Analysis of the error correction capability of LDPC and MDPC codes under parallel bit-flipping decoding and application to cryptography

Paolo Santini, Massimo Battaglioni, Marco Baldi and Franco Chiaraluce
Dipartimento di Ingegneria dell'Informazione
Università Politecnica delle Marche
Ancona, Italy, 60131
Email: p.santini@pm.univpm.it, {m.battaglioni, m.baldi, f.chiaraluce}@staff.univpm.it

Abstract

Iterative decoders used for decoding low-density parity-check (LDPC) and moderate-density parity-check (MDPC) codes are not characterized by a deterministic decoding radius and their error rate performance is usually assessed through intensive Monte Carlo simulations. However, several applications, like code-based cryptography, need guaranteed low values of the error rate, which are infeasible to assess through simulations, thus requiring the development of theoretical models for the error rate of these codes under iterative decoding. Some models of this type already exist, but become computationally intractable for parameters of practical interest. Other approaches approximate the code ensemble behaviour through some assumptions, which may not hold true for a specific code. We propose a theoretical analysis of the error correction capability of LDPC and MDPC codes that allows deriving tight bounds on the error rate at the output of parallel bit-flipping decoders. Special attention is devoted to the case of codes with small girth; moreover, single-iteration decoding is investigated through a rigorous approach, which does not require any assumption and hence results in a guaranteed error correction capability for any single code. We show an example of application

The material in this paper has been presented in part at the 2019 IEEE International Conference on Communications, Shanghai (China) [1].

of the new bound to the context of code-based cryptography, where guaranteed error rates are needed to achieve some strong security levels.

Index Terms

Bit flipping decoder, code-base cryptography, error correction capability, LDPC codes, MDPC codes.

I. Introduction

Contrary to bounded distance decoders, iterative decoders commonly used for low-density parity-check (LDPC) and moderate-density parity-check (MDPC) codes are not characterized by a deterministic decoding radius. This implies the existence of a residual error rate that is difficult to model theoretically, and is hence usually assessed through Monte Carlo simulations. Nevertheless, there are applications in which extremely low error rates are required. One of these cases is in the area of code-based cryptography, where error rates as low as 2^{-80} or less are required to avoid some types of attacks [2]–[5]. Obviously, such low values of the error rate are infeasible to assess through numerical simulations.

Therefore, an important research challenge is represented by the development of analytical tools able to foresee the number of errors that an iterative decoder can correct. A vast body of literature exists on this subject [6]–[10], which permits to determine lower and upper bounds on the guaranteed error correction capability of the code. Many of these approaches use expander graph based arguments [8], [9], whose application, however, is known to be NP-hard [11] and can be used for a limited number of cases and under specific constraints. Moreover, the bounds these methods provide are often loose, particularly in case of small girths.

To overcome these limitations, recently, in [12] and [1], a new approach has been proposed to evaluate the guaranteed error correction capability of LDPC and MDPC codes. In [12], in particular, a majority-logic decoder is considered and it is shown that its error correction capability depends on the maximum number of superimpositions between any two columns of the code parity-check matrix. This allows deriving conditions under which a single iteration of this decoder corrects all errors up to a given weight. These results are extended in [1], where a more general decoder is considered and tighter bounds are derived.

The latter results, however, are obtained under some assumptions. As a first contribution, this paper improves the analysis in [1], by providing tighter bounds. For such a purpose, we

focus attention on Gallager's bit flipping (BF) decoder [13], because of its high computational efficiency, due to a relatively low algorithmic complexity.

Low-complexity iterative decoders are important in many applications where high throughputs have to be achieved. Starting from its basic principle, several variants of Gallager's BF algorithm have been proposed. Among them, in this paper we focus on the so-called parallel BF. Roughly speaking, the parallel BF algorithm operates as follows. At each iteration, all parity checks are computed: all bits involved in a number of unsatisfied parity-check equations overcoming some suitably chosen threshold are flipped, and the syndrome is accordingly updated. The procedure is iterated, until a null syndrome is obtained or a maximum number of iterations is reached. Following a more general approach than [14], where parallel BF is introduced, we consider a threshold that is not fixed, but rather depends on some features of the code under investigation.

In principle, other families of iterative decoding algorithms could achieve better error correction performance than BF decoding. However, we focus on channel models without soft information, where decoding algorithms working with discrete values are a natural choice. Moreover, the parallel BF algorithm is characterized by a very high algorithmic efficiency, which is an important requirement in code-based cryptography [15], [16]. Such an area of application is experiencing an increasing interest by the scientific community due to the standardization initiative of post-quantum cryptosystems started in 2016 by the US National Institute of Standards and Technology (NIST) [17]. In this context, state-of-the-art schemes based on LDPC and MDPC codes such as LEDAcrypt [18] and BIKE [19] employ decoders such as BF or some of its variants. This is all the more evident by considering that in these applications very large codes are usually required and the adoption of more complex decoding algorithms would yield unacceptable delays.

When LDPC or MDPC codes are used in code-based cryptosystems, the structure of their parity-check matrix is mainly dictated by security issues. This may yield unavoidable short cycles in the Tanner graph describing the code. More precisely, in these systems the sparse parity-check matrix of an LDPC or MDPC code is used as a secret key and it usually has quasi-cyclic (QC) structure. Starting from a code ensemble, according to the chosen QC structure, the parity-check matrix of the code is randomly picked from the ensemble, thus often yielding a large number of cycles of length 6 or even 4. Accurate evaluation of the guaranteed error correction capability of codes with small girth has not been extensively investigated in previous literature. This is another relevant contribution of this paper, as we show that the new bounds are particularly tight if the girth of the considered codes is small.

We devote our attention to the first iteration of BF decoding. For it, we provide an upper bound on the error rate of LDPC and MDPC codes which does not rely on any specific assumption. We note that some lower and upper bounds on the error rate under BF decoding are also proposed in [20], but their computation requires pre-processing of all possible initial error patterns with weight up to a certain value; thus, the approach becomes quickly unfeasible as the error probability of the channel decreases or error patterns with too large weight have to be considered. The same remark holds for the approaches proposed in [21]–[24], which allow estimating the error rate of LDPC codes under BF decoding. Our approach instead is fully analytical, and does not require any preliminary simulation or assumption. To the best of our knowledge, this is the first time in which this problem is faced in exact analytical terms.

The paper is organized as follows. In Section II we introduce the notation used throughout the paper and recall some basic notions of LDPC and MDPC codes. In Section III we discuss the error correction capability of codes with small girth under BF decoding. In Section IV we provide an upper bound on the error rate of LDPC and MDPC codes under BF decoding. In Section V we present the results of numerical simulations and show an application of the derived bounds to code-based cryptography. Finally, we draw some conclusions in Section VI.

II. Notation and definitions

We use capital letters to denote sets, adopting caligraphic fonts for sets of vectors. The cardinality of a set A (or $\mathcal{A}$) is denoted as $|A|$ (or $|\mathcal{A}|$). Given a set A, we use $a \leftarrow A$ to express the fact that a is randomly extracted, with uniform law, among all the elements of A, and the same notation is used for sets of vectors.

The binary Galois field is denoted as $\mathbb{F}_2$. We use small bold letters to denote vectors, and capital bold letters to denote matrices. Given a matrix $\mathbf{H}$, its entry at position (i, j) is denoted as $h_{i,j}$ and its k-th column is denoted as $\mathbf{h}_k$. Given a vector $\mathbf{e}$, we refer to its j-th entry as e_j. Given a set A, we have $\mathbf{e}^{(A)} = \{e_i \text{ s.t. } i \in A\}$. The AND, OR and ex-OR operations are denoted as $\wedge$, $\vee$ and $\oplus$, respectively. The Hamming weight and the support of any vector $\mathbf{e}$ are referred to as $\mathrm{wt}(\mathbf{e})$ and $S(\mathbf{e})$, respectively. The set of integers between a and b, extremes included, is indicated as $[a, b]$. We denote the set of all binary vectors of length n and Hamming weight m as $\mathcal{B}_m$.

A. LDPC and MDPC codes

A binary LDPC code is the null space of a binary parity-check matrix $\mathbf{H}$ containing a small number of ones compared to the total number of entries. Denoting the code block length as n and the code dimension as k, $\mathbf{H}$ has $r \geq n-k$ rows and n columns and the design rate is $R = \frac{1}{2}$. The *syndrome* of a binary vector $\mathbf{e}$ is defined as $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$, where $^\top$ denotes transposition and the product is performed over $\mathbb{F}_2$. Any codeword belonging to the code defined by $\mathbf{H}$ has an all-zero syndrome. The i-th column and j-th row of $\mathbf{H}$ have weight v_i and w_j, respectively. The code is said to be (v,w)-*regular* if each column of $\mathbf{H}$ contains exactly v ones and each row contains exactly w ones. Regular LDPC codes are generally characterized by $w = O(\log n)$, whereas regular MDPC codes have $w = O(\sqrt{n})$. These two families of codes allow the same decoding principle, based on the sparsity of their parity-check matrices. Let us introduce two classes of QC codes that will be considered throughout the paper (in particular, in Sections IV-C and V). Codes in the first class are defined by parity-check matrices in the following form

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_1 \end{bmatrix}, \tag{1}$$

where each $\mathbf{H}_i$, $i \in \{0,1\}$, is a circulant matrix of size p and row/column weight v. The resulting codes are $(v,2v)$-regular, have block length $2p$ and design rate $R = \frac{1}{2}$.

Codes in the second class, also named *monomial codes* [25], are defined by parity-check matrices in the following form

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}^p(i_{0,0}) & \dots & \mathbf{I}^p(i_{0,w-1}) \\ \vdots & \ddots & \vdots \\ \mathbf{I}^p(i_{v-1,0}) & \dots & \mathbf{I}^p(i_{v-1,w-1}) \end{bmatrix}, \tag{2}$$

where $\mathbf{I}^p(i)$ is the identity matrix of size p whose columns have been cyclically shifted downwards by i positions.

Definition 1 *Given a matrix $\mathbf{H} \in \mathbb{F}_2^{r\times n}$, the* adjacency matrix *of $\mathbf{H}$, denoted as $\mathbf{\Gamma}$, is the $n \times n$ matrix whose element in position (i,j) is such that*

$$\gamma_{i,j} = \begin{cases} |S(\mathbf{h}_i) \cap S(\mathbf{h}_j)| & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}.$$

The adjacency matrix is commonly employed in graph theory: given a multigraph with n nodes, the adjacency matrix can be defined as the $n \times n$ matrix whose element in position (i,j) is equal to the number of edges connecting nodes i and j. Obviously, starting from a

parity-check matrix $\mathbf{H}$, we can construct a graph[1] with n nodes, such that the i-th and the j-th node are connected by $|S(\mathbf{h}_i) \cap S(\mathbf{h}_j)|$ edges.

B. Bit flipping decoding

Let us describe a general version of the parallel BF algorithm, which performs a single iteration. Decoder inputs are a syndrome $\mathbf{s} \in \mathbb{F}_2^r$ and a vector of integers $\mathbf{b} = [b_0, \cdots, b_{n-1}]$, such that $b_i \in [1, v_i]$, $\forall i$. For each $i \in [0, n-1]$, the number of unsatisfied parity-check equations involving the i-th bit is computed; we denote such a number as σ_i. The decoder considers as "error affected" all bits for which $\sigma_i \geq b_i$ and, thus, returns as output a vector $\mathbf{e}'$ with support $S(\mathbf{e}') = \{i \text{ s.t. } \sigma_i \geq b_i\}$. So, b_i has the meaning of a *decision threshold* for the i-th bit. Clearly, decoding is successful if $\mathbf{e}'$ coincides with the actual error vector. An important special case considered next is that in which $b_i = b, \forall i$, which boils down to a majority-logic decoder when $b = \lfloor \frac{v}{2} \rfloor + 1$. The decoding procedure we consider is reported in Algorithm 1.

