Distance Properties of Punctured Simplex Codes and Design of High-Rate PRC-LDPC Codes for Complexity-Constrained Applications

Massimo Battaglioni, *Member, IEEE*, Marco Baldi, *Senior Member, IEEE*, Franco Chiaraluce, *Senior Member, IEEE*, and Giovanni Cancellieri

***Abstract*—Cyclic simplex codes are usually deemed as impractical because of their extremely low rate and large codeword length for practical values of the code dimension. To address these limitations, researchers have resorted to punctured simplex codes, focusing on primitive polynomials and employing statistical analyses to investigate their properties. This paper delves deeper into the properties of punctured binary simplex codes, also focusing on the recently introduced family of Primitive Rate-Compatible Low-Density Parity-Check (PRC-LDPC) codes. We study the average behavior of punctured simplex codes in terms of minimum distance properties. Furthermore, our results highlight the potential of high-rate PRC-LDPC codes to reach or even surpass the performance of state-of-the-art code families. We show how to design good codes by applying puncturing and shortening operations to cyclic simplex codes, also considering complexity-constrained scenarios.**

***Index Terms*—LDPC codes, minimum distance, pseudo-noise sequences, simplex codes.**

I. Introduction

SIMPLEX codes [2], also known as Hadamard codes, are dual codes of Hamming codes. For any given code dimension k, a binary simplex code contains $2^k - 1$ non-zero codewords, each having Hamming weight 2^{k-1}. These codewords are cyclic shifts of one another, making simplex codes cyclic. Notably, these codewords can be generated using a single Linear-Feedback Shift Register (LFSR) with k memory elements, where each codeword results from a different initial seed for the LFSR [2]. The analysis of cyclic simplex codes is thus closely related to that of m-sequences, or maximum-length linear recursive sequences, which are a subset of Pseudo-Noise (PN) sequences. However, cyclic simplex codes on their own are often deemed impractical due to their codeword length of 2^k-1 for a given code dimension k, resulting in a code rate of $\frac{k}{2^k-1}$, which becomes too low for practical values of k. Some notable exceptions are given by coded distributed systems, where simplex codes with small codeword lengths are employed [3], and by turbo-Hadamard [4] and low-density parity-check (LDPC)-Hadamard [5] coding systems, where simplex codes are used in conjunction with state-of-the art codes. On the other hand, PN sequences remain widely used. For example, they are extensively employed in multiple access communications, as well as in ranging and synchronization applications [6], [7], [8], [9]. The primary reason for this apparent contradiction is that practical error-correcting codes usually require values of k that range from several tens to tens of thousands, whereas PN sequences typically involve values of k in the range of tens. In this paper, we exploit the link between m-sequences and simplex codes, since it allows us to exploit many results related to these sequences to analyze simplex codes, which we treat with a polynomial approach.

Techniques such as puncturing and shortening, which enable the design of new linear codes starting from "classical" ones, are crucial to design codes with practical parameters derived from cyclic simplex codes. However, this flexibility severely complicates the analysis of the resulting codes. In this paper we contribute to such a line of research as follows:

- we analyze punctured binary simplex codes, mainly for relatively low code rates, having codewords that coincide with subsequences of m-sequences;
- we extend the design of a special class of practical binary punctured simplex codes, named Primitive Rate-Compatible Low-Density Parity-Check (PRC-LDPC) codes, mainly for moderate to high code rates.

Regarding the second point, the values of code rate are very important since, for moderate to high code rates, enhancing the minimum distance properties of a code can significantly improve its error rate performance. Consequently, the combined use of puncturing and shortening techniques, on which our code design is based, becomes highly impactful.

Received 19 October 2024; revised 15 October 2025; accepted 25 November 2025. Date of publication 28 November 2025; date of current version 23 December 2025. The work of Massimo Battaglioni, Marco Baldi, and Franco Chiaraluce was supported in part by European Union—Next Generation EU under Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP J33C22002880001, Partnership on "Telecommunications of the Future" under Grant PE00000001—Program "RESTART." An earlier version of this paper was presented in part at the IEEE International Symposium on Information Theory 2024 [DOI: 10.1109/ISIT57864.2024.10619325]. *(Corresponding author: Massimo Battaglioni.)*

Massimo Battaglioni and Marco Baldi are with the Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, 60131 Ancona, Italy (e-mail: m.battaglioni@univpm.it).

Franco Chiaraluce is with the Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, 60131 Ancona, Italy, and also with the Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 43124 Parma, Italy.

Giovanni Cancellieri, retired, was with the Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, 60131 Ancona, Italy. He resides in 60015 Falconara, Italy

Communicated by V. Skachek, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2025.3638947.

Digital Object Identifier 10.1109/TIT.2025.3638947

A. Related Works

Rate-Compatible (RC) codes allow adjusting the code rate to adapt to channel conditions without changing the code. Among them, Rate-Compatible Low-Density Parity-Check (RC-LDPC) codes [10] stand out as a promising option.

The emergence of RC-LDPC codes addresses the limitation on the code rate by offering a framework that reconciles the excellent error-correction properties of LDPC codes with the need for rate flexibility [11], [12], [13]. PRC-LDPC codes obtained by puncturing simplex codes were introduced in [14], [15]. Each family of such codes is uniquely represented by a primitive polynomial.

The design of PRC-LDPC codes starts from simplex codes, thus from the choice of a proper primitive parity-check polynomial $h(x)$. In order to obtain an LDPC code, $h(x)$ must be relatively sparse, meaning that it has many more null coefficients than non-null coefficients. Additionally, as shown in [16], the support vector of its coefficients vector must be a Golomb ruler to avoid the presence of short cycles in the associated Tanner graph [17]. The main advantages of these rate-adaptive codes over comparable solutions stand in their low encoding/decoding complexity, in the extremely low representation cost and in the existence of convenient techniques to compute their minimum distance, or even their weight spectrum, which is useful to characterize the code error correction performance.

Good puncturing patterns for simplex codes can be found in [18]. The rateless performance of punctured simplex codes is studied in [19]. Punctured simplex codes for error detection are investigated in [20]. In all these papers, punctured simplex codes are not seen as LDPC codes, and they do not allow efficient decoding algorithms for large values of k and n. LDPC codes based on punctured simplex codes are instead studied in [1], [15], and [16]. The effect of random puncturing patterns on minimum distance properties and iterative decoding thresholds of LDPC code ensembles (not obtained from simplex codes) is studied in [21].

To some extent, this work is also related to the literature on the properties of subsequences of m-sequences. In [22], the authors study m-sequences through the moments of their weight distributions. A comprehensive treatment of linear recurring sequences can be found in [23, Chapter 8]. More recently, a detailed characterization of the weight distribution of subsequences of m-sequences has been developed in [16], [19], and [24]. Furthermore, [25, Section II] provides valuable insight into the distance properties of linear phase detection schemes, including their asymptotic behavior, where m-sequences are also employed.[1] Our analysis draws inspiration from several of the key results presented in these works. In particular, we build upon insights on the weight distribution to carry out an asymptotic analysis of simplex codes subjected to a specific puncturing operation. It is important to emphasize that our goal is not to improve or extend the existing results on m-sequences, but rather to leverage them in support of a distinct coding-theoretic analysis (and construction).

[1]Strictly speaking, phase detection sequences must include the all-zero subsequence of length m, which is not present in an m-sequence. Nevertheless, aside from this detail, the two constructions are essentially equivalent.

B. Our Contribution

This paper aims to investigate novel properties of punctured binary simplex codes, focusing on their ability to achieve unique asymptotic performance as k increases (a similar result was proven for shortened and punctured polar codes in [26]). Specifically, we demonstrate that punctured simplex codes exhibit asymptotic performance that is equal to or potentially better than that of simplex codes with the same codeword length. Establishing a relationship between these asymptotic performances permits us to use simplex codes as a benchmark for evaluating the performance of punctured simplex codes. However, it is important to note that simplex codes are restricted to codeword lengths of 2^k-1, leading to increasingly larger gaps between usable values of n as the code dimension grows, whereas punctured simplex codes can fill this gap, since they are inherently characterized by a fine rate adaptability. The analysis we present regarding this issue is completely novel with respect to [1].

Additionally, our study enables practical applications of these codes beyond extremely low code rates or specific codeword lengths, which is typically the case for their parent cyclic simplex codes. The importance of the structure of the code parity-check polynomial $h(x)$ in the design of codes of this kind is emphasized. In fact, as an additional contribution, we provide insights into the design of the special family of punctured simplex codes named PRC-LDPC codes, for scenarios where constrained hardware and software resources require low complexity. In other words, we propose a method to adapt PRC-LDPC codes optimized in an unconstrained setting, to a scenario where the decoding complexity is upper bounded. To reach this goal, we combine puncturing and shortening operations. This idea was already proposed in [1]. However, following some insights presented in [27], we here propose a new, and more efficient, shortening procedure with respect to [1]. Through numerical assessments, we evaluate the error rate performance of high-rate codes in complexity-constrained settings, demonstrating that PRC-LDPC codes achieve good performance under belief propagation decoding while maintaining low decoding complexity, also compared to state-of-the-art solutions, such as LDPC codes employed in the 5G standard [28].

C. Paper Outline

The paper is organized as follows. In Section II, we introduce the mathematical notation and recall some preliminary concepts on cyclic simplex codes, punctured simplex codes and PRC-LDPC codes. In Section III, we analyze the minimum distance of families of punctured simplex codes. In Section IV, we compare some asymptotic and finite-length properties of cyclic simplex codes and punctured simplex codes with the same codeword length; in the same section, we provide some bounds on the error rate performance of

these codes. In Section V, we provide some more numerical examples, introducing the general principles we adopt for the code design of high-rate PRC-LDPC codes. In Section VI, we deal with the practical design of PRC-LDPC codes in complexity-constrained scenarios. Section VII shows the error rate performance of some PRC-LDPC codes under iterative decoding, assessed through Monte Carlo analysis of simulated transmissions. Finally, Section VIII concludes the paper.

II. Notation and Background

In this section we introduce the notation we use throughout the paper and we provide some background notions.

A. Notation

Given two integers a and b, we denote by $[a, b]$ the set of integers $\{y : a \le y \le b\}$. We use bold upper case letters (resp., lower case) to denote matrices (resp., vectors). For a vector $\mathbf{a}$ of length n, the i-th entry is denoted as a_i. The support vector Supp($\mathbf{a}$) of a vector $\mathbf{a}$ of length n is defined as the vector containing the positions of the non-zero entries of $\mathbf{a}$. For an $m \times n$ matrix $\mathbf{A}$, the entry at position (i, j) is denoted as $a_{i,j}$, the i-th row as $\mathbf{a}_{i,:}$ and the j-th column as $\mathbf{a}_{:,j}$. Given an $m \times n$ matrix $\mathbf{A}$, and a set $K \subset [0, n-1]$, $\mathbf{A}_K$ is the submatrix of $\mathbf{A}$ formed by the columns of $\mathbf{A}$ with indices in K. The binary entropy function is defined as $H_2(x) = x \log_2\left(\frac{1}{x}\right) + (1-x)\log_2\left(\frac{1}{1-x}\right)$.

The Hamming distance (simply called distance in the rest of the paper) between two vectors $\mathbf{u}$ and $\mathbf{v}$ is the number of positions in which the corresponding entries differ and is denoted as $D(\mathbf{u}, \mathbf{v})$. The Hamming weight (simply called weight in the rest of the paper) of a vector $\mathbf{v}$ is the number of non-zero symbols it contains and is denoted as wt($\mathbf{v}$), or w_v for brevity. Similarly, the weight of a polynomial is the number of its non-zero coefficients. To every polynomial $h(x) = h_0 + h_1 x + \cdots + h_k x^k$, we associate a coefficients vector $\mathbf{h} = (h_0, \ldots, h_k)$. The reciprocal of a polynomial $h(x)$ of degree k is denoted as $h^*(x) = x^k h(x^{-1})$. We denote the finite field with order q as $\mathbb{F}_q$. A polynomial of degree k in $\mathbb{F}_2[x]$ is said to be primitive if it is the minimal polynomial of a primitive element of $\mathbb{F}_{2^k}$. Straightforwardly, primitive polynomials in $\mathbb{F}_2[x]$ have odd weight.

We say that a sequence is circular if its first and last entries are considered consecutive, and use the symbol $\hat{}$ over the sequence letter to distinguish these sequences from conventional ones. For example, $\mathbf{p}$ denotes a non-circular sequence, whereas $\hat{\mathbf{p}}$ denotes a circular one. An m-sequence (maximal length sequence) is a binary sequence of period $2^m - 1$ that can be generated by an m-stage LFSR with a primitive characteristic polynomial. We define the sliding-window sequence extraction function $W_{i,n}(\cdot)$ that takes as input a circular sequence, say $\hat{\mathbf{p}}$, and returns $W_{i,n}(\hat{\mathbf{p}}) = [\hat{p}_i, \ldots, \hat{p}_{i+n-1}]$. The probability of an event E is denoted as $\Pr\{E\}$.

B. Punctured Simplex Codes and PRC-LDPC Codes

For some finite field $\mathbb{F}_q$, a linear block code C is defined as a $k-$dimensional subspace of $\mathbb{F}_q^n$, where $k < n$, which can be described as the kernel of a full-rank matrix $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, having size $r \times n$, where $r = n - k$. Such a matrix is known as the code parity-check matrix. A vector $\mathbf{c} \in \mathbb{F}_q^n$ belongs to the code C, and is called a codeword, if and only if $\mathbf{c}\mathbf{H}^\top = \mathbf{0}$, where $^\top$ denotes transposition. Hence, $C = \{\mathbf{c} \in \mathbb{F}_q^n : \mathbf{c}\mathbf{H}^\top = \mathbf{0}\}$. The code rate, denoted as R, is defined as $R = \frac{k}{n}$. In the rest of the paper, we will consider only binary codes, thus assuming $q = 2$. The number of codewords of weight w is denoted as $A(w)$. The minimum distance of a linear block code, $d_{\min}$, is the smallest positive value of w such that $A(w) > 0$. Analogously, we define the maximum distance of the code, denoted as $d_{\max}$, as the largest positive value of w such that $A(w) > 0$. We will often describe a code by the triplet $(n, k, d_{\min})$, and omit the third parameter when unnecessary. For the sake of clarity, when referring to parent cyclic codes, we denote the codeword length by N. A quantity that can be useful to consider as a code performance measure is its asymptotic coding gain over uncoded binary phase shift keying (BPSK) modulation, expressed as $G_\infty = 10\log_{10}(R \cdot d_{\min})$ for soft-decision maximum-likelihood decoding [29, Chapter 1].

An (N, k) linear block code of length N is cyclic if any cyclic shift of each one of its codewords results in another codeword. It is convenient to express the codewords of cyclic codes as polynomials (named code polynomials), each having degree $N-1$ or less. There exists one and only one code polynomial of degree $N-k$, named $g(x)$, which is called the generator polynomial of the code (since each code polynomial can be expressed as a multiple of $g(x)$). The generator polynomial of a binary (N, k) cyclic code divides x^N+1, i.e., $x^N+1 = g(x)f(x)$, where $f(x)$ is a polynomial of degree k. The reciprocal of $f(x)$, denoted as $h(x)$, is the parity-check polynomial of the code (and it is also a factor of $x^N + 1$) [29].

A binary cyclic simplex code with dimension k has codeword length $N = 2^k - 1$, and thus rate $R = \frac{k}{2^k-1}$, $d_{\min} = d_{\max} = 2^{k-1}$ and $A(2^{k-1}) = 2^k - 1$. Given a primitive parity-check polynomial $h(x)$ with coefficients vector $\mathbf{h}$, the parity-check matrix of the corresponding simplex code is

$$\mathbf{H} = \begin{pmatrix} h_0 & h_1 & \cdots & h_k & & \\ & h_0 & h_1 & \cdots & h_k & \\ & & \ddots & \ddots & \vdots & \\ & & & h_0 & h_1 & \cdots & h_k \\ & & & & h_0 & h_1 & \cdots & h_k \end{pmatrix} \Bigg\updownarrow r = N - k \qquad (1)$$

Every non-zero codeword of a cyclic simplex code is an m-sequence whose length equals its period, namely $N = 2^k - 1$, since $h(x)$ is primitive [30]. Therefore, since the code is cyclic, one of its codewords can be obtained by generating an m-sequence of length N, denoted as $\hat{\mathbf{p}}$ in the following, with an LFSR whose connections are determined by $h(x)$. All the other non-zero codewords can then be obtained by cyclic shifts of $\hat{\mathbf{p}}$.

