Stopping Sets for MDS-based Product Codes

Fanny Jardel*, Joseph J. Boutros†, and Mireille Sarkiss‡
*Telecom ParisTech, 75013 Paris, France
†Texas A&M University at Qatar, 23874 Doha, Qatar
‡CEA List, 91191 Gif Sur Yvette, France
fannjard@gmail.com, boutros@tamu.edu, mireille.sarkiss@cea.fr

***Abstract*—Stopping sets for MDS-based product codes under iterative row-column algebraic decoding are analyzed in this paper. A union bound to the performance of iterative decoding is established for the independent symbol erasure channel. This bound is tight at low and very low error rates. We also proved that the performance of iterative decoding reaches the performance of Maximum-Likelihood decoding at vanishing channel erasure probability. Numerical results are shown for product codes at different coding rates.**

I. Introduction

Many constructions are known in coding theory for building efficient compound codes out of simple elementary codes. Besides concatenated convolutional (turbo) codes and low-density parity-check (LDPC) codes, product codes exhibit interesting algebraic properties and admit elegant graphical representations [1], [2]. Given the great amount of research on coding for erasure channels, motivated by recent results on Reed-Muller codes found by H. Pfister, R. Urbanke, and their teams, we study in this paper the performance of product codes in presence of erasures. Our product codes are built from maximum-distance separable (MDS) non-binary components. Stopping sets were first introduced for LDPC codes [4] and later studied for linear block codes [5]. We characterize and enumerate stopping sets for iterative decoding of product codes with MDS constituents. Iterative decoding performance of our codes is compared to maximum-likelihood (ML) performance. Our work [8] is a continuation of previous works on product codes, mainly the work by Rosnes on stopping sets for binary product codes [6] and the enumeration of specific erasure patterns by Sendrier for MDS-based product codes [7].

Section II introduces notation and main properties of non-binary product codes. The enumeration of stopping sets is given in Section III. Performance analysis with independent erasures is made in Section IV before concluding the paper.

II. Non-binary product codes

A linear block code C with parameters $[n, k, d]_q$ is a sub-space of dimension k of $\mathbb{F}_q^n$, where $\mathbb{F}_q$ is the finite field of order q and d is the minimum Hamming distance of C. The sub-space C is also referred to as a q-ary linear code of length n. C is MDS, i.e. Maximum Distance Separable, if it satisfies $d = n - k + 1$. Binary MDS codes are the trivial repetition codes and the single parity-check codes. In this paper, we only consider non-trivial non-binary MDS codes where $q > n > 2$.

The support of C, denoted by $\mathcal{X}(C)$, is the set of ℓ distinct positions $\{i_1, i_2, \ldots, i_\ell\} = \{i_j\}_{j=1}^{\ell}$, $1 \leq i_j \leq n$, such that, for all j, there exists a codeword $c = (c_1 \ldots c_n) \in C$ with $c_{i_j} \neq 0$. This notion of support $\mathcal{X}$ is applied later to rows and columns in a product code.

Consider two linear q-ary codes $C_1[n_1, k_1, d_1]$ and $C_2[n_2, k_2, d_2]$. A product code $C_P = C_1 \otimes C_2$ is a compound code built from the column component code C_1 and the row component code C_2 as follows: $v = [v_{ij}] \in \mathbb{F}_q^{n_1 \times n_2}$ is a codeword of C_P if all columns in v belong to C_1 and all rows in v belong to C_2. Here, $\mathbb{F}_q^{n_1 \times n_2}$ denotes the set of all $n_1 \times n_2$ matrices with entries from $\mathbb{F}_q$. The product code C_P has dimension $K = k_1 k_2$ and length $N = n_1 n_2$. In addition, its minimum Hamming distance is $d_1 d_2$. More product code properties are found in [1], [3].

Now, we define a rectangular support which will be useful later to characterize a stopping set in a row-column (bi-dimensional) product code. Let $\mathcal{S} \subseteq \{1, \ldots, n_1\} \times \{1, \ldots, n_2\}$ be a set of symbol positions in the product code. The set of row positions associated to $\mathcal{S}$ is $\mathcal{R}_1(\mathcal{S}) = \{i_1, \ldots, i_{\ell_1}\}$ where $|\mathcal{R}_1(\mathcal{S})| = \ell_1$ and for all $i \in \mathcal{R}_1(\mathcal{S})$ there exists $(i, \ell) \in \mathcal{S}$. The set of column positions associated to $\mathcal{S}$ is $\mathcal{R}_2(\mathcal{S}) = \{j_1, \ldots, j_{\ell_2}\}$ where $|\mathcal{R}_2(\mathcal{S})| = \ell_2$ and for all $j \in \mathcal{R}_2(\mathcal{S})$ there exists $(\ell, j) \in \mathcal{S}$. The rectangular support of $\mathcal{S}$ is

$$\mathcal{R}(\mathcal{S}) = \mathcal{R}_1(\mathcal{S}) \times \mathcal{R}_2(\mathcal{S}), \tag{1}$$

i.e. the smallest $\ell_1 \times \ell_2$ rectangle including all columns and all rows of $\mathcal{S}$.

The channel model considered in this paper is an i.i.d. symbol-erasure channel (SEC). It is assumed that the symbols of a product code codeword are independently erased each with a probability ϵ, $0 < \epsilon < 1$. The set of positions of erased symbols is called an *erasure pattern*. On a $SEC(q, \epsilon)$, an erasure-filling decoder for C_P fails in finding the original codeword when specific erasure patterns occur. More precisely, we list three decoding methods:

- Type I: ML decoder. This is a non-iterative decoder. It is based on a Gaussian reduction of the parity-check matrix of the product code.
- Type II: Iterative algebraic decoder. At odd decoding iterations, component codes C_1 on each column are decoded via an algebraic decoder (bounded-distance) that fills up to $d - 1$ erasures. Similarly, at even decoding iterations, component codes C_2 on each row are decoded via an algebraic decoder.

- Type III: Iterative ML-per-component decoder. This decoder was considered by Rosnes in [6] for binary product codes. At odd (resp. even) decoding iterations, column codes C_1 (resp. row codes C_2) are decoded via an optimal ML component decoder.

The ML decoder (type I) fails if the erasure pattern covers a product code codeword. Iterative algebraic decoder (type II) and iterative ML-per-component decoder (type III) fail for special erasure patterns, those covering a stopping set as defined below. Indeed, as shown in the sequel for MDS-based product codes, type-II stopping sets and type-III stopping sets are identical.

Definition 1: Consider a product code $C_P = C_1 \otimes C_2$. Let $\mathcal{S} \subseteq \{1,\ldots,n_1\} \times \{1,\ldots,n_2\}$ with $|\mathcal{R}_1(\mathcal{S})| = \ell_1$ and $|\mathcal{R}_2(\mathcal{S})| = \ell_2$. Consider the ℓ_1 rows of $\mathcal{S}$ given by $\mathcal{S}_r^{(i)} = \{j : (i,j) \in \mathcal{S}\}$ and the ℓ_2 columns of $\mathcal{S}$ given by $\mathcal{S}_c^{(j)} = \{i : (i,j) \in \mathcal{S}\}$. The set $\mathcal{S}$ is a stopping set of type II for C_P if $|\mathcal{S}_r^{(i)}| \geq d_2$ and $|\mathcal{S}_c^{(j)}| \geq d_1$, for all $i \in \mathcal{R}_1(\mathcal{S})$ and for all $j \in \mathcal{R}_2(\mathcal{S})$.

Definition 2: The set $\mathcal{S}$ is a stopping set of type III for C_P if there exist linear subcodes $C_c^{(j)} \subseteq C_1$ and $C_r^{(i)} \subseteq C_2$ such that $\mathcal{X}(C_c^{(j)}) = \mathcal{S}_c^{(j)}$ and $\mathcal{X}(C_r^{(i)}) = \mathcal{S}_r^{(i)}$, $\forall i \in \mathcal{R}_1(\mathcal{S})$ and $\forall j \in \mathcal{R}_2(\mathcal{S})$.

Before enumerating stopping sets of a product code, let us recall some fundamental results regarding the decoding of its row and column component codes. An erasure pattern is said to be *ML-correctable* if the ML decoder is capable of solving all its erased symbols. The ML erasure-filling capability of a linear code satisfies the following property.

Proposition 1: Let $C[n,k,d]_q$ be a linear code with $q \geq 2$. Assume that C is not MDS and the n symbols of a codeword are transmitted on an erasure channel. Then, there exists an erasure pattern of weight greater than $d-1$ that is ML-correctable.

Proof: Let H be an $(n-k) \times n$ parity-check matrix of C with rank $n-k > d-1$. For any integer w in the range $[d, n-k]$, there exists a set of w linearly independent columns in H. Choose an erasure pattern of weight w with erasures located at the positions of the w independent columns. Then, the ML decoder is capable of solving all these erasures by simple Gaussian reduction of H. ∎

For MDS codes, in a way similar to the above proposition, we state a known result in the following corollary.