III. Guaranteed error correction capability of bit flipping

Let us provide some preliminary definitions taken from [1], with some adaptations.

Definition 2 *Given* $\mathbf{H}$*, let us consider the rows of* $\mathbf{H}$ *indexed by* $S(\mathbf{h}_i)$ *and put them into a matrix* $\mathbf{H}^{(i)}$*. Following [1], we define* $\mathbf{H}^{(i)}$ *as the* i*-th* partial parity-check matrix. *The* j*-th column of* $\mathbf{H}^{(i)}$ *is denoted as* $\mathbf{h}_j^{(i)}$*. We also define*

$$\delta^{(i)}(\mathbf{H}^{(i)}, z) = \max_{M,\, |M|=z,\, i \notin M} \left\{ \mathrm{wt}\left(\bigoplus_{j \in M} \mathbf{h}_j^{(i)} \right) \right\},$$

where M *is a set containing the indexes of* z *columns of* $\mathbf{H}^{(i)}$*, except for the* i*-th. We call the* maximum column intersection of order z*, and denote as* $\delta(\mathbf{H}, z)$*, the quantity defined as*

$$\delta(\mathbf{H}, z) = \max_{0 \leq i \leq n-1} \left\{ \delta^{(i)}(\mathbf{H}^{(i)}, z) \right\}.$$

When $z = 1$, we call $\delta(\mathbf{H}, 1)$ the *maximum column intersection* and, for simplicity, we denote it as δ; it is easy to see that δ corresponds to the maximum number of set positions in which two columns of $\mathbf{H}$ overlap. We remark that, if the code has girth larger than 4, then the supports of any two columns intersect in at most one position, thus we have $\delta = 1$.

[1]We remark that this graph, which is not bipartite, is different from the Tanner graph [26] of the code.

Algorithm 1 BFdecoder

Input: $\mathbf{H} \in \mathbb{F}_2^{r \times n}$, $\mathbf{s} \in \mathbb{F}_2^r$, $i_{\max} \in \mathbb{N}$, $\mathbf{b} = [b_0, \ldots, b_{n-1}]$, $b_i \in [1, v_i]$, $\forall i$

Output: $\mathbf{e}' \in \mathbb{F}_2^n$

$\mathbf{e}' \leftarrow \mathbf{0}_n$
$F \leftarrow \varnothing$
for $i \leftarrow 0$ **to** $n-1$ **do**
    $\sigma_i \leftarrow 0$
    for $l \in S(\mathbf{h}_i)$ **do**
        $\sigma_i \leftarrow \sigma_i + s_l$
    end for
    if $\sigma_i \geq b_i$ **then**
        $F \leftarrow F \cup i$ ▷ Position i is estimated as error affected
    end if
end for
for $i \in F$ **do**
    $e'_i \leftarrow e'_i \oplus 1$ ▷ Error estimation update
end for
return $\{\mathbf{e}'\}$

The above notions can be easily related to the entries of the adjacency matrix. For instance, the weight of the j-th column of the i-th partial parity-check matrix is equal to the (i, j)-th element of the matrix $\mathbf{\Gamma}$, $\gamma_{i,j}$, and the maximum column intersection corresponds to the largest entry of $\mathbf{\Gamma}$. For a code with girth larger than 4, the entries of the adjacency matrix belong to $[0, 1]$.

Definition 3 *Given* $\mathbf{H}$ *and the corresponding adjacency matrix* $\mathbf{\Gamma}$*, we denote as* $\tilde{\boldsymbol{\gamma}}^{(i)}$ *the vector formed by the elements of the* i*-th row of* $\mathbf{\Gamma}$*, except for the* i*-th one. We define* $\mu^{(i)}(z)$ *as the sum of the* z *largest entries of* $\tilde{\boldsymbol{\gamma}}^{(i)}$*. We then define the* maximum column union of order z*, denoted as* $\mu(\mathbf{H}, z)$*, the quantity*

$$\mu(\mathbf{H}, z) = \max_{0 \leq i \leq n-1} \left\{ \mu^{(i)}(z) \right\}. \tag{3}$$

A. Bounds on the error correction capability

The following theorem, from [12], shows that the error correction capability of a code decoded with a majority-logic decoder is related to the maximum column intersection.

Theorem 1 [12] *Let us consider a code defined by a parity-check matrix for which every column has weight at least v and whose maximum column intersection is δ. Majority-logic decoding on this matrix allows the correction of all error vectors with weight $t \leq t_M$, where $t_M = \left\lfloor \frac{v}{2\delta} \right\rfloor$.*

Corollary 1 Let us consider a code with $g > 4$ defined by a parity-check matrix for which every column has weight at least v^*. Majority-logic decoding on this matrix allows the correction of all error vectors with weight $t \leq t_M$, where $t_M = \left\lfloor \frac{v^*}{2} \right\rfloor$.

Proof: It is a straightforward consequence of the fact that, if $g > 4$, the maximum column intersection is equal to 1. ■

As mentioned in the Introduction, these preliminary results are generalized in [1], where it is shown that the guaranteed error correction capability under BF decoding can actually be expressed by taking into account the interplay of more than two columns, that is, assuming $z > 1$.

Theorem 2 [1] *Let us consider a code defined by a parity-check matrix $\mathbf{H}$ in which every column has weight at least v^*. Let t be an integer such that*

$$v^* > \delta(\mathbf{H}, t) + \delta(\mathbf{H}, t-1).$$

Then a BF decoder with variable decoding thresholds

$$b_i \in [\delta(\mathbf{H}, t) + 1, v_i - \delta(\mathbf{H}, t-1)], \quad \forall i \in [0, n-1], \quad v_i \geq v^*,$$

(or fixed decoding threshold $b \in [\delta(\mathbf{H}, t) + 1, v^* - \delta(\mathbf{H}, t-1)]$) corrects all the error vectors of weight t in one iteration.

If we denote by t_M the largest integer t such that Theorem 2 is satisfied, and assume that $\delta(\mathbf{H}, i) \leq \delta(\mathbf{H}, j)$,[2] $\forall i < j \leq t_M$, then Theorem 2 allows correction of all the error vectors with weight smaller than or equal to t_M. Let us now specialize Theorem 2 to (v, w)-regular codes with girth $g > 4$. When $g > 4$, the weight of the columns of any partial parity-check matrix is either 0 or 1. In particular, any partial parity-check matrix contains one column

[2]This condition may be satisfied or not, depending on the structure of $\mathbf{H}$.

with weight v, $(w-1)v$ columns with weight 1 and $n-(w-1)v-1$ all-zero columns. As any partial parity-check matrix has v rows, it follows that

$$\delta(\mathbf{H}, z) = z \quad \forall z \leq v,$$

which is obtained by considering z different columns. Then, according to Theorem 2, we have that

$$t_M = \max_t \{t \text{ s.t. } v > t + t - 1\} = \max_t \left\{t \text{ s.t. } t \leq \left\lfloor \frac{v}{2} \right\rfloor \right\} = \left\lfloor \frac{v}{2} \right\rfloor,$$

with threshold $b = \left\lfloor \frac{v}{2} \right\rfloor + 1$ if v is even (corresponding to a majority-logic decoder), and $b \in [\left\lfloor \frac{v}{2} \right\rfloor + 1, \left\lceil \frac{v}{2} \right\rceil + 1]$ if v is odd.

In other words, when $g > 4$, Theorem 1 and Theorem 2 express the same error correction capability, with Theorem 2 giving an additional choice on the decision threshold when v is odd. When $g = 4$, instead, as proved in [1], the bound given in Theorem 2 is never smaller than that given in Theorem 1, which means that the new bound is tighter.

Theorem 2 guarantees correction of all error vectors up to a given weight t_M only if $\delta(\mathbf{H}, t)$ is a non-decreasing function for all $t \leq t_M$. This assumption is reasonable for sparse parity-check matrices, but it may be not verified for any choice of $\mathbf{H}$; thus, we state the following Theorem 3, based on the adjacency matrix $\mathbf{\Gamma}$, which does not rely on any assumption. Theorem 3 provides an upper bound on the error correction capability that is smaller than or equal to the one given by Theorem 2, but larger than or equal to the one given by Theorem 1.

Theorem 3 *Let us consider a code defined by a parity-check matrix* $\mathbf{H}$ *in which every column has weight at least* v^**. Let* t *be an integer smaller than or equal to* t_M*, where* t_M *is the largest integer such that*

$$v^* > \mu(\mathbf{H}, t_M) + \mu(\mathbf{H}, t_M - 1). \tag{4}$$

Then a BF decoder with decoding thresholds

$$b_i \in [\mu(\mathbf{H}, t) + 1, v_i - \mu(\mathbf{H}, t - 1)] \tag{5}$$

corrects all the error vectors of weight smaller than or equal to t *in one iteration.*

Proof: Let σ_i denote the number of unsatisfied parity-check equations in which the i-th bit participates, and v_i denote the weight of the i-th column in $\mathbf{H}$. Let us denote by $\mathbf{e}$ the error vector and assume that $\mathrm{wt}\,(\mathbf{e}) = t$; if $e_i = 1$, then we have

$$\begin{aligned}\sigma_i^{(1)} &= v_i - \mathrm{wt}\left(\bigoplus_{j\in S(\mathsf{e})\setminus i} \mathbf{h}_j^{(i)}\right)\\ &\geq v_i - \sum_{j\in S(\mathsf{e})\setminus i} \gamma_{i,j}\\ &\geq v_i - \mu(\mathbf{H}, t-1).\end{aligned} \tag{6}$$

In the same way, when the i-th bit is error free, that is, $e_i = 0$, we have

$$\begin{aligned}\sigma_i^{(0)} &= \mathrm{wt}\left(\bigoplus_{j\in S(\mathsf{e})} \mathbf{h}_j^{(i)}\right)\\ &\leq \sum_{j\in S(\mathsf{e})} \gamma_{i,j}\\ &\leq \mu(\mathbf{H}, t).\end{aligned} \tag{7}$$

Clearly, one iteration of BF decoding can correct any error vector $\mathbf{e}$ of weight t if, $\forall i$, there exists a value of b_i such that

$$\min_{\mathsf{e}}\{\sigma_i^{(1)}\} \geq b_i > \max_{\mathsf{e}}\{\sigma_j^{(0)}\},\ \ \forall i \in S(\mathbf{e}),\ \ \forall j \notin S(\mathbf{e}). \tag{8}$$

Inserting (6) and (7) into (8), we obtain

$$v_i - \mu(\mathbf{H}, t-1) \geq b_i > \mu(\mathbf{H}, t), \tag{9}$$

which implies

$$v^* - \mu(\mathbf{H}, t-1) > \mu(\mathbf{H}, t). \tag{10}$$

According to (9), any $b_i \in [\mu(\mathbf{H}, t) + 1, v_i - \mu(\mathbf{H}, t-1)]$ guarantees that all bits such that $e_i = 0$ are characterized by values of $\sigma_i^{(0)}$ that never exceed b_i and, thus, are not flipped; oppositely, all bits such that $e_i = 1$ are characterized by values of $\sigma_i^{(1)}$ larger than or equal to b_i, and thus are flipped. ■

B. Comparison with previous approaches

In [8], explicit formulas for bounds on the error correction capability are presented, thus we use them as a benchmark for our approach. We remark that our bounds are referred to a single decoding iteration, whereas those in [8] are referred to an unspecified number of

decoding iterations. Despite this, as shown in the following, for small values of g our bounds are tighter than those in [8]. The latter are specified through the following theorem.