By definition, the parity-check matrices of LDPC codes are sparse, meaning that the number of their non-zero entries is much smaller than that of their zero entries. The girth of a code is the length of the shortest cycle(s) in its Tanner graph. Soft-decision decoding algorithms, such as the widely-used sum-product algorithm [31], face convergence issues when applied to LDPC codes with parity-check matrices with short girth.

The Row-Column Constraint (RCC) in the parity-check matrix of an LDPC code defines the absence of four non-zero entries forming the vertices of a rectangle, preventing the existence of length-4 cycles in the associated Tanner graph [17].

In the process of puncturing an (N, k) code, the number k of information bits associated to each codeword remains unchanged while some entries, say ρ, of the codeword are deleted (punctured) and therefore not transmitted. The result is an $(N-\rho, k)$ linear block code with a higher rate and usually a lower minimum distance than the original code. It is, in fact, evident that the minimum distance of a punctured code cannot exceed that of the original code.

So, the parent cyclic simplex code is characterized by a parity-check matrix of the form given in (1), which is non-systematic, with codeword length $N = 2^k - 1$. As discussed in [29, Section 5.2], the corresponding generator matrix $\mathbf{G}$ also exhibits a doubly-triangular structure, with an upper-triangular left portion and a lower-triangular right portion. Thanks to this structure, it is straightforward to transform $\mathbf{G}$ into systematic form via Gauss-Jordan elimination over $\mathbb{F}_2$, yielding a matrix of the form $\mathbf{G}_{\text{sys}} = [\mathbf{I}_k \mid \mathbf{P}]$. In this representation, the first k positions of each codeword correspond to information symbols, while the remaining $N-k$ symbols correspond to parity. This structural property is the key to our puncturing strategy. We specifically remove the rightmost ρ symbols of each codeword, which, under the systematic form, correspond to redundancy. Since this operation corresponds to the removal of the last ρ columns from the generator matrix $\mathbf{G}$, the resulting matrix retains its upper-triangular structure. In parallel, puncturing corresponds to removing the last ρ rows and ρ columns from $\mathbf{H}$, and thus the parity-check matrix $\mathbf{H}$ also preserves its doubly-triangular form. Therefore, the punctured code inherits the desirable structural properties of the original code: $\mathbf{H}$ remains doubly-triangular, and $\mathbf{G}$ remains upper-triangular and can still be easily brought into systematic form.

Given the sequence $\hat{\mathbf{p}}$, which unambiguously defines the parent cyclic simplex code, we denote as $d^{\hat{\mathbf{p}}}_{\min}(n)$, $d^{\hat{\mathbf{p}}}_{\max}(n)$ and $A^{\hat{\mathbf{p}}}(w, n)$, $\forall w : d^{\hat{\mathbf{p}}}_{\min}(n) \le w \le d^{\hat{\mathbf{p}}}_{\max}(n)$, the minimum distance, the maximum distance and the weight distribution, respectively, of any punctured code of length $k+1 \le n \le N$, where $n = N-\rho$, obtained from a cyclic simplex code. The ensemble of codes formed by a parent cyclic simplex code and all its possible punctured versions (using the contiguous puncturing patterns described above) is called *code family*. Any code in such a family can be represented without ambiguity as $\mathcal{C}^{\hat{\mathbf{p}}}(n)$. For the sake of readability, when clear from the context, the dependence on $\hat{\mathbf{p}}$ will be omitted.

The $2^k - 1$ non-zero codewords of $\mathcal{C}^{\hat{\mathbf{p}}}(n)$, as for the parent simplex code, can be obtained as all the vectors selected by a sliding window of length $n = N-\rho < 2^k - 1$ that circularly spans over $\hat{\mathbf{p}}$.

We say that a code family of punctured simplex codes defines a family of PRC-LDPC codes if the parent cyclic simplex code is designed in such a way that all the punctured codes have a sparse parity-check matrix, fulfilling the RCC. It is proven in [14, Theorem 1] that any PRC-LDPC code satisfies the RCC if and only if the support of the coefficients vector of the corresponding primitive polynomial is a Golomb ruler. It is proven in [16, Theorem 2] that the girth of any non-trivial PRC-LDPC code cannot exceed 6.

Given a PRC-LDPC code of length n, we also define families of codewords: two (or more) codewords $\mathbf{t}$ and $\mathbf{c}$ belong to the same family if and only if $\text{Supp}(\mathbf{c}) = \text{Supp}(\mathbf{t}) + \gamma$, for some $\gamma \in \mathbb{Z}_{\neq 0}$, with

$$\gamma \in \left[-\min\left\{\text{Supp}(\mathbf{t})\right\} ; n-1-\max\left\{\text{Supp}(\mathbf{t})\right\}\right] \setminus \{0\}.$$

The shortening operation for PRC-LDPC codes follows the same structural principles that govern the design of the puncturing strategy. The generator matrix $\mathbf{G}$ can be brought into systematic form through simple row operations applied to its upper-triangular form, so that the first k positions of each codeword correspond to information symbols. Shortening by s symbols therefore involves removing s information symbols from these positions. This is accomplished by deleting s columns from the first k columns of the parity-check matrix $\mathbf{H}$. As a result, the codeword length is reduced from N to $N-s$, and the code dimension from k to $k-s$. This property holds true for both the original (N, k) cyclic simplex code and its punctured versions, provided that puncturing is performed on the rightmost code symbols.

III. Minimum Distance of Punctured Simplex Codes

In this section we wish to analyze the minimum and maximum distance properties of punctured simplex codes. We thus study the functions $d_{\min}(n)$ and $d_{\max}(n)$.

Let us define

$$\Delta(n) = \frac{d_{\max}(n) - d_{\min}(n)}{2},$$

and

$$\bar{d}(n) = \frac{d_{\max}(n) + d_{\min}(n)}{2},$$

such that

$$d_{\min}(n) = \bar{d}(n) - \Delta(n), \quad d_{\max}(n) = \bar{d}(n) + \Delta(n). \tag{2}$$

When applying a single puncturing operation to a punctured simplex code $\mathcal{C}(n+1)$, turning it into $\mathcal{C}(n)$, $d_{\min}$ and $d_{\max}$ can either decrease by one unit, or remain the same [16, Lemma 7]. Thus,

- $\Delta(n) = \Delta(n+1)$ when either $d_{\min}$ and $d_{\max}$ do not change, or they both decrease by one unit, after the puncturing operation;
- $\Delta(n) = \Delta(n+1) - \frac{1}{2}$, when $d_{\min}(n) = d_{\min}(n+1)$ and $d_{\max}(n) = d_{\max}(n+1) - 1$;
- $\Delta(n) = \Delta(n+1) + \frac{1}{2}$, when $d_{\min}(n) = d_{\min}(n+1) - 1$ and $d_{\max}(n) = d_{\max}(n+1)$.

Let us also define

$$\omega(n) = \begin{cases} \frac{n}{2} & \text{if } n \le \frac{N-1}{2}, \\ \frac{n}{2} + \frac{1}{2} & \text{if } \frac{N-1}{2} + 1 \le n \le N \end{cases} \tag{3}$$

and

$$\delta(n) = \bar{d}(n) - \omega(n). \tag{4}$$

Proposition 1: After a single puncturing operation on $\mathcal{C}(n+1)$, the following may happen, when $n \neq \frac{N-1}{2}$,

1) $\delta(n) = \delta(n+1) + 1/2$ if $d_{\max}(n) = d_{\max}(n+1)$ and $d_{\min}(n) = d_{\min}(n+1)$;
2) $\delta(n) = \delta(n+1)$ if $d_{\max}(n) = d_{\max}(n+1)-1$ and $d_{\min}(n) = d_{\min}(n+1)$, or if $d_{\max}(n) = d_{\max}(n+1)$ and $d_{\min}(n) = d_{\min}(n+1)-1$;
3) $\delta(n) = \delta(n+1) - 1/2$ if $d_{\max}(n) = d_{\max}(n+1) - 1$ and $d_{\min}(n) = d_{\min}(n+1) - 1$.

Proof: We analyze $\delta(n)$ in terms of $\delta(n+1)$ in each case, assuming $n \neq \frac{N-1}{2}$.

Case 1: $d_{\max}(n) = d_{\max}(n+1)$ and $d_{\min}(n) = d_{\min}(n+1)$. We have

$$\omega(n+1) = \omega(n) + \frac{1}{2}.$$

This is because both branches of the piecewise definition of $\omega(n)$ are linear with slope $\frac{1}{2}$, and for $n \neq \frac{N-1}{2}$, the same rule applies in both n and $n+1$. Thus

$$\delta(n) = \bar{d}(n) - \omega(n) \tag{5}$$
$$= \bar{d}(n+1) - \left(\omega(n+1) - \tfrac{1}{2}\right) \tag{6}$$
$$= \delta(n+1) + \tfrac{1}{2}. \tag{7}$$

Case 2: Either $d_{\max}(n) = d_{\max}(n+1)-1$ and $d_{\min}(n) = d_{\min}(n+1)$, or $d_{\max}(n) = d_{\max}(n+1)$ and $d_{\min}(n) = d_{\min}(n+1) - 1$. In both cases, $\bar{d}(n) = \bar{d}(n+1) - \frac{1}{2}$, while $\omega(n+1) = \omega(n) + \frac{1}{2}$ as before. Hence

$$\delta(n) = \bar{d}(n+1) - \frac{1}{2} - \left(\omega(n+1) - \frac{1}{2}\right) = \delta(n+1).$$

Case 3: $d_{\max}(n) = d_{\max}(n+1)-1$ and $d_{\min}(n) = d_{\min}(n+1)-1$. Then $\bar{d}(n) = \bar{d}(n+1) - 1$, and again $\omega(n+1) = \omega(n) + \frac{1}{2}$. Thus

$$\delta(n) = \bar{d}(n+1) - 1 - \left(\omega(n+1) - \frac{1}{2}\right) = \delta(n+1) - \frac{1}{2}.$$

This completes the proof. ∎

In order to treat the case where $n = \frac{N-1}{2}$ (see Lemma 2), we first need to make some additional considerations.

Let us remind the fundamental concept that all the non-zero codewords of $\mathcal{C}^{\hat{\mathbf{p}}}(n)$ can be obtained as all the vectors selected by a sliding window of length $n = N - \rho < 2^k - 1$ that circularly spans over $\hat{\mathbf{p}}$. Therefore, the minimum distance of this code is the weight of the length-n subsequence(s) of $\hat{\mathbf{p}}$ with the smallest weight, whereas its maximum distance is the weight of the length-n subsequence(s) of $\hat{\mathbf{p}}$ with the largest weight. In other words, $d_{\min}(n) = \min_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}}))$ and $d_{\max}(n) = \max_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}}))$.

On the one hand, for a given n, we thus have that $d_{\min}(n) = d_{\min}(n+1) - 1$ if and only if at least one of the subsequences of weight $d_{\min}(n+1)$ ends with a symbol 1. On the other hand, $d_{\max}(n) = d_{\max}(n+1) - 1$ if and only if all the subsequences of weight $d_{\max}(n+1)$ end with a symbol 1.

Lemma 1: Given an m-sequence $\hat{\mathbf{p}}$, we have that

- $\min_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}})) = 0$ for $0 \le n \le k-1$
- $\max_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}})) = n$ for $0 \le n \le k$
- $\min_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}})) = 2^{k-1} - (N-n)$ for $N-k \le n \le N$
- $\max_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}})) = 2^{k-1}$ for $N-k+1 \le n \le N$

Proof: An m-sequence derived from a polynomial of degree k contains all the possible non-zero subsequences of length k (each occurring only once), and it also contains the all-zero subsequence of length $k-1$. Therefore, the minimum weight of the length-n subsequences of $\hat{\mathbf{p}}$ cannot be larger than 0, if $n \le k-1$ (because the subsequences 0, 00, 00…0 up to length $k-1$ exist). Similarly, if $n \le k$, the maximum weight of the length-n subsequences of $\hat{\mathbf{p}}$ is exactly n, since the subsequences 1, 11, 11…1 of length up to k exist. This proves that $\min_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}})) = 0$ when $n < k$, and $\max_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}})) = n$, when $n \le k$. The other statements can be proven with identical arguments. ∎

Next, we study how the minimum and maximum distances vary with codeword length when puncturing a punctured simplex code. This is done in probabilistic terms. Namely, we briefly analyze the probability that a single puncturing operation on a punctured simplex code of length n, i.e., $\mathcal{C}(n)$, causes a drop of the minimum/maximum distance by one unit in the resulting code $\mathcal{C}(n-1)$. For the minimum distance and the maximum distance, this probability is represented by

$$\Pr\{d_{\min}(n-1) = d_{\min}(n) - 1\},$$
$$\Pr\{d_{\max}(n-1) = d_{\max}(n) - 1\},$$

respectively. Thus, given a family of punctured simplex codes represented by $\hat{\mathbf{p}}$, if we set

$$d_{\min}(n) = \min_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}}))$$
$$d_{\max}(n) = \max_{0\le i\le n-1} \mathrm{wt}(W_{i,n}(\hat{\mathbf{p}}))$$

also for the "degenerate" codes for which $n \le k$, then Lemma 1 allows us to claim that:

- $\Pr\{d_{\min}(n-1) = d_{\min}(n) - 1\} = 0$ when $n \le k-1$;
- $\Pr\{d_{\min}(n-1) = d_{\min}(n) - 1\} = 1$ when $N-k \le n \le N$;
- $\Pr\{d_{\max}(n-1) = d_{\max}(n) - 1\} = 0$ when $N-k+1 \le n \le N$;
- $\Pr\{d_{\max}(n-1) = d_{\max}(n) - 1\} = 1$ when $n \le k$.

In fact, for "degenerate" codes ($n \le k$), it immediately follows from the proof of Lemma 1 that

$$d_{\min}(n) = \begin{cases} 0, & n < k, \\ 1, & n = k, \end{cases} \qquad d_{\max}(n) = n \quad (n \le k). \tag{8}$$

Up to this point, we have only studied some extreme values of n. Let us provide the following more general results, holding for any choice of the primitive polynomial $h(x)$.

Corollary 1: On average, over all values of $k \le n \le 2^k - 1$, we have that $\Pr\{d_{\min}(n-1) = d_{\min}(n) - 1\} = \frac{2^{k-1}-1}{2^k-k-1}$. Furthermore, on average, over all values of $k+1 \le n \le 2^k - 1$, $\Pr\{d_{\max}(n-1) = d_{\max}(n) - 1\} = \frac{2^{k-1}-k}{2^k-k-2}$.