Corollary 1: Let $C[n,k,d]_q$ be an MDS code. All erasure patterns of weight greater than $d-1$ are not ML-correctable.

We conclude from the previous corollary that an algebraic decoder for an MDS code attains the word-error performance of its ML decoder. What about symbol-error performance? Indeed, for general binary and non-binary codes, the ML decoder may outperform an algebraic decoder since it is capable of filling some of the erasures when dealing with a pattern which is not ML-correctable. In the MDS case, the answer comes from the absence of spectral holes for any MDS code beyond its minimum distance. This basic result is proven via standard tools from algebraic coding theory [1]:

Proposition 2: Let $C[n,k,d]_q$ be a non-binary MDS code ($q > n > 2$). For any w satisfying $d \leq w \leq n$ and any support $\mathcal{X} = \{i_1, i_2, \ldots, i_w\}$, where $1 \leq i_j \leq n$, there exists a codeword in C of weight w having $\mathcal{X}$ as its own support.

Proof: By assumption we have $w > r = n-k$. Let H be a parity-check matrix of C with rank $r = n-k$. Recall that the MDS property makes full-rank any set of $n-k$ columns of H [1]. w is written as $w = r + \ell$, where $\ell = 1 \ldots k$. The w positions of $\mathcal{X}$ are anywhere inside the range $[1,n]$, but for simplicity let us denote $h_1 \ldots h_r$ the r columns of H in the first r positions. The last ℓ columns are denoted $\zeta_1 \ldots \zeta_\ell$. For any $j = 1 \ldots \ell$, we have

$$\zeta_j = \sum_{i=1}^{r} a_{i,j} h_i,$$

where $a_{i,j} \in \mathbb{F}_q \setminus \{0\}$ otherwise it contradicts $d = n-k+1$. Now, select $\alpha_1 \ldots \alpha_\ell$ from $\mathbb{F}_q \setminus \{0\}$ such that: α_1 is arbitrary, α_2 is chosen outside the set $\{-\alpha_1 a_{i,1}/a_{i,2}\}_{i=1}^{r}$, then α_3 is chosen outside the set $\{(-\alpha_1 a_{i,1} - \alpha_2 a_{i,2})/a_{i,3}\}_{i=1}^{r}$, and so on, up to α_ℓ which is chosen outside the set $\{-\sum_{u=1}^{\ell-1} \alpha_u a_{i,u}/a_{i,\ell}\}_{i=1}^{r}$. The equality

$$\sum_{j=1}^{\ell} \alpha_j \zeta_j = \sum_{i=1}^{r} \sum_{j=1}^{\ell} \alpha_j a_{i,j} h_i$$

produces a codeword of Hamming weight w. Hence, there exists a codeword of weight w with non-zero symbols in all positions given by $\mathcal{X}$. ∎

Now, at the symbol level for an MDS code and an erasure pattern which is not ML-correctable ($w > d-1$), we conclude from Proposition 2 that the ML decoder cannot solve any of the w erasures because they are covered by a codeword. Consequently, an algebraic decoder for an MDS code also attains the symbol-error performance of the ML decoder. This behavior will have a direct consequence on the iterative decoding of a product code with MDS components: stopping sets are identical when dealing with algebraic and ML-per-component decoders, i.e. type-II and type-III stopping sets are identical thanks to Corollary 1 and Proposition 2. In the next sections, component codes C_1 and C_2 of a product code are assumed to be MDS.

III. Stopping sets analysis

Many of the properties of type-II and type-III stopping sets can be found in [8]. It is important to mention obvious stopping sets, also known as rank-1 sets. A stopping set $\mathcal{S}$ is *obvious* if $\mathcal{S} = \mathcal{R}(\mathcal{S})$. In the remaining material of this paper, we restrict our study to type-II stopping sets.

For a fixed non-zero integer w, the number of stopping sets of size $|\mathcal{S}| = w$, denoted as τ_w, falls in two different cases. Firstly, $\tau_w = 0$ if w is small with respect to the minimum Hamming distance of the product code. Also, $\tau_w = 0$ for special erasure patterns obtained by adding a small neighborhood to a smaller obvious set. Secondly, for both obvious and non-obvious stopping sets, τ_w is non-zero and

the weight w may correspond to many rectangular supports of different height and width. The code performance over erasure channels is dominated by not-so-large stopping sets. Non-empty stopping sets of the second case satisfy the general property stated in the following lemma.

Lemma 1: Given a weight $w \leq (d_1+1)(d_2+1)$ and assuming $\tau_w > 0$, then $\exists \mathcal{S}^0$ such that $\forall \mathcal{S}$ with $|\mathcal{S}| = w$, we have $\|\mathcal{R}(\mathcal{S})\| \leq \|\mathcal{R}(\mathcal{S}^0)\| = (\ell_1^0, \ell_2^0)$, where

$$\ell_1^0 \leq d_1 + 1 + \left\lfloor \frac{d_1+1}{d_2} \right\rfloor, \quad \ell_2^0 \leq d_2 + 1 + \left\lfloor \frac{d_2+1}{d_1} \right\rfloor. \quad (2)$$

Proof: Let w be equal to $(d_1+1)(d_2+1)$. In order to establish an upper bound of the height ℓ_1, we build the highest possible rectangular support for this weight w. Assume the rectangle is $\ell_1^0 \times \ell_2$, each of its rows should have at least d_2 erasures to make the type-II decoder fail. Then $d_2\ell_1^0 \leq (d_1+1)(d_2+1)$ which becomes the upper bound given by (2). Now, if w is less than $(d_1+1)(d_2+1)$, the rectangular support of the stopping set can only shrink in size. The upper bound of the width in (2) is proven in a similar way. ∎

The above lemma states the existence of a maximal rectangular support for a given stopping set size. The next lemma gives an upper bound of the size of $\mathcal{R}(\mathcal{S})$ by stating a limit to the number of zeros (non-erased positions) inside the rectangle $\mathcal{R}(\mathcal{S})$.

Lemma 2: Let $\mathcal{R}(\mathcal{S})$ be the $\ell_1 \times \ell_2$ rectangular support of a stopping set $\mathcal{S}$ of size w. Let $\beta = \ell_1\ell_2 - w$ be the number of zero positions, or equivalently β is the size of the set $\mathcal{R}(\mathcal{S}) \backslash \mathcal{S}$. Then

$$\beta \leq \min((\ell_1 - d_1)\ell_2, \ell_1(\ell_2 - d_2)). \quad (3)$$

The enumeration of stopping sets represented as matrices of a given distribution of row weight and column weight is equivalent to enumerating bipartite graphs where left vertices stand for rows and right vertices stand for columns. Stopping sets enumeration in the next theorem is based on β, the number of zeros or the number of non-erased positions. Hence, we shall use the opposite rule. A stopping set of weight w and having a $\ell_1 \times \ell_2$ rectangular support shall be represented by a bipartite graph with ℓ_1 left vertices, ℓ_2 right vertices, and a total of $\beta = \ell_1\ell_2 - w$ edges. Notice that these bipartite graphs have no length-2 cycles because parallel edges are forbidden.

In the sequel, the open interval between two real numbers a and b will be denoted $]a, b[\; = \{x \in \mathbb{R} : a < x < b\}$.

Theorem 1: Let C_P be a product code $[n_1, k_1, d_1]_q \otimes [n_2, k_2, d_2]_q$ built from row and column MDS component codes, where the alphabet size q is greater than $\max(n_1, n_2)$. Let τ_w be the number of stopping sets of size w. We write $\tau_w = \tau^a + \tau^b$, where τ^a counts obvious stopping sets and τ^b counts non-obvious stopping sets. Under (type-II) iterative algebraic decoding and for $d_1 = d_2 = d \geq 2$, stopping sets are characterized as follows:

- For $w < d^2$, $\tau^a = \tau^b = 0$.
- For $w = d^2$, $\tau^a = \binom{n_1}{d}\binom{n_2}{d}$ and $\tau^b = 0$.
- For $w \in]d^2, d(d+1)[$, $\tau^a = \tau^b = 0$.
- For $w = d(d+1)$,

$$\tau^a = \binom{n_1}{d}\binom{n_2}{d+1} + \binom{n_1}{d+1}\binom{n_2}{d},$$
$$\tau^b = (d+1)!\binom{n_1}{d+1}\binom{n_2}{d+1}.$$

- For $w \in \;]d(d+1), d(d+2)[$.
  Let us write $w = d^2 + d + \lambda$, where $\lambda \in [1, d-1]$.

$$\tau^a = 0,$$
$$\tau^b = (d+1-\lambda)!\binom{d+1}{\lambda}^2\binom{n_1}{d+1}\binom{n_2}{d+1}.$$

- For $w = d(d+2)$,

$$\tau^a = \binom{n_1}{d}\binom{n_2}{d+2} + \binom{n_1}{d+2}\binom{n_2}{d},$$
$$\tau^b = (d+1)^2\binom{n_1}{d+1}\binom{n_2}{d+1} + \sum_{2r_0+r_1=d}\binom{d+1}{r_0}\binom{d+1-r_0}{r_1}\frac{(d+2)!}{2^{r_2}} \left[\binom{n_1}{d+1}\binom{n_2}{d+2} + \binom{n_1}{d+2}\binom{n_2}{d+1}\right] + x_{d+2}\binom{n_1}{d+2}\binom{n_2}{d+2},$$

  where $\sum_{2r_0+r_1=d}$ is a summation over r_0 and r_1, both being non-negative and satisfying $2r_0 + r_1 = d$, $r_2 = d + 1 - r_0 - r_1$, and x_{d+2} is the number of degree-2 bipartite graphs as given by Lemma 3 in [8].

- For $w = (d+1)(d+1)$

$$\tau^a = \binom{n_1}{d+1}\binom{n_2}{d+1},$$
$$\tau^b = \sum_{2r_0+r_1=d+1}\binom{d+1}{r_0}\binom{d+1-r_0}{r_1}\frac{(d+2)!}{2^{r_0}} \left[\binom{n_1}{d+1}\binom{n_2}{d+2} + \binom{n_1}{d+2}\binom{n_2}{d+1}\right] + y_{d+2}\binom{n_1}{d+2}\binom{n_2}{d+2},$$

  where y_{d+2} is the number of degree-2 bipartite graphs, except for one left vertex and one right vertex having degree 1. The number y_{d+2} is given by Lemma 4 in [8].

The detailed proof of Theorem 1 is found in [8] (Theorem 2 in the journal version). The proof cannot be included in this short paper due to lack of space. However, we give an illustration on how to compute τ_w for $w = 15$ and $d = 3$ as a first step to help the reader in understanding the general proof. Results for $d_1 \neq d_2$ are given in Theorem 3 in [8].

For a stopping set of size $w = d(d+2)$, a $d \times (d+2)$ rectangular support corresponds to obvious sets enumerated by τ^a. The latter also counts sets with a $(d+2) \times d$ rectangular support. As seen in Theorem 1, for $w = d(d+2)$, non-obvious

sets counted by τ^b correspond to three different sizes of $\mathcal{R}(\mathcal{S})$: $(d+1)\times(d+1)$, $(d+1)\times(d+2)$, and $(d+2)\times(d+2)$.

Consider the $(d+1)\times(d+2)$ rectangle in the practical case $d=3$ and $w=15$. Here we get $\beta=(d+1)(d+2)-d(d+2)=d+2=5$. In this stopping set, every column must have one zero (the non-erased symbol). A row may carry up to two zeros. Let r_0, r_1, and r_2 denote the number of rows with 0, 1, and 2 zeros respectively. Then $r_0+r_1+r_2=d+1$ and $0\cdot r_0+r_1+2r_2=\beta$ so we obtain the condition $2r_0+r_1=d$ as stated in the theorem. For $d=3$, two pairs satisfy this condition, $(r_0,r_1)=(1,1)$ and $(r_0,r_1)=(0,3)$. An illustration of the 4×5 stopping sets is given below.

$$\begin{pmatrix}1&1&1&1&1\\0&1&1&1&1\\1&0&0&1&1\\1&1&1&0&0\end{pmatrix} \qquad \begin{pmatrix}0&1&1&1&1\\1&0&1&1&1\\1&1&0&1&1\\1&1&1&0&0\end{pmatrix}$$

$(r_0,r_1)=(1,1)$ $\qquad$ $(r_0,r_1)=(0,3)$

In the first case on the left, r_0 rows must be selected among $d+1=5$ rows, followed by r_1 rows to be chosen among the remaining $d+1-r_0$ rows. In addition, the $d+2=5$ columns can be permuted leading to an extra factor equal to $(d+2)!$. Finally, each double-zero row creates two identical columns, i.e. the final number should be divided by 2^{r_2}. We obtain the general expression for the number of these matrices

$$\sum_{2r_0+r_1=d}\binom{d+1}{r_0}\binom{d+1-r_0}{r_1}\frac{(d+2)!}{2^{r_2}}.$$

The exact number of 4×5 matrices is 360 for $(r_0,r_1)=(1,1)$ and 240 for $(r_0,r_1)=(0,3)$. This number is added to τ_{15} in (7) after multiplication by $\binom{n_1}{d+1}\binom{n_2}{d+2}+\binom{n_1}{d+2}\binom{n_2}{d+1}$ for $n_1=n_2=16$.

• Extension to w beyond $(d+1)^2$.

In [8], authors considered high-rate product codes suitable to storage applications. For low-rate product codes, e.g. $R=K/N$ around $1/2$, it is necessary to enumerate stopping sets beyond $(d+1)^2$. Take $w=(d+1)^2+\lambda<d(d+3)$, τ_w is directly derived from the last case in Theorem 1. Thanks to Lemma 2, all rectangular supports are excluded except for $(d+1)\times(d+2)$ and $(d+2)\times(d+2)$, where the second rectangular size makes the dominating term in τ^b. For a $(d+2)\times(d+2)$ rectangle, a larger stopping set is built by adding λ ones among the β original zeros which yields $\tau_w=\tau^b=y_{d+2}\binom{n_1}{d+2}\binom{n_2}{d+2}\binom{\beta}{\lambda}$, where the number of zeros in the original stopping set is $\beta=(d+2)^2-(d+1)^2=2d+3$. Finally, take $w=d(d+3)$, the width of a rectangular support cannot exceed $d+3$ thanks to Lemma 1. Rectangles of size $d(d+3)$ (obvious sets), $(d+1)\times(d+2)$, $(d+1)\times(d+3)$, $(d+2)\times(d+2)$, and $(d+2)\times(d+3)$ should be considered. We get $\tau^a=\binom{n_1}{d}\binom{n_2}{d+3}+\binom{n_1}{d+3}\binom{n_2}{d}$ and $\tau^b=(d+1)^3(d+2)/2\left[\binom{n_1}{d+1}\binom{n_2}{d+2}+\binom{n_1}{d+2}\binom{n_2}{d+1}\right]$ for the $(d+1)\times(d+2)$ rectangle. In this case too, the number of stopping sets is dominated by the $(d+2)\times(d+2)$ rectangle with $\tau^b=y_{d+2}\binom{n_1}{d+2}\binom{n_2}{d+2}\binom{2d+3}{d-1}$ for non-obvious sets.

IV. Performance with independent erasures

Consider the i.i.d. erasure channel $SEC(q,\epsilon)$. The N symbols of a codeword are independently erased by the channel. A symbol is erased with a probability ϵ and is correctly received with a probability $1-\epsilon$. Before studying the performance on the $SEC(q,\epsilon)$, we state a result about obvious stopping sets in the following proposition.

Proposition 3: Let $C_P=C_1\otimes C_2$ be a product code with non-binary MDS components. All obvious stopping sets are supports of product code codewords.

Proof: Consider an $\ell_1\times\ell_2$ obvious stopping set. Its rectangular support is $\mathcal{R}(\mathcal{S})=\mathcal{R}_1(\mathcal{S})\times\mathcal{R}_2(\mathcal{S})$. We have $\ell_1\geq d_1$ and $\ell_2\geq d_2$. From Proposition 2, there exists a column codeword $x=(x_1,x_2,\ldots,x_{n_1})\in C_1$ of weight ℓ_1 with support $\mathcal{R}_1(\mathcal{S})\times\{j_1\}$, where $j_1\in\mathcal{R}_2(\mathcal{S})$. Similarly, there exists a row codeword $y=(y_1,y_2,\ldots,y_{n_2})\in C_2$ of weight ℓ_2 with support $\{i_1\}\times\mathcal{R}_2(\mathcal{S})$, where $i_1\in\mathcal{R}_1(\mathcal{S})$. The Kronecker product of x and y satisfies $\mathcal{X}(x\otimes y)=\mathcal{S}$. ■

Corollary 2: Consider a product code $C_P=C_1\otimes C_2$ with non-binary MDS component codes. Assume the symbols of C_P are transmitted over a $SEC(q,\epsilon)$ channel. Let $P_{ew}^{\mathcal{G}}$ be the word error probability of an iterative (type-II) decoder and P_{ew}^{ML} be the word error probability of ML decoding. Then, for $\epsilon\ll 1$, the error probabilities satisfy $P_{ew}^{\mathcal{G}}\sim P_{ew}^{ML}$.

Proof: On the $SEC(q,\epsilon)$, the word error probabilities are given by [5],

$$P_{ew}^{ML}=\sum_{i=d_1d_2}^{N}\Psi_i(ML)\epsilon^i(1-\epsilon)^{N-i}, \tag{4}$$

where $\Psi_i(ML)$ is the number of weight-i erasure patterns covering a product code codeword, and

$$P_{ew}^{\mathcal{G}}=\sum_{i=d_1d_2}^{N}\Psi_i(\mathcal{G})\epsilon^i(1-\epsilon)^{N-i}, \tag{5}$$

where $\Psi_i(\mathcal{G})$ is the number of weight-i erasure patterns covering a stopping set. Asymptotic length analysis is not considered in this paper, i.e. $N=n_1n_2$ is fixed. We write $P_{ew}^{ML}=\Psi_{d_1d_2}(ML)\epsilon^{d_1d_2}+o(\epsilon^{d_1d_2})$ and $P_{ew}^{\mathcal{G}}=\Psi_{d_1d_2}(\mathcal{G})\epsilon^{d_1d_2}+o(\epsilon^{d_1d_2})$. From Proposition 3, we get the equality $\Psi_{d_1d_2}(\mathcal{G})=\Psi_{d_1d_2}(ML)$ and so we obtain $\lim_{\epsilon\to 0}P_{ew}^{\mathcal{G}}/P_{ew}^{ML}=1$. ■

Numerical evaluations of $\Psi_i(\mathcal{G})$ are tractable for very short codes ($N\leq 25$) and become very difficult for codes of moderate size and beyond, e.g. for $N\geq 100$. For this reason, expressions (4) and (5) are not practical to predict the $SEC(q,\epsilon)$ performance of product codes.

For $P_{ew}^{\mathcal{G}}$, thanks to Theorem 1, a union bound can be easily established. Indeed, we have

$$P_{ew}^{\mathcal{G}}=P(\exists\mathcal{S}\ covered)\leq\sum_{w}P(\exists\mathcal{S}:|\mathcal{S}|=w,\mathcal{S}\ covered),$$

leading to

$$P_{ew}^{\mathcal{G}}\leq P^U(\epsilon)=\sum_{w=d_1d_2}^{N}\tau_w\epsilon^w. \tag{6}$$

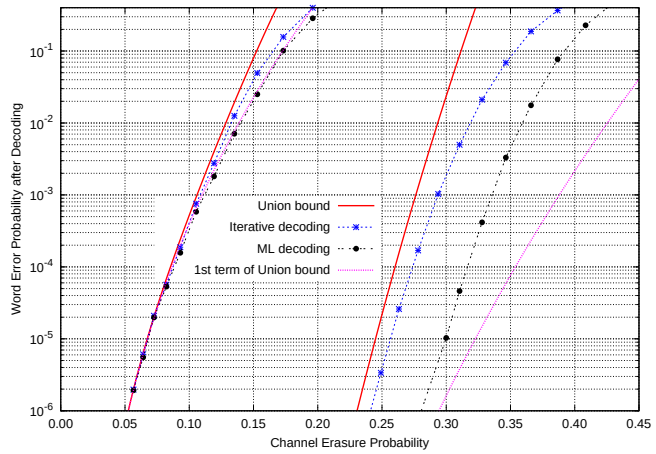

Fig. 1: Product codes $[16, 14]_q^{\otimes 2}$ (left) and $[16, 12]_q^{\otimes 2}$ (right). Word error rate performance for iterative decoding versus its union bound and ML decoding.

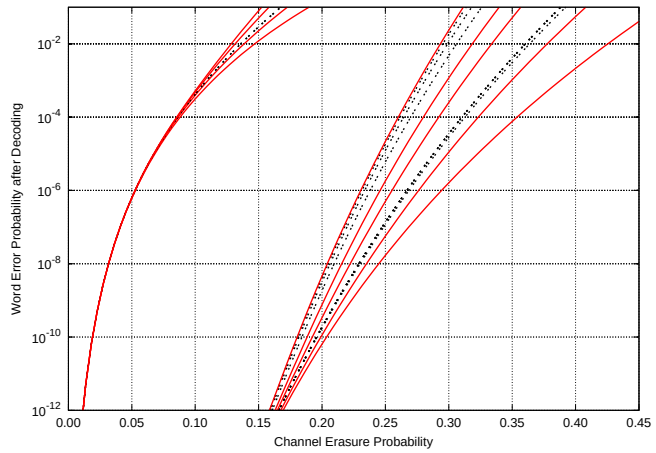

Fig. 2: Product codes $[16, 14]_q^{\otimes 2}$ (left) and $[16, 12]_q^{\otimes 2}$ (right). Contribution of union bound terms to P_{ew}. Union bound truncated at a term with $\tau^a = 0$ is drawn in dashed black.

From Theorem 1, the union bound $P^U(\epsilon)$ for the $[16, 14, 3]_q^{\otimes 2}$ product code is, for $w \leq (d+1)^2 = 16$,

$$\begin{aligned} P^U(\epsilon) = \; & 313600\epsilon^9 + 81536000\epsilon^{12} + 317990400\epsilon^{13} \\ & + 238492800\epsilon^{14} + 48519627520\epsilon^{15} \\ & + 448369776400\epsilon^{16} + o(\epsilon^{16}). \end{aligned} \tag{7}$$

The performance of this code on the $SEC(q, \epsilon)$ channel is shown in Figure 1 (left curves). Its coding rate is $R = K/N = 0.7656$. We used the standard finite field of size $q = 256$.

As observed in the plot of Figure 1, the union bound is sufficiently tight. Furthermore, the performance of the iterative algebraic row-column decoding is very close to that of ML decoding in the whole range of ϵ. For small ϵ, the curves are superimposed as predicted by Corollary 2.

The union bound $P^U(\epsilon)$ for the $[16, 12, 5]_q^{\otimes 2}$ product code is, for $w \leq d(d+3) = 40$,

$$\begin{aligned} P^U(\epsilon) = & 19079424\epsilon^{25} + 462E8\epsilon^{30} + 277E9\epsilon^{31} + 346E9\epsilon^{32} \\ & + 153E9\epsilon^{33} + 28E9\epsilon^{34} + 430E12\epsilon^{35} + 617E13\epsilon^{36} \\ & + 79E15\epsilon^{37} + 47E16\epsilon^{38} + 17E17\epsilon^{39} + 43E17\epsilon^{40} + o(\epsilon^{40}). \end{aligned}$$

The performance of this code on the $SEC(q, \epsilon)$ channel is shown in Figure 1 (right curves). Its coding rate is $R = K/N = 0.5625$. The union bound is relatively tight but the gap between ML and iterative decoding performance is larger. Increasing d from 3 to 5 increases the erasure-filling capacity of a component decoder by 2 but the minimum Hamming distance of C_P jumps from 9 to 25 thus giving a greater efficiency to ML decoding.

Figure 2 shows truncated union bounds for the same codes where the bound summation is truncated at all sizes w used in the final expressions of $P^U(\epsilon)$. It allows us to observe the contribution of each term to the union bound.

V. Conclusions

We enumerated stopping sets for MDS-based product codes under iterative row-column algebraic decoding. A union bound was established for the i.i.d. symbol erasure channel. This bound is tight enough in predicting the performance of iterative decoding at low and very low error rates. Furthermore, we proved that the performance of iterative decoding reaches the performance of ML decoding at vanishing channel erasure probability. Finally, as a rule of thumb, we state that a stopping set size $w = (d+1)^2$ is sufficient in bounding the performance at high coding rate (e.g. $R = K/N$ around $3/4$) while it is necessary to enumerate stopping sets up to $w = d(d+3)$ or $(d+2)^2$ for low and very low coding rates.

Acknowledgment

The work of Joseph J. Boutros was supported by the Qatar National Research Fund (QNRF), a member of Qatar Foundation, under NPRP project 6-784-2-329.

References

[1] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.

[2] L.M.G.M. Tolhuizen, "More results on the weight enumerator of product codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2537-2577, Sep. 2002.

[3] F.R. Kschischang, Product Codes, J.G. Proakis (ed), *Wiley encyclopedia of telecommunications*, pp. 2007-2012, vol. 4, Hoboken, NJ, 2003.

[4] C. Di, D. Proietti, I.E. Telatar, T.J. Richardson, and R.L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570-1579, Jun. 2002.

[5] M. Schwartz and A. Vardy, "On the stopping distance and the stopping redundancy of codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 922-932, Mar. 2006.

[6] E. Rosnes, "Stopping set analysis of iterative row-column decoding of product codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1551-1560, Apr. 2008.

[7] N. Sendrier, "Codes correcteurs d'erreurs à haut pouvoir de correction," Thèse de Doctorat de l'Université Paris 6, in French, Dec. 1991.

[8] F. Jardel and J.J. Boutros, "Edge Coloring and Stopping Sets Analysis in Product Codes with MDS components," submitted to the *IEEE Trans. Inf. Theory*, Dec. 2015. ArXiv 1603.01468.