Theorem 4 [8] *For a code defined by a parity-check matrix* $\mathbf{H}$ *with girth* g *in which every column has weight* v*, BF decoding with decoding threshold* $b = \left\lfloor \frac{v}{2} \right\rfloor + 1$ *allows correction of all error patterns of weight less than*

$$\begin{cases} \frac{1}{2} + \frac{v}{4} \sum_{i=0}^{k-1} \left(\frac{v-2}{2}\right)^i & \text{if } g = 4k+2, \\ \sum_{i=0}^{k-1} \left(\frac{v-2}{2}\right)^i & \text{if } g = 4k. \end{cases} \tag{11}$$

For $g = 4$, $g = 6$ and $g = 8$, the bounds on the error correction capability computed according to (11) are 0, $\lceil \frac{v+2}{4} \rceil - 1$ and $\lceil \frac{v}{2} \rceil - 1$, respectively. So, for $g = 4$ (11) is useless. On the contrary, the error correction capability given by Theorem 2 is not null on condition that $\delta(\mathbf{H}, 0) + \delta(\mathbf{H}, 1) < v$, that is, being $\delta(\mathbf{H}, 0) = 0$ by definition, if $\delta < v$. So, contrary to (11), as long as $\mathbf{H}$ does not contain repeated columns, Theorem 2 guarantees a significant error correction capability, just after one decoding iteration. Several examples are reported in [1], where it is shown that even the values resulting from Theorem 3 (that, we remind, are more conservative than those from Theorem 2) are often significantly larger than those obtained from Theorem 1.

For $g = 6$, we have $\delta = 1$ and the error correction capability given by Theorem 2 coincides with that given by Theorem 3, resulting in $t_M = \lfloor \frac{v}{2} \rfloor \geq \lceil \frac{v+2}{4} \rceil - 1$. Notice that the previous inequality, which compares the error correction capability given in Theorem 3 (left hand side) and that resulting from (11) (right hand side), holds with the equality sign only for $v = 1$ and $v = 3$. To be more explicit, the gap between the correction capability foreseen by Theorem 2 and that obtained through (11) becomes higher and higher for increasing v, which is a significant issue in view of the application to code-based cryptography, where v may assume relatively large values. Finally, for $g = 8$, Theorem 2 and Theorem 3 result in $\lfloor \frac{v}{2} \rfloor$, whereas (11) results in $\lceil \frac{v}{2} \rceil - 1$. So, since $\lfloor \frac{v}{2} \rfloor - \left(\lceil \frac{v}{2} \rceil - 1 \right) = 1 - v \bmod 2$, the bounds are the same for odd values of v, whereas the bound we provide in Theorem 2 and Theorem 3 is larger by 1 than that given in (11) for even values of v.

The comparison between the bounds we propose and those in [8] is summarized in Table I, where by "range of improvement" we mean the values of v for which our bound is strictly tighter than that in [8]. The case of $g = 10$ has been also included in the table, for which the advantage of our approach is limited to the case of $v = 2$. The advantage disappears for $g > 10$ that, however, is not of interest in this paper.

TABLE I

COMPARISON OF BOUNDS ON THE ERROR CORRECTION CAPABILITY OF LDPC AND MDPC CODES FOR DIFFERENT VALUES OF THE GIRTH.

g	Bound on t_M given by Theorem 2	Eq. (11)	Range of improvement
4	$\geq \lfloor \frac{v}{2\delta} \rfloor$	0	$\forall v$
6	$\lfloor \frac{v}{2} \rfloor$	$\lceil \frac{v+2}{4} \rceil - 1$	$\forall v \neq 1, 3$
8	$\lfloor \frac{v}{2} \rfloor$	$\lceil \frac{v}{2} \rceil - 1$	$\forall v > 2$, v even
10	$\lfloor \frac{v}{2} \rfloor$	$\lceil \frac{v^2+4}{8} \rceil - 1$	$v = 2$

So, based on the above considerations, we can conclude that the major impact of the present analysis and, similarly, of the analyses in [1], [12], occurs for codes with $g = 4$ and $g = 6$.

IV. ANALYSIS OF THE DECODING FAILURE PROBABILITY FOR THE FIRST ITERATION OF BF DECODING

In this section we derive a conservative bound for the decoding failure probability, denoted as P_f,[3] of the first and only iteration of a BF decoder, with decoding thresholds $[b_0, b_1, \cdots, b_{n-1}]$, applied on a syndrome $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$, where $\mathbf{e} \leftarrow \mathcal{B}_t$. Having a fixed number of errors (t) is a scenario of interest in code-based cryptography, in which encryption is performed by intentionally corrupting a codeword with a constant number of errors. Nevertheless, once having characterized the decoder performance for a given number of errors, it is easy to extend such a characterization to channel models (like the binary symmetric channel (BSC)) in which the statistic of the number of errors is known. In fact, a BSC with crossover probability ρ can be straightforwardly studied by considering that the probability that the channel introduces exactly t errors is equal to $\Pr\{\mathrm{wt}(\mathbf{e}) = t\} = \binom{n}{t}\rho^t(1-\rho)^{n-t}$. So, denoting the error vector after the first iteration as $\mathbf{e}'$, the decoding failure probability over the BSC can be computed as

$$P_f = \sum_{l=0}^{n} \Pr\left\{\mathbf{e}' \neq \mathbf{e} \mid \mathrm{wt}(\mathbf{e}) = l\right\} \Pr\left\{\mathrm{wt}(\mathbf{e}) = l\right\}, \tag{12}$$

where $\Pr\left\{\mathbf{e}' \neq \mathbf{e} \mid \mathrm{wt}(\mathbf{e}) = l\right\}$ can be upper bounded through the method we describe next. $\Pr\left\{\mathrm{wt}(\mathbf{e}) = l\right\}$, instead, defines the adopted channel model. For the sake of conciseness, we

[3]Notice that the decoding failure probability coincides with the expected value of the frame error rate (FER).

only study the case in which

$$\begin{cases} \Pr\{\mathrm{wt}(\mathbf{e}) = l\} = 1, & l = t, \\ \Pr\{\mathrm{wt}(\mathbf{e}) = l\} = 0, & \forall l \neq t. \end{cases}$$

that models the application to code-based cryptography (where a fixed number t of intentional errors is used for encryption). However, our analysis can be easily extended to other channel models (like the BSC) by changing the definition of $\Pr\{\mathrm{wt}(\mathbf{e}) = l\}$.

For $i \in [0, n-1]$, we define f_i as the binary variable obtained through the following rule

$$f_i = \begin{cases} 0 & \text{if } [(\sigma_i < b_i) \wedge (e_i = 0)] \vee [(\sigma_i \geq b_i) \wedge (e_i = 1)], \\ 1 & \text{if } [(\sigma_i \geq b_i) \wedge (e_i = 0)] \vee [(\sigma_i < b_i) \wedge (e_i = 1)]. \end{cases} \tag{13}$$

In other words, when $f_i = 0$, the decoder takes a right decision on the i-th bit, i.e., it flips a bit affected by an error or it does not flip an error-free bit. Conversely, when $f_i = 1$, the decoder takes a wrong decision on the i-th bit; a wrong decision can either be the flip of an error-free bit or the missing flip of a bit affected by an error. The error patterns that cause a decoding error in the i-th position, that is, those for which $f_i = 1$, are defined by the so-called *error sets*, which we introduce below.

Definition 4 *Let $\mathbf{H} \in \mathbb{F}_2^{r \times n}$ be the parity-check matrix of a code with block length n. We consider the first and only iteration of a BF decoder as in Algorithm 1, with decoding thresholds $[b_0, \cdots, b_{n-1}]$. Let f_i be the binary variable defined as in* (13)*, for $i \in [0, n-1]$. Then, for $z \in \{0, 1\}$, we define the* error set *for the i-th bit as follows*

$$\mathcal{E}_{i,t,b_i}^{z} = \{\mathbf{e} \in \mathcal{B}_t \ \ s.t. \ \ f_i = 1 | \, e_i = z\}.$$

As we show in the following section, the cardinality of each error set represents a fundamental quantity for assessing the error correction capability of the first iteration of a BF decoder as in Algorithm 1. Notice that the cardinality computation for each error set is strictly related to a subset sum problem, which in our case can be defined as follows: for a generic set, determine the number of subsets with given size having the property that the sum of their entries exceeds some target value. The precise subset sum problem variant that we consider in this paper is formalized in the following definition.

Definition 5 *Let $\mathbf{a} \in \mathbb{N}^l$ be a length-l vector. For $m \leq l$, let $P_{l,m} = \{p_0, \cdots, p_{m-1}\}$ be a size-m set of distinct integers in $[0, l-1]$ such that $p_0 < p_1 < \cdots p_{m-1}$. Let $\mathcal{P}_{l,m}$ be the ensemble containing all such sets; clearly, $|\mathcal{P}_{l,m}| = \binom{l}{m}$. For $\alpha \in \mathbb{N}$, we define*

$$\mathcal{N}_{m,\alpha}^{\mathsf{a}} = \left\{ P_{l,m} \in \mathcal{P}_{l,m} \ s.t. \ \sum_{i=0}^{m-1} a_{p_i} > \alpha \right\}.$$

A. Decoding failure probability analysis based on the error sets

Let us introduce a property of the error sets that will then be used to derive the main result reported in Theorem 5.

Lemma 1 Let $\mathbf{H} \in \mathbb{F}_2^{r \times n}$ be a parity-check matrix, and let $\mathcal{E}^z_{i,t,b_i}$, for $z \in \{0,1\}$, be the error set for the i-th bit. We denote with $\tilde{\boldsymbol{\gamma}}^{(i)}$ the vector formed by the entries of the i-th row of the adjacency matrix $\boldsymbol{\Gamma}$, defined in Section II, except for the i-th one. Then, we have

$$\left|\mathcal{E}^1_{i,t,b_i}\right| \leq \left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t-1,v_i-b_i}\right|, \tag{14}$$

$$\left|\mathcal{E}^0_{i,t,b_i}\right| \leq \left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t,b_i-1}\right|. \tag{15}$$

Proof: We focus on the i-th bit, characterized by a certain value of σ_i and flipping threshold b_i, and derive the conditions upon which the decoder takes a wrong decision (i.e., $f_i = 1$). We first consider the case of $e_i = 1$: a wrong decision is taken if the decoder does not flip the bit, i.e., if $\sigma_i < b_i$. From (6), we know that the value of σ_i is not lower than the difference between the weight of the i-th column (that is, v_i) and the sum of the values $\gamma_{i,j}$ indexed by $S(\mathbf{e})$, except the i-th index (that is, $\sum_{j \in S(\mathbf{e})\setminus i} \gamma_{i,j}$). If such a difference is not lower than b_i, then $\sigma_i \geq b_i$ and the decoder flips the i-th bit. On the other hand, if $v_i - \sum_{j \in S(\mathbf{e})\setminus i} \gamma_{i,j} < b_i$, σ_i might be lower than b_i and the decoder might not flip the i-th bit. Hence, a necessary (but not sufficient) condition to have a wrong decision on the i-th bit is

$$\sum_{j \in S(\mathbf{e})\setminus i} \gamma_{i,j} > v_i - b_i. \tag{16}$$

Because of the above reasoning, $\mathcal{E}^1_{i,t,b_i}$ is a subset of the error vectors satisfying (16). The set $S(\mathbf{e}) \setminus i$ in (16) corresponds to a subset of $[0, i-1] \cup [i+1, n-1]$, of size $t-1$; furthermore, the values $\gamma_{i,j}$ that are possibly selected by $S(\mathbf{e}) \setminus i$ are entries of $\tilde{\boldsymbol{\gamma}}^{(i)} = [\gamma_{i,0}, \cdots, \gamma_{i,i-1}, \gamma_{i,i+1}, \cdots, \gamma_{i,n-1}]$, which has length $n-1$. Let $P_{n-1,t-1}$ be a subset of $[0, n-1]$ such that the sum of the entries in $\tilde{\boldsymbol{\gamma}}^{(i)}$ indexed by $P_{n-1,t-1}$ is larger than $v_i - b_i$. According to Definition 4, the number of such sets corresponds to the cardinality of $\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t-1,v_i-b_1}$. Furthermore, to each one of these subsets, we can associate an error vector satisfying (16), with support

$$\left\{j \in P_{n-1,t-1} \,|j < i\right\} \cup i \cup \left\{j+1 \in P_{n-1,t-1} \,|j > i\right\}.$$

Thus, we obtain

$$\begin{aligned}\left|\mathcal{E}^1_{i,t,b_i}\right| &\leq \left|\left\{\mathbf{e} \in \mathcal{B}_t \text{ s.t. } (e_i = 1) \wedge \left(\sum_{j \in S(\mathbf{e})\setminus i} \gamma_{i,j} > v_i - b_i\right)\right\}\right| \\ &= \left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t-1,v_i-b_i}\right|.\end{aligned}$$

Similarly, for the case of $e_i = 0$, we can derive from (7) that a necessary but not sufficient condition for $f_i = 1$ is $b_i \leq \sigma_i \leq \sum_{j \in S(e)} \gamma_{i,j}$. Similarly to the case of $e_1 = 1$, we have

$$\begin{aligned}\left|\mathcal{E}^0_{i,t,b_i}\right| &\leq \left| \left\{ \mathbf{e} \in \mathcal{B}_t \text{ s.t. } (e_i = 0) \wedge \left(\sum_{j \in S(e)} \gamma_{i,j} > b_i - 1 \right) \right\} \right| \\ &= \left| \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,b_i-1} \right|.\end{aligned}$$

■

Based on these relationships, we can now prove the following main theorem.

Theorem 5 *Let $\mathbf{H} \in \mathbb{F}_2^{r \times n}$ be a parity-check matrix. Let $\mathbf{e} \in \mathcal{B}_t$, and $\mathbf{s} = \mathbf{e}\mathbf{H}^T$ be the corresponding syndrome. We consider a single BF iteration applied on $\mathbf{s}$, with decoding threshold for the i-th bit denoted as b_i. Let $\tilde{\boldsymbol{\gamma}}^{(i)}$ denote the vector formed by the elements in the i-th row of $\boldsymbol{\Gamma}$, except for the i-th one. The probability that the decoder fails to decode, starting from $\mathbf{s}$, is upper bounded as follows*

$$P_f \leq \min \left\{ 1; \frac{\sum_{i=0}^{n-1} \left(|\mathcal{N}^{\tilde{\gamma}^{(i)}}_{t-1,v_i-b_i}| + |\mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,b_i-1}| \right)}{\binom{n}{t}} \right\}. \tag{17}$$

Proof: Let us start from an arbitrary position $i \in [0, n-1]$. Let $\mathcal{E}_{i,t,b_i}$ be the set of error vectors of weight t such that, when the decoding threshold for the i-th bit is b_i, the decoder decision results in $f_i = 1$ (i.e., the decoder flips the bit if $e_i = 0$ or does not flip the bit if $e_i = 1$). Clearly $\mathcal{E}_{i,t,b_i} = \mathcal{E}^0_{i,t,b_i} \cup \mathcal{E}^1_{i,t,b_i}$. Moreover, the sets $\mathcal{E}^1_{i,t,b_i}$ and $\mathcal{E}^0_{i,t,b_i}$ are disjoint, since the vectors in $\mathbf{e} \in \mathcal{E}^1_{i,t,b_i}$ are such that $e_i = 1$ and those in $\mathcal{E}^0_{i,t,b_i}$ are such that $e_i = 0$. Taking into account (14) and (15), we obtain

$$\begin{aligned}|\mathcal{E}_{i,t,b_i}| &= \left|\mathcal{E}^0_{i,t,b_i}\right| + \left|\mathcal{E}^1_{i,t,b_i}\right| \\ &\leq \left| \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t-1,v_i-b_i} \right| + \left| \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,b_i-1} \right|.\end{aligned} \tag{18}$$

For all values $j \in [0, n-1]$ such that $\mathcal{E}_{j,t,b_j}$ contains $\mathbf{e}$, we have $f_j = 1$, i.e., a wrong decoder decision is taken on the j-th bit. Then, the probability that decoding fails can be upper bounded by means of the following chain of inequalities

$$\begin{aligned}P_f &= \frac{\left|\bigcup_{i=0}^{n-1} \mathcal{E}_{i,t,b_i}\right|}{|\mathcal{B}_t|} \\ &\leq \frac{\sum_{i=0}^{n-1} |\mathcal{E}_{i,t,b_i}|}{|\mathcal{B}_t|} \\ &\leq \frac{\sum_{i=0}^{n-1} \left(\left| \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t-1,v_i-b_i} \right| + \left| \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,b_i} \right| \right)}{|\mathcal{B}_t|}.\end{aligned} \tag{19}$$

The thesis of the theorem is finally proved by considering that $|\mathcal{B}_t| = \binom{n}{t}$ and that, by definition, $P_f \leq 1$ (while the bound in (19) is not guaranteed to be smaller than or equal to 1). ■

In order to compute the bound given in the theorem above, we need to solve instances of the subset sum problem according to Definition 5. Clearly, the naive approach of testing all possible subsets of vectors $\tilde{\boldsymbol{\gamma}}^{(i)}$ is computationally unfeasible. Fortunately, in our case of interest, the problem can be eased by considering that, due to the sparsity of the parity-check matrix, $\tilde{\boldsymbol{\gamma}}^{(i)}$ is likely to contain a large number of very small entries (the majority of which being actually null). This peculiarity of sparsity makes the problem efficiently solvable; a low complexity approach to perform this computation is described in Appendix A.

The expression of P_f derived above is coherent with the results given in Section III-A and, in particular, in Theorem 3. Indeed, the following corollary holds.

Corollary 2 Let us suppose that $t \leq t_M$, where t_M is the largest integer such that (4) holds. If the decoding threshold is chosen as follows

$$b_i \in \left[\mu(\mathbf{H}, t) + 1, v_i - \mu(\mathbf{H}, t-1)\right], \quad \forall i, \tag{20}$$

then $\left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t-1, v_i - b_i}\right| = \left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t, b_i}\right| = 0$, $\forall i$ and, consequently, $P_f = 0$.

Proof: By definition,

$$\begin{aligned}\left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t-1, v_i - b_i}\right| &= \left|\left\{P_{n,t-1} \in \mathcal{P}_{n,t-1} \text{ s.t. } \sum_{i=0}^{t-2} \tilde{\boldsymbol{\gamma}}^{(i)}_{p_i} > v_i - b_i\right\}\right| \\ &= \left|\left\{P_{n,t-1} \in \mathcal{P}_{n,t-1} \text{ s.t. } b_i > v_i - \sum_{i=0}^{t-2} \tilde{\boldsymbol{\gamma}}^{(i)}_{p_i}\right\}\right|.\end{aligned}$$

However, it follows from the definition of $\mu(\mathbf{H}, t-1)$ and from (20) that

$$b_i \leq v_i - \mu(\mathbf{H}, t-1) \leq v_i - \sum_{i=0}^{t-2} \tilde{\boldsymbol{\gamma}}^{(i)}_{p_i}$$

for any choice of the indexes p_i and, thus, $\left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t-1, v_i - b_i}\right| = 0$. Similarly, we have

$$\begin{aligned}\left|\mathcal{N}^{\tilde{\boldsymbol{\gamma}}^{(i)}}_{t, b_i - 1}\right| &= \left|\left\{P_{n,t} \in \mathcal{P}_{n,t} \text{ s.t. } \sum_{i=0}^{t-1} \tilde{\boldsymbol{\gamma}}^{(i)}_{p_i} > b_i - 1\right\}\right| \\ &= \left|\left\{P_{n,t-1} \in \mathcal{P}_{n,t-1} \text{ s.t. } b_i < \sum_{i=0}^{t-1} \tilde{\boldsymbol{\gamma}}^{(i)}_{p_i} + 1\right\}\right|.\end{aligned}$$

It also follows from (20) that

$$b_i \geq \mu(\mathbf{H}, t) + 1 \geq \sum_{i=0}^{t-1} \tilde{\boldsymbol{\gamma}}_{p_i}^{(i)} + 1$$

for any choice of the indexes p_i, and thus $\left|\mathcal{N}_{t,b_i-1}^{\tilde{\boldsymbol{\gamma}}^{(i)}}\right| = 0$. Finally, the fact that $P_f = 0$ is a straightforward consequence of (17). ■

In the particular case of regular codes, which implies to have equal decoding threshold values, noted as b, assuming v is odd and $b = \lceil \frac{v}{2} \rceil$, the bound on P_f provided by Theorem 5 can be rewritten as

$$P_f \leq \min \left\{ 1; \frac{\sum_{i=0}^{n-1} \left|\mathcal{N}_{t,\frac{v-1}{2}}^{\boldsymbol{\gamma}^{(i)}}\right|}{\binom{n}{t}} \right\}. \tag{21}$$

The proof is reported in Appendix B.

Equation (21) can be used for any regular code with $g \geq 4$. For regular codes with $g \geq 6$, however, (21) can be further elaborated as discussed next.

B. Regular codes with girth larger than 4

When $g \geq 6$, we have

$$\gamma_{i,j} \in \{0, 1\}, \ \ \forall i, j. \tag{22}$$

In particular, for (v, w)-regular codes, each row and each column of $\boldsymbol{\Gamma}$ contain exactly $v(w-1)$ non-zero entries. The following lemma holds.

Lemma 2 Let $\mathbf{a} \in \mathbb{F}_2^l$ be a vector of weight m; then, we have $\left|\mathcal{N}_{x,\alpha}^{\mathtt{a}}\right| = \theta(l, x, m, \alpha)$, with

$$\theta(l, x, m, \alpha) = \begin{cases} 0 & \text{if } \alpha \geq m \quad \text{or} \quad x \leq \alpha \\ \sum_{j=\alpha+1}^{\min\{m,x\}} \binom{m}{j}\binom{l-m}{x-j} & \text{otherwise} \end{cases}. \tag{23}$$

The following Theorem 6 specializes Theorem 5 to the case of a regular code with girth larger than 4, and reformulates (21) for such a case.