Proof: It follows from Lemma 1 that $d_{\min}(k) = 1$. Furthermore we already know that $d_{\min}(N) = 2^{k-1}$. Therefore, for $k \le n \le N$, on average, $\Pr\{d_{\min}(n-1) = d_{\min}(n) - 1\} = \frac{2^{k-1}-1}{(2^k-1)-k}$. Similarly, $d_{\max}(k+1) = k$, since the length-$(k+1)$ all-ones subsequence cannot exist (otherwise the all-ones subsequence of length k would appear twice in $\hat{\mathbf{p}}$, contradicting the hypothesis that it is an m-sequence). Moreover, we know

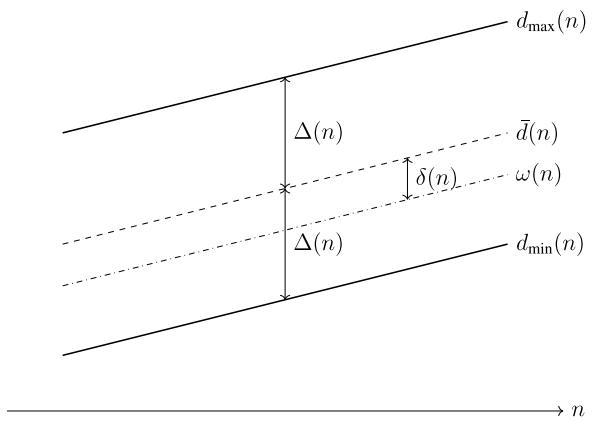


Fig. 1. Exemplary behavior of $\Delta(n)$, $\delta(n)$, $\bar{d}(n)$ and $\omega(n)$.

that $d_{\max}(N) = 2^{k-1}$. We thus have that, on average over $k+1 \le n \le N$, $\Pr\{d_{\max}(n-1) = d_{\max}(n) - 1\} = \frac{2^{k-1}-k}{2^k-1-(k+1)}$. ∎

Clearly, for increasing values of k, the average probabilities described in the above corollary tend to $1/2$.

By substitution of (4) into (2) we obtain the following relationships, holding for any family of punctured codes

$$\begin{aligned} d_{\min}(n) &= \omega(n) + \delta(n) - \Delta(n), \\ d_{\max}(n) &= \omega(n) + \delta(n) + \Delta(n). \end{aligned} \tag{9}$$

Equation (9) is particularly relevant for PRC-LDPC codes, because we will be able to derive insights regarding the functions $\Delta(n)$ and $\delta(n)$ for such codes, which might not be applicable to general codes. In Fig. 1, we provide an intuitive view of the parameters featured in this section in a hypothetical situation in which the values of $d_{\min}(n)$ and $d_{\max}(n)$ are increasing with n. We will denote the average value of any of these parameters over all possible primitive parity-check polynomials by using the brackets $\langle\cdot\rangle$.

Theorem 1: For punctured simplex codes of length $n = N - \rho > k$, where $1 \le \rho \le N - (k+1)$, it holds that $\Delta(n) = \Delta(\rho)$.

Proof: First of all, we need to prove that, for punctured simplex codes, $d_{\min}(N-\rho) + d_{\max}(\rho) = 2^{k-1}$, for $1 \le \rho \le N - (k+1)$. The values of $d_{\min}(\rho)$ and $d_{\max}(\rho)$ for $1 \le \rho \le k$ are specified in (8). Let us suppose that the code $\mathcal{C}(n)$ has a codeword of weight w, i.e., the sliding window of size n intercepts a codeword of weight w in $\hat{\mathbf{p}}$, which has weight 2^{k-1}, being an m-sequence. Let us assume that such a codeword, $\mathbf{c}_1$, is selected by $W_{i_1,n}(\hat{\mathbf{p}})$.

Then, there also exists a codeword of the code $\mathcal{C}(N-n)$ of weight $2^{k-1} - w$, selected in $\hat{\mathbf{p}}$ by the sliding window of size $N - n = \rho$. Such a codeword, $\mathbf{c}_2$, is selected by $W_{i_2,N-n}(\hat{\mathbf{p}})$, where $i_2 = i_1 + n$.

If $\mathbf{c}_1$ is a minimum weight codeword for $\mathcal{C}(n)$, then necessarily $\mathbf{c}_2$ is a maximum weight codeword for $\mathcal{C}(N-n)$, since minimization of w implies maximization of $2^{k-1} - w$. Therefore, $d_{\min}(N-\rho) = 2^{k-1} - d_{\max}(\rho)$, which proves the initial statement. Analogous reasoning yields $d_{\max}(N-\rho) = 2^{k-1} - d_{\min}(\rho)$.

It follows that

$$d_{\min}(N-\rho) + d_{\max}(\rho) = d_{\max}(N-\rho) + d_{\min}(\rho) = 2^{k-1}, \tag{10}$$

from which

$$\frac{d_{\max}(N-\rho) - d_{\min}(N-\rho)}{2} = \frac{d_{\max}(\rho) - d_{\min}(\rho)}{2},$$

implying that $\Delta(N-\rho) = \Delta(n) = \Delta(\rho)$. ∎

Theorem 2: For punctured simplex codes of length $n = N - \rho$, where $1 \le \rho \le N - (k+1)$, the following holds: $\delta(n) = -\delta(\rho)$.

Proof: Let us assume that $\rho < \frac{N}{2}$, and thus $\omega(\rho) = \rho/2$, and $\omega(N-\rho) = \frac{N-\rho+1}{2}$, according to (3). If $\rho \ge \frac{N}{2}$, the proof is identical. Let us also recall (10) from the proof of Theorem 1. Then, we can write, sequentially,

$$\begin{aligned} 2^k - 1 &= N \\ 2 \cdot 2^{k-1} &= N + 1 \\ 2\big[d_{\max}(N-\rho) + d_{\min}(\rho)\big] &= N + 1 \\ d_{\max}(N-\rho) + d_{\min}(\rho) &= -d_{\max}(N-\rho) - d_{\min}(\rho) \\ &\quad + N + 1 \\ d_{\min}(N-\rho) + d_{\max}(\rho) &= -d_{\max}(N-\rho) - d_{\min}(\rho) \\ &\quad + N + 1 \\ d_{\max}(\rho) + d_{\min}(\rho) &= -\big[d_{\max}(N-\rho) + d_{\min}(N-\rho)\big] \\ &\quad + N + 1 \\ \frac{d_{\max}(\rho) + d_{\min}(\rho)}{2} - \frac{\rho}{2} &= -\frac{d_{\max}(N-\rho) + d_{\min}(N-\rho)}{2} \\ &\quad + \frac{N+1}{2} - \frac{\rho}{2} \\ \frac{d_{\max}(\rho) + d_{\min}(\rho)}{2} - \frac{\rho}{2} &= -\frac{d_{\max}(N-\rho) + d_{\min}(N-\rho)}{2} \\ &\quad + \frac{N-\rho+1}{2} \\ \bar{d}(\rho) - \omega(\rho) &= -\big(\bar{d}(N-\rho) - \omega(N-\rho)\big) \\ \delta(\rho) &= -\delta(N-\rho) \end{aligned}$$

where again (8) provides the values of $d_{\min}(\rho)$ and $d_{\max}(\rho)$ when $1 \le \rho \le k$. ∎

Let us now study the values of $\delta(n)$ around the symmetry point. To this end, we define

$$\begin{aligned} n' &= \frac{N-1}{2}, \\ n'' &= \frac{N+1}{2}. \end{aligned} \tag{11}$$

Lemma 2: Either $\delta(n') = -\delta(n'') = 1/2$, or $\delta(n') = -\delta(n'') = 0$.

Proof: Substituting n' and n'' into the upper equation in (9), we obtain

$$\begin{aligned} d_{\min}(n') &= \frac{(N-1)/2}{2} + \delta(n') - \Delta(n') \\ &= \frac{(N-1)}{4} - \Delta(n') + \delta(n'), \\ d_{\min}(n'') &= \frac{(N+1)/2}{2} + \frac{1}{2} - \Delta(n') - \delta(n') \\ &= \frac{(N-1)}{4} - \Delta(n') - \delta(n') + 1, \end{aligned}$$

where we have exploited the fact that $\Delta(n') = \Delta(n'')$ and $\delta(n') = -\delta(n'')$. Moreover, since the code $\mathcal{C}(n')$ can be obtained from $\mathcal{C}(n'')$ by applying a single puncturing operation, either

- $d_{\min}(n') = d_{\min}(n'')$, from which we obtain that $\delta(n') = -\delta(n') + 1$, i.e., $\delta(n') = \frac{1}{2}$, or
- $d_{\min}(n') = d_{\min}(n'') - 1$, from which we obtain that $\delta(n') = -\delta(n')$, i.e., $\delta(n') = 0$.

This proves the thesis. ∎

Summarizing, the largest variation of $\delta(\cdot)$ caused by a single puncturing operation is 1/2, except when $\mathcal{C}\left(\frac{N+1}{2}\right)$ is punctured, in which case $\delta(\cdot)$ can decrease by one unit, as proved in Lemma 2. The following corollary is a straightforward consequence of the above reasoning (this result is also partially proven in [18] and, for this reason, we omit the proof), and provides quite useful bounds on the minimum and maximum distances of PRC-LDPC codes, and of punctured simplex codes in general.

Corollary 2: Given a punctured simplex code $\mathcal{C}^{\hat{\mathbf{p}}}(n)$ of dimension k, it holds that

$$d_{\min}(n) \le \omega(n) \le d_{\max}(n).$$

We can also provide bounds for the values of $\Delta(n)$ and $\delta(n)$, according to the following corollary.

Corollary 3: Given a punctured simplex code $\mathcal{C}^{\hat{\mathbf{p}}}(n)$ of dimension k, it holds that

$$0 \le \Delta(n) \le 2^{k-2} - \frac{1}{2}, \tag{12}$$

$$|\delta(n)| \le 2^{k-3}. \tag{13}$$

Proof: In order to prove the corollary, it is sufficient to observe that:

- $\Delta(n)$ increases by at most 1/2, for each unitary increase of n, starting from 0 up to $\frac{N-1}{2}$; then, it cannot increase further, because it has even parity with respect to $n = \frac{N}{2}$. Therefore, the largest possible value of Δ is $\frac{N-1}{2} \cdot \frac{1}{2} = \frac{2^k-2}{4} = 2^{k-2} - \frac{1}{2}$. Moreover, Δ cannot be negative, since $d_{\max} \ge d_{\min}$.
- $\delta(n)$ increases by 1/2, or decreases by the same quantity, for each unitary increase of n, starting from 0 up to $\frac{N+1}{4}$. Then it cannot increase, or decrease, further, because it has odd parity with respect to $n = \frac{N}{2}$. Therefore, the largest possible absolute value of δ is $\frac{N+1}{8} = 2^{k-3}$. ∎

Remark 1: Corollary 1 and Theorems 1 and 2 give us hints on the average behavior of $\Delta(n)$ and $\delta(n)$. In fact, based on them, we expect $\Delta(n)$ to have a "trapezoid-like" shape, with a decreasing trend for large values of n, followed by a rather stable trend as the window size gets smaller and, because of the symmetry, an increasing trend for small values of n. Instead, for $\delta(n)$ we expect a rather "flat" behavior, since it is odd and there is no prevailing trend that makes it decrease (or increase) significantly when the window size changes, especially for intermediate values of n. Numerical examples supporting this reasoning will be provided in Section V.

By using an exhaustive numerical search, we have computed the values of $d_{\min}$, $d_{\max}$, Δ and δ when the codeword length takes the values n' and n'' in (11), which are of particular interest since they are the closest ones to the symmetry axes of Δ and δ, for many primitive polynomials characterized by degree between 7 and 14. The numerical results are reported in Appendix A. For $7 \le k \le 9$, all existing primitive polynomials have been considered. Instead, to save space, only a subset of the whole set of primitive polynomials has been reported for $k > 9$. However, note that polynomials with different weights have been considered, in order to show that, for low code rates, the polynomial weight does not influence significantly the minimum distance properties. The maximum value of k has been chosen to allow exhaustive computation of all the variables under analysis, particularly the minimum and maximum distance, which becomes too expensive for larger values of k. Results reported in the tables of Appendix A confirm the theoretical results previously provided in this section. In fact, we notice that either $\delta(n') = \delta(n'') = 0$, or $\delta(n') = -\delta(n'') = 1/2$. Instead, due to parity, $\Delta(n') = \Delta(n'')$. Then, as expected, stepping from n' to n'', the minimum and maximum distance either remain the same, or increase by one unit.

Lemma 3: Given the set of primitive polynomials of degree k, let us consider the corresponding ensemble of punctured simplex codes of length n' and dimension k. If this ensemble is sufficiently large, it holds that

$$\langle \Delta(n') \rangle \approx 2^{k/2-1}. \tag{14}$$

Proof: For the sake of simplicity, but safeguarding accuracy, let us approximate the average weight distribution of the ensemble of punctured simplex codes of length n' with the ideal Hamming weight distribution of the non-zero subsequences of length n' of an m-sequence [30], that is,[2]

$$A(w) \approx \frac{2^k - 1}{2^{n'} - 1} \binom{n'}{w}.$$

This is possible because, as discussed in Section II, the non-zero codewords of any punctured simplex code of length n can be obtained as all the vectors selected by a sliding window of length n that circularly spans over the m-sequence $\hat{\mathbf{p}}$. We thus have

$$A(w) \approx \frac{2^k - 1}{2^{\frac{2^k-2}{2}} - 1} \binom{\frac{2^k-2}{2}}{w} \approx \frac{\binom{2^{k-1}}{w}}{2^{2^{k-1}-k-1}}.$$

By using Stirling's approximation for the binomial, we get

$$A(w) \approx 2^{2^{k-1}H_2(w/2^{k-1}) - \frac{1}{2}\log_2\left[2\pi w\left(1-\frac{w}{2^{k-1}}\right)\right] - 2^{k-1} + k + 1}.$$

In order to get $\langle d_{\min}(n') \rangle$, we need to consider that for weights such that $2^{k-1}H_2(w/2^{k-1}) < \frac{1}{2}\log_2\left[2\pi w\left(1-\frac{w}{2^{k-1}}\right)\right] + 2^{k-1} - k - 1$, the expectation of $A(w)$ is smaller than 1, whereas for weights such that $2^{k-1}H_2(w/2^{k-1}) > \frac{1}{2}\log_2\left[2\pi w\left(1-\frac{w}{2^{k-1}}\right)\right] + 2^{k-1} - k - 1$, the expectation of $A(w)$ is larger than 1. We can thus expect the average minimum distance to be close to the (smallest) value of w for which $A(w) = 1$. A similar reasoning holds for $\langle d_{\max}(n') \rangle$. We thus study the following equation

$$2^{k-1}H_2\left(\frac{w}{2^{k-1}}\right) = \frac{1}{2}\log_2\left[2\pi w\left(1-\frac{w}{2^{k-1}}\right)\right] + 2^{k-1} - k - 1. \tag{15}$$

By setting $x = \frac{w}{2^{k-1}}$ we get

$$2^{k-1}H_2(x) = \frac{1}{2}\log_2[\pi 2^k x(1-x)] + 2^{k-1} - k - 1.$$

[2]More accurate approximations of the weight distribution, such as those presented in [19] and [22], can be employed, at the expense of significantly increased complexity in the theoretical analysis.

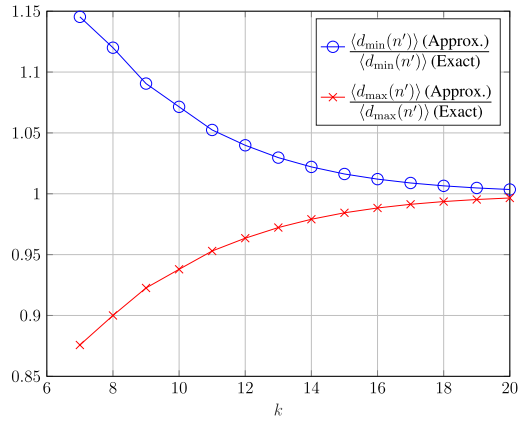


Fig. 2. Ratio of approximate and exact solutions of (15), for some values of k.

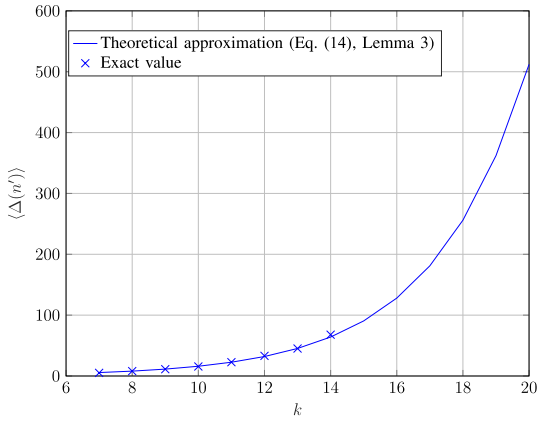


Fig. 3. Approximated (according to Lemma 3) and exact values of the average $\Delta(n')$, for some values of k.

By solving this transcendental equation numerically, we get that it has two solutions (as expected, since $A(w)$ has binomial shape), which can be approximated by

$$x_1 \approx \frac{1}{2} - \frac{1}{2^{k/2}},$$
$$x_2 \approx \frac{1}{2} - \frac{1}{2^{k/2}} + \frac{1}{2^{k/2-1}}.$$

Since, $x_1 < x_2$, we can derive $\langle d_{\min}(n')\rangle = x_1 2^{k-1}$ and $\langle d_{\max}(n')\rangle = x_2 2^{k-1}$, i.e.,

$$\langle d_{\min}(n')\rangle \approx 2^{k-2} - 2^{k/2-1},$$
$$\langle d_{\max}(n')\rangle \approx 2^{k-2} - 2^{k/2-1} + 2^{k/2}.$$

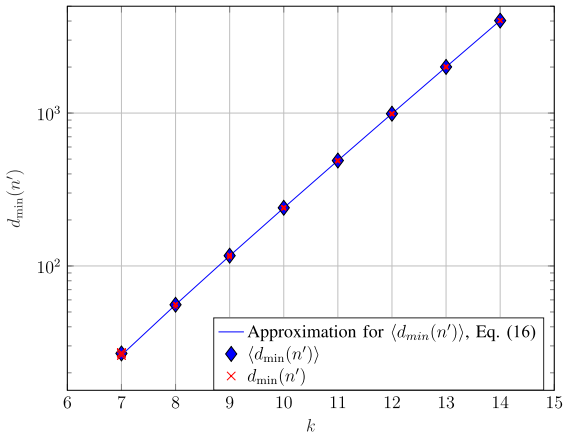
We finally obtain

$$\langle \Delta(n')\rangle = \frac{\langle d_{\max}(n')\rangle - \langle d_{\min}(n')\rangle}{2} \approx \frac{2^{k/2}}{2} = 2^{k/2-1}.$$

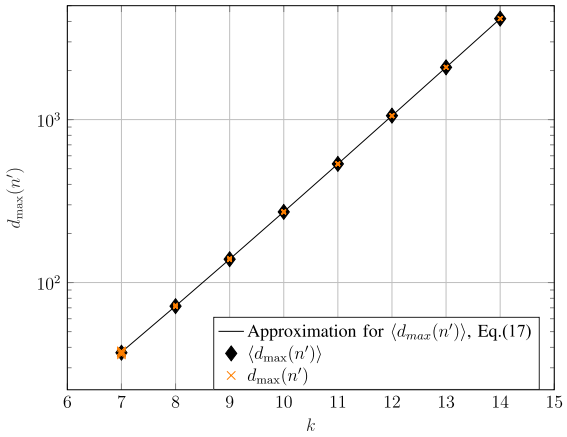
■

In Fig. 2 we report the ratio of the approximate and exact solutions of (15), for some values of k. We notice that the approximations get better and better as k increases, but are quite close to the exact solution even for small values of k.

From (14) we notice that the average value of $\Delta(n')$ is quite smaller than the (loose) bound on its absolute value derived in Corollary 3. In Fig. 3, we compare the average value of



(a)



(b)

Fig. 4. Approximated values of the average of (a) $d_{\min}(n')$ and (b) $d_{\max}(n')$, along with their exact values (markers only), for different values of k.

$\Delta(n')$, computed over all the primitive polynomials of degree $k \in [7, 14]$, with the theoretical value discussed in Lemma 3, and we note that they substantially match for all the considered values of k.

To further evaluate the accuracy of our approximations, in Fig. 4 we show the actual value of $\langle d_{\min}(n')\rangle$ and $\langle d_{\max}(n')\rangle$, the approximations

$$\langle d_{\min}(n')\rangle \approx \langle d_{\min}(n'')\rangle \approx \omega(n') - 2^{k/2-1} \tag{16}$$

and

$$\langle d_{\max}(n')\rangle \approx \langle d_{\max}(n'')\rangle \approx \omega(n') + 2^{k/2-1}, \tag{17}$$

and the actual values of $d_{\min}(n')$ and $d_{\max}(n')$. We observe that the approximation is remarkably tight for $k \in [7, 14]$, which supports its validity also for values of k where an exhaustive analysis is not feasible. We also notice that the actual values of the minimum distance are not significantly dispersed around the actual average value. Let us study this aspect more in depth.

Since, as discussed previously in this section, $\Delta(n)$ varies rather slowly with n (at most $\pm 1/2$ for a unit increment or decrement of n), we can expect (14) to be valid also around n'. In Table I we report the average and standard deviation

TABLE I
AVERAGE AND STANDARD DEVIATION VALUES OF $d_{\min}(n')$, $d_{\max}(n')$, $\Delta(n')$, FOR $k \in [7, 14]$

k	$\langle d_{\min}(n') \rangle$	$\sigma_{d_{\min}(n')}$	$\langle d_{\max}(n') \rangle$	$\sigma_{d_{\max}(n')}$	$\langle \Delta(n') \rangle$	$\sigma_{\Delta(n')}$	$2^{k/2-1}$	2^{k-2}
7	26.78	0.44	37.11	0.60	5.17	0.50	5.66	32
8	55.75	0.71	71.63	0.74	7.94	0.68	8	64
9	116.71	1.14	139.07	1.41	11.18	1.05	11.31	128
10	240.08	1.19	271.31	1.25	15.62	1.19	16	256
11	489.15	2.15	534.54	2.07	22.69	2.10	22.63	512
12	990.23	3.72	1057.08	3.64	33.42	3.67	32	1024
13	2003.70	5.31	2092.00	5.29	44.15	5.30	45.25	2048
14	4027.85	8.39	4163.62	8.63	67.88	8.50	64	4096

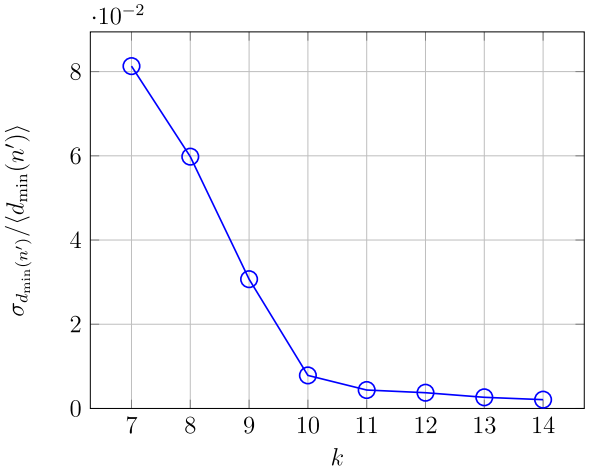


Fig. 5. Coefficient of variation of $d_{\min}(n')$.

values (noted by σ) of $d_{\min}(n')$, $d_{\max}(n')$, and $\Delta(n')$ for all the values of k considered in Appendix A. Table I also includes the values of $2^{k/2-1}$ and 2^{k-2}, which are approximations of $\langle \Delta(n') \rangle$ and $\langle \Delta(n') \rangle + \langle d_{\min}(n') \rangle$, respectively. The former approximation has been proven in Lemma 3, whereas the second one follows from (9), by considering that $\omega(n') = \frac{2^{k-1}-1}{2}$ and that $\delta(n')$ is either 0 or 1/2. We note that (14) and also the aforementioned approximation on $\langle \Delta(n') \rangle + \langle d_{\min}(n') \rangle$ are indeed tight. Furthermore, in Fig. 5 we show the coefficient of variation of $d_{\min}(n')$, defined as its standard deviation divided by its average value. We notice that, for increasing k, the coefficient of variation decreases, and that it assumes relatively small values. This allows us to claim that most of the families of punctured simplex codes have similar minimum distance properties, at least when the codeword length equals n'.

We provide a more explicit example next.

Example 1: In Fig. 6 we show the evolution of some of the parameters defined in the previous section for the code family associated to the parity-check polynomial $h(x) = 1 + x + x^4$, which is unpractical due to the small value of k, but allows a manageable graphical representation. In particular, in the figure:

- black circles represent the values of the minimum distance,
- blue circles represent the values of the maximum distance,
- hollow circles represent the values of $\bar{d}(n)$, that is, the average value between $d_{\min}(n)$ and $d_{\max}(n)$,

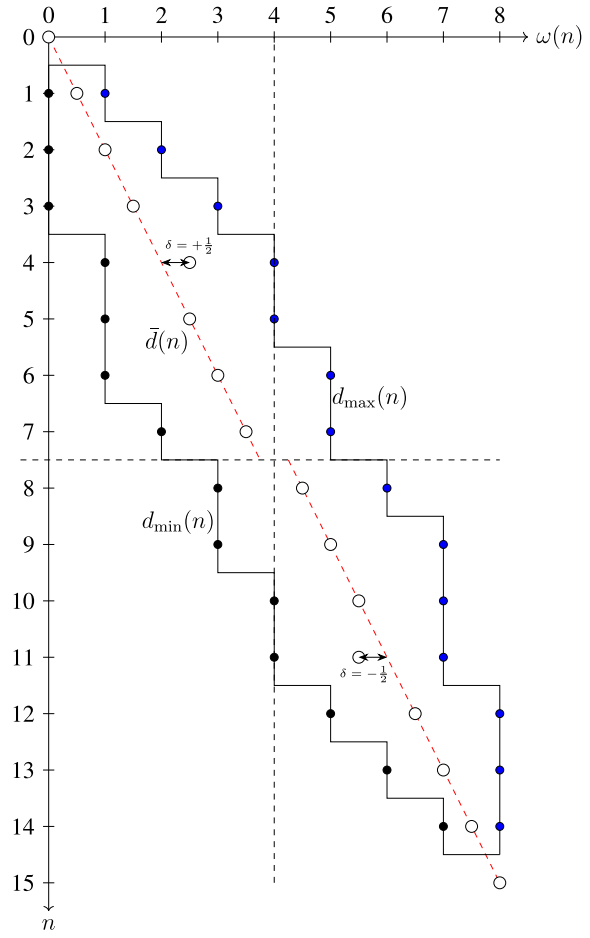


Fig. 6. Behavior of $d_{\min}(n)$ and $d_{\max}(n)$ (along with other parameters), for the code family described by $h(x) = 1 + x + x^4$.

- the dashed red line represents $\omega(n)$.

The step-like behavior of $d_{\min}(n)$ and $d_{\max}(n)$ is apparent. We mention again that, when $n \leq k$, the code is "degenerate". However, it is interesting to take also degenerate codes into account, to acknowledge symmetry properties. There is an anti-symmetrical horizontal axis between $n = 7$ and $n = 8$. In general, this exists between n' and n'', as discussed earlier. We can also observe a vertical axis of anti-symmetry corresponding to $A(2^{k-2})$. The parent cyclic simplex code, defined by $n = N = 15$, is not included in these observations about symmetry. It might align with a further degenerate code with $n = 0$ and all codewords being null. It can be observed (and

it is very easy to prove) that, for any code family, when $n \in [0,k] \cup [N-k,N]$, we have $\delta(n) = 0$. Moreover, and this is also a general and easily verifiable result, for $n \in [0,k]$ we have $\Delta(n) = n$, and for $n \in [N-k,N]$ we have $\Delta(n) = N - n$.

We remark that our results are consistent with those reported in [20], despite some important differences in the respective code constructions and analysis frameworks. In [20], a single, deterministically constructed punctured simplex code obtained by shortening a Hamming code is considered. This yields a fixed minimum distance equal to

$$d_{\min}(n''+1) = 2^{k-1} - 2^{k-2} = 2^{k-2}. \tag{18}$$

In contrast, our approach considers an ensemble of punctured simplex codes, where each code corresponds to a different primitive polynomial of degree k. As a consequence, our reported minimum distances are ensemble averages, and the analysis must take into account the variability across the ensemble. From this standpoint, our theoretical framework is more general than that of [20]. In particular, from (16) we obtain

$$\langle d_{\min}(n'')\rangle \approx 2^{k-2} - 2^{k/2-1} + \frac{1}{2}. \tag{19}$$

The difference between (18) and (19) vanishes exponentially fast with k. Specifically, the relative gap behaves as

$$\frac{2^{k/2-1} - \frac{1}{2}}{2^{k-2}} = O(2^{-k/2}),$$

which implies that for even moderately large k, both constructions exhibit essentially equivalent distance properties. Overall, the consistency between our results and those of [20] provides further support for the ensemble-based approach adopted in this work.

IV. Asymptotic Performance of Punctured Simplex Codes

In this section, we first compare the asymptotic properties of punctured simplex codes and cyclic simplex codes. Then, we compute some bounds on the error rate performance of low-rate punctured simplex codes.

A. Comparing Punctured Simplex Codes and Cyclic Simplex Codes

Cyclic simplex codes are characterized by the triplet $[2^k - 1, k, 2^{k-1}]$. Their average asymptotic coding gain (which is also the asymptotic coding gain of all the codes in the ensemble) is

$$G_{\infty,C} = \frac{k}{2^k - 1} 2^{k-1} \approx \frac{k}{2},$$

tending to ∞ when $k \to \infty$. However, for each unit increase in k, the codeword length doubles.

Let us now consider the ensemble of cyclic simplex codes of dimension k and codeword length $N_C = 2^k - 1$, and the punctured simplex codes of dimension $k+1$ and codeword length $n' = \frac{N-1}{2} = \frac{2^{k+1}-2}{2} = N_C$, obtained by puncturing the parent cyclic simplex code of dimension $k+1$ and codeword length $N = 2^{k+1} - 1$. We denote the average asymptotic coding gain of these punctured simplex codes as $G_{\infty,P}$. These two code ensembles have slightly different code rates: $R_C = \frac{k}{2^k-1}$ for the former one, and $R_P = \frac{k+1}{2^k-1}$ for the latter. Therefore, $\frac{R_C}{R_P} = \frac{k}{k+1}$. We are interested in studying for which values of k the asymptotic gain of the ensemble of punctured simplex codes is larger than that of the cyclic simplex codes, i.e., $G_{\infty,C} < G_{\infty,P}$.

Let us also consider that, from (9), we have

$$d_{\min}(n') = \omega(n') + \delta(n') - \Delta(n').$$

By taking into account Lemma 2, Lemma 3 and $\omega(n') = \frac{n'}{2}$ (according to (3)), and considering the average ensemble behavior, we get

$$\langle d_{\min}(n')\rangle \approx 2^{k-1} - \langle \Delta(n')\rangle.$$

So, we can compute

$$\frac{G_{\infty,C}}{G_{\infty,P}} = \frac{\frac{k}{2^k-1}2^{k-1}}{\frac{k+1}{2^k-1} d_{\min}(n')} \approx \frac{k}{k+1} \cdot \frac{2^{k-1}}{2^{k-1} - \langle \Delta(n')\rangle},$$

where we have substituted $d_{\min}(n')$ with its average value. The following theorem holds.

Theorem 3: Given the ensemble of cyclic simplex codes of dimension k, codeword length $2^k - 1$ and asymptotic coding gain $G_{\infty,C}$, and the corresponding ensemble of punctured simplex codes of dimension $k+1$, codeword length $2^k - 1$, and average asymptotic coding gain $G_{\infty,P}$, it holds that

$$\lim_{k\to\infty} \frac{G_{\infty,C}}{G_{\infty,P}} \approx 1.$$

Proof: According to Lemma 3, we have

$$\frac{G_{\infty,C}}{G_{\infty,P}} \approx \frac{k}{k+1} \cdot \frac{2^{k-1}}{2^{k-1} - \langle \Delta(n')\rangle} \approx \frac{k}{k+1} \cdot \frac{2^{k-1}}{2^{k-1} - 2^{\frac{k+1}{2}-1}}.$$

When $k \to \infty$, we have that $2^{\frac{k+1}{2}}$ becomes negligible with respect to 2^{k-1}.

Therefore,

$$\lim_{k\to\infty} \frac{G_{\infty,C}}{G_{\infty,P}} \approx \lim_{k\to\infty} \frac{k}{k+1} \cdot \frac{2^{k-1}}{2^{k-1} - 2^{\frac{k+1}{2}-1}} \tag{20}$$

$$= \lim_{k\to\infty} \frac{k}{k+1} = 1. \tag{21}$$

■

Thus, in the asymptotic setting, cyclic simplex codes with dimension k and punctured simplex codes with dimension $k+1$, both with the same codeword length $2^k - 1$, tend to have the same, optimal behavior.

After such reasoning in the asymptotic regime, we want to remark that we do not need infinitely large values of k to get $G_{\infty,C} < G_{\infty,P}$. Actually, we show the behavior of $\frac{G_{\infty,C}}{G_{\infty,P}}$ in Fig. 7, where we see that the ratio becomes smaller than 1 just for k in the order of 7, while, for increasing k, the ratio tends rapidly to 1, thus confirming the claim of Theorem 3.

We can generalize the above reasoning, by considering the ensemble of punctured simplex codes with dimension $k+m$, obtained by puncturing cyclic simplex codes of the same dimension until their codeword length is $2^k - 1$, and a cyclic simplex code of dimension k and codeword length $2^k - 1$. The average minimum distance of the former ensemble is $2^k - 1 - \langle \Delta\left(\left\lfloor \frac{2^{k+m}-1}{2^m} \right\rfloor\right)\rangle + \langle \delta\left(\left\lfloor \frac{2^{k+m}-1}{2^m} \right\rfloor\right)\rangle$. We can state a result

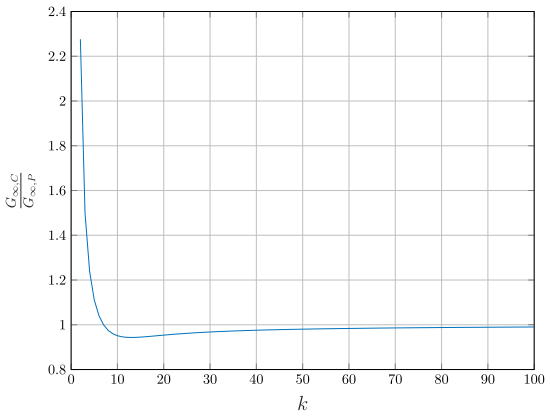


Fig. 7. Ratio of the asymptotic coding gain of simplex codes with dimension k and the asymptotic coding gain of punctured simplex codes with dimension $k+1$, having the same codeword length 2^k-1.

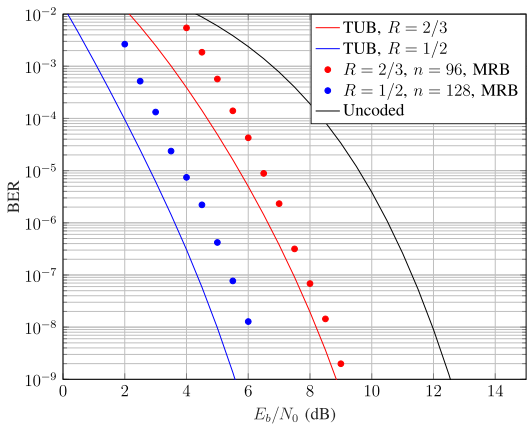


Fig. 8. Truncated union bound and error rate under order-4 MRB decoding for PRC-LDPC codes with dimension $k=64$, and different codeword lengths.

that is similar to Theorem 3, but holds under more stringent assumptions.

Corollary 4: Given the ensemble of cyclic simplex codes of dimension k, codeword length 2^k-1 and asymptotic coding gain $G_{\infty,C}$, and the ensemble of punctured simplex codes of dimension $k+m$, codeword length 2^k-1, and average asymptotic coding gain $G_{\infty,P}$, if $\langle\Delta\left(\left\lfloor\frac{2^{k+m}-1}{2^m}\right\rfloor\right)\rangle = o(2^{k-1})$, and $\langle\delta\left(\left\lfloor\frac{2^{k+m}-1}{2^m}\right\rfloor\right)\rangle \ll \langle\Delta\left(\left\lfloor\frac{2^{k+m}-1}{2^m}\right\rfloor\right)\rangle$, then it holds that

$$\lim_{k\to\infty}\frac{G_{\infty,C}}{G_{\infty,P}} \approx 1.$$

Proof: If $\langle\Delta\left(\left\lfloor\frac{2^{k+m}-1}{2^m}\right\rfloor\right)\rangle = o(2^{k-1})$, then the proof follows that of Theorem 3, and is therefore omitted. ∎

A sufficient condition for Corollary 4 to hold is, for example, $\langle\Delta\left(\left\lfloor\frac{2^{k+m}-1}{2^m}\right\rfloor\right)\rangle \approx \langle\Delta\left(\left\lfloor\frac{2^{k+m}-1}{2}\right\rfloor\right)\rangle$. According to Remark 1, this is reasonable, at least for small values of m.

B. Bounds on the Error-Rate Performance of Punctured Simplex Codes

Let us consider, as an example, the primitive polynomial

$$h(x) = 1 + x + x^4 + x^{28} + x^{33} + x^{47} + x^{64},$$

and the corresponding family of punctured simplex codes (which are also PRC-LDPC codes, since the support vector of the coefficients vector corresponding to $h(x)$ is a Golomb ruler) with

1) $R=2/3$, characterized by the triplet $[96, 64, 3]$.
2) $R=1/2$, characterized by the triplet $[128, 64, 9]$.

The codewords of these codes have been found by using the tool in [32], and the method described in [33], which clearly only allow to estimate the minimum distance, rather than exactly computing it.

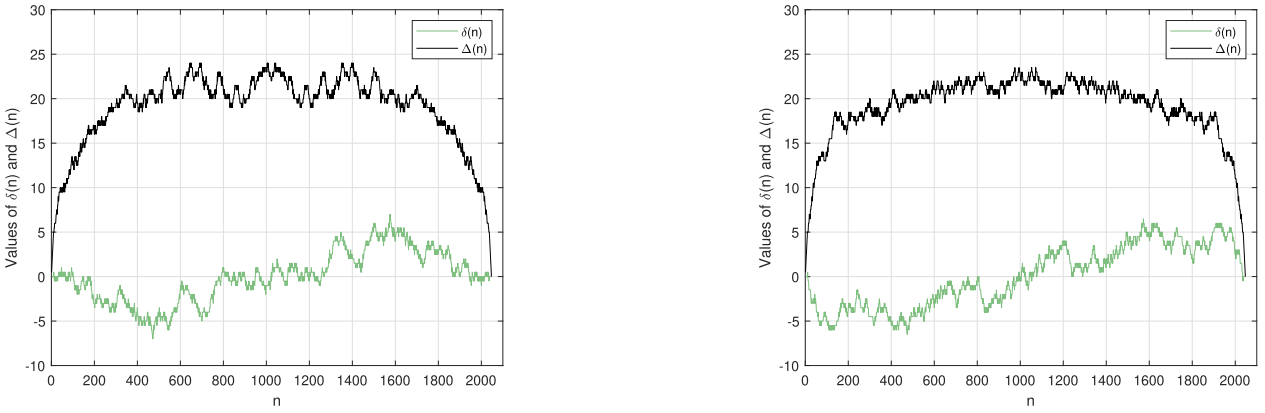
In Fig. 8 we show the BER truncated union bound (TUB) for these codes, computed as

$$\mathrm{BER}_{\mathrm{TUB}} \approx \sum_{w=d_{\min}}^{d^*}\frac{1}{2}\frac{w}{n}A(w)\mathrm{erfc}\left(\sqrt{w\frac{k}{n}\frac{E_b}{N_0}}\right),$$

where $d_{\min} \le d^* \le d_{\max}$. Clearly, the larger d^*, the tighter the TUB to the complete union bound, which corresponds to $d^* = d_{\max}$. The considered values of d^* are 7 and 53 for the codes with rate $2/3$ and $1/2$, respectively. We remind that the complexity of finding codewords increases with their weight and with the codeword length.

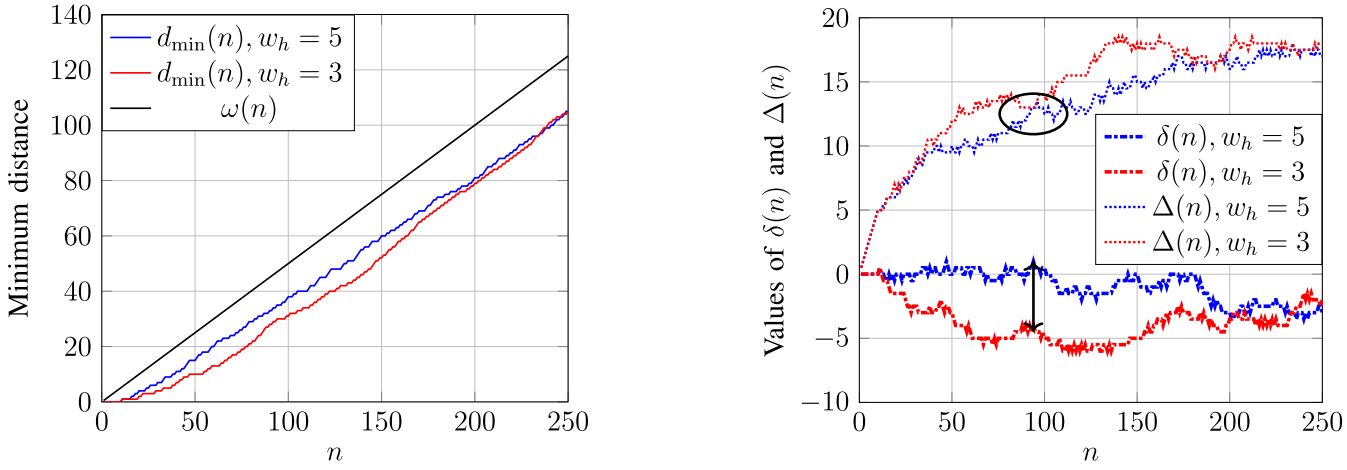
We observe that, as expected, the (truncated) union bounds shift to the left as fewer symbols are punctured. To verify the tightness of such bounds to the performance actually achievable by these codes, the error rate performance of the codes with rate $2/3$ and $1/2$ has been assessed through Monte Carlo simulations of BPSK-modulated transmissions over the additive white Gaussian noise (AWGN) channel. Decoding has been performed through the Most Reliable Basis (MRB) algorithm [34] which, as its order increases, approaches performance of the maximum-likelihood decoder. Complexity of the MRB algorithm, however, is higher than that of the iterative decoding algorithms commonly used for LDPC codes, which usually prevents us from considering high values of its order. In particular, MRB decoding of order 4 was employed to obtain the performance shown in Fig. 8.

Given these promising results, in the rest of the paper we focus solely on PRC-LDPC codes. To achieve good performance under belief propagation (BP) decoding, PRC-LDPC codes should be carefully designed by leveraging both puncturing and shortening operations (whereas, up till now in the paper, only puncturing operations have been considered). Effective code design also requires optimizing the specific primitive parity-check polynomial that defines the parity-check matrix. The choice of the primitive polynomial influences the cycle distribution, weight distribution, and minimum distance properties of the code. These factors, in turn, impact the error rate performance in both the waterfall and error floor regions. In this paper, and particularly in Section V, we focus on the minimum distance properties. Consequently, in the rest of the paper, we shift our focus from low- (or moderate-) rate codes to more practical (relatively) high-rate codes. Specifically, we address practical code design in Section VI and show the performance of shortened and punctured PRC-LDPC codes in Section VII.



(a) (b)

Fig. 9. $\delta(n)$ and $\Delta(n)$ for (a) $h(x) = 1 + x + x^4 + x^9 + x^{11}$ and (b) $h(x) = 1 + x^2 + x^{11}$.



(a) (b)

Fig. 10. (a) Minimum distance and (b) values of $\delta(n)$ and of $\Delta(n)$ for the codes in $\mathcal{C}_{\hat{\mathbf{p}}}$, considering (22) and (23) as primitive polynomials.

V. Towards High-Rate Codes: Numerical Examples

In this section, we provide some additional numerical examples, by considering primitive polynomials for which the support vector of the coefficients vector is a Golomb ruler, thus yielding PRC-LDPC codes [16]. Specifically, we investigate the minimum distance properties of two codes constructed using different primitive polynomials of the same degree but with distinct weights.

Let us consider

$$h(x) = 1 + x + x^4 + x^9 + x^{11}, \tag{22}$$

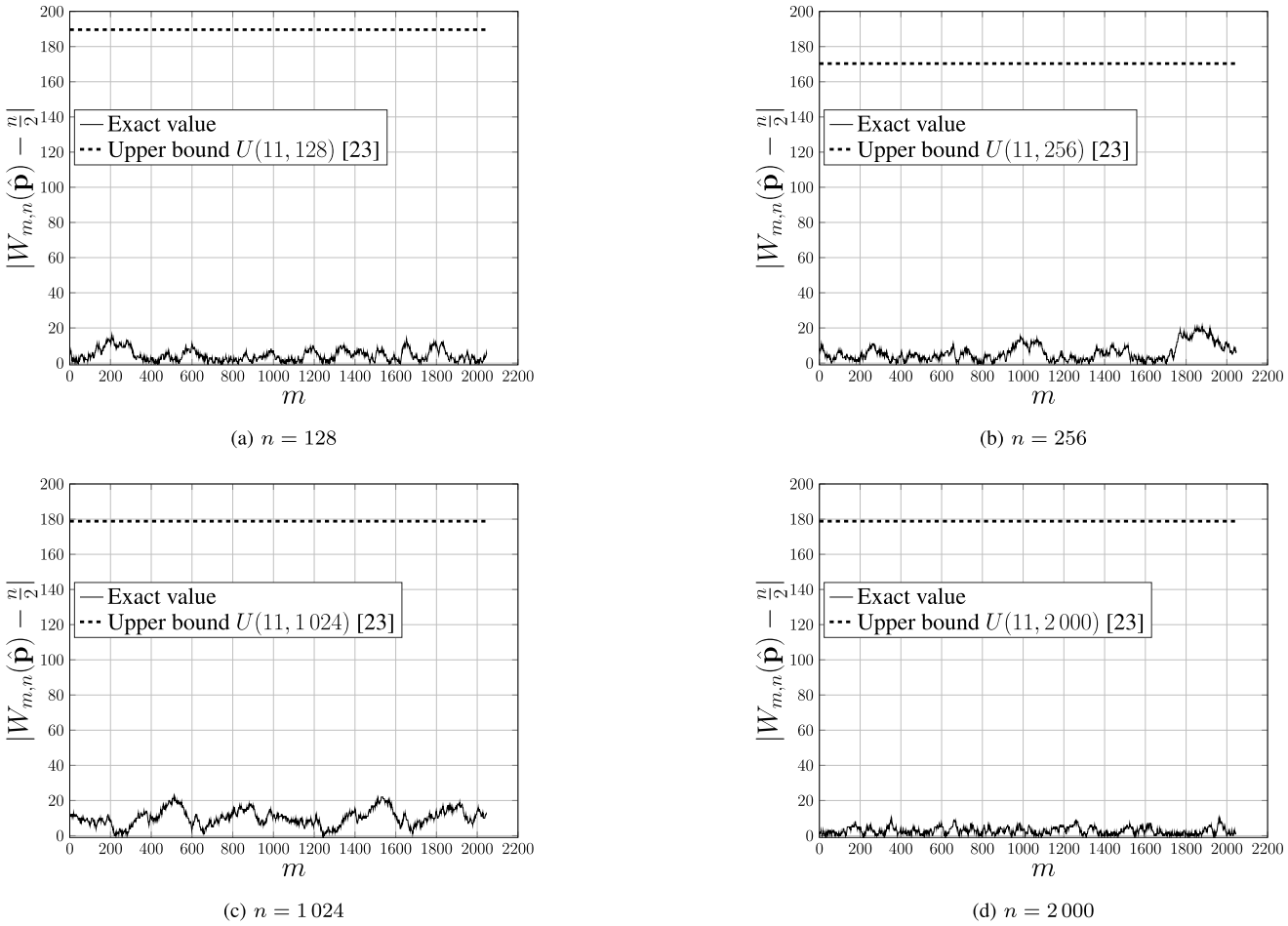
for which $\mathrm{Supp}(\mathbf{h}) = [0, 1, 4, 9, 11]$ is a Golomb ruler. Any code in $\mathcal{C}_{\hat{\mathbf{p}}}$ thus satisfies the RCC. We show the behavior of $\Delta(n)$ and $\delta(n)$ in Fig. 9(a).

We compare the behavior of this family of codes to that of the family associated to the primitive polynomial with $w_h = 3$

$$h(x) = 1 + x^2 + x^{11}, \tag{23}$$

for which $\mathrm{Supp}(\mathbf{h}) = [0, 2, 11]$ is also a Golomb ruler. The behavior of $\delta(n)$ and $\Delta(n)$ for this family of codes is also shown in Fig. 9(b). We note that, as discussed in Remark 1 and thus expected, $\Delta(n)$ has an initial rapid growing phase, a relatively stationary phase and, finally, a rapid decrease. We also notice that the actual values of $\Delta(n')$ (which equals 22 in both cases) are very close to the approximation $2^{\frac{k}{2}-1} = 22.63$. Moreover, we observe that the approximation of $\langle\Delta(n')\rangle$ derived in Section III, specifically $2^{k/2-1}$, serves as a good estimate for general values of $\Delta(n)$ as well, provided that the ratio n/n' is sufficiently small. Moreover, especially for relatively small values of n, and thus moderate to large values of the code rate, the latter family of codes exhibits many negative values of $\delta(n)$, and a rather quick increase of $\Delta(n)$. According to (9), the former family of codes is expected to yield better minimum distance properties for such values of n.

A portion of the minimum distance profile for these two families of PRC-LDPC codes, along with $\omega(n)$, is shown in Fig. 10(a); an excerpt of $\delta(n)$ and $\Delta(n)$ is instead shown in Fig. 10(b). In particular, for each value of n, the minimum distance has been computed as the smallest weight of any subsequence of length n of $\hat{\mathbf{p}}$. We observe that, as expected, for relatively small values of n, the codes with $w_h = 5$



(a) $n = 128$ (b) $n = 256$

(c) $n = 1\,024$ (d) $n = 2\,000$

Fig. 11. Values of and upper bound on the deviation between the weight of the subsequences of $\hat{\mathbf{p}}$ of length n, and the expected value $n/2$, for $m \in [0, N-1]$, different values of n, and $h(x) = 1 + x + x^4 + x^9 + x^{11}$.

exhibit larger minimum distances than those with $w_h = 3$. In particular, as evidenced in Fig. 10(a), in the zones where $\Delta(n)$ is comparable for the two families, which are highlighted by an ellipse in Fig. 10(b), the greatest contribution to the differences in terms of minimum distance for the considered codes is given by $\delta(n)$ (arrow in Fig. 10(b)). Instead, when the values of $\Delta(n)$ for the two families are approximately the same, and the same holds for the values of $\delta(n)$, then their minimum distances converge to (approximately) the same values, as apparent for $n \gtrapprox 200$. In general, code families characterized by the same values of k have comparable minimum distance when $|\Delta(n)| \gg |\delta(n)|$, even though they might need smaller or larger values of n to reach the stationary behavior of $\Delta(n)$.

In order to further validate our analysis, let us consider the upper bound proposed in [23, Theorem 8.85], stating that, $\forall m \in [0, N-1]$:

$$\left|W_{m,n}(\hat{\mathbf{p}}) - \frac{n}{2}\right| \leq 2^{k/2-1}\left(\frac{2}{\pi}\log_2(2^k - 1) + \frac{2}{5} + \frac{n}{2^k - 1}\right) = U(k, n). \tag{24}$$

So, given $h(x) = 1 + x + x^4 + x^9 + x^{11}$, for some arbitrarily chosen values of n, i.e., $n \in \{128, 256, 1\,024, 2\,000\}$, and $\forall m \in [0, N-1]$, we have computed both

$$\left|W_{m,n}(\hat{\mathbf{p}}) - \frac{n}{2}\right|$$

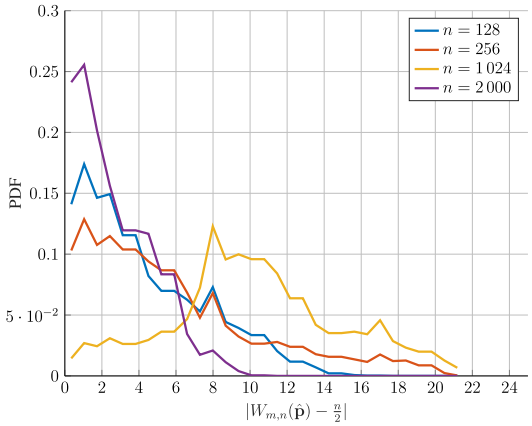
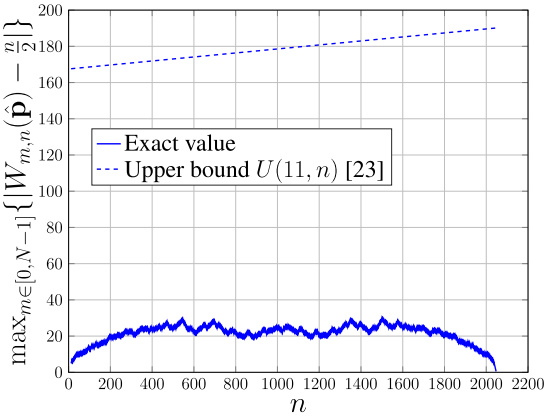


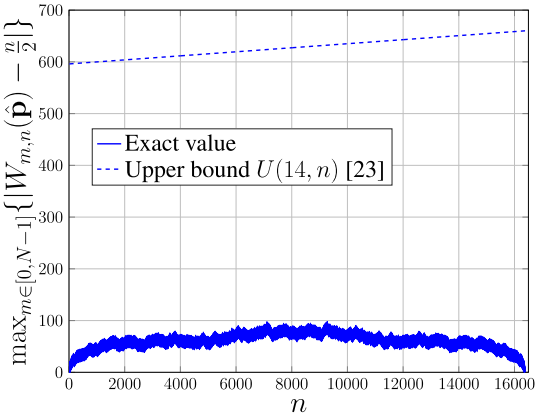
Fig. 12. PDF of $\left|W_{m,n}(\hat{\mathbf{p}}) - \frac{n}{2}\right|$, for $h(x) = 1 + x + x^4 + x^9 + x^{11}$ and different values of n.

and $U(11, n)$, and compared them. The results are shown in Fig. 11. We have also plotted in Fig. 12 the PDF of $\left|W_{m,n}(\hat{\mathbf{p}}) - \frac{n}{2}\right|$ for the selected values of n.

Furthermore, $\forall n \in [k+1, 2^k - 1]$, we have computed $\max_{m\in[0,N-1]}\left\{\left|W_{m,n}(\hat{\mathbf{p}}) - \frac{n}{2}\right|\right\}$ and compared it again with the upper bound, considering $h(x) = 1 + x + x^4 + x^9 + x^{11}$ (that



(a) $h(x) = 1 + x + x^4 + x^9 + x^{11}$



(b) $h(x) = 1 + x + x^6 + x^8 + x^{14}$

Fig. 13. Values of and upper bound on the maximum deviation between the weight of the subsequences of $\hat{\mathbf{p}}$ of length n, and the expected value $n/2$, for two different primitive parity-check polynomials with (a) $k = 11$, and (b) $k = 14$.

is, $k = 11$) and $h(x) = 1 + x + x^6 + x^8 + x^{14}$ (that is, $k = 14$). The results are shown in Fig. 13.

Fig.s 11 and 12 show that, as expected, for values of n which are relatively far from $\frac{N}{2}$, the weights of all the subsequences of $\hat{\mathbf{p}}$ of length n concentrate around $n/2$, as the deviation $\left|W_{m,n}(\hat{\mathbf{p}}) - \frac{n}{2}\right|$ becomes small for most values of m. This is also supported by Fig. 13, with regards to the maximum deviation between these weights and the expected value.

VI. Design of PRC-LDPC Codes for Complexity-Constrained Scenarios

Several degrees of freedom can be exploited in the selection of a PRC-LDPC code. Suppose that a code with rate R is required. Also suppose that the largest decoding complexity allowed by the application context in which the code needs to be used is $\Gamma_{\max}$, defined as the largest number of binary operations required to decode a codeword.

The sparsity of the parity-check matrix of PRC-LDPC codes enables efficient decoding through low-complexity iterative algorithms. Here, we make use of the implementation of the Sum-Product Algorithm (SPA) proposed in [35]. To achieve a good trade-off between complexity and performance, we consider 8-bit quantization for the decoder variables. In order to decode a codeword, whether decoding is successful or not,

$$\Gamma = nI_{\text{avg}} f(\langle w_c\rangle, R) \tag{25}$$

binary operations are needed [36], on average, where

$$f(\langle w_c\rangle, R) = 8(8\langle w_c\rangle + 12R - 11) + \langle w_c\rangle, \tag{26}$$

I_{avg} is the average number of decoding iterations, and $\langle w_c\rangle$ is the average column weight of the parity-check matrix.

As apparent from (25), having assumed $\Gamma_{\max}$ and R to be imposed by the application, the degrees of freedom in the code design are $h(x)$, which determines $\langle w_c\rangle$, and n.

The method we propose to design a PRC-LDPC code with the desired R, meeting the constraint on the largest decoding complexity allowed, is to start from an optimized PRC-LDPC with rate R, which does not necessarily comply with the constraint on the complexity, and then apply puncturing and shortening operations until the constraint is satisfied. Therefore, we need to exploit the inherent flexibility and adaptability of optimized PRC-LDPC codes to obtain codes with the same rate, but smaller codeword length. A similar idea was proposed in [37], with the difference that the codeword length was the only relevant parameter. Here, we also need to take into account the average column weight, which significantly complicates the code design.

Next, we describe the procedure we apply (called Procedure A in the following), by denoting the number of punctured positions as ρ (coherent with the notation used for this parameter in the previous sections), and that of shortened positions as s.

Given a code $\mathcal{C}_{\hat{\mathbf{p}}}(n)$ with rate $R = \frac{k}{n}$, for which $\Gamma > \Gamma_{\max}$:

1) set $n^* = \left\lfloor \frac{\Gamma_{\max}}{I_{\text{avg}} f(\langle w_c\rangle, R)} \right\rfloor$;
2) set $k^* = \lfloor Rn^* \rfloor$ and $s = k - k^*$;
3) puncture the last $\rho = n - n^* - s$ symbols of $\mathcal{C}_{\hat{\mathbf{p}}}(n)$, just as described in Section II-B, obtaining a new parity-check matrix $\mathbf{H}'$, of size $(n - k - \rho) \times (n - \rho)$;
4) shorten s symbols of the code corresponding to $\mathbf{H}'$, thus obtaining an $(n - k - \rho) \times (n - \rho - s)$ parity-check matrix $\mathbf{H}''$;
5) compute Γ for the code represented by $\mathbf{H}''$: if $\Gamma < \Gamma_{\max}$, set $n^* = n^* + 1$, $\mathbf{H}_{\text{out}} = \mathbf{H}''$, and go back to step 2); else, return $\mathbf{H}_{\text{out}}$.

We remark that, regarding step 4), in [1] we proposed to shorten the first $\left\lfloor \frac{s}{2} \right\rfloor$ and the last $\left\lceil \frac{s}{2} \right\rceil$ columns of $\mathbf{H}'$. However, this is a sub-optimal choice, especially from the weight distribution standpoint; this was observed in [27], in the unconstrained scenario. So, let us suppose that s positions need to be shortened, as defined in step 2) of the above procedure. Given a set of codewords $\mathcal{D} = \mathbf{c}_0, \ldots, \mathbf{c}_{D-1}$, let us compute

$$\mathbf{b}^{(\mathcal{D})} = \sum_{i=0}^{D-1} \mathbf{c}_i,$$

where the sum is performed over $\mathbb{Z}_{\geq 0}$. Then, we define $I_j^{(\mathcal{D})} = \{i \in [0, k-1] : b_i^{(\mathcal{D})} = j\}$.

Aiming at eliminating as many low-weight codewords as possible, the following shortening procedure might be followed:

1) set $d = d_{\min}$;
2) find as many codewords of weight d as possible (using, for example, the tool in [32]) and separate these codewords into families, as defined in Section II-B;
3) for each family found, cyclically shift the non-zero pattern characterizing it towards right, and check whether the obtained vector $\mathbf{v}$ is a codeword, by computing $\mathbf{v}\mathbf{H}^\top$. Repeat until $\mathbf{v}\mathbf{H}^\top \neq \mathbf{0}$, or $v_{n-1} = 1$;
4) for each group, cyclically shift the non-zero pattern characterizing it towards left, and check whether the obtained vector $\mathbf{v}$ is a codeword, by computing $\mathbf{v}\mathbf{H}^\top$. Repeat until $\mathbf{v}\mathbf{H}^\top \neq \mathbf{0}$, or $v_0 = 1$;
5) form the set $\mathcal{D}$, containing all the codewords found in steps 2), 3) and 4);
6) partition $[0; k-1]$ as $\bigcup_{j=0}^{M} I_j^{(\mathcal{D})}$, where M is the largest index j for which $I_j^{(\mathcal{D})}$ is not empty;
7) shorten s positions by first picking those in $I_M^{(\mathcal{D})}$, then those in $I_{M-1}^{(\mathcal{D})}$, and so on up to $I_1^{(\mathcal{D})}$. If, for some $1 \leq j \leq M$, the residual number of positions to be shortened, say l, is smaller than the number of elements in $I_j^{(\mathcal{D})}$, randomly choose l entries of $I_j^{(\mathcal{D})}$ and shorten those positions. If $s > |\bigcup_{j=0}^{M} I_j^{(\mathcal{D})}|$, after having shortened all the positions in $\bigcup_{j=1}^{M} I_j^{(\mathcal{D})}$, set $d = d_{\min}+1$, $s = s - |\bigcup_{j=0}^{M} I_j^{(\mathcal{D})}|$ and go back to step 2).

We remark that goals different from eliminating low-weight codewords can be pursued. In particular, it might be possible to design the shortening pattern is such a way that the code has a good asymptotic threshold. Instead, to mitigate error floors, another possibility is to find the most harmful objects (trapping sets, absorbing sets, etc.) for the given code and decoder and shorten the code in such a way that these objects are eliminated. This issue is left for future research.

Instead, we prove next that the gap between the actual computational complexity obtained using Procedure A and the threshold $\Gamma_{\max}$ decreases for increasing codeword lengths.

Theorem 4: Let $\Gamma_{\max}$ be a target decoding complexity constraint. Let $\tilde{n}$ denote the largest codeword length of the punctured and shortened simplex code (denoted as $\mathcal{C}(\tilde{n})$) obtained as output of Procedure A, such that the corresponding decoding complexity

$$\tilde{\Gamma} = \tilde{n}\, I_{\text{avg}}\, f\left(\langle \tilde{w_c} \rangle, \frac{\tilde{k}}{\tilde{n}}\right)$$

satisfies

$$\tilde{\Gamma} \leq \Gamma_{\max},$$

where $\tilde{k} = [R\tilde{n}]$ and $\langle \tilde{w_c} \rangle$ is its average column weight. Then, the relative error

$$\epsilon = \frac{\Gamma_{\max} - \tilde{\Gamma}}{\Gamma_{\max}}$$

can be made arbitrarily small as $\tilde{n}$ grows, i.e.,

$$\lim_{\tilde{n}\to\infty} \epsilon = 0.$$

Proof: By construction, it holds that

$$\tilde{\Gamma} \leq \Gamma_{\max} < \Gamma^+,$$

where

$$\Gamma^+ = (\tilde{n}+1)\, I_{\text{avg}}\, f\left(\langle w_c^+ \rangle, \frac{k^+}{\tilde{n}+1}\right)$$

is the decoding complexity associated with the punctured and shortened simplex code, obtained by running steps 2)-4) of Procedure A, feeding codeword length $\tilde{n}+1$; k^+ and $\langle w_c^+ \rangle$ represent the dimension and the parity-check matrix average column weight of this code, respectively. The decoding complexity difference $\Gamma^+ - \tilde{\Gamma}$ can be expressed, after straightforward computations following from (25) and (26), as

$$\Gamma^+ - \tilde{\Gamma} = \tag{27}$$

$$I_{\text{avg}}\Big[65\,\tilde{n}\big(\langle w_c^+ \rangle - \langle \tilde{w_c} \rangle\big) + 65\,\langle w_c^+ \rangle + 96\,(k^+ - \tilde{k}) - 88\Big]. \tag{28}$$

Upon increasing the codeword length from $\tilde{n}$ to $\tilde{n}+1$, two cases can arise:

- If $[R(\tilde{n}+1)] = [R\tilde{n}]$, then the number of shortened symbols remains unchanged, whereas the number of punctured symbols decreases by one unit. In this case, with respect to the parity-check matrix of $\mathcal{C}(\tilde{n})$, the parity-check matrix contains an additional row and an additional column. This row carries exactly w_h ones; therefore, the total number of ones increases by exactly w_h. As a result, the new total number of ones is

  $$\tilde{n}\langle \tilde{w_c} \rangle + w_h,$$

  and the new number of columns is $\tilde{n}+1$. Thus, the new average column weight is

  $$\langle w_c^+ \rangle = \frac{\tilde{n}\langle \tilde{w_c} \rangle + w_h}{\tilde{n}+1},$$

  and the corresponding variation is

  $$\langle w_c^+ \rangle - \langle \tilde{w_c} \rangle = \frac{w_h - \langle \tilde{w_c} \rangle}{\tilde{n}+1}.$$

  Thus, the variation scales as $\mathcal{O}(1/\tilde{n})$ and becomes negligible for large $\tilde{n}$.
- If $[R(\tilde{n}+1)] = [R\tilde{n}] + 1$, then the number of shortened symbols decreases by one, while the number of punctured symbols remains unchanged. In this case, with respect to the parity-check matrix of $\mathcal{C}(\tilde{n})$, an additional column must be included. This column typically has a small number of ones, denoted by d_s, since shortening is applied to columns located within the low-density band of the matrix (i.e., among the first k codeword symbols). Since the total number of ones increases by d_s, and the number of columns increases by one, the new average column weight is

  $$\langle w_c^+ \rangle = \frac{\tilde{n}\langle \tilde{w_c} \rangle + d_s}{\tilde{n}+1},$$

  and the corresponding variation is

  $$\langle w_c^+ \rangle - \langle \tilde{w_c} \rangle = \frac{d_s - \langle \tilde{w_c} \rangle}{\tilde{n}+1}.$$

TABLE II

PARAMETERS OF THE CONSIDERED PRC-LDPC CODES WITH $k = 553$, FOR $I_{\text{avg}} = 100$, IN AN UNCONSTRAINED SETTING

Code	n	$\langle w_c \rangle$	Supp($\mathbf{h}$)	$d_{\min}$	Γ
$\mathcal{C}_{\frac{3}{4}}$	737	3.75	$[0, 3, 66, 97, 142, 220, 221, 295, 330, 354, 382, 402, 486, 546, 553]$	9	$1.65 \cdot 10^7$
$\mathcal{C}_{\frac{4}{5}}$	691	3.8	$[0, 3, 41, 95, 97, 152, 220, 221, 242, 295, 330, 338, 382, 415, 486, 504, 523, 546, 553]$	7	$1.60 \cdot 10^7$
$\mathcal{C}_{\frac{5}{6}}$	663	3.5	$[0, 3, 15, 41, 97, 106, 142, 152, 220, 242, 295, 338, 382, 388, 402, 415, 486, 504, 523, 546, 553]$	5	$1.43 \cdot 10^7$

TABLE III

PARAMETERS OF THE CONSIDERED CODES, FOR $I_{\text{avg}} = 100$, AND $\Gamma_{\max} = 1.5 \cdot 10^7$

Code	R	n	$\langle w_c \rangle$	ρ	s	$d_{\min}$	Γ
$\mathcal{C}'_{\frac{3}{4}}$	$\frac{3}{4}$	680	3.64	14	43	8	$1.4995 \cdot 10^7$
$\mathcal{C}'_{\frac{4}{5}}$	$\frac{261}{326}$	652	3.71	8	31	8	$1.4989 \cdot 10^7$

Again, since d_s and $\langle \tilde{w_c} \rangle$ are constants, the variation scales as $\mathcal{O}(1/\tilde{n})$ and becomes negligible for large $\tilde{n}$.

In both cases, the difference $\langle w_c^+ \rangle - \langle \tilde{w_c} \rangle$ behaves like $\mathcal{O}(1/\tilde{n})$. Moreover, the change in the dimension, $k^+ - \tilde{k}$, is either 0 or 1 depending on whether the rounding function increases, and thus the term involving $k^+ - \tilde{k}$ remains bounded independent of $\tilde{n}$. Therefore, the overall variations induced by increasing the codeword length by one unit are bounded and asymptotically negligible. Thus, $\Gamma^+ - \tilde{\Gamma}$ remains finite as $\tilde{n}$ grows, and

$$\limsup_{\tilde{n} \to \infty} (\Gamma^+ - \tilde{\Gamma}) < +\infty.$$

Since $\Gamma_{\max}$ lies between $\tilde{\Gamma}$ and Γ^+, it follows that

$$0 \leq \Gamma_{\max} - \tilde{\Gamma} < \Gamma^+ - \tilde{\Gamma}.$$

Hence, the relative error satisfies

$$0 \leq \epsilon < \frac{\Gamma^+ - \tilde{\Gamma}}{\Gamma_{\max}}.$$

Finally, noting that $\Gamma_{\max}$ grows linearly with $\tilde{n}$ (see (25)) while $\Gamma^+ - \tilde{\Gamma}$ remains bounded, we conclude

$$\lim_{\tilde{n} \to \infty} \epsilon = 0,$$

which proves the claim. ∎

VII. ERROR RATE PERFORMANCE OF HIGH-RATE CODES UNDER BELIEF PROPAGATION DECODING

Let us first consider an unconstrained setting, for which we have designed some PRC-LDPC codes with $k = 553$, by following the design method in [16]. Their parameters are shown in Table II. Notice that all the support vectors are Golomb rulers.

Let us now suppose that the application context enforces $\Gamma_{\max} = 1.5 \cdot 10^7$. This value of $\Gamma_{\max}$ has been chosen because it approaches the average complexity required to decode a codeword for the 5G codes we will compare in the following. The same approach can be obviously applied to other choices of $\Gamma_{\max}$, though they might require a heavy employment of puncturing and shortening operations. We notice that all the PRC-LDPC codes considered in Table II, except for $\mathcal{C}_{\frac{5}{6}}$, do not comply with the considered constraint. We thus apply the puncturing and shortening procedure described in Section VI, and the parameters of the obtained complexity-constrained PRC-LDPC codes are shown in Table III.

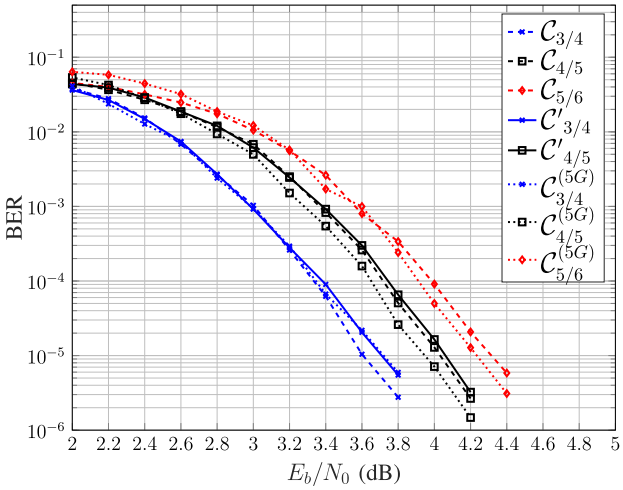


Fig. 14. BER vs SNR per information bit, for codes with different rates, in both an unconstrained and a complexity-constrained setting.

TABLE IV

PARAMETERS OF THE CONSIDERED 5G CODES

Code	k	n	P	$\langle w_c \rangle$	Γ
$\mathcal{C}^{(5G)}_{\frac{3}{4}}$	510	728	48	3.63	$1.57 \cdot 10^7$
$\mathcal{C}^{(5G)}_{\frac{4}{5}}$	522	701	48	3.36	$1.43 \cdot 10^7$
$\mathcal{C}^{(5G)}_{\frac{5}{6}}$	553	716	52	3.14	$1.38 \cdot 10^7$

We note that the difference between the original code rate and the selected one is null or negligible; also, in all the considered cases, the average column weight slightly decreases, due to the combination of puncturing and shortening on $\mathbf{H}$. Moreover, we remark that a significant portion of symbols were punctured and shortened (about 7.7% for $C_{\frac{3}{4}}$, and 5.6% for $C_{\frac{4}{5}}$); finally, notice that it is possible to obtain actual values of the complexity which are quite close to the bound, even in finite-length scenarios.

Performance of the codes in Tables II and III in terms of BER has been assessed through Monte Carlo simulations of BPSK transmissions over the AWGN channel. As mentioned in Section VI, we have employed the SPA decoder described in [35], running at most 100 iterations. The simulation results are shown in Fig. 14. In the unconstrained scenario, the BER follows the obvious trend, according to which the codes with smallest rate have the best performance. It is remarkable that

TABLE V

VALUES OF $d_{\min}$, $d_{\max}$, Δ AND δ FOR DIFFERENT PARITY-CHECK POLYNOMIALS WITH $k \in \{7, 8\}$

$h(x)$	$d_{\min}(n')$	$d_{\max}(n')$	$\delta(n')$	$d_{\min}(n'')$	$d_{\max}(n'')$	$\delta(n'')$	$\Delta(n') = \Delta(n'')$
[0, 1, 7]	27	37	0.5	27	37	-0.5	5
[0, 3, 7]	27	37	0.5	27	37	-0.5	5
[0, 1, 2, 3, 7]	27	37	0.5	27	37	-0.5	5
[0, 2, 3, 4, 7]	27	36	0	28	37	0	4.5
[0, 1, 2, 3, 4, 5, 7]	27	37	0.5	27	37	-0.5	5
[0, 1, 3, 6, 7]	27	37	0.5	27	37	-0.5	5
[0, 2, 4, 6, 7]	27	37	0.5	27	37	-0.5	5
[0, 2, 5, 6, 7]	26	38	0	26	38	-0.5	6
[0, 1, 2, 4, 5, 6, 7]	26	38	0.5	26	38	-0.5	6
[0, 2, 3, 4, 8]	56	71	0	57	72	0	7.5
[0, 1, 3, 5, 8]	56	72	0.5	56	72	-0.5	8
[0, 1, 2, 3, 4, 6, 8]	56	71	0	57	72	0	7.5
[0, 1, 5, 6, 8]	56	71	0	57	72	0	7.5
[0, 2, 5, 6, 8]	56	71	0	57	72	0	7.5
[0, 3, 5, 6, 8]	54	73	0	55	74	0	9.5
[0, 1, 6, 7, 8]	56	72	0.5	56	72	-0.5	8
[0, 1, 2, 5, 6, 7, 8]	56	72	0.5	56	72	-0.5	8

TABLE VI

VALUES OF $d_{\min}$, $d_{\max}$, Δ AND δ FOR DIFFERENT PARITY-CHECK POLYNOMIALS WITH $k \in \{9, 10\}$

$h(x)$	$d_{\min}(n')$	$d_{\max}(n')$	$\delta(n')$	$d_{\min}(n'')$	$d_{\max}(n'')$	$\delta(n'')$	$\Delta(n') = \Delta(n'')$
[0, 4, 9]	116	140	0.5	116	140	-0.5	12
[0, 2, 3, 5, 9]	118	138	0.5	118	138	-0.5	10
[0, 3, 4, 6, 9]	117	139	0.5	117	139	-0.5	11
[0, 1, 2, 3, 5, 6, 9]	117	139	0.5	117	139	-0.5	11
[0, 1, 2, 4, 5, 6, 9]	117	139	0.5	117	139	-0.5	11
[0, 1, 3, 4, 6, 7, 9]	115	141	0.5	115	141	-0.5	13
[0, 1, 4, 8, 9]	116	139	0	117	140	0	11.5
[0, 4, 5, 8, 9]	118	138	0.5	118	138	-0.5	10
[0, 5, 6, 8, 9]	118	138	0.5	118	138	-0.5	10
[0, 1, 3, 5, 6, 8, 9]	117	139	0.5	117	139	-0.5	11
[0, 2, 7, 8, 9]	117	138	0	118	139	0	10.5
[0, 1, 2, 3, 7, 8, 9]	114	141	0	115	142	0	13.5
[0, 1, 5, 6, 7, 8, 9]	117	139	0.5	117	139	-0.5	11
[0, 3, 5, 6, 7, 8, 9]	117	139	0.5	117	139	-0.5	11
[0, 3, 10]	239	272	0	240	273	0	16.5
[0, 1, 3, 4, 10]	240	271	0	241	272	0	15.5
[0, 1, 2, 3, 5, 6, 10]	242	269	0	243	270	0	13.5
[0, 2, 3, 8, 10]	240	271	0	241	272	0	15.5
[0, 3, 4, 8, 10]	240	272	0.5	240	272	-0.5	16
[0, 1, 5, 8, 10]	241	270	0	242	271	0	14.5
[0, 4, 5, 8, 10]	240	271	0	241	272	0	15.5
[0, 1, 3, 4, 5, 6, 7, 8, 10]	239	272	0	240	273	0	16.5
[0, 1, 4, 9, 10]	241	271	0.5	241	271	-0.5	15
[0, 1, 2, 3, 4, 5, 6, 9, 10]	242	270	0.5	242	270	-0.5	14
[0, 2, 3, 6, 8, 9, 10]	239	272	0	240	273	0	16.5
[0, 1, 5, 6, 8, 9, 10]	240	272	0.5	240	272	-0.5	16
[0, 3, 4, 5, 6, 7, 8, 9, 10]	238	274	0.5	238	274	-0.5	18

overall, as also expected, puncturing and shortening weaken the code, but the loss in terms of $\frac{E_b}{N_0}$ is almost negligible. We also observe that, beyond decreasing the decoding complexity, reducing the codeword length also lowers the encoding complexity and reduces the decoding latency (defined as the number of symbols that must be received before the decoding process can start). We have also selected some 5G LDPC codes [28], reported in Table IV, and compared their performance to that of the considered PRC-LDPC codes. For 5G codes, we have $R = \frac{k}{n-P}$, where P is the number of punctured information symbols. In contrast to our approach, the decoder for 5G codes attempts to recover even the symbols that are not transmitted, since they are information symbols. In our case, given the lower triangular form of the parity-check matrix of PRC-LDPC codes, we always puncture control symbols and thus we can avoid reconstructing them.[3] In Fig. 14, we observe that the performance loss of PRC-LDPC codes with

[3]This distinction motivates the use of the symbol P to represent the number of punctured symbols, instead of ρ, as previously done.

TABLE VII

VALUES OF $d_{\min}$, $d_{\max}$, Δ AND δ FOR DIFFERENT PARITY-CHECK POLYNOMIALS WITH $k \in \{11, 12\}$

$h(x)$	$d_{\min}(n')$	$d_{\max}(n')$	$\delta(n')$	$d_{\min}(n'')$	$d_{\max}(n'')$	$\delta(n'')$	$\Delta(n') = \Delta(n'')$
[0, 2, 11]	490	534	0.5	490	534	-0.5	22
[0, 1, 3, 5, 11]	487	537	0.5	487	537	-0.5	25
[0, 2, 3, 5, 11]	489	535	0.5	489	535	-0.5	23
[0, 1, 5, 6, 11]	491	532	0	492	533	0	20.5
[0, 2, 3, 7, 11]	492	532	0.5	492	532	-0.5	20
[0, 2, 5, 8, 11]	489	534	0	490	535	0	22.5
[0, 1, 4, 5, 6, 8, 11]	484	539	0	485	540	0	27.5
[0, 1, 2, 3, 4, 5, 6, 8, 11]	489	534	0	490	535	0	22.5
[0, 1, 4, 9, 11]	490	534	0.5	490	534	-0.5	22
[0, 1, 4, 7, 8, 9, 11]	489	535	0.5	489	535	-0.5	23
[0, 2, 3, 10, 11]	491	533	0.5	491	533	-0.5	21
[0, 1, 3, 4, 7, 10, 11]	491	533	0.5	491	533	-0.5	21
[0, 1, 3, 4, 5, 7, 8, 10, 11]	487	537	0.5	487	537	-0.5	25
[0, 1, 4, 6, 12]	992	1055	0	993	1056	0	31.5
[0, 2, 3, 9, 12]	994	1054	0.5	994	1054	-0.5	30
[0, 1, 2, 3, 8, 9, 12]	995	1052	0	996	1053	0	28.5
[0, 2, 6, 8, 9, 10, 12]	997	1050	0	998	1051	0	26.5
[0, 2, 4, 5, 6, 8, 9, 10, 12]	993	1055	0.5	993	1055	-0.5	31
[0, 1, 2, 4, 6, 11, 12]	990	1058	0.5	990	1058	-0.5	34
[0, 1, 3, 5, 9, 11, 12]	989	1059	0.5	989	1059	-0.5	35
[0, 4, 6, 7, 9, 11, 12]	987	1060	0	988	1061	0	36.5
[0, 5, 6, 7, 9, 11, 12]	985	1062	0	986	1063	0	38.5
[0, 4, 7, 8, 9, 11, 12]	990	1057	0	991	1058	0	33.5
[0, 1, 2, 5, 7, 8, 9, 11, 12]	986	1061	0	987	1062	0	37.5
[0, 1, 2, 5, 10, 11, 12]	988	1059	0	989	1060	0	35.5
[0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12]	987	1060	0	988	1061	0	36.5

TABLE VIII

VALUES OF $d_{\min}$, $d_{\max}$, Δ AND δ FOR DIFFERENT PARITY-CHECK POLYNOMIALS WITH $k \in \{13, 14\}$

$h(x)$	$d_{\min}(n')$	$d_{\max}(n')$	$\delta(n')$	$d_{\min}(n'')$	$d_{\max}(n'')$	$\delta(n'')$	$\Delta(n') = \Delta(n'')$
[0, 1, 3, 4, 13]	1994	2102	0.5	1994	2102	-0.5	54
[0, 1, 2, 3, 4, 5, 7, 9, 13]	2009	2087	0.5	2009	2087	-0.5	39
[0, 1, 5, 7, 8, 9, 13]	2006	2090	0.5	2006	2090	-0.5	42
[0, 4, 5, 7, 9, 10, 13]	2002	2094	0.5	2002	2094	0	46
[0, 1, 2, 3, 6, 8, 9, 10, 13]	2008	2087	0	2009	2088	-0.5	39.5
[0, 1, 4, 7, 8, 11, 13]	2002	2094	0.5	2002	2094	-0.5	46
[0, 2, 4, 8, 9, 12, 13]	2005	2090	0	2006	2091	-0.5	42.5
[0, 2, 3, 4, 6, 8, 10, 12, 13]	2005	2091	0.5	2005	2091	0	43
[0, 1, 2, 5, 11, 12, 13]	1996	2099	0	1997	2100	-0.5	51.5
[0, 1, 4, 6, 7, 8, 11, 12, 13]	2010	2086	0.5	2010	2086	0	38
[0, 1, 6, 8, 14]	4026	4165	0	4027	4166	0	69.5
[0, 1, 6, 10, 14]	4030	4161	0	4031	4162	0	65.5
[0, 1, 3, 4, 6, 7, 9, 10, 14]	4028	4164	0.5	4028	4164	-0.5	68
[0, 1, 6, 11, 14]	4012	4180	0.5	4012	4180	-0.5	84
[0, 2, 5, 6, 9, 11, 14]	4018	4174	0.5	4018	4174	-0.5	78
[0, 1, 3, 5, 6, 7, 8, 9, 11, 12, 14]	4032	4160	0.5	4032	4160	-0.5	64
[0, 3, 4, 7, 9, 10, 11, 12, 14]	4036	4155	0	4037	4156	0	59.5
[0, 1, 3, 5, 6, 13, 14]	4023	4169	0.5	4023	4169	-0.5	73
[0, 1, 2, 3, 4, 5, 7, 8, 10, 13, 14]	4040	4151	0	4041	4152	0	55.5
[0, 1, 2, 4, 5, 6, 11, 13, 14]	4039	4152	0	4040	4153	0	56.5
[0, 1, 2, 3, 5, 8, 11, 13, 14]	4029	4162	0	4030	4163	0	66.5
[0, 1, 6, 7, 10, 11, 12, 13, 14]	4031	4161	0.5	4031	4161	-0.5	65
[0, 5, 6, 9, 10, 11, 12, 13, 14]	4018	4173	0	4019	4174	0	77.5

respect to 5G codes is very small. This slight loss is mainly due to the fact that $\Gamma_{\max} = 1.5 \cdot 10^7$ has been chosen as the actual decoding complexity of these 5G codes. Larger values of $\Gamma_{\max}$ would advantage PRC-LDPC codes, since they would be less weakened by the additional puncturing and shortening operations, whereas smaller values of $\Gamma_{\max}$ would make the comparison unfair, because it would be necessary to modify in some way also 5G codes, in such a way that they also comply with the requirement. In other words, in our comparison, 5G codes have some intrinsic advantage.

VIII. CONCLUSION AND FUTURE WORKS

In this paper we have conducted a thorough analysis of the minimum distance of punctured binary simplex codes. We have studied performance metrics in both the asymptotic and the finite-length regime.

Simulation results show that the performance of the special class of punctured simplex codes, named PRC-LDPC codes, is comparable to that of state-of-the-art LDPC codes, like those used in the 5G standard. The rates considered in the design examples are rather large. As a spark for future works, we propose the design of low-rate PRC-LDPC codes with good asymptotic thresholds, obtained in their turn by a careful choice of puncturing (as done, for example, in [18]) and shortening patterns. Future investigations will also focus on increasing the density of the primitive parity-check polynomial and exploring hybrid decoding strategies that combine iterative decoding with ordered statistic decoding.

We also foresee that it is possible to exploit the theoretical analysis developed in this paper to design sequences with good partial-period correlation properties, which are particularly interesting in situations where short correlation windows are necessary due to strict requirements on synchronization time or scarce hardware availability. Leveraging the error correction model, it should be possible to select sequences that exhibit good out-of-peak partial-period autocorrelation, even when the size of the correlation window changes. Furthermore, our analysis examined both the minimum and maximum distances of the codes under consideration, which could be highly relevant for applications utilizing PN sequences and non-coherent receivers.

APPENDIX A
ANALYSIS OF DISTANCE PROPERTIES FOR $n \in \{n', n''\}$

With reference to the notation introduced throughout the paper, in this appendix we show in Tables 5–8 the exact values of $d_{\min}$, $d_{\max}$, Δ and δ, when the codeword length $n \in \{n', n''\}$, for many primitive polynomials characterized by degree between 7 and 14. In particular, coherent with the notation introduced in Section II, the polynomial $h(x)$ is expressed in terms of its support.

REFERENCES

[1] M. Battaglioni, M. Amagliani, M. Baldi, F. Chiaraluce, and G. Cancellieri, "Design and analysis of a family of complexity-constrained LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2024, pp. 428–433.

[2] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[3] M. Aktas, G. Joshi, S. Kadhe, F. Kazemi, and E. Soljanin, "Service rate region: A new aspect of coded distributed system design," *IEEE Trans. Inf. Theory*, vol. 67, no. 12, pp. 7940–7963, Dec. 2021.

[4] L. Ping, W. K. Leung, and K. Y. Wu, "Low rate turbo-Hadamard codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2001, p. 211.

[5] G. Yue, L. Ping, and X. Wang, "Generalized low-density parity-check codes based on Hadamard constraints," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1058–1079, Mar. 2007.

[6] S. Yeom, Y. Jung, and S. Lee, "An adaptive threshold technique for fast PN code acquisition in DS-SS systems," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2870–2875, Jul. 2011.

[7] Consultative Committee for Space Data Systems. (2014). *Data Transmission and PN Ranging for 2 GHz CDMA Link Via Data Relay Satellite*. [Online]. Available: https://public.ccsds.org/Pubs/415x1b1.pdf

[8] T. Li, Y. Shangguan, and G. Shao, "A new method for PN code synchronization in the direct sequence spread spectrum communication systems," in *Proc. 11th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2016, pp. 92–96.

[9] Consultative Committee for Space Data Systems. (2022). *Pseudo-Noise (PN) Ranging Systems*. [Online]. Available: https://public.ccsds.org/Pubs/414x1b3.pdf

[10] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[11] M. R. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 159–161, Mar. 2004.

[12] T. Tian and C. R. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 5, pp. 789–795, Dec. 2005.

[13] F. Babich, M. Noschese, and F. Vatta, "Analysis and design of rate compatible LDPC codes," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.

[14] M. Battaglioni and G. Cancellieri, "Punctured binary simplex codes as LDPC codes," in *Proc. 61st FITCE Int. Congr. Future Telecommunications: Infrastructure Sustainability (FITCE)*, Sep. 2022, pp. 1–6.

[15] M. Battaglioni, M. Baldi, F. Chiaraluce, and G. Cancellieri, "Rate-adaptive LDPC codes obtained from simplex codes," in *Proc. ICC-IEEE Int. Conf. Commun.*, May 2023, pp. 142–147.

[16] M. Battaglioni, M. Baldi, F. Chiaraluce, and G. Cancellieri, "Rate-compatible LDPC codes based on primitive polynomials and Golomb rulers," *IEEE Trans. Commun.*, vol. 72, no. 12, pp. 7361–7373, Dec. 2024.

[17] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[18] S. Jiang, F. C. M. Lau, W. M. Tam, and C.-W. Sham, "Design and error performance of punctured Hadamard codes," in *Proc. 23rd Asia–Pacific Conf. Commun. (APCC)*, Dec. 2017, pp. 1–5.

[19] M. Shirvanimoghaddam, "Primitive rateless codes," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6395–6408, Oct. 2021.

[20] M. Baldi, M. Bianchi, F. Chiaraluce, and T. Klove, "A class of punctured simplex codes which are proper for error detection," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3861–3880, Jun. 2012.

[21] D. G. M. Mitchell, M. Lentmaier, A. E. Pusane, and D. J. Costello, "Randomly punctured LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 408–421, Feb. 2016.

[22] S. Wainberg and J. Wolf, "Subsequences of pseudorandom sequences," *IEEE Trans. Commun.*, vol. COM-18, no. 5, pp. 606–612, Oct. 1970.

[23] R. Lidl and H. Niederreiter, *Finite Fields* (Encyclopedia of Mathematics and Its Applications), vol. 20. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[24] M. Shirvanimoghaddam, "On the Hamming weight distribution of subsequences of pseudorandom sequences," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 1671–1675.

[25] L. Wang, S. Hu, and O. Shayevitz, "Quickest sequence phase detection," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5834–5849, Sep. 2017.

[26] B. Shuval and I. Tal, "Strong polarization for shortened and punctured polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2024, pp. 2198–2203.

[27] M. Battaglioni, G. Greganti, and G. Cancellieri, "Optimized rate-adaptive error correction through puncturing and shortening of simplex codes," in *Proc. AEIT Int. Annu. Conf. (AEIT)*, Sep. 2024, pp. 1–6.

[28] *Technical Specification Group Radio Access Network; NR; Multiplexing and Channel Coding (Release 16)*, document TS 38.212, 3GPP, Jun. 2020.

[29] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2004.

[30] S. Fredricsson, "Pseudo-randomness properties of binary shift register sequences," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 1, pp. 115–120, Jan. 1975.

[31] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[32] D. J. C. MacKay. (2008). *Source Code for Approximating the Mindist Problem of LDPC Codes*. [Online]. Available: http://www.inference.eng.cam.ac.uk/mackay/MINDISTECC.html

[33] J. Stern, "A method for finding codewords of small weight," in *Proc. Int. Colloq. Coding Theory Appl.*, 1989, no. 388, pp. 106–113.

[34] Y. Wu and C. N. Hadjicostis, "Soft-decision decoding using ordered recodings on the most reliable basis," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 829–836, Feb. 2007.

[35] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommun. Conf.*, vol. 2, Nov. 2001, pp. 1036–1036E.

[36] M. H. Tadayon, A. Tasdighi, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Efficient search of compact QC-LDPC and SC-LDPC convolutional codes with large girth," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1156–1159, Jun. 2018.

[37] M. Battaglioni and G. Cancellieri, "A family of error correcting codes for automotive applications," in *Proc. AEIT Int. Conf. Electr. Electron. Technol. Automot. (AEIT AUTOMOTIVE)*, Jul. 2023, pp. 1–6.

Massimo Battaglioni (Member, IEEE) received the Laurea and Laurea Magistrale (summa cum laude) degrees in electronic engineering and the Ph.D. degree in information engineering from Università Politecnica delle Marche (UNIVPM), Ancona, Italy, in 2013, 2015, and 2019, respectively. In 2017, he was a Visiting Student with the Electrical and Information Technology Department, LTH, Lund University, Sweden. In 2018, he was a Visiting Student with the Klipsch School of Electrical and Computer Engineering, New Mexico State University (NMSU), Las Cruces, NM, USA; and the School of Electrical and Electronic Engineering, University College Dublin, Ireland. Since 2024, he has been a Tenure-Track Researcher with the Department of Information Engineering, UNIVPM. He has co-authored more than 40 scientific articles. His research interests include coding techniques for communications reliability. He serves as an Editor for IEEE COMMUNICATIONS LETTERS.

Marco Baldi (Senior Member, IEEE) received the Laurea degree (Hons.) in electronics engineering and the Ph.D. degree in electronics, computer, and telecommunications engineering from Università Politecnica delle Marche (UNIVPM), Ancona, Italy. Since 2019, he has been an Associate Professor with the Department of Information Engineering, UNIVPM, where he also coordinates the local node of the CINI Cybersecurity National Laboratory and takes part in the Research and Service Center for Privacy and Cybersecurity (CRiSPY). He has co-authored more than 200 scientific articles, one book, and four patents. His research interests include coding and cryptography for information reliability and security. He serves as an Area Editor in coding for IEEE COMMUNICATIONS LETTERS, a Senior Area Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS.

Franco Chiaraluce (Senior Member, IEEE) received the Laurea degree (summa cum laude) in electronic engineering from the University of Ancona, in 1985. Since 1987, he has been with the Department of Electronics and Automatics, University of Ancona. He is currently a Full Professor of telecommunications with Università Politecnica delle Marche (UNIVPM), Ancona, Italy, where he is also the Director of the Department of Information Engineering (DII). He has co-authored more than 350 scientific articles and three books and holds three patents. On his research topics, he collaborates with national and international companies. His research interests include various aspects of communication systems theory and design, with a special emphasis on error-correcting codes, cryptography, and physical layer security.

Giovanni Cancellieri received the degree in electronic engineering and the degree in physics from the University of Bologna. From 1986 to 2022, he was a Full Professor of telecommunications with Università Politecnica delle Marche, and he is now retired. His main research activities have been focused on optical fibres, radio communications, and wireless systems, with special emphasis on channel coding and modulation systems. He is the co-author of about 150 articles, five books of scientific content, and two international patents.