



Received July 31, 2018, accepted August 26, 2018, date of publication September 6, 2018, date of current version September 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868963

Algebra-Assisted Construction of Quasi-Cyclic LDPC Codes for 5G New Radio

HUAAN LI¹, BAOMING BAI¹ , (Member, IEEE), XIJIN MU², JI ZHANG¹, AND HENGZHOU XU¹ ³

¹State Key Laboratory of ISN, Xidian University, Xi'an 710071, China

²Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

³School of Network Engineering, Zhoukou Normal University, Henan 466001, China

Corresponding author: Baoming Bai (bmbai@mail.xidian.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 61771364, 61701368, and 61801527, in part by the Innovation Fund of Xidian University under Grant JB182001, in part by the Key Scientific Research Projects of the Henan Educational Committee under Grant 19A510028, and in part by the China Academy of Telecommunications Technology.

ABSTRACT Quasi-cyclic LDPC (QC-LDPC) codes have been accepted as the standard codes of 5G enhanced mobile broadband data channel. These standard codes are designed to support multiple lifting sizes and possess rate-compatible property, which can help adapt various information lengths and code rates well. In this paper, we propose an *algebra-assisted* method for constructing QC-LDPC codes with such properties. We will first review the encoding mechanism and requirements of 5G LDPC codes, and present *cycle analysis* for such emerging codes. We then propose a metric, referred to as *weighted average number of cycles* (WANC), from the perspective of cycle structure for constructing the QC-LDPC codes that can support *multiple lifting sizes*. Based on the WANC metric and algebraic methods, we develop a simple and practical algorithm to construct this kind of QC-LDPC codes. We finally apply the proposed algorithm to construct the exponent matrices for cases of 5G LDPC codes and the standard LDPC codes of consultative committee for space data systems, respectively. Simulation results show that the proposed WANC metric and designed algorithm are feasible and effective, and thus can be utilized to design other similar QC-LDPC codes.

INDEX TERMS Quasi-cyclic LDPC code, 5G channel codes, algebra-assisted construction, multiple lifting size, cycle analysis.

I. INTRODUCTION

Low-density parity-check (LDPC) codes, proposed by Robert G. Gallager in the early 1960s [1] and rediscovered in the late 1990s, are shown to form a class of capacity-approaching channel codes [2] and perform amazingly well with iterative decoding based on sum-product algorithm (SPA) or minimum-sum algorithm (MSA) [3]. Since their rediscovery, research effort at such codes mainly focuses on construction, encoding/decoding algorithms with low complexity, performance analysis, and applications. Especially, many of these codes have been successively used in various communication systems, including digital video broadcast [4], 10G BASE-T Ethernet [3] (Chapter 11), and NASA near-earth and deep-space missions [5].

As a class of linear block codes, LDPC codes are completely specified by their sparse parity-check matrices, and the construction and structural properties of the parity-check matrices play an important role in the error performance, as well as the structure and cost of encoder/decoder in terms of computation and hardware.

Based on the construction methods, LDPC codes can be roughly classified into two categories: random-like and structured codes. The best known method to construct random-like LDPC codes is the progressive edge growth (PEG) algorithm [6]. Structured LDPC codes are generally constructed based on the protograph or algebraic methods [7]–[10]. The protograph construction is to copy the base graph (i.e., protograph) \mathcal{G} Z times, and then connect these copies by permuting the edges of the individual copies. Here the parameter Z is called the *expansion factor* or *lifting size*. Algebraic methods include the superposition (SP) construction as a representative based on finite fields, finite geometries, and combinatorial designs. The SP construction of LDPC codes is to disperse every element of the base matrix into a sparse matrix \mathbf{Q} or zero matrix of the same size. It has been shown in [11] that the protograph construction of LDPC codes is a special case of the SP construction when \mathbf{Q} 's are $Z \times Z$ matrices. As a special kind of structured codes, quasi-cyclic LDPC (QC-LDPC) codes [12] are generally specified by an array of circulants. Commonly used circulant for

QC-LDPC codes is the circulant permutation matrix (CPM). Due to the fact that they possess various advantages such as easy hardware implementation of encoder and decoder, fast decoding convergence, and lower error-floor, QC-LDPC codes have been extensively investigated and found wide applications in different digital communication and storage systems [3], [7]–[20].

Most recently, QC-LDPC codes have been recommended by 3GPP as the channel coding scheme for the enhanced mobile broadband (eMBB) data channel of 5G communication [21]. Besides QC structure, these standard codes simultaneously possess rate-compatible property [22], and can support *multiple lifting sizes*. These properties make such codes easily adapt various information lengths and rate matching. In this paper, we will discuss the construction of QC-LDPC codes with such properties, with emphasis on the study of 5G LDPC codes and their properties. Certainly, the authors of this paper have being participated in the design of 5G LDPC codes, and one *co-designed* exponent matrix has been accepted by 3GPP for the standard 5G LDPC codes.

In this paper, we first systematically summarize the standard 5G LDPC codes and introduce their encoding/decoding process, which involves *puncturing* and *shortening* techniques [23]. These will be useful for the applications of such codes in the future. Motivated by the fantastic property of 5G LDPC codes, we propose a new metric, referred to as *weighted average number of cycles* (WANC), from the perspective of cycle structure for constructing the QC-LDPC codes whose exponent matrices can support *multiple lifting sizes*. Based on the WANC metric and algebraic methods, we develop a simple and practical algorithm for constructing this kind of QC-LDPC codes, and summarize our method as algebra-assisted construction method. Ingredients of our proposed method contain a *circulant coefficient table*, which can be easily determined by the *algebraic methods*. This indicates that our method is different from greedy search, and hence has lower complexity. By using this method, our designed exponent matrices for the smaller base matrix have been accepted by 3GPP for 5G LDPC codes, while most other candidate exponent matrices are based on the combination of greedy search and large-scale simulation. As examples in this paper, we apply the proposed WANC metric and relevant algorithm to design the exponent matrices for cases of 5G LDPC codes (same lifting size set but another base matrix) and the standard LDPC codes of consultative committee for space data systems (CCSDS). Numerical results determine the feasibility and effectiveness of our method. In addition, we provide *cycle analysis* for 5G LDPC codes to simplify the construction of their exponent matrices. This will help understand these emerging codes from the perspective of cycle structure, and guide the design of other QC-LDPC codes, base/exponent matrices of which have similar structure.

The rest of this paper is organized as follows. Section II presents the definitions, basic concepts and some earlier results on QC-LDPC codes which will be used in

later sections. Section III reviews the defined standard LDPC codes of 5G eMBB data channel. Section IV presents our proposed metric, relevant algorithms, and their extension for constructing the exponent matrices of 5G LDPC codes, as well as cycle analysis for such codes to simplify their design. Section V gives numerical results. Finally, Section VI concludes this paper.

II. PRELIMINARY

In this section, we first review a general matrix-theoretic method (which has been adopted for 5G LDPC codes) for constructing QC-LDPC codes based on arbitrary integer sets. We also present some structural properties of LDPC codes for analysis later.

A. MATRIX DISPERSION

Let Z be a positive integer. Consider the set of integers $\mathbb{Z}_Z = \{0, 1, 2, \dots, Z - 1\}$. For each element $p \in \mathbb{Z}_Z$, we represent it by a binary circulant permutation matrix (CPM) of size $Z \times Z$ (with both rows and columns labeled from 0 to $Z - 1$) whose top row has a single nonzero component at position p . We denote this binary CPM by $\mathbf{Q}(p)$, and all the nonzero entries in $\mathbf{Q}(p)$ are “1”. It is not hard to see that the representation of $p \in \mathbb{Z}_Z$ by $\mathbf{Q}(p)$ is *unique*. That is to say, the mapping between $p \in \mathbb{Z}_Z$ and $\mathbf{Q}(p)$ is *one-to-one*. And $\mathbf{Q}(p)$ becomes more sparse with the increase of Z . For convenience, we introduce $\mathbf{Q}(-1)$ and define it as the zero matrix (ZM) of the same size. We refer to such matrix representation for each element over \mathbb{Z}_Z as *matrix dispersion* or *Z-fold CPM-dispersion* [11], and parameter Z as *lifting size* [24]. Take \mathbb{Z}_3 as an example,

$$\mathbf{Q}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q}(1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$\mathbf{Q}(2) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{Q}(-1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In the following discussion, by CPM-dispersion of an element in \mathbb{Z}_Z , we mean dispersing it into a CPM of size $Z \times Z$. And such one-to-one CPM-dispersion of an integer will be utilized to construct all QC-LDPC codes presented in this paper.

B. QC-LDPC CODES AND SUPERPOSITION CONSTRUCTION

A binary (n, k) QC-LDPC code is defined by the null space of an $m \times n$ sparse parity-check matrix \mathbf{H} over the finite field $\text{GF}(2)$, which consists of an array of circulants and has a rank of $n - k (\leq m)$. If \mathbf{H} has a constant column weight d_v and a constant row weight d_c , the resultant LDPC code is called (d_v, d_c) -regular QC-LDPC code. If the rows and/or columns of \mathbf{H} have various weights, then the null space of \mathbf{H} defines an irregular QC-LDPC code. Taking into account the implementation, the parity-check matrix \mathbf{H} of a QC-LDPC code is commonly

composed of an array of CPMs and/or ZMs of the same size, e.g.,

$$\mathbf{H} = \begin{bmatrix} \mathbf{Q}(p_{0,0}) & \mathbf{Q}(p_{0,1}) & \cdots & \mathbf{Q}(p_{0,n_b-1}) \\ \mathbf{Q}(p_{1,0}) & \mathbf{Q}(p_{1,1}) & \cdots & \mathbf{Q}(p_{1,n_b-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q}(p_{m_b-1,0}) & \mathbf{Q}(p_{m_b-1,1}) & \cdots & \mathbf{Q}(p_{m_b-1,n_b-1}) \end{bmatrix} \quad (1)$$

where $p_{i,j} \in \{-1\} \cup \mathbb{Z}_Z$ for $0 \leq i < m_b$, $0 \leq j < n_b$, $n = n_b \times Z$, and $m = m_b \times Z$. \mathbf{H} shares a *one-to-one* mapping with the following matrix

$$\mathbf{P} = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n_b-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n_b-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m_b-1,0} & p_{m_b-1,1} & \cdots & p_{m_b-1,n_b-1} \end{bmatrix}.$$

Here we call \mathbf{P} the *exponent matrix* of \mathbf{H} , and each entry in \mathbf{P} is referred to as *shift value*, in particular. We denote the *base matrix* of \mathbf{P} by $\mathbf{B} = [b_{i,j}]_{0 \leq i < m_b, 0 \leq j < n_b}$, where $b_{i,j} = 0$ if $p_{i,j} = -1$ and $b_{i,j} = 1$ otherwise. Note that the method of constructing \mathbf{H} from \mathbf{B} or \mathbf{P} in (1) is known as *superposition construction* or *protograph construction* [11].

C. QC-LDPC CODES WITH DIAGONAL STRUCTURE

Parity-check matrices with diagonal structures can be used to perform LDPC encoding directly [25]. Such structures include diagonal and bidiagonal structures, as well as their combinations. Assume that the exponent matrix for a bidiagonal parity-check matrix is \mathbf{P} of size $m_b \times n_b$ and can be divided into two parts, i.e., $\mathbf{P} = [\mathbf{P}' \ \mathbf{P}'']$, in which \mathbf{P}' is an $m_b \times (n_b - m_b + 1)$ matrix as follows

$$\mathbf{P}' = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n_b-m_b} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n_b-m_b} \\ p_{2,0} & p_{2,1} & \cdots & p_{2,n_b-m_b} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m_b-2,0} & p_{m_b-2,1} & \cdots & p_{m_b-2,n_b-m_b} \\ p_{m_b-1,0} & p_{m_b-1,1} & \cdots & p_{m_b-1,n_b-m_b} \end{bmatrix} \quad (2)$$

and \mathbf{P}'' is an $m_b \times (m_b - 1)$ matrix given by

$$\mathbf{P}'' = \begin{bmatrix} 0 & -1 & \cdots & -1 & -1 \\ 0 & 0 & \cdots & -1 & -1 \\ -1 & 0 & \cdots & -1 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \cdots & 0 & 0 \\ -1 & -1 & \cdots & -1 & 0 \end{bmatrix} \quad (3)$$

where p_{0,n_b-m_b} and p_{m_b-1,n_b-m_b} are nonnegative, and for $0 < i < m_b - 1$, one and only one p_{i,n_b-m_b} is nonnegative and others are set to “-1”.

What the base/exponent matrices of 5G LDPC codes adopt is a hybrid structure that combines the characteristics of both diagonal and bidiagonal structures. More details are presented in the next section.

D. CYCLE AND GIRTH

An LDPC code can also be described by a bipartite graph, termed Tanner graph [26], which has a one-to-one correspondence with the parity-check matrix \mathbf{H} . Two classes of nodes in a Tanner graph are usually referred to as variable nodes (VNs) and check nodes (CNs), corresponding to the columns and rows of \mathbf{H} , respectively. Suppose the parity-check matrix $\mathbf{H} = [h_{i,j}]_{0 \leq i < m, 0 \leq j < n}$, in which $h_{i,j}$ is equal to 0 or 1. Then in its Tanner graph, the i -th CN is connected to the j -th VN by an edge if and only if $h_{i,j} = 1$. It is worth noting that a pair of VN and CN in a Tanner graph is either not connected or connected by one and only one edge in this work, i.e., the Tanner graph does not permit multiple edges or self loops.

A *cycle* in a Tanner graph refers to a closed path that begins and ends at the same node, and its length is defined as the number of edges on this path. The *girth* of a Tanner graph, denoted by g , is the length of the shortest cycle, which is even and no less than 4.

The impact of Tanner graph on code performance is complex. There are some conclusions that a Tanner graph with girth 6 or 8 is sufficient for good performance. Many examples of such codes can be found in [11], [13]–[17], and [23]. The error floor performance of an LDPC code is related to the short cycles in its Tanner graph and the sizes of relevant trapping sets. In general, cycles of length 4 will degrade the error performance of an LDPC code decoded with iterative decoding algorithms. Hence, if possible, all constructions of LDPC codes require that their parity-check matrices should satisfy the following *row&column constraint* [14]: *no two rows (or columns) have more than one nonzero element in the same position*. This constraint can ensure that the Tanner graph of designed code has no cycles of length 4 and achieves girth at least 6 [3], [14], [23].

Fig. 1 shows the Tanner graph for a (6, 3) LDPC code specified by the following 4×6 parity-check matrix (with rank 3)

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (4)$$

\mathbf{H}_1 has constant column weight 2 and constant row weight 3, and satisfies *row&column constraint*. Hence, its Tanner graph in Fig. 1 has no cycles of length 4 and girth at least 6. Actually its girth is exactly 6. A cycle of length 6 has been depicted by the bold line in the figure.

III. STRUCTURE AND PROPERTIES OF 5G LDPC CODES

A. CODING FOR 5G EMBB DATA CHANNEL

The ITU-R has outlined the performance requirements for 5G communication (IMT-2020) in [27] and defined three typical usage scenarios for 5G: eMBB, ultra reliable low latency communications (URLLC), and massive machine type communications (mMTC). According to their applications, URLLC and mMTC are sensitive to the latency

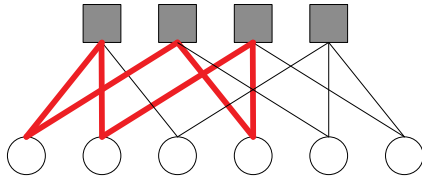


FIGURE 1. Tanner graph of a (6, 3) LDPC code specified by H_1 in (4).

TABLE 1. Coding parameters for performance evaluation in eMBB scenario [28].

Index	Assumption(s)
Channel	AWGN
Modulation	QPSK, 16QAM, 64QAM
Code Rate	1/5, 1/3, 2/5, 1/2, 2/3, 3/4, 5/6, 8/9
Information Length	100, 400, 1000, 2000, 4000, 6000, 8000

¹ Fading channels will be considered in the next stage.
² These code rates and information lengths should be evaluated at least, and only for initial performance evaluations. Other code rates and information lengths are not precluded. The evaluation is limited to similar code rate and information length.

and hence use short data package with more reliable communication, e.g., requiring no visible error-floor down to BLER of 10^{-5} . eMBB is the most obvious extension of 4G Long Term Evolution (LTE) and remains the most critical scenario with the continued increase of user demands (e.g., larger user density and better user experience). Channel coding is one of key technologies expected to satisfy the demands of eMBB scenario and needs to support a much wider range of code rates, code lengths, and modulation formats than 4G LTE. In particular, it is suggested by 3GPP [30] that eMBB code lengths range from about 100 to 8000 bits and code rates range from 1/5 to 8/9 (as shown in Table 1), and the target codes can support the information throughput up to 20 Gbps. Furthermore, regarding the error-correcting performance, good performance at the BLER of 10^{-2} and invisible error-floor down to BLER of 10^{-4} are required.

There are many channel coding schemes with capacity-approaching performance at the large code lengths, such as LDPC, spatially coupled LDPC [31], Turbo [32], and Polar [33] codes. After the comprehensive assessment of error-correcting performance, achievable throughput, implementation complexity, and processing energy consumption, QC-LDPC codes are accepted by 3GPP as the channel coding scheme for 5G eMBB data channel [21].

B. BASE MATRICES OF STANDARD 5G LDPC CODES

As mentioned above, