the left-end (repetition) code $\left\{ {}_0^g \right\}$, while the right-end code $\left\{ {}_h^h \right\}$ includes 2^h bits. We can use the unit $(2^h \times 2^h)$-generator matrix to encode these 2^h bits.

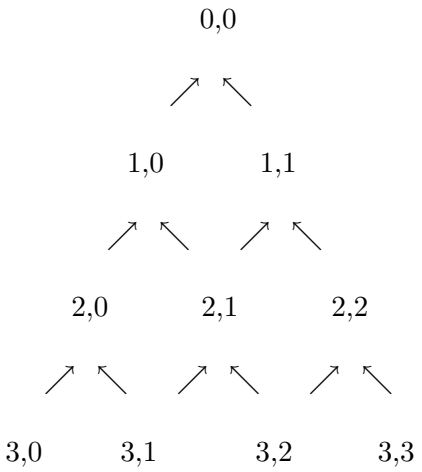

Fig. 1.1: Full decomposition

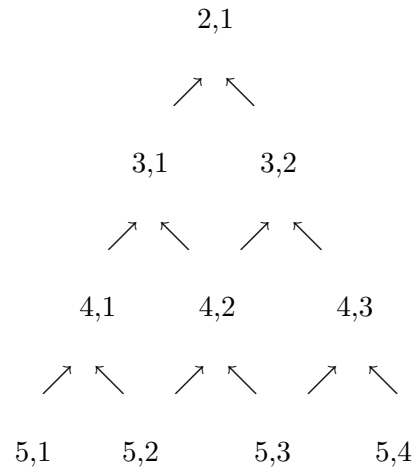

Fig. 1.2: Partial decomposition

Given any algorithm ψ, we use notation $|\psi|$ for its complexity. Let ψ_r^m denote the above encoding for code $\left\{ {}_r^m \right\}$. Following [2] and [4], we arrive at the following lemma.

Lemma 3 *RM codes* $\left\{ {}_r^m \right\}$ *can be recursively encoded with complexity*

$$|\psi_r^m| \leq n \min(r, m - r). \tag{7}$$

Now consider an information bit a_j associated with a left node $\left\{ {}_0^g \right\}$, where $g \in [1, m - r]$. We will map a_j onto a specific "binary path"

$$\xi \stackrel{\text{def}}{=} (\xi_1, ..., \xi_m)$$

of length m leading from the origin $\left\{ {m \atop r} \right\}$ to the end node $\left\{ {g \atop 0} \right\}$. To do so, we first define the senior bit

$$\xi_1 = \begin{cases} 0, \text{ if } a_j \in \mathbf{a}_{r-1}^{m-1}, \\ \\ 1, \text{ if } a_j \in \mathbf{a}_{r}^{m-1}. \end{cases}$$

Next, we take $\xi_2 = 0$ if a_j belongs to the left descendant subcode. Otherwise, $\xi_2 = 1$. For example, given $\xi_1 = 0$, we take

$$\xi_2 = \begin{cases} 0, \text{ if } a_j \in \mathbf{a}_{r-2}^{m-2}, \\ \\ 1, \text{ if } a_j \in \mathbf{a}_{r-1}^{m-2}. \end{cases}$$

Similar procedures are then repeated at the following steps and give subpath ξ^{m-g} of length $m-g$ that arrives at $\left\{ {g \atop 0} \right\}$. We then take g right-hand steps and add g ones. Thus, we obtain a full path ξ of length m that arrives at the node $\left\{ {0 \atop 0} \right\}$. Considering all g, we obtain a subset of $\binom{m-1}{r}$ paths, all of which have $m-r$ ones including the last symbol $\xi_m = 1$.

Now consider any information bit a_j associated with a right-end node $\left\{ {h \atop h} \right\}$, where $h \in [1, r]$. The same mapping procedure gives a subpath

$$\xi \stackrel{\text{def}}{=} (\xi_1, ..., \xi_{m-h})$$

of length $m-h$ that also includes $m-r$ ones and ends with $\xi_{m-h} = 1$. Note that there are 2^h information bits a_j associated with this node. Also, for the full length m, we can take $h = 0$ for any left end path ξ. Thus, for any (left or right) end node, we can define ξ as the shortest path of weight $m-r$.

Next, we proceed with the full set of the above paths. By taking all the paths of length $m-h$ with multiplicity 2^h, we obtain the overall number of information symbols

$$\sum_{h=0}^{r} \binom{m-h-1}{m-r-1} 2^h = \sum_{h=0}^{r} \binom{m}{h} = k. \tag{8}$$

From now on, Γ is the complete set of paths ξ ordered lexicographically. Given two paths of different lengths, here we compare their prefixes of the maximum common length. Also, we use notation a_ξ for any information symbol associated with a given path ξ. Finally, following [5], note that the code distance d of the original code and distances d^v and d^u of its two descendants satisfy an obvious recursion

$$d = \min(2d^u, d^v).$$

Here we also take into account that code $\left\{ {m-1 \atop r-1} \right\}$ belongs to $\left\{ {m-1 \atop r} \right\}$. Given the

original code distances 2^g and 1 of the trivial end codes $\left\{ {g \atop 0} \right\}$ and $\left\{ {h \atop h} \right\}$, we see that $\left\{ {m \atop r} \right\}$-code has distance $d = 2^{m-r}$.

Next, we prove the following useful lemma.

Lemma 4 *There exists a generator matrix G of the $\left\{ {m \atop r} \right\}$-code that consists of the codewords of minimum weight 2^{m-r}.*

Proof. Consider the generator matrix G of an RM code defined by the $(\mathbf{u}, \mathbf{u}+\mathbf{v})$-construction. Let c_ξ be a codeword in G associated with some information symbol a_ξ and its path ξ of some length $m-h$. Recall that a codeword c_ξ is obtained from some polynomial g_ξ. First, consider this polynomial g_ξ associated with a path ξ. In particular, equality $\xi_1 = 1$ holds for all information symbols a_ξ that form a codeword $\mathbf{u} \in \left\{ {m-1 \atop r} \right\}$.

Second, (6) shows that vectors $(\mathbf{u}, \mathbf{u})$ correspond to the polynomials f^{m-1}_r that exclude the variable x_1. Proceeding in the same way, we see that a path ξ has symbol $\xi_l = 1$ if a polynomial g_ξ excludes a variable x_l for any $l \le m-h$. As a result, any path ξ of length $m-h$ is associated with the polynomial

$$g_\xi = \prod_{l=1}^{m-h} x_l^{1-\xi_l}.$$

Now assume that the unit matrix is used on the end code $\left\{ {h \atop h} \right\}$. Equivalently, this requires the set of 2^h polynomials $\{f_h\}$ each of which gives a codeword c_ξ of weight 1. Thus, every codeword c_ξ is generated by some polynomial $g_\xi f_h$. We complete the proof noting that polynomial $g_\xi f_h$ gives a codeword c_ξ of weight $\mathrm{wt}(c_\xi) = 2^{\mathrm{wt}(\xi)}$. □

Remark. It is also easy to see that any polynomial f_h is defined by h Boolean variables $i_l \in \{0, 1\}$ and has the form

$$f_h = \prod_{l=m-h+1}^{m} (i_l + x_l).$$

Thus, in this minimum-weight representation, any information symbol a_ξ can be associated with the polynomial

$$\prod_{l=1}^{m-h} x_l^{1-\xi_l} \prod_{l=m-h+1}^{m} (i_l + x_l).$$

3 Two recursive algorithms

Below we use the mapping $a \Longleftrightarrow (-1)^a$ for any binary symbol a. Obviously, the binary sum $a+b$ of two symbols a, b is being mapped onto the product of their images α, β:

$$a+b \Longleftrightarrow \alpha\beta.$$

Thus, we assume that all code vectors belong to $\{1,-1\}^n$ and have the form $(\mathbf{u}, \mathbf{uv})$. Let a codeword $\mathbf{c}=(\mathbf{u}, \mathbf{uv})$ be transmitted over a binary symmetric channel with crossover probability $p<1/2$. The received block $\mathbf{y} \in \{1,-1\}^n$ consists of two halves $\mathbf{y}'$ and $\mathbf{y}''$ corrupted by noise. More generally, we will use any vector $\mathbf{y} \in \mathbb{R}^n$. We start with a basic algorithm $\Psi_{\rm rec}(\mathbf{y})$ that will be later used in recursive decoding. Our description here repeats that of [13].

Step 1. We first try to find the codeword $\mathbf{v}$ from $\left\{ {m-1 \atop r-1} \right\}$. Here we first find its "channel estimate"

$$\mathbf{y}^v = \mathbf{y}'\mathbf{y}'' \tag{9}$$

(which gives the binary sum of vectors $\mathbf{y}'$ and $\mathbf{y}''$ in the former notation). Next, we employ some decoding $\Psi(\mathbf{y}^v)$, which will be specified later. The output is some vector $\widehat{\mathbf{v}} \in \left\{ {m-1 \atop r-1} \right\}$ and its information block $\widehat{\mathbf{a}}^v$.

Step 2. We try to find the block $\mathbf{u} \in \left\{ {m-1 \atop r} \right\}$ given $\widehat{\mathbf{v}}$ from Step 1. Here we take two corrupted versions of vector $\mathbf{u}$, namely $\mathbf{y}'$ in the left half and $\mathbf{y}''\widehat{\mathbf{v}}$ in the right half. These two vectors are combined as

$$\mathbf{y}^u = (\mathbf{y}' + \mathbf{y}''\widehat{\mathbf{v}})/2. \tag{10}$$

(Note that the above addition is performed over real numbers and corresponds to the "soft-decision" majority voting.) Then we use a decoding $\Psi(\mathbf{y}^u)$ specified later. The output is some vector $\widehat{\mathbf{u}} \in \left\{ {m-1 \atop r} \right\}$ and its information block $\widehat{\mathbf{a}}^u$. So, decoding $\Psi_{\rm rec}(\mathbf{y})$ performs as follows.

Algorithm $\Psi_{\rm rec}(\mathbf{y})$.

1. Calculate vector $\mathbf{y}^v = \mathbf{y}'\mathbf{y}''$.

Find $\hat{\mathbf{v}} = \Psi(\mathbf{y}^v)$ and $\widehat{\mathbf{a}}^v$.

2. Calculate vector $\mathbf{y}^u = (\mathbf{y}' + \mathbf{y}''\hat{\mathbf{v}})/2$.

Find $\hat{\mathbf{u}} = \Psi(\mathbf{y}^u)$ and $\widehat{\mathbf{a}}^u$.

3. Output decoded components:

$\widehat{\mathbf{a}} := (\widehat{\mathbf{a}}^v \mid \widehat{\mathbf{a}}^u)$; $\quad \hat{\mathbf{c}} := (\hat{\mathbf{u}} \mid \hat{\mathbf{u}}\hat{\mathbf{v}})$.

In a more general scheme Ψ_r^m, we repeat this recursion by decomposing sub-blocks $\mathbf{y}^v$ and $\mathbf{y}^u$ further. On each intermediate step, we only recalculate the newly defined vectors $\mathbf{y}^v$ and $\mathbf{y}^u$ using (9) when decoder moves left and (10) when it goes right. Finally, we decode the recalculated vectors $\mathbf{y}^v$ and $\mathbf{y}^u$, once we reach the end nodes $\left\{ {g \atop 0} \right\}$ and $\left\{ {h \atop h} \right\}$. Given any end code C of length l and any estimate $\mathbf{z} \in \mathbb{R}^l$, we employ the (soft decision) minimum-distance (MD) decoding $\Psi(\mathbf{z}) = \hat{\mathbf{c}}$ that outputs a codeword $\hat{\mathbf{c}}$ that maximizes the inner product $(\mathbf{c}, \mathbf{z})$. The algorithm is described below.

Algorithm $\Psi_r^m(\mathbf{y})$.

1. If $0 < r < m$, perform $\Psi_{\rm rec}(\mathbf{y})$

using $\Psi(\mathbf{y}^v) = \Psi_{r-1}^{m-1}$ and $\Psi(\mathbf{y}^u) = \Psi_r^{m-1}$.

2. If $r = 0$, perform MD decoding

$\Psi(\mathbf{y}^v)$ for code $\left\{ {r \atop 0} \right\}$.

3. If $r = m$, perform MD decoding

$\Psi(\mathbf{y}^u)$ for code $\left\{ {r \atop r} \right\}$.

In the following algorithm Φ_r^m, we terminate decoding $\Psi_{\rm rec}$ at the biorthogonal codes $\left\{ {g \atop 1} \right\}$.

Algorithm $\Phi_r^m(\mathbf{y})$.

1. If $1 < r < m$, find $\Psi_{\rm rec}(\mathbf{y})$
using $\Psi(\mathbf{y}^v) = \Phi_{r-1}^{m-1}$ and $\Psi(\mathbf{y}^u) = \Phi_r^{m-1}$.

2. If $r = 1$, perform MD decoding
$\Phi(\mathbf{y}^v)$ for code $\left\{ {r \atop 1} \right\}$.

3. If $r = m$, perform MD decoding
$\Phi(\mathbf{y}^u)$ for code $\left\{ {r \atop r} \right\}$.

Thus, procedures Ψ_r^m and Φ_r^m have a recursive structure that calls itself until MD decoding is applied on the end nodes. Now the complexity estimate follows, similarly to the estimates in [4], [13].

Lemma 5 *Algorithms Ψ_r^m and Φ_r^m decode RM codes $\left\{ {m \atop r} \right\}$ with complexity*

$$|\Psi_r^m| \le 3n \min(r, m-r) + n \tag{11}$$
$$|\Phi_r^m| \le 2n \min(r, m-r) + n(m-r+1). \tag{12}$$

Remark. Note that every new recalculation (10) almost doubles the size of our original alphabet $\{\pm 1\}$. For example, starting with l consecutive estimates (10), we obtain the alphabet $\{\pm t/2^l\}$, where t runs from 0 to 2^l. Then the first "left-hand" recalculation (9) will "square" this alphabet to $\{\pm t/2^{2s}\}$, where t runs from 0 to 2^{2s}. In general, the alphabet size at any node $\left\{ {g \atop h} \right\}$ depends on a specific path connecting this node with the origin $\left\{ {m \atop r} \right\}$.

Note that both algorithms have complexity upper-bounded by the order of $(3n \log_2 n)/2$. The following simple lemma from [13] shows that recursive decoding follows lexicographic order of our paths $\xi \in \Gamma$.

Lemma 6 *For two paths ξ' and ξ'', the bit $a(\xi'')$ is decoded after $a(\xi')$ if $\xi'' > \xi'$.*

Proof. Given two paths ξ'' and ξ', let l be the first (senior) position where they disagree. If $\xi'' > \xi'$, then $\xi''_l > \xi'_l$. The latter implies that ξ' represents the left-hand step, while ξ' does the right one. Correspondingly, ξ' proceeds first. $\square$

Let

$$\xi = (\xi_1, ..., \xi_l), \quad \underline{\xi} = (\xi_1, ..., \xi_{l-1}), \quad l \in [1, m-h]$$

denote any subpath of some length l and its prefix $\underline{\xi}$, so that $\xi = (\underline{\xi}, \xi_l)$. Below we show that the above algorithms admit bounded distance decoding. Our proof is similar to that used in [4] for a different recursive algorithm.

Lemma 7 *Both algorithms Ψ_r^m and Φ_r^m perform bounded distance decoding of RM codes $\left\{ {m \atop r} \right\}$.*

Proof. Without loss of generality, let the (former all-zero) codeword $\mathbf{c} = \mathbf{1}$ be transmitted. For any received vector $\mathbf{y}$, let $d(\xi)$ and $\mathbf{y}(\xi)$ denote the code distance and the vector obtained on any (possibly, incomplete) subpath ξ. Let $\mathbf{z}(\xi) = \mathbf{1} - \mathbf{y}(\xi)$ be the corresponding "discrepancy" vector [1]. Given ξ, let $I = I(\xi)$ be any subset of positions on $\mathbf{y}(\xi)$ such that $|I| \geq d(\xi)$. Obviously, decodings $\Psi(\cdot)$ and $\Phi(\cdot)$ are correct on some *end* path ξ if for any I,

$$\sum_{i \in I(\xi)} y_i(\xi) > 0, \tag{13}$$

or, equivalently,

$$\sum_{i \in I(\xi)} z_i(\xi) < |I|.$$

Below, we replace the latter by a stronger inequality

$$\sum_{i \in I(\xi)} z_i(\xi) < d(\xi). \tag{14}$$

In bounded-distance decoding, the original block $\mathbf{y}$ has fewer than $d/2$ errors and satisfies (14). Next, we prove that (14) always holds on a subpath $\xi = (\underline{\xi}, \xi_l)$ if it holds on the preceding block $\mathbf{y}(\underline{\xi})$. First, we take $\xi_l = 0$ and consider the corresponding subblock $\mathbf{y}^v(\underline{\xi})$. Here $d(\xi) = d(\underline{\xi})$. According to (9), $\mathbf{y}^v(\underline{\xi})$ has discrepancies

$$z_i^v = z_i' + z_i'' - z_i' z_i'',$$

where z_i' and z_i'' are the discrepancies on positions i' and i'' of both halves. Consider a subset $I(\xi)$ of size $|I| \geq d(\underline{\xi})$ on $\mathbf{y}^v$ and let I' and I'' be the corresponding subsets on vectors $\mathbf{y}'(\underline{\xi})$ and $\mathbf{y}''(\underline{\xi})$. Then inequality (14) holds on $\mathbf{y}^v$, since

$$\sum_{i \in I} z_i^v \leq \sum_{i \in I'} z_i' + \sum_{i \in I''} z_i'' = \sum_{i \in I' \cup I''} z_i < d(\xi).$$

Similarly, for $\xi_l = 1$, we have distance $d(\xi) = d(\underline{\xi})/2$. According to (10), $\mathbf{y}^u$ gives discrepancies

$$z_i^u = (z_i' + z_i'')/2.$$

[1] Any correct symbol y_i is replaced by $-y_i$ if an error occurs in position i. Therefore z is not an error vector.

Then for any subset $I(\xi)$ on $\mathbf{y}^u$, such that $|I(\xi)| \geq d(\underline{\xi})/2$, we have(14), since

$$\sum_{i \in I} z_i^u = \sum_{i \in I'} z_i'/2 + \sum_{i \in I''} z_i''/2 = \sum_{i \in I' \cup I''} z_i/2 < d(\underline{\xi})/2.$$

□

4 Preliminary probabilistic analysis of recursive algorithms

4.1 Recalculation of the outputs

In the following two sections, we consider the algorithm Ψ_r^m. In Section 6, similar analysis will be applied to the algorithm Φ_r^m. For both algorithms, it will turn out that the output bit error rate (BER) significantly varies on different paths, including those that lead to the same node. Therefore our main goal is to define the most error-prone paths ξ and estimate the output BER for the corresponding information symbols $a(\xi)$.

Without loss of generality, below we always assume that the codeword $\mathbf{c} = \mathbf{1}$ is transmitted. On any node ξ, the algorithm $\Psi_r^m(\mathbf{y})$ gives some vector $\mathbf{y}(\xi)$ of length 2^{m-l}. Here the previous estimate $\mathbf{y}(\underline{\xi})$ is first split into the halves $\mathbf{y}'(\underline{\xi})$ and $\mathbf{y}''(\underline{\xi})$. These halves are multiplied according to (9), if $\xi_l = 0$. By contrast, we use recalculation (10) at this node, if $\xi_l = 1$. Note that (10) also includes the estimate $\widehat{\mathbf{v}} = \widehat{\mathbf{v}}(\xi')$, obtained on the preceding node $\xi' = (\underline{\xi}, 0)$.

Below we show that decoding analysis on any path ξ can be performed given that all preceding decodings are correct. Note that in this case we can take $\mathbf{v}(\xi')= \mathbf{1}$ for all subpaths $\xi' < \xi$, and rewrite equations (9) and (10) as follows

$$\mathbf{y}^v = \mathbf{y}'\mathbf{y}'', \quad \mathbf{y}^u = (\mathbf{y}' + \mathbf{y}'')/2, \tag{15}$$

This also allows us to recurrently recalculate the intermediate outputs in the following way:

$$\mathbf{y}(\xi) \stackrel{\text{def}}{=} \begin{cases} \mathbf{y}'(\underline{\xi}) \cdot \mathbf{y}''(\underline{\xi}), & \text{if } \xi_l = 0, \\ \mathbf{y}'(\underline{\xi})/2 + \mathbf{y}''(\underline{\xi})/2, & \text{if } \xi_l = 1. \end{cases} \tag{16}$$

Our next step is to redefine the decoding results obtained on the end paths ξ.

Lemma 8 *For any end path ξ, the algorithm decodes the output $y(\xi)$ into the information block*

$$\widehat{\mathbf{a}}(\xi) = \mathit{sign}(\mathbf{y}(\xi)). \tag{17}$$

Proof. Consider a right-end path ξ that ends at some code $\left\{ {h \atop h} \right\}$. The corresponding output $\mathbf{y}(\xi)$ consists of 2^h symbols $y(\xi)$. In this case, MD decoding Ψ_h^h makes bit-by-bit decisions (17). Consider a left-end path ξ that is decoded on a repetition code $\left\{ {g \atop 0} \right\}$. Let $\underline{\xi}$ denote its prefix that enters this code and $\mathbf{y}(\underline{\xi})$ be the corresponding vector of length 2^g. Note that the symbol $y(\xi)$ obtained at the end node $\left\{ {0 \atop 0} \right\}$, is

$$y(\xi) = \sum_{i=1}^{2^g} y_i(\underline{\xi})/2^g. \tag{18}$$

Thus, decoding Ψ_0^g takes the sign of $y(\xi)$ by the same decoding rule (17). $\square$

According to our decoding rule (17), the error event $\{\widehat{a}(\xi) = -1\}$ has probability

$$P(\xi) = \Pr\{y(\xi) < 0\}.$$

Let 0^t or 1^t denote the subpaths that consist of t zeros and ones, respectively. Recall that all paths ξ are ordered lexicographically, and the first path is

$$\xi_* = 0^r, 1^{m-r}. \tag{19}$$

To use recurrent calculations (16), we wish to consider the *conditional* probability given that all preceding decodings $\mathbf{v}(\xi')$ are correct for all paths $\xi' < \xi$:

$$p(\xi) = \Pr\{y(\xi) < 0 \,|\, \mathbf{v}(\xi') = \mathbf{1} \text{ for all } \xi' < \xi\}. \tag{20}$$

Thus, we first need to estimate unconditional probabilities $P(\xi)$ using their conditional counterparts $p(\xi)$.

Lemma 9 *For any path $\xi \in \Gamma$, its bit error rate $P(\xi)$ satisfies inequality*

$$P(\xi) \le \sum_{\xi' \le \xi} p(\xi'). \tag{21}$$

Block error probability P satisfies inequalities

$$p(\xi_*) \le P \le \sum_{\xi'} p(\xi'). \tag{22}$$

Sketch of the proof. This simple statement is formally proven in [13]. Informally, the upper bound (21) assumes incorrect decoding of an information symbol $a(\xi)$ whenever failure occurs on any previous step $\xi' \le \xi$. Then the corresponding probabilities are added together in the union bound. Similarly, the upper bound in (22) sums up probabilities of decoding failures on all k paths ξ'. Also, the lower bound implies that the block is always incorrect given decoding failure on the first step ξ_*. $\square$

4.2 Asymptotic setting

In our setting, the received vector $\mathbf{y}(\xi) = \mathbf{y}$ consists of n independent identically distributed (i.i.d.) random variables (rv) $y_i = y$, each of which has probability distribution

$$\Pr\{y_i\} = \begin{cases} 1-p, \text{ if } y_i = +1, \\ p, \qquad \text{if } y_i = -1, \end{cases} \tag{23}$$

Our next goal is to estimate the error probability

$$p(\xi) = \Pr\{y(\xi) < 0\} \tag{24}$$

where $y(\xi)$ is the rv, which is obtained on some path ξ after performing recurrent recalculations (16). Note that these transformations (16) can be mixed in an arbitrary (irregular) order depending on a particular path ξ. In [13], our approach was to find the first two moments of variables $y(\xi)$. However, as mentioned above, this approach is sufficient only for codes of fixed order r. Therefore below we estimate $p(\xi)$ using high-order power and central moments of $y(\xi)$.

1. First, note that the blocks $\mathbf{y}'$ and $\mathbf{y}''$ used in (15) always include different channel bits y_i. Consequently, their descendants $\mathbf{y}'(\underline{\xi})$ and $\mathbf{y}''(\underline{\xi})$ used in (16) are also obtained from different channel bits. These bits are combined in the same operations. Therefore all symbols $y_i(\xi)$ of the vector $\mathbf{y}(\xi)$ are i.i.d. rv. This simple observation also allows to simplify our notation and remove indices i. Namely, given any subpath ξ of some length l, $y(\xi)$ will denote one of the random variables $y_i(\xi)$ obtained on the path ξ by recalculations (16).

2. Let $a(\xi) = \mathsf{E}\ y(\xi)$ denote the expectation of any rv $y(\xi)$. Below we will study the normalized random variables

$$z(\xi) = y(\xi)/a(\xi), \tag{25}$$

with expected values 1. Namely, given some positive even integer s, we wish to estimate the s-th power and central moments

$$E_s(\xi) \stackrel{\text{def}}{=} \mathsf{E}\ \{z^s(\xi)\}, \quad D_s(\xi) \stackrel{\text{def}}{=} \mathsf{E}\ \{(z(\xi)-1)^s\}. \tag{26}$$

3. Given these moments (26), we will prove Theorem 2 using Chebyshev inequality, which gives the upper bound

$$\begin{aligned} p(\xi) &= \Pr\{z(\xi) < 0\} \\ &\le \Pr\{(z(\xi)-1)^s > 1\} \le D_s(\xi). \end{aligned} \tag{27}$$

4.3 Recalculation of the power and central moments

Given any subpath ξ, our next goal is to obtain the recurrent formulas for the moments $E_s(\cdot)$ on the two descendant paths $(\xi, 0)$ and $(\xi, 1)$. To do so, we first define the recursions for the rv $z(\xi)$.

Lemma 10 *On any subpath $\xi = (\underline{\xi}, \xi_l)$ of length l, random variables $z(\xi)$ satisfy recursions:*

$$z(\xi) = \begin{cases} z'(\underline{\xi}) \cdot z''(\underline{\xi}), & \text{if } \xi_l = 0, \\ z'(\underline{\xi})/2 + z''(\underline{\xi})/2, & \text{if } \xi_l = 1, \end{cases} \tag{28}$$

Proof. Obviously, for any $\xi = (\underline{\xi}, \xi_l)$, the means $a(\xi)$ satisfy the recursion

$$a(\xi) = \begin{cases} a^2(\underline{\xi}), & \text{if } \xi_l = 0, \\ a(\underline{\xi}), & \text{if } \xi_l = 1, \end{cases} \tag{29}$$

which follows from (16). Here we simply replace all three rv used in (16) by their expectations. Also, we use the fact that vectors $\mathbf{y}'(\underline{\xi})$ and $\mathbf{y}''(\underline{\xi})$ are independent and have the same expectation $\mathsf{E}(y(\underline{\xi}))$ on their symbols. Then recursion (28) directly follows from equalities (16) for $y(\xi)$ and (29) for $a(\xi)$. □

By replacing rv in (28) with their moments $E_s(\xi)$, we obtain the recursion

$$E_s(\xi) = \begin{cases} E_s^2(\underline{\xi}), & \text{if } \xi_l = 0, \\ 2^{-s} \sum_{i=0}^{s} \binom{s}{i} E_i(\underline{\xi}) E_{s-i}(\underline{\xi}), & \text{if } \xi_l = 1, \end{cases} \tag{30}$$

Note that equalities (30) yield rather cumbersome iterations even after very few steps. Below, we will simplify the problem in two different ways. First, we will find the *weakest* paths ξ that give the biggest central moments $D_s(\xi)$. It will turn out that the same weakest paths can be considered for all $s > 1$. Second, we will estimate $D_s(\xi)$ for $m \to \infty$ using the second moments $D_2(\xi)$.

5 The weakest paths

In (31) below, we consider two subpaths ξ_- and ξ_+ of *the same Hamming weight that differ only in the last two positions*:

$$\begin{cases} \xi_- = (\xi_1, ..., \xi_{l-2}, 0, 1), & \\ & \\ \xi_+ = (\xi_1, ..., \xi_{l-2}, 1, 0). & \end{cases} \qquad \begin{array}{c} \nearrow\nwarrow \\ 0 \;\nwarrow\!\!\searrow\!\!\nearrow\; 1 \\ \xi_1, ..., \xi_{l-2} \end{array} \tag{31}$$

In other words, these paths diverge and submerge on the two last steps if placed on the triangle of Fig. 1.1. We say that ξ_- and ξ_+ are the left-loop and right-loop paths, correspondingly. We also say that the two descending paths $\xi_{\text{left}} = (\xi_-, \overline{\xi})$ and $\xi_{\text{right}} = (\xi_+, \overline{\xi})$ with the same suffix $\overline{\xi}$ are the neighbors. The following theorem is central to our analysis of the weakest paths.

Theorem 11 *For any even s, any two neighbors ξ_{left} and ξ_{right} satisfy inequality*

$$D_s(\xi_{left}) \geq D_s(\xi_{right}). \tag{32}$$

Proof. The proof consists of two parts.

1. We first prove this property for the paths ξ_- and ξ_+. Suppose that the original subpath $\xi_1, ...\xi_{l-2}$ in (31) outputs 4 different i.i.d. rv z_1, z_2, z_3, and z_4. It is readily verified that ξ_- and ξ_+ have the outputs

$$\begin{aligned} z(\xi_-) &= z_1 z_2/2 + z_3 z_4/2, \\ z(\xi_+) &= (z_1 + z_2)(z_3 + z_4)/4. \end{aligned}$$

Obviously, we keep the same moment $D_s(\xi_-)$, by considering the new rv

$$Z(\xi_-) = z_1 z_3/2 + z_2 z_4/2.$$

This can be rewritten as

$$Z(\xi_-) = z(\xi_+) + \widetilde{z}, \tag{33}$$

where

$$\widetilde{z} = (z_1 - z_2)(z_3 - z_4)/4.$$

Note that $\widetilde{z}$ is *symmetric* conditioned on any value of $z(\xi_+)$. The latter definition means that we have the equality

$$\Pr\{\widetilde{z} \,|\, z(\xi_+)\} = \Pr\{-\widetilde{z} \,|\, z(\xi_+)\}. \tag{34}$$

Indeed, $z_1 - z_2$ has symmetric distribution given the sum $z_1 + z_2$ (though these two variables are obviously dependent). The same fact holds for $z_3 - z_4$ given

$z_3 + z_4$. Therefore the *conditional* moments of the rv $\widetilde{z}$ satisfy the following:

$$E_{2i+1}(\widetilde{z} \,|\, z(\xi_+)) = 0, \quad E_{2i}(\widetilde{z} \,|\, z(\xi_+)) > 0, \quad i = 1, 2, ...$$

We can also rewrite (33) for the two unbiased rv $Z(\xi_-) - 1$ and $z(\xi_+) - 1$ and consider their power expansion:

$$(Z(\xi_-) - 1)^s = (z(\xi_+) - 1 + \widetilde{z})^s. \tag{35}$$

Then we can take expectations of both sides in (35) and obtain the following:

$$D_s(\xi_-) = D_s(\xi_+) + \mathsf{E} \sum_{i=0}^{s-1} \binom{s}{i} (z(\xi_+) - 1)^i \, E_{s-i}(\widetilde{z} \,|\, z(\xi_+)) > D_s(\xi_+).$$

The last inequality follows from the two facts. First, for odd i, we can remove all the summands from the latter sum since $E_{s-i}(\widetilde{z} \,|\, z(\xi_+)) = 0$. Second, the remaining summands include only even moments $D_i(\xi_+)$ and $E_{s-i}(\widetilde{z} \,|\, z(\xi_+))$, which are both positive.

2. Next, we prove general property (32) for arbitrary neighbors ξ' and ξ''. Note that Part 1 of the proof only used the fact that $\widetilde{z}$ is a symmetric rv which satisfied condition (34) in representation (33). Obviously, it suffices to prove that this property holds for the two immediate suffixes $\overline{\xi} = 0$ and $\overline{\xi} = 1$. This directly follows from (28) and (33). Indeed, for $\overline{\xi} = 1$ we have the following equalities:

$$\begin{aligned} z(\xi_{\text{left}}) &= z'(\xi_-) + z''(\xi_-) = z'(\xi_+) + z''(\xi_+) + \widetilde{z}' + \widetilde{z}'' \\ &= z(\xi_{\text{right}}) + \widetilde{z}' + \widetilde{z}''. \end{aligned}$$

Note that $\widetilde{z}' + \widetilde{z}''$ is a symmetric rv. Similarly, for $\overline{\xi} = 0$, we have equalities

$$\begin{aligned} z(\xi_{\text{left}}) &= z'(\xi_-) \cdot z''(\xi_-) \\ &= z(\xi_{\text{right}}) + \widetilde{z}' \cdot z''(\xi_+) + \widetilde{z}'' \cdot z'(\xi_+) + \widetilde{z}' \cdot \widetilde{z}''. \end{aligned}$$

It is easy to verify that the last three summands again represent a symmetric rv for any given value of the product $z(\xi_{\text{right}}) = z'(\xi_+)z''(\xi_+)$. Thus, both descendant subpaths also satisfy conditions (33) and (34). □

Remark. For all $s > 1$, a more general inequality reads $|D_s(\xi_-)| \geq |D_s(\xi_+)|$. This can be obtained using the fact that all random variables $\overline{z}(\xi)$ have negative odd moments.

Now we see that any path ξ becomes weaker if a permutation $(1, 0) \Rightarrow (0, 1)$ is performed on two adjacent symbols 0 and 1. Given a subset of paths $I \subseteq \Gamma$, we now say that $\xi_*(I)$ is the weakest path in I if

$$D_s(\xi_*(I)) = \max_{\xi \in I} D_s(\xi).$$

In particular, let the subset Γ_0^g include all left-end paths ξ, which pass through the node $\left\{ {g \atop 0} \right\}$ and let Γ_h^h be the subset of the right-end paths that pass through the node $\left\{ {h \atop h} \right\}$. Then we have the following corollary.

Corollary 12

1. The first (leftmost) path $\xi_* = 0^r, 1^{m-r}$ *from (19) is the weakest path on the entire set* Γ.

2. For any $g \in [1, m-r-1]$, *the weakest path on the subset* Γ_0^g *is its leftmost path*

$$\xi_0^g = 0^{r-1}, 1^{m-r-g}, 0, 1^g. \tag{36}$$

3. For any $h \in [1, r]$, *the weakest path on the subset* Γ_h^h *is its leftmost path*

$$\xi_h^h = 0^{r-h}, 1^{m-r}.$$

Proof. All left-end paths have the same length m. Correspondingly, we only need to prove that both ξ_* and ξ_0^g are the leftmost paths on Γ and Γ^g. Recall that Γ includes all paths of weight $m-r$ or more. Then (19) is the leftmost path on Γ since all r zeros form its prefix. Next, recall that each path $\xi \in \Gamma_0^g$ ends with zero and g ones. Also, each left-end path ξ has r zeros. Thus, ξ_0^g has the leftmost prefix in Γ_0^g, with $r-1$ zeros preceding $m-r-g$ ones.

Similarly, all right-end paths ξ ending at the node $\left\{ {h \atop h} \right\}$ have the same length $m-h$. Here ξ_h^h is the leftmost path. □

6 Decoding Thresholds

6.1 Algorithm Ψ_r^m

Our next goal is to estimate the parameter $D_s(\xi_*)$ for the weakest path ξ_*. More generally, below we also consider the moments $D_s(\xi_0^g)$, where $g \leq m-r$. Then ξ_* is a special case of ξ_0^g with $g = m-r$. As we noted before, subsequent recalculations performed for $D_s(\xi_0^g)$ are rather cumbersome. Therefore we use a different approach. Namely, let $m \to \infty$ and

$$(m-r)/\ln m \to \infty. \tag{37}$$

Consider any path $\xi = (\underline{\xi}, 1^g)$, which ends with g (or more) ones. In this case, any rv $z(\xi)$ is the sum of 2^g i.i.d. rv $z(\underline{\xi}_i)$:

$$z(\xi) = \sum_{i=1}^{2^g} z(\underline{\xi}_i).$$

Therefore $z(\xi)$ has pdf that tends to the Gaussian distribution $\mathcal{N}(1, D_2(\xi))$. Now suppose that the second moment $D_2(\xi)$ is given. We then use the following approximation

$$D_s(\xi) \sim (s-1)!! \cdot (D_2(\xi))^{s/2}, \quad m \to \infty, \tag{38}$$

valid for the Gaussian rv. Here we use the fact [9] that approximation (38) is tight for the sum of 2^g i.i.d. rv if $s \le 2^{g/6}$. In turn, the latter restriction holds true given our restriction (37).

Summarizing, for any path ξ, we invoke Theorem 11 and upper-bound $D_s(\xi)$ by $D_s(\xi_*)$. Thus, we need to find $D_2(\xi_*)$ and then use approximation (38). Below, we extensively use the parameter

$$\varepsilon = 1 - 2p.$$

Lemma 13 *The weakest paths ξ_*, ξ_0^g, and ξ_h^h yield the second moments*

$$D_2(\xi_*) = 2^{-(m-r)}(\varepsilon^{-2^{r+1}} - 1), \tag{39}$$
$$D_2(\xi_0^g) = 2^{-g}\{[(\varepsilon^{-2^r} - 1)2^{-(m-r-g)} + 1]^2 - 1\}, \tag{40}$$
$$D_2(\xi_h^h) = 2^{-(m-r)}(\varepsilon^{-2^{r-h+1}} - 1). \tag{41}$$

Proof. Recall that the original channel outputs y_i have the means $\mathsf{E}y_i = \varepsilon$, in which case rv $z_i = y_i/\varepsilon$ give the moments

$$D_2 = \varepsilon^{-2} - 1.$$

Second, note that for any path $\xi = (\underline{\xi}, \xi_l)$, recursion (28) shows that the moments $D_2(\xi)$ satisfy the recursions

$$D_2(\xi) + 1 = (D_2(\underline{\xi}) + 1)^2, \quad \text{if } \xi_l = 0, \tag{42}$$

$$D_2(\xi) = D_2(\underline{\xi})/2, \quad \text{if } \xi_l = 1. \tag{43}$$

Given the prefix $\underline{\xi} = 0^r$, equality (43) gives

$$D_2(\underline{\xi}) = \varepsilon^{-2^{r+1}} - 1.$$

Then we proceed with the suffix 1^{m-r} on the path $\xi_* = \underline{\xi}, 1^{m-r}$. Here equality (42) gives (39). Equality (41) is almost identical.

Finally, another prefix $\underline{\xi} = 0^{r-1}1^{m-r-g}0$ gives

$$D_2(\underline{\xi}) = (\varepsilon^{-2^r} - 1)2^{-(m-r-g)}.$$

Then (42) gives the expression in braces in (40). The final step in (40) is obtained by adding the suffix 1^g. □

Theorem 14 *For long RM codes* $\left\{ {m \atop r} \right\}$ *that satisfy restriction (37), algorithm* Ψ_r^m *has complexity order bounded by* $3n \min(r, m-r) + n$, *and*

- *corrects all but a vanishing fraction of errors of weight bounded by*

$$n(1-\theta)/2, \tag{44}$$

- *fails on a nonvanishing fraction of errors of weight*

$$n(1-\theta')/2$$

or less, where

$$\theta = (2m/d)^{1/2^{r+1}}, \quad \theta' = (1/d)^{1/2^{r+1}}. \tag{45}$$

Proof. The proof consists of 3 parts.

1. Consider a binary channel with crossover probability $p = (1-\theta)/2$. Our first goal is to prove that algorithm Ψ_r^m gives a vanishing block error probability on this channel as $m \to \infty$. Indeed, we substitute $\varepsilon = \theta$ in (39) and obtain equality

$$D_2(\xi_*) = (2m)^{-1} - 2^{r-m}. \tag{46}$$

For $m \to \infty$, equalities (38) and (46) give the moment

$$D_{2m}(\xi_*) \sim (2m-1)!!(2m)^{-m} \lesssim e^{-m}.$$

Here we use the fact that $(2m-1)!!$ is upper-bounded by the order of $(2m/e)^m$. Now we see that any path ξ satisfies inequality

$$p(\xi) < D_{2m}(\xi_*) \lesssim e^{-m}. \tag{47}$$

The output block error probability P of the algorithm Ψ_r^m has the order $P \leq k \max p(\xi)$. Here the number of information symbols (paths) k has the order at most 2^m. This gives a vanishing probability

$$P \lesssim (e/2)^{-m}. \tag{48}$$

2. Next, note that the error patterns of weight pn or less occur with a nonvanishing probability on the channel with crossover probability p. Indeed,

$$\sum_{i=0}^{pn} \binom{n}{i} p^i (1-p)^{n-i} \to 1/2.$$

Due to inequality (48), only a vanishing fraction of error patterns of weight pn or less is left uncorrected for any $p \le (1-\theta)/2$. Therefore, the threshold δ is lower bounded by $n(1-\theta)/2$.

3. Now let $p' = n(1-\theta')/2$. Then (39) shows that $D_2(\xi_*) = 1-2^{r-m}$. Recall that rv $z(\xi_*)$ has pdf that tends to the Gaussian distribution $\mathcal{N}(1, D_2(\xi_*))$.Then we use equalities in (27) to obtain the following estimate:

$$p(\xi_*) = \Pr\{z(\xi_*) < 0\} \to F(-1), \quad m \to \infty,$$

where $F(\cdot)$ is the cumulative probability function for Gaussian distribution. Thus, Ψ_r^m fails to correct nonvanishing fraction $F(-1)$ of errors of weight $n(1-\theta')/2$ or less. □

Corollary 15 *For long RM codes* $\left\{ {m \atop r} \right\}$, *algorithm* Ψ_r^m *has complexity order bounded by* $3n \min(r, m-r) + n$ *and achieves the following thresholds and residuals*

$$\begin{aligned} &\delta = \frac{n}{2}, \qquad \varepsilon = \left(\frac{2m}{d}\right)^{1/2^{r+1}}, \quad \textit{if} \quad \frac{r}{\ln m} \to 0, \\ &\delta = \frac{d \ln d}{4}, \quad \varepsilon' = \frac{\ln(2m)}{\ln d}, \qquad \textit{if} \quad \frac{\min(r, m-r)}{\ln m} \to \infty. \end{aligned} \tag{49}$$

Proof. It is readily verified from (45) that both θ' and θ vanish if $r/\ln m \to 0$. Therefore this case gives the threshold $\delta = n/2$ in (49). By contrast, $\theta' \to d(\ln d)/4$ and $\theta \to d(\ln d - \ln(2m))/4$ if $r/\ln m \to \infty$. This gives the second line in (49). □

Remarks.

1. For majority decoding, the results of [6] are similar to the estimates (49) of recursive decoding. This is due to the fact that both algorithms process the weakest path ξ_* in a similar way. Indeed, both algorithms first estimate the product of 2^r channel symbols. In the second step, the majority estimate (18) is taken over all 2^{m-r} different estimates.

2. A substantial difference between the two algorithms is that any other path ξ is processed in recursive decoding Ψ_r^m using the previous estimates. Because

of this, the algorithm outperforms majority decoding in both the complexity and BER $p(\xi)$ for any $\xi \neq \xi_*$. In the next section, we will see that recursive decoding is further enhanced by using the algorithm Φ_r^m.

6.2 Algorithm Φ_r^m

To proceed with a proof of Theorem 2, we summarize the similarities and differences between the algorithms Φ_r^m and Ψ_r^m that will be used below.

1. Let $\varrho = \underline{\varrho}, 1^g$ be any left-end subpath that has prefix $\underline{\varrho}$ and arrives at some biorthogonal code $\left\{ {g+1 \atop 1} \right\}$ of length $n = 2^{g+1}$. Let c_t^g be the tth codeword of $\left\{ {g+1 \atop 1} \right\}$. Here $t = 1, ..., l$, and $l = 2^{g+2}$. Below I_t^g denotes the support of the codeword c_t^g, which includes all positions with symbols -1. Also, we order the codewords in such a way that

$$c_1^g = 1^n, \quad c_l^g = -c_1^g.$$

By contrast, any right-end subpath ϱ ends at some code $\left\{ {h \atop h} \right\}$ as before. Obviously, the received rv y_i and their recalculations $y(\varrho)$ are performed in (16) similarly in both algorithms Φ_r^m and Ψ_r^m. The same holds for rv $z(\varrho)$ recalculated in (28).

2. Let the all-one codeword $\mathbf{c} = 1^n$ be transmitted and let $\mathbf{y}(\underline{\varrho})$ be the corresponding vector obtained on prefix $\underline{\varrho}$. Then $\mathbf{y}(\underline{\varrho})$ is correctly decoded into the vector c_1^g of the code $\left\{ {g+1 \atop 1} \right\}$ if and only if the event

$$\Omega(\varrho) = \{\mathbf{y} : \sum_{i \in I_t^g} y_i(\underline{\varrho}) > 0, \quad t = 2, ..., l\} \tag{50}$$

takes place. Here $y_i(\underline{\varrho})$ are i.i.d. rv, $|I_t| = 2^g$ for all $t = 2, ..., l-1$, and $|I_l| = 2^{g+1}$.

3. We proceed with the paths ϱ using the arguments of Lemma 9. Namely, the probability of decoding failure on the path ϱ is

$$P(\varrho) = \Pr\{\overline{\Omega}(\varrho)\}.$$

Then we consider the two events

$$A(\varrho) = \bigcap_{\varrho' \leq \varrho} \Omega(\varrho'), \quad B(\varrho) = \bigcap_{\varrho' < \varrho} \Omega(\varrho')$$

Error probability $p(\varrho)$ is again replaced by its conditional counterpart

$$p(\varrho) = \Pr\{\overline{A}(\varrho) \,|\, B(\varrho)\}.$$

Lemma 9 then carries over to the newly defined probabilities $p(\varrho)$. Now we can proceed with the main theorem of this section.

Theorem 16 *Long RM codes* $\left\{ \begin{smallmatrix} m \\ r \end{smallmatrix} \right\}$ *that satisfy restriction (37) can be decoded with complexity order of* $(3n \log_2 n)/2$ *or less, and*

- *correct all but a vanishing fraction of errors of weight up to*

$$n(1-\theta)/2,$$

- *fail on a nonvanishing fraction of errors of weight*

$$n(1-\theta')/2$$

or less, where

$$\theta = (4m/d)^{1/2^r}, \quad \theta' = (1/d)^{1/2^r}. \tag{51}$$

Proof . We take any path $\varrho = (\underline{\varrho}, 1^g)$. Also, all rv $y_i(\underline{\varrho})$ in (50) are considered given correct results c_1^g on all previous paths. Then we can use all intermediate recalculations (30) without any changes. For any vector $\mathbf{y}(\underline{\varrho})$, consider any subset I of 2^g positions and the sum

$$y_I(\varrho) = \sum_{i \in I} y_i(\underline{\varrho}).$$

Here $y_i(\underline{\varrho})$ form 2^g i.i.d. rv. Thus, the sum $y_I(\varrho)$ has the same pdf for any I. In turn, this allows us to remove index I, and use the common notation $y_I(\underline{\varrho}) \equiv y(\underline{\varrho})$. This gives the union bound in the following form:

$$p(\varrho) \le \sum_{t=2}^{l} \Pr\{\sum_{i \in I_t^g} y_i(\underline{\varrho}) \le 0\} \le l \Pr\{y(\underline{\varrho}) \le 0\} \tag{52}$$

Also, the probability $\Pr\{y(\underline{\varrho}) \le 0\}$ of incorrect decoding into any codeword c_t^g, $t \ne l$, gives the lower bound:

$$p(\varrho) > \Pr\{y(\underline{\varrho}) \le 0\}.$$

Next, we replace rv $y(\underline{\varrho})$ by normalized rv $z(\underline{\varrho})$ similarly to (25) and apply estimates (27) using parameter $D_s(\varrho)$:

$$p(\varrho) \le l D_s(\varrho).$$

Now we can invoke Theorem 11 and Lemma 13. In particular, the leftmost paths

$$\begin{aligned} \varrho_* &= (0^{r-1}, 1^{m-r}), \\ \varrho_1^g &= (0^{r-2}, 1^{m-r-g}, 0, 1^g). \end{aligned} \tag{53}$$

are the weakest paths on the entire set and on the node $\left\{ {g+1 \atop 1} \right\}$, respectively. Similarly to (39) and (40), we then find

$$D_2(\varrho_*) = 2^{-(m-r)}(\varepsilon^{-2^r} - 1), \tag{54}$$

$$D_2(\varrho_1^g) = 2^{-g}\{((\varepsilon^{-2^{r-1}} - 1)2^{-(m-r-g)} + 1)^2 - 1\}. \tag{55}$$

Consider a channel with crossover probability $p = (1-\theta)/2$, where θ is defined in (51). Direct substitution $\varepsilon = \theta$ in (54) gives

$$D_2(\varrho_*) = (4m)^{-1} - 2^{r-m}. \tag{56}$$

Then we use equalities (38) and (46) to obtain the moment

$$D_{2m}(\varrho_*) \sim (2m-1)!! \cdot (4m)^{-m} \lesssim (2e)^{-m}. \tag{57}$$

By using (52) for any $l = 2^{g+2} \le 2^{m+1}$, we obtain inequality

$$p(\varrho) < lD_{2m}(\varrho_*) \lesssim 2e^{-m}.$$

Thus, the output block error probability P of the algorithm Φ_r^m satisfies inequalities

$$P \le k \max p(\varrho) \lesssim 2(e/2)^{-m}. \tag{58}$$

Now we see that (58) is similar to the former estimate (48) from Theorem 14 and also gives vanishing block error probability. In this case, we can entirely repeat the proof of Theorem 14. □

Now the proof of Theorem 2 follows from Theorem 16 in the same way that is used to prove Corollary 15.

Remark. Note that the above restriction (37) on the maximum order of RM codes is essential. In particular, consider very high orders $r \ge m - \log_2 m$, which give inequality $d \le m$. Then Theorem 16 becomes invalid, giving the residual $\varepsilon' > 1$ in (5).

7 Concluding remarks

In this paper, we found the decoding thresholds achieved by recursive algorithms for general RM codes. Our calculations include three important steps.

First, decoding is being separated into different paths ξ. Each path is associated with a specific information bit and yields one rv. We also establish a partial order on all paths ξ related to their (high-order) moments D_s. The step is being completed by founding the weakest paths ξ_0^g.

In the second step, we calculate the moments D_s. This is done using the second-order moments D_2 and applying the Gaussian approximation for the higher orders. Finally, only the weakest path ξ_* is being used in the third step to find the correct thresholds.

An important question that arises in this regard is how decoding procedures can be improved further. To do this, we can simply eliminate a few weakest paths, starting from ξ_* for the algorithm Ψ_r^m, or ϱ_* for the algorithm Φ_r^m. Thus, we replace the original RM code with its subcode, in which a few information bits are eliminated.

For subcodes, we can proceed in a similar fashion, by finding the weakest remaining paths and calculating their thresholds according to Theorems 2 and 14. It is for this reason that calculations of moments D_s are also important on other paths. To date, however, the above ordering is rather incomplete. Therefore it is an open question as to which information bits should be removed from the original code.

Another decoding enhancement can be obtained by using soft-decision decoding instead of its hard-decision counterpart described above. The main difference arises from the fact that in the soft-decision case the parameters p and ε become random variables, whose values depend on the received symbols.

Thirdly, decoding performance can be greatly enhanced by using the lists of a few most probable codewords used in all intermediate decoding steps. To date, theoretical analysis of recursive list decoding is also an open area.

Finally, the above analysis presents only the first cut to the decoding problem. Namely, this analysis allows us to obtain only the thresholds, below which the decoding error probability becomes arbitrarily small for long codes. However, it is yet unclear how this error probability can be calculated or even how fast it declines. A solution to this problem is important from both the theoretical and practical perspective.

Acknowledgment. The authors wish to thank M. Burnashev for many helpful discussions.

References

[1] I.S. Reed, "A class of multiple error correcting codes and the decoding scheme," *IEEE Trans. Info. Theory*, vol. IT-4, pp. 38-49, 1954.

[2] S.N. Litsyn, "On decoding complexity of low-rate Reed-Muller codes," *Proc. 9^{th} All-Union Conf. on Coding Theory and Info. Transmission*, Part 1, Odessa, USSR, pp. 202-204, 1988 (in Russian).

[3] F. Hemmati, "Closest coset decoding of $u|u+v|$ codes," *IEEE Selected Areas Commun.*, vol. 7, pp. 982-988, 1989.

[4] G.A. Kabatyanskii, "On decoding of Reed-Muller codes in semicontinuous channels," *Proc. 2^{nd} Int. Workshop "Algebr. and Comb. Coding Theory"*, Leningrad, USSR, 1990, pp. 87-91.

[5] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1981.

[6] R.E. Krichevskiy, "On the Number of Reed-Muller Code Correctable Errors," *Dokl. Soviet Acad. Sciences*, vol. 191, pp. 541-547, 1970.

[7] V. Sidel'nikov and A. Pershakov, "Decoding of Reed-Muller codes with a large number of errors," *Probl. Info. Transmission*, vol. 28, no. 3, pp. 80-94, 1992 .

[8] G.D. Forney, "Coset codes-part II: Binary lattices and related codes," *IEEE Trans. Info. Theory*, vol. 34, pp. 1152-1187, 1987.

[9] W. Feller, *An Introduction to Probability Theory and its Applications.* New York: Wiley, vol. 2, 1971.

[10] G. Schnabl and M. Bossert, "Soft-decision decoding of Reed-Muller Codes as generalized multiple concatenated codes," *IEEE Trans. Info. Theory*, vol. 41, pp. 304-308, 1995.

[11] I. Dumer and R. Krichevskiy, "Soft Decision Majority Decoding of Reed-Muller Codes," *IEEE Trans. Info. Theory*, vol. 46, pp. 258-264, Jan. 2000.

[12] I. Dumer, "Recursive decoding of Reed-Muller codes," *Proc. 37^{th} Allerton Conf. on Commun., Cont., and Comp.*, Monticello, IL, Sept. 22-24, 1999, pp. 61-69.

[13] I. Dumer, "Recursive decoding and its performance for low-rate Reed-Muller codes," *IEEE Trans. Info. Theory*, vol. 50, pp. 811-823, May 2004.