Theorem 6 *Let $\mathbf{H} \in \mathbb{F}_2^{r \times n}$ be the parity-check matrix of a (v, w)-regular code with girth $g \geq 6$. Let $\mathbf{e} \in \mathcal{B}_t$, and $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$. We consider a single iteration of BF decoding applied to $\mathbf{s}$, with a unique decoding threshold b. If v is odd and $b = \lceil \frac{v}{2} \rceil$, we have*

$$\begin{cases} P_f = 0 & \textit{if } t \leq \frac{v-1}{2} \\ P_f \leq \min \left\{ 1; \frac{n\theta(n,t,v(w-1),\frac{v-1}{2})}{\binom{n}{t}} \right\} & \textit{otherwise} \end{cases}, \tag{24}$$

where, using (23)*,*

$$\theta\Big(n,t,v(w-1),\frac{v-1}{2}\Big)=\sum_{j=\frac{v+1}{2}}^{\min\{v(w-1),t\}}\binom{v(w-1)}{j}\binom{n-v(w-1)}{t-j}.$$

Proof: The proof is quite similar to that of Theorem 5 and its specialization to the case of regular codes (reported in Appendix B), by taking into account Lemma 2. ■

C. A special class of QC codes

In this section we consider QC codes with parity-check matrix as in (1), which is interesting for cryptographic applications, as will be discussed in Section V. By considering the QC nature of these codes, described by parity-check matrices made of circulant blocks, the bounds introduced in the previous sections can be further specialized. It can be easily verified that, for these codes, the matrix $\mathbf{\Gamma}$ is QC as well; this property can be exploited to further speed-up the computation of the error sets required to calculate the bounds.

The following well-known result holds.

Lemma 3 Any circulant matrix with weight larger than 2 has girth $g\leq 6$.

Proof: The proof is omitted for brevity. See [27, Lemma 4.2]. ■

It follows from Lemma 3 that a parity-check matrix as in (1) cannot have girth larger than 6.

In this case, the matrix $\mathbf{\Gamma}$ can be written as

$$\mathbf{\Gamma}=\begin{bmatrix}\mathbf{\Gamma}_{0,0} & \mathbf{\Gamma}_{0,1}\\ \mathbf{\Gamma}_{1,0} & \mathbf{\Gamma}_{1,1}\end{bmatrix}, \tag{25}$$

where each $\mathbf{\Gamma}_{i,j}$ is a $p\times p$ matrix; in particular, $\mathbf{\Gamma}$ is symmetric, and this means that $\mathbf{\Gamma}_{0,0}$ and $\mathbf{\Gamma}_{1,1}$ are symmetric as well, while $\mathbf{\Gamma}_{0,1}^{\top}=\mathbf{\Gamma}_{1,0}$. Moreover, each block $\mathbf{\Gamma}_{i,j}$ is circulant. In particular, let $\boldsymbol{\gamma}^{(i)}$ be the i-th row of $\mathbf{\Gamma}$; then, all rows $\boldsymbol{\gamma}^{(j)}$ such that $\lfloor i/p\rfloor=\lfloor j/p\rfloor$ are identical up to a quasi-cyclic shift; this means that

$$\left|\mathcal{E}_{i,t,b}^{z}\right|=\left|\mathcal{E}_{j,t,b}^{z}\right|,\ \ \forall b,t,\ \ \forall i,j \text{ s.t. } \lfloor i/p\rfloor=\lfloor j/p\rfloor, \tag{26}$$

with $z\in\{0,1\}$. Then, from Theorem 5 we obtain

$$P_f\leq\min\left\{1;p\,\frac{\mathcal{N}_{\text{tot}}}{\binom{n}{t}}\right\}, \tag{27}$$

with

$$\mathcal{N}_{\text{tot}}=\left|\mathcal{N}_{t-1,v-b}^{\tilde{\gamma}^{(0)}}\right|+\left|\mathcal{N}_{t,b-1}^{\tilde{\gamma}^{(0)}}\right|+\left|\mathcal{N}_{t-1,v-b}^{\tilde{\gamma}^{(p)}}\right|+\left|\mathcal{N}_{t,b-1}^{\tilde{\gamma}^{(p)}}\right|.$$

V. Application to cryptography

In this section we assess the accuracy of our bound through numerical simulations. Then, we make some considerations on the connections of the proposed bound with the security levels of code-based cryptosystems.

A. Numerical simulations

There is a recent trend in post-quantum cryptography regarding the use of quasi-cyclic low-density parity-check (QC-LDPC) and quasi-cyclic moderate-density parity-check (QC-MDPC) codes [15], [16], [28] defined in Section IV-C, since they enable the design of McEliece cryptosystem variants with very small public keys. We remark that, in code-based cryptography, a decoding failure yields a decryption failure; thus, the FER coincides with the so-called decryption failure rate (DFR).

Let us first consider some codes defined by parity-check matrices as in (1). In order to show the tightness of the provided bounds, let us consider different choices of code parameters. First, we analyze some specifically designed codes, whose column weight is chosen in such a way as to approach or reach the expected guaranteed error correction capability through Monte Carlo simulations. Then, we also consider codes that have actually been proposed for cryptographic applications, whose column weight must be sufficiently large to withstand key recovery attacks [16, Section 5.2].

In order to assess the behaviour of codes with similar parameters and different girth, let us consider a first code, $\mathcal{C}_0$, with length $n = 19\,702$, design rate $R = \frac{1}{2}$, $p = 9\,851$, $v = 25$, $g = 4$ and a second code, $\mathcal{C}_1$, with $n = 17\,558$, design rate $R = \frac{1}{2}$, $p = 8\,779$, $v = 13$ and girth $g = 6$. A compact representation of their parity-check matrices is available in Appendix C. We assess the DFR achieved by a single-iteration BF decoder with different threshold values through Monte Carlo simulations; for each value of t, the DFR has been estimated through the observation of 100 wrong decoding instances. The comparison of the simulation results with our bounds is shown in Figs. 1 and 2, respectively. From the figures we observe that for both codes the bound becomes tighter and tighter for decreasing values of t.

Let us now consider a $(45, 90)$-regular code, $\mathcal{C}_2$, with block length $n = 9\,602$, circulant block size $p = 4\,801$, design rate $R = \frac{1}{2}$ and girth $g = 4$. These parameters are suitable for cryptographic applications [16]. A compact representation of its parity-check matrices is available in Appendix C. Also in this case, its error rate performance is compared to the bound, considering different thresholds. The results are shown in Fig. 3. We notice that, also in this case, the bound becomes tighter and tighter for decreasing values of t.

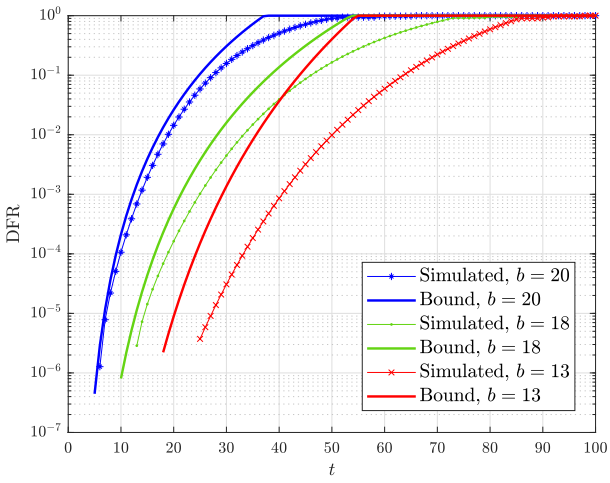

Fig. 1. Comparison of the DFR resulting from Monte Carlo simulations with our bound for a $(25, 50)$-regular code with block length $n = 19\,702$, $R = \frac{1}{2}$, $p = 9\,851$, $v = 25$, $g = 4$, and different threshold values.

In order to assess the effect of the parity-check matrix column weight, let us consider three $(v, 2v)$-regular codes, $\mathcal{C}_3$, $\mathcal{C}_4$ and $\mathcal{C}_5$, defined by parity-check matrices as in (1), with the same block length, $n = 23\,434$, circulant block size $p = 11\,717$, and design rate $R = \frac{1}{2}$, but different values of the column weight: $v = 9$, $v = 15$ and $v = 47$ for $\mathcal{C}_3$, $\mathcal{C}_4$ and $\mathcal{C}_5$, respectively. A compact representation of their parity-check matrices is available in Appendix C. The decoding threshold is chosen as $b = \lfloor \frac{v}{2} \rfloor + 1$. The simulation results are shown in Fig. 4. Also in these cases, the bound becomes tighter and tighter for decreasing values of t. We also remark that the bound is tight for both LDPC and MDPC codes; in fact, $\mathcal{C}_3$ and $\mathcal{C}_4$ are LDPC codes, whereas $\mathcal{C}_5$ is an MDPC code.

In order to assess the effect of the block length, let us fix the parity-check matrix row and column weight and consider three $(25, 50)$-regular codes, $\mathcal{C}_6$, $\mathcal{C}_7$ and $\mathcal{C}_8$, defined by parity-check matrices as in (1), with block length $n = 9\,946$, $n = 13\,766$ and $n = 29\,734$, respectively. A compact representation of their parity-check matrices is available in Appendix C. Also in this case, the threshold is $b = \lfloor \frac{v}{2} \rfloor + 1$. A comparison of their DFR with the proposed bound is shown in Fig. 5. In all these cases, the bound becomes tighter and tighter for decreasing values of t, as in the previously considered cases.

Finally, let us consider a different family of codes, that is, monomial codes defined in Section II-A. It is shown in [29] that, for a proper choice of the shifts and of the code parameters, monomial codes can be used in code-based cryptosystems. Thus, we consider QC-LDPC codes of this type designed through the technique suggested in [29, Section

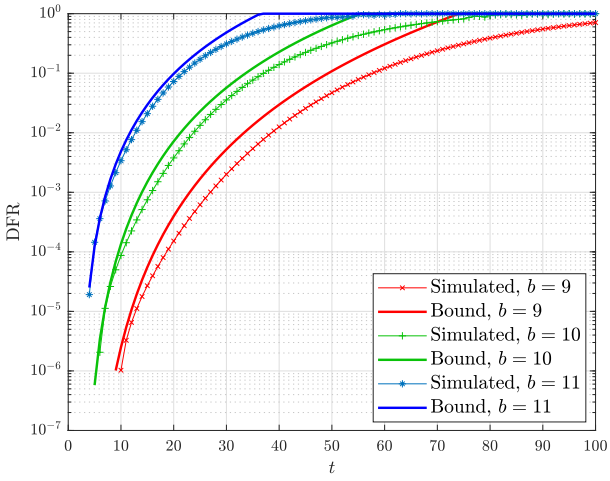

Fig. 2. Comparison of the DFR resulting from Monte Carlo simulations with our bound, for a $(13, 26)$-regular code with block length $n = 17\,558$, $R = \frac{1}{2}$, $p = 8\,779$, $v = 13$, $g = 6$, and different threshold values.

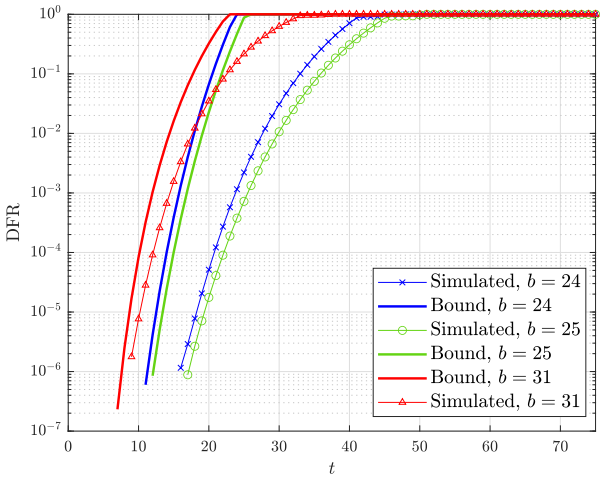

Fig. 3. Comparison of the DFR resulting from Monte Carlo simulations with our bound for a $(45, 90)$-regular code with block length $n = 9\,602$, $R = \frac{1}{2}$, $p = 4\,801$, $g = 4$, and different threshold values.

IV-C] with some modifications, in such a way as to obtain codes with variable rate and row/column weight. These codes have girth 6 and design rate $R = 1 - \frac{v}{w}$, and we assess their error rate performance considering $b = \lceil \frac{v}{2} \rceil$, as imposed by Theorem 6. In particular, let us consider three parameter sets, described in Table II, and for each parameter set, i.e., for each code ensemble, we randomly generate three monomial codes and compare their error rate performance with the bound given by (24). Results are shown in Fig. 6. We observe that there is no appreciable difference between the performance of codes in the same ensemble. We also observe that the bound is tight for monomial codes as well.

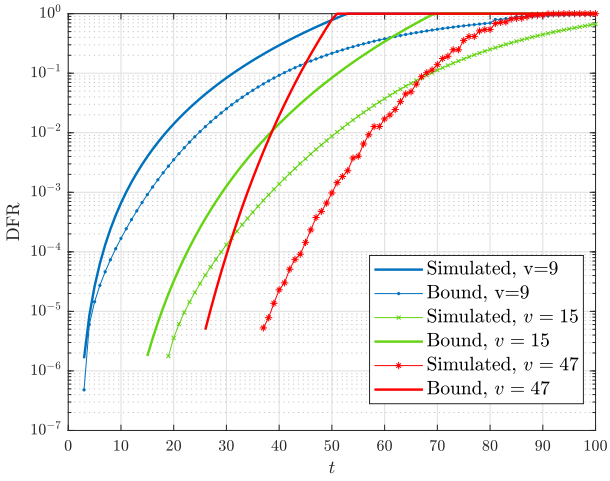

Fig. 4. Comparison of the DFR resulting from Monte Carlo simulations with our bound, for $(v, 2v)$-regular codes with block length $n = 23\,434$, $R = \frac{1}{2}$, $p = 11\,717$, $v \in \{9, 15, 47\}$, $g = 4$, and $b = \lfloor \frac{v}{2} \rfloor + 1$.

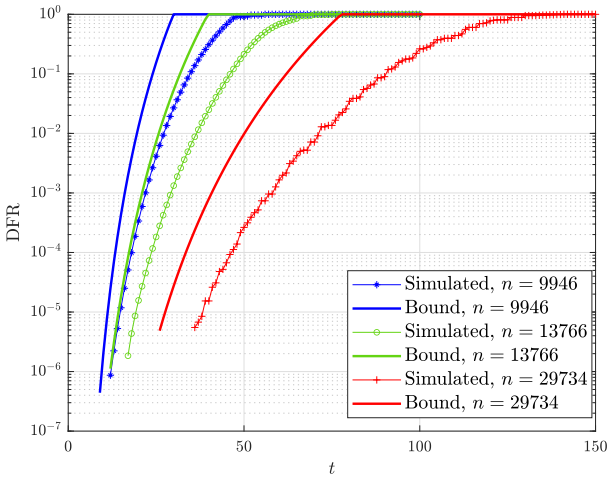

Fig. 5. Comparison of the DFR resulting from Monte Carlo simulations with our bound, for $(25, 50)$-regular codes with block length $n \in \{9\,946, 13\,766, 29\,734\}$, $R = \frac{1}{2}$, $p = \frac{n}{2}$, $g = 4$, and $b = 13$.

TABLE II

PARAMETERS OF THE CONSIDERED MONOMIAL CODES.

Parameter Set	n	r	v	w	p	g	Design rate
# 1	4 171	1 455	15	43	97	6	0.65
# 2	8 517	5 177	31	51	167	6	0.39
# 3	3 937	2 921	23	31	127	6	0.26

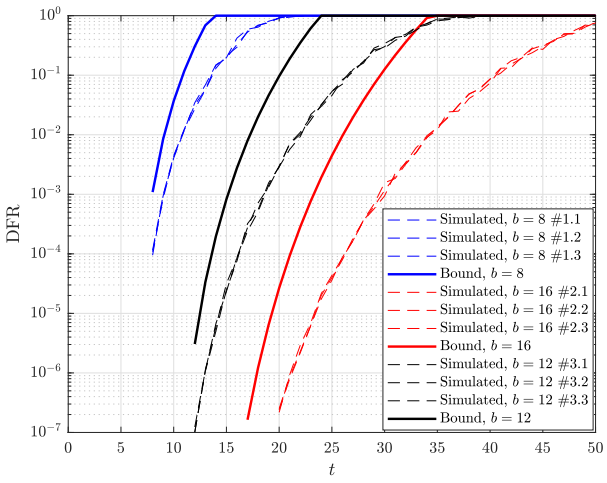

Fig. 6. Comparison of the DFR resulting from Monte Carlo simulations with our bound, for monomial codes described by the parameters in Table II.

B. Design of codes with given DFR

When codes as in (1) are used in code-based cryptosystems that support key reuse, the required values of P_f are much smaller than those reported in the figures of Section V-A, and are impossible to assess through Monte Carlo simulations. In particular, in order to avoid key recovery attacks based on decryption failures, such as those in [30], [5], also called *reaction attacks*, a cryptosystem designed for a 2^λ security level (expressed as number of binary operations) must have $\mathrm{DFR} < 2^{-\lambda}$ [31] with values of λ not smaller than 80. A negligible decoding failure probability is also required to achieve the desirable security condition known as indistinguishability under adaptive chosen ciphertext attack (IND-CCA) [31].

This makes the derived bounds particularly useful in this case. In fact, by assuming the QC code structure specified in Section IV-C, we can use (27) to design code parameters able to achieve the desired small values of P_f without requiring any simulation. To show an example, let us consider the case of a security level of 2^{80} binary operations, for which QC-MDPC codes with $v \geq 45$ and $t \geq 84$ are needed [16]. The matrices proposed in [16] have $p = 4801$, which however leads to a decoding failure probability too large to resist reaction attacks and to achieve IND-CCA. A decoding failure probability lower than 2^{-80} is instead required for such a purpose.

Indeed, the bound given in (27) allows achieving such a requirement through a classic rejection sampling approach: for each randomly generated parity-check matrix in the form (1), the bound (27) is computed and the matrix discarded if such a value is above the target

P_f. The procedure is repeated until a matrix with the desired property is obtained. In order to verify the feasibility of such an approach, we consider different parameter sets and, for each set, we generate $1\,000$ parity-check matrices at random and compute the bound on P_f given by (27). The choice of b is optimized by choosing its value for which the bound takes its minimum.

The results of this experiment are reported in Table III. We notice that, for all tested parameter sets, a significant percentage of matrices satisfies the constraint $P_f < 2^{-80}$. This fact guarantees that the time required to generate a valid matrix is limited. In other words, it is not difficult to find a matrix for which we can be sure that the desired security level is reached.

We point out that, despite the codes obtained through the above approach are significantly larger than those originally proposed, they still lead to public key sizes that are smaller than those of other competing cryptosystems, while achieving IND-CCA. For instance, considering binary Goppa codes as in the original McEliece cryptosystem, the public key size equals $460\,647$ bits [32] for 80 bits security, while the parameters we have found lead to a reduction in the public key size by a factor ranging between 1.64 and 3.57. Additionally, the parameter sets we propose represent a concrete worst case estimate of the key size increase which is needed in order to ensure IND-CCA. Indeed, we obviously expect that if more than one decoding iteration is performed, the minimum value of p which is necessary to fulfill $P_f < 2^{-80}$ decreases, thus further reducing the key size and allowing more significant improvements with respect to other cryptosystems. However, extending the bound to the case of multiple iterations goes beyond the scope of this paper and is left for future works.

TABLE III

NUMBER OF SELECTED KEYS FOR DIFFERENT PARAMETER SETS.

p	v	Keys achieving $P_f < 2^{-80}$
279 991	45	158 out of 1 000
194 989	65	990 out of 1 000
160 499	75	792 out of 1 000
149 993	85	971 out of 1 000
138 389	95	847 out of 1 000
130 043	105	226 out of 1 000

VI. Conclusion

We have studied the error correction capability of LDPC and MDPC codes under BF iterative decoding, with the aim of finding theoretical models for its characterization without resorting to computation-intensive simulations.

Under the simplifying setting of a single-iteration BF decoder, we have shown that a per-code upper bound on the error rate can indeed be found. Such a bound provides an important tool in those contexts where very small error rates have to be guaranteed for each specific code.

One of these scenarios is that of code-based cryptography, and we have shown how our bound can be succesfully applied to such a context, allowing the design of cryptosystems based on QC-LDPC and QC-MDPC codes able to achieve strong security notions while keeping the size of the public keys smaller than that of classic systems employing algebraic codes and bounded-distance decoders.

Appendix A

In this appendix we describe an efficient way to compute the cardinalities of the sets introduced in Definition 5. To this end, we first formalize the problem and then describe a method that, for the cases we are interested in, significantly improves upon the naive exhaustive search approach.

Problem 1 *Let* $\mathbf{a} \in \mathbb{N}^l$ *be a length-*l *vector of non negative integers, and let* $B \subseteq [0, l-1]$ *be a set of size* $m \leq l$*. Given* $\alpha \in \mathbb{N}$*,* $\alpha > 0$*, compute*

$$N_B = \left| \left\{ B \subseteq [0, l-1],\ |B| = m \ \textit{s.t.}\ \sum_{i \in B} a_i > \alpha \right\} \right|.$$

It is clear that an exhaustive search would require to generate all subsets of size m: thus, the corresponding complexity will be equal to $\binom{l}{m}$. As we show with combinatorial arguments, a simple algorithm can be devised, with a complexity that may be significantly lower.

In particular, we obtain the number of sets that are complementary to those defined in Problem 1, that is,

$$\bar{N}_B = \left| \left\{ B \subseteq [0, l-1],\ |B| = m \ \text{s.t.}\ \sum_{i \in B} a_i \leq \alpha \right\} \right|,$$

from which the value of N_B can be straightforwardly obtained as

$$N_B = \binom{l}{m} - \bar{N}_B. \tag{28}$$

For a set B, we denote with $\mathbf{a}^{(B)}$ the vector formed by the entries of $\mathbf{a}$ that are indexed by B; we define $\bar{N}_B^{(j)}$ as the number of subsets B for which the corresponding sub-vector $\mathbf{a}^{(B)}$ contains m elements, j of which are distinct, whose sum is smaller than or equal to α. We have

$$\bar{N}_B = \sum_{j=1}^{m} \bar{N}_B^{(j)}. \tag{29}$$

The values of $\bar{N}_B^{(j)}$ can be easily obtained, as we show next.

First of all, let ω be the number of distinct values in $\mathbf{a}$, with $Y = \{y_0, y_1, \cdots, y_{\omega-1}\}$ being the set of such values in ascending order. In the same way, we define $\lambda_u = |\{i \text{ s.t. } a_i = y_u\}|$. As we show below, the computation of $\bar{N}_B$ depends only on these quantities.

Let Y_B be the set of distinct values that are contained in $\mathbf{a}^{(B)}$. When $j = 1$, we easily have

$$\bar{N}_B^{(1)} = \sum_{0 \le i \le \omega-1 \,:\, y_i \le \left\lfloor \frac{\alpha}{m} \right\rfloor} \binom{\lambda_i}{m}, \tag{30}$$

where, as usual, $\binom{\lambda_i}{m} = 0$ if $m > \lambda_i$. When $j > 1$, some further considerations must be taken into account. For a set B, let $y_{i_0}, y_{i_1}, \cdots, y_{i_{j-1}}$ be the distinct values assumed by the entries of $\mathbf{a}^{(B)}$, and denote the corresponding multiplicities as $m_0, m_1, \cdots, m_{j-1}$. If $B \in \bar{N}_B^{(j)}$, we must have

$$\sum_{u=0}^{j-1} m_u y_{i_u} \le \alpha. \tag{31}$$

We clearly have $m = \sum_{u=0}^{j-1} m_u$, from which we obtain $m_0 = m - \sum_{u=1}^{j-1} m_u$; then, (31) can be rewritten as

$$m y_{i_0} + \sum_{u=1}^{j-1} m_u (y_{i_u} - y_{i_0}) \le \alpha. \tag{32}$$

It is obvious that

$$m y_{i_0} + \sum_{u=1}^{j-1} m_u (y_{i_u} - y_{i_0}) \ge m y_{i_0} + \sum_{u=1}^{j-1} (y_{i_u} - y_{i_0}). \tag{33}$$

The above condition can be turned into the following criterion: a set B associated to the values $y_{i_0}, y_{i_1}, \cdots, y_{i_{j-1}}$ of $\mathbf{a}^{(B)}$, whose sum is smaller than or equal to α, exists if and only if

$$\sum_{u=1}^{j-1} (y_{i_u} - y_{i_0}) \le \alpha - m y_{i_0}. \tag{34}$$

Let us now fix an index $q \in [1, j-2]$, and suppose that we are looking at all sets B such that $\mathbf{a}^{(B)}$ contains the values $y_{i_0}, \cdots, y_{i_{q-1}}$ with respective multiplicities $m_1, m_2, \cdots, m_{q-1}$. Then, imposing the constraint and summing over all subsets, we obtain

$$\begin{aligned}\alpha &\geq m y_{i_0} + \sum_{u=1}^{q-1} m_u (y_{i_u} - y_{i_0}) + m_q (y_{i_q} - y_{i_0}) + \sum_{z=q+1}^{j-1} m_z (y_{i_z} - y_{i_0}) \\ &\geq m y_{i_0} + \sum_{u=1}^{q-1} m_u (y_{i_u} - y_{i_0}) + m_q (y_{i_q} - y_{i_0}) + \sum_{z=q+1}^{j-1} (y_{i_z} - y_{i_0}).\end{aligned}$$

Then, the maximum value for m_q is obtained as

$$m_q^{(\max)} = \min \left\{ \lambda_q, \left\lfloor \frac{\alpha - m y_{i_0} - \sum_{u=1}^{q-1} m_u (y_{i_u} - y_{i_0}) - \sum_{z=q+1}^{j-1} (y_{i_z} - y_{i_0})}{y_{i_q} - y_{i_0}} \right\rfloor \right\}. \tag{35}$$

Finally, $\bar{N}_B^{(j)}$ can be computed as

$$\bar{N}_B^{(j)} = \sum_{i_0=0}^{\omega-j} \sum_{i_1=i_0+1}^{\omega-j+1} \cdots \sum_{i_{j-1}=i_{j-2}+1}^{\omega-1} d(i_0, \cdots, i_{j-1}), \tag{36}$$

where

$$d(i_0, \cdots, i_{j-1}) = \begin{cases} 0 & \text{if } \sum_{u=1}^{j-1} (y_{i_u} - y_{i_0}) > \alpha - m y_{i_0} \\ \sum_{m_1=1}^{m_1^{(\max)}} \cdots \sum_{m_{j-1}=1}^{m_{j-1}^{(\max)}} \binom{\lambda_{i_0}}{m - \sum_{i=1}^{j-1} m_i} \prod_{u=1}^{j-1} \binom{\lambda_{i_u}}{m_u} & \text{otherwise} \end{cases}. \tag{37}$$

We point out that when $\mathbf{a}$ contains a small number of distinct elements (i.e., $\omega \ll l$) this approach becomes significantly faster than the exhaustive search on all subsets. Indeed, first of all we clearly have $\bar{N}_B^{(j)} = 0$ when $j > \omega$; moreover, the number of configurations tested by using (37) is surely smaller than m^{j-1}. Then, for a specific value of j, the computation of $\bar{N}_B^{(j)}$ requires to test no more than $m^{j-1}\binom{\omega}{j}$ configurations. Thus, we can roughly upper bound the total number of configurations that are considered as

$$\sum_{j=1}^{\omega} m^{j-1} \binom{\omega}{j} \leq \sum_{j=1}^{\omega} m^{j-1} \left(\frac{\omega e}{j} \right)^j \leq \omega m^{\omega-1} e^{\omega}, \tag{38}$$

where e is the basis of the natural logarithmic. It can be verified that, when $m, w \ll l$, the above upper bound is significantly smaller than $\binom{l}{m}$.

APPENDIX B

In this appendix we consider the case of regular codes, for which the decoding threshold values can be assumed constant and equal to b, and we demonstrate that when v is odd and $b = \lceil \frac{v}{2} \rceil$, the bound (17) can be reformulated as in (21).

Let $\mathbf{H}$ be the parity-check matrix of a (v, w)-regular code with block length n and odd v. Let us denote as $\boldsymbol{\gamma}^{(i)}$ the i-th row of the adjacency matrix $\boldsymbol{\Gamma}$. Moreover, let $\mathbf{e} \in \mathcal{B}_t$, and $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$. We consider a single iteration of BF decoding applied to $\mathbf{s}$, with a unique decoding threshold $\left\lceil \frac{v}{2} \right\rceil$.

In order to determine a bound for P_f in these conditions, we can basically repeat the steps in the proof of Theorem 5. In this case, however, (18) can be specialized as follows

$$\begin{aligned}\left|\mathcal{E}_{i,t,\left\lceil \frac{v}{2} \right\rceil}\right| &= \left|\mathcal{E}^1_{i,t-1,v-\left\lceil \frac{v}{2} \right\rceil} \cup \mathcal{E}^0_{i,t,\left\lceil \frac{v}{2} \right\rceil - 1}\right| \\ &= \left|\mathcal{E}^1_{i,t-1,\frac{v-1}{2}} \cup \mathcal{E}^0_{i,t,\frac{v-1}{2}}\right| \\ &\leq \left|\mathcal{N}^{\tilde{\gamma}^{(i)}}_{t-1,\frac{v-1}{2}}\right| + \left|\mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,\frac{v-1}{2}}\right|, \end{aligned} \tag{39}$$

where we have exploited the fact that, since v is odd, we have $\left\lceil \frac{v}{2} \right\rceil = \frac{v+1}{2}$. Now, if we consider $\boldsymbol{\gamma}^{(i)}$ and a set $S \in \mathcal{N}^{\gamma^{(i)}}_{t,\frac{v-1}{2}}$, we have only two possibilities:

1) If $i \in S$, since $\gamma_{i,i} = 0$, we have $\sum_{j \in S \setminus i} \gamma_{i,j} > \frac{v-1}{2}$, from which $\{S \setminus i\} \in \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t-1,\frac{v-1}{2}}$.
2) If $i \notin S$, we have $\sum_{j \in S} \gamma_{i,j} > \frac{v-1}{2}$, from which $S \in \mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,\frac{v-1}{2}}$.

Then, we can state

$$\left|\mathcal{N}^{\tilde{\gamma}^{(i)}}_{t-1,\frac{v-1}{2}}\right| + \left|\mathcal{N}^{\tilde{\gamma}^{(i)}}_{t,\frac{v-1}{2}}\right| = \left|\mathcal{N}^{\gamma^{(i)}}_{t,\frac{v-1}{2}}\right|. \tag{40}$$

By replacing this equality in (17), the simpler (21) is eventually obtained.

APPENDIX C

In this appendix we give the parity-check matrices used in the Monte Carlo simulations. All the considered matrices are in form (1) and $\mathbf{H}_0$ and $\mathbf{H}_1$ are circulant matrices. The support of their first columns, which is $S(\mathbf{h}^{(0)}_0)$ and $S(\mathbf{h}^{(1)}_0)$, but is denoted for simplicity as S_0 and S_1, respectively, compactly describes the whole parity-check matrix.

The parity-check matrix of $\mathcal{C}_0$ is represented by

$$S_0 = \Big[\ 16 \quad 364 \quad 572 \quad 1166 \quad 1726 \quad 2231 \quad 2518 \quad 2555 \quad 2565 \quad 3334 \quad 3806 \quad 3818 \quad 4126$$
$$4590 \quad 4852 \quad 5425 \quad 5502 \quad 5536 \quad 5576 \quad 5880 \quad 7923 \quad 8296 \quad 8788 \quad 9035 \quad 9179\ \Big],$$

$$S_1 = \Big[\ 246 \quad 406 \quad 1732 \quad 1855 \quad 1871 \quad 2254 \quad 2297 \quad 2320 \quad 2474 \quad 3333 \quad 3513 \quad 4042$$
$$4511 \quad 5260 \quad 6037 \quad 6673 \quad 6716 \quad 7334 \quad 7766 \quad 7940 \quad 8036 \quad 8136 \quad 8802 \quad 8881 \quad 9384\ \Big].$$

The parity-check matrix of the code $\mathcal{C}_1$ is represented by

$$S_0 = \Big[\ 934 \quad 1750 \quad 3485 \quad 4040 \quad 4117 \quad 4639 \quad 4838 \quad 4879 \quad 5874 \quad 5886 \quad 6041 \quad 6874 \quad 7425\ \Big],$$

$$S_1 = \Big[\ 2043 \quad 2184 \quad 2619 \quad 2715 \quad 3190 \quad 3359 \quad 4163 \quad 4327 \quad 4705 \quad 5188 \quad 5335 \quad 7629 \quad 7879 \ \Big].$$

The parity-check matrix of $\mathcal{C}_2$ is represented by

$$S_0 = \Big[\ 168 \quad 229 \quad 309 \quad 405 \quad 464 \quad 507 \quad 668 \quad 888 \quad 893 \quad 908 \quad 984 \quad 1015$$
$$1143 \quad 1178 \quad 1299 \quad 1311 \quad 1368 \quad 1380 \quad 1433 \quad 1478 \quad 1675 \quad 1728 \quad 1800$$
$$1936 \quad 2069 \quad 2084 \quad 2215 \quad 2530 \quad 2632 \quad 2842 \quad 3090 \quad 3103 \quad 3282 \quad 3332$$
$$3532 \quad 3595 \quad 3657 \quad 3882 \quad 3919 \quad 3929 \quad 4077 \quad 4138 \quad 4160 \quad 4654 \quad 4698 \ \Big],$$

$$S_1 = \Big[\ 263 \quad 271 \quad 277 \quad 369 \quad 381 \quad 641 \quad 689 \quad 754 \quad 792 \quad 935 \quad 1153 \quad 1415$$
$$1551 \quad 1727 \quad 1732 \quad 1743 \quad 1988 \quad 2065 \quad 2099 \quad 2102 \quad 2139 \quad 2159 \quad 2205$$
$$2249 \quad 2443 \quad 2566 \quad 2586 \quad 2737 \quad 2932 \quad 3041 \quad 3140 \quad 3337 \quad 3504 \quad 3613$$
$$3632 \quad 3946 \quad 3953 \quad 4047 \quad 4097 \quad 4218 \quad 4233 \quad 4315 \quad 4329 \quad 4486 \quad 4506 \ \Big].$$

The parity-check matrix of $\mathcal{C}_3$ is represented by

$$S_0 = \Big[\ 864 \quad 3551 \quad 4164 \quad 5538 \quad 8013 \quad 8487 \quad 8846 \quad 8986 \quad 10925 \ \Big],$$
$$S_1 = \Big[\ 2256 \quad 6346 \quad 6495 \quad 6959 \quad 7551 \quad 8409 \quad 8725 \quad 10317 \quad 11554 \ \Big].$$

The parity-check matrix of $\mathcal{C}_4$ is represented by

$$S_0 = \Big[\ 1106 \quad 1985 \quad 2497 \quad 3036 \quad 3394 \quad 5118 \quad 5136 \quad 5276$$
$$6506 \quad 6523 \quad 7450 \quad 8338 \quad 8472 \quad 9662 \quad 11434 \ \Big],$$

$$S_1 = \Big[\ 471 \quad 974 \quad 1775 \quad 5048 \quad 5595 \quad 5617 \quad 6805 \quad 8861$$
$$8894 \quad 9009 \quad 9158 \quad 9416 \quad 11071 \quad 11379 \quad 11404 \ \Big].$$

The parity-check matrix of $\mathcal{C}_5$ is represented by

$$S_0 = \Big[\ 242 \quad 432 \quad 447 \quad 784 \quad 1040 \quad 1669 \quad 1786 \quad 2430 \quad 2496 \quad 2643 \quad 2682 \quad 3161 \quad 3173$$
$$3952 \quad 4461 \quad 5319 \quad 5336 \quad 5369 \quad 5423 \quad 5678 \quad 5768 \quad 5891 \quad 6906 \quad 6943 \quad 7207 \quad 7535$$
$$7740 \quad 7743 \quad 8435 \quad 8496 \quad 8608 \quad 8765 \quad 8824 \quad 9251 \quad 9463 \quad 9635 \quad 9637 \quad 9659 \quad 9685$$
$$9969 \quad 9971 \quad 10052 \quad 10284 \quad 10397 \quad 10525 \quad 10821 \quad 11367 \ \Big],$$

$$S_1 = \Big[\ 144\ \ 284\ \ 722\ \ 724\ \ 821\ \ 1403\ \ 1465\ \ 1546\ \ 2028\ \ 2277\ \ 2569\ \ 2916\ \ 3108$$
$$3286\ \ 3400\ \ 3460\ \ 3759\ \ 3844\ \ 3983\ \ 4252\ \ 4600\ \ 4631\ \ 5289\ \ 5323\ \ 5587\ \ 6004$$
$$6403\ \ 7380\ \ 7427\ \ 7826\ \ 7899\ \ 7998\ \ 8106\ \ 8960\ \ 9004\ \ 9196\ \ 9348\ \ 9508\ \ 9803$$
$$10058\ \ 10497\ \ 10671\ \ 10751\ \ 10865\ \ 11092\ \ 11362\ \ 11394\ \Big].$$

The parity-check matrix of $\mathcal{C}_6$ is represented by

$$S_0 = \Big[\ 516\ \ 739\ \ 988\ \ 1332\ \ 1408\ \ 1503\ \ 1668\ \ 1671\ \ 1743\ \ 1983\ \ 2042\ \ 2110\ \ 2466$$
$$2583\ \ 2661\ \ 2808\ \ 2863\ \ 2918\ \ 2976\ \ 3388\ \ 3551\ \ 3828\ \ 4337\ \ 4533\ \ 4741\ \Big],$$

$$S_1 = \Big[\ 132\ \ 448\ \ 502\ \ 769\ \ 868\ \ 1063\ \ 1436\ \ 1457\ \ 1511\ \ 1676\ \ 2023\ \ 2422\ \ 2469$$
$$2613\ \ 2620\ \ 3197\ \ 3499\ \ 3754\ \ 4020\ \ 4054\ \ 4211\ \ 4286\ \ 4528\ \ 4599\ \ 4930\ \Big].$$

The parity-check matrix of $\mathcal{C}_7$ is represented by

$$S_0 = \Big[\ 709\ \ 792\ \ 854\ \ 907\ \ 1548\ \ 1608\ \ 2062\ \ 2152\ \ 2158\ \ 2359\ \ 2625\ \ 2981\ \ 3372$$
$$3572\ \ 3664\ \ 3716\ \ 3726\ \ 4283\ \ 5311\ \ 5551\ \ 6014\ \ 6432\ \ 6569\ \ 6595\ \ 6636\ \Big],$$

$$S_1 = \Big[\ 824\ \ 934\ \ 1220\ \ 1570\ \ 2129\ \ 2244\ \ 2526\ \ 2629\ \ 3533\ \ 3557\ \ 3708\ \ 3833\ \ 3862$$
$$4147\ \ 4252\ \ 4556\ \ 4636\ \ 4662\ \ 5254\ \ 5286\ \ 5375\ \ 5691\ \ 5738\ \ 6347\ \ 6785\ \Big].$$

The parity-check matrix of $\mathcal{C}_8$ is represented by

$$S_0 = \Big[\ 1383\ \ 1783\ \ 1940\ \ 2117\ \ 2834\ \ 3216\ \ 3347\ \ 4168\ \ 4267\ \ 6118\ \ 7683\ \ 8431\ \ 9114$$
$$9191\ \ 9562\ \ 10170\ \ 10515\ \ 10874\ \ 11604\ \ 12110\ \ 13137\ \ 13202\ \ 13508\ \ 14658\ \ 14687\ \Big],$$

$$S_1 = \Big[\ 189\ \ 272\ \ 753\ \ 938\ \ 1372\ \ 1940\ \ 1984\ \ 2524\ \ 3072\ \ 4414\ \ 4637\ \ 4807\ \ 4971$$
$$6029\ \ 6360\ \ 6931\ \ 6970\ \ 7653\ \ 8817\ \ 9193\ \ 11761\ \ 11981\ \ 12242\ \ 12549\ \ 13846\ \Big].$$

REFERENCES

[1] P. Santini, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Hard-decision iterative decoding of LDPC codes with bounded error rate," in *Proc. IEEE International Conference on Communications (ICC 2019)*, Shanghai, China, May 2019.

[2] T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson, "A reaction attack on the QC-LDPC McEliece cryptosystem," in *Post-Quantum Cryptography, PQCrypto 2017*, ser. Lecture Notes in Computer Science, T. Lange and T. Takagi, Eds. Springer International Publishing, 2017, vol. 10346, pp. 51–68.

[3] T. Paiva and R. Terada, "Improving the efficiency of a reaction attack on the QC-MDPC McEliece," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E101.A, pp. 1676–1686, Oct 2018.

[4] E. Eaton, M. Lequesne, A. Parent, and N. Sendrier, "QC-MDPC: A timing attack and a CCA2 KEM," in *Post-Quantum Cryptography, PQCrypto 2018*, ser. Lecture Notes in Computer Science, T. Lange and R. Steinwandt, Eds., vol. 10786. Springer, Cham, 2018, pp. 47–76.

[5] P. Santini, M. Battaglioni, F. Chiaraluce, and M. Baldi, "Analysis of reaction and timing attacks against cryptosystems based on sparse parity-check codes," in *Code-Based Cryptography Workshop (CBC 2019)*, ser. Lecture Notes in Computer Science, M. Baldi, E. Persichetti, and P. Santini, Eds. Springer, Cham, 2019, vol. 11666, pp. 115–136.

[6] V. V. Zyablov and M. S. Pinsker, "Estimation of the error-correction complexity for Gallager low-density codes," *Problems of Information Transmission*, vol. 11, pp. 23–26, 1975.

[7] D. Burshtein, "On the error correction of regular LDPC codes using the flipping algorithm," *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 517–530, Feb. 2008.

[8] S. K. Chilappagari, D. V. Nguyen, B. Vasic, and M. W. Marcellin, "On the guaranteed error correction capability of LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT 2008)*, Toronto, Canada, Jul. 2008, pp. 434–438.

[9] S. K. Chilappagari, B. Vasic, and M. W. Marcellin, "Guaranteed error correction capability of codes on graphs," in *Proc. 2009 Information Theory and Applications Workshop*, San Diego, CA, Feb. 2009, pp. 50–55.

[10] W.-Y. Chen and C.-C. Lu, "On error correction capability of bit-flipping algorithm for LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT 2011)*, Saint-Petersburg, Russia, Jul. 2011, pp. 1283–1286.

[11] N. Alon, "Spectral techniques in graph algorithms," in *LATIN'98: Theoretical Informatics. LATIN 1998*, ser. Lecture Notes in Computer Science, C. L. Lucchesi and A. V. Moura, Eds. Springer, Berlin, Heidelberg, 1998, vol. 1380, pp. 206–215.

[12] J.-P. Tillich, "The decoding failure probability of MDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT 2018)*, Vail, CO, Jun. 2018, pp. 941–945.

[13] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[14] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[15] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "LEDAkem: A post-quantum key encapsulation mechanism based on QC-LDPC codes," in *Post-Quantum Cryptography, PQCrypto 2018*, ser. Lecture Notes in Computer Science, T. Lange and R. Steinwandt, Eds. Springer, Cham, 2018, vol. 10786, pp. 3–24.

[16] R. Misoczki, J. P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," in *Proc. IEEE International Symposium on Information Theory (ISIT 2013)*, Istanbul, Turkey, Jul. 2013, pp. 2069–2073.

[17] National Institute of Standards and Technology. (2016, Dec.) Post-quantum crypto project. [Online]. Available: http://csrc.nist.gov/groups/ST/post-quantum-crypto/

[18] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "LEDAkem and LEDApkc website," https://www.ledacrypt.org/.

[19] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Güneysu, A. C. Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor, "BIKE website," https://bikesuite.org/.

[20] H. Xiao and A. H. Banihashemi, "Hard-decision performance of LDPC codes on binary symmetric channels with small crossover probabilities," in *Proc. 23rd Biennial Symp. Communications*, Kingston, Ontario, Canada, May 2006, pp. 1–4.

[21] N. Miladinovic and M. P. C. Fossorier, "Improved bit-flipping decoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1594–1606, Apr. 2005.

[22] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error floors of LDPC codes on the binary symmetric channel," in *Proc. IEEE International Conference on Communications (ICC 2006)*, vol. 3, Istanbul, Turkey, Jun. 2006, pp. 1089–1094.

[23] H. Xiao and A. H. Banihashemi, "Estimation of bit and frame error rates of finite-length low-density parity-check codes on binary symmetric channels," *IEEE Transactions on Communications*, vol. 55, no. 12, pp. 2234–2239, Dec. 2007.

[24] ——, "Error rate estimation of low-density parity-check codes on binary symmetric channels using cycle enumeration," *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1550–1555, Jun. 2009.

[25] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[26] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[27] M. Baldi, "Quasi-cyclic low-density parity-check codes," in *QC-LDPC Code-Based Cryptography*. Springer International Publishing, 2014.

[28] M. Baldi, M. Bianchi, and F. Chiaraluce, "Security and complexity of the McEliece cryptosystem based on QC-LDPC codes," *IET Information Security*, vol. 7, no. 3, pp. 212–220, Sep. 2012.

[29] P. Santini, M. Baldi, G. Cancellieri, and F. Chiaraluce, "Hindering reaction attacks by using monomial codes in the McEliece cryptosystem," in *Proc. IEEE International Symposium on Information Theory (ISIT 2018)*, Vail, CO, Jun. 2018, pp. 951–955.

[30] Q. Guo, T. Johansson, and P. Stankovski, "A key recovery attack on MDPC with CCA security using decoding errors," in *Advances in Cryptology - ASIACRYPT 2016. ASIACRYPT 2016*, ser. Lecture Notes in Computer Science, J. Cheon and T. Takagi, Eds. Springer, Berlin, Heidelberg, 2016, vol. 10031, pp. 789–815.

[31] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in *Theory of Cryptography, TCC 2017*, ser. Lecture Notes in Computer Science, Y. Kalai and L. Reyzin, Eds. Springer, Cham, 2017, vol. 10677, pp. 341–371.

[32] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography. PQCrypto 2008*, ser. Lecture Notes in Computer Science, J. Buchmann and J. Ding, Eds. Springer, Berlin, Heidelberg, 2008, vol. 5299, pp. 31–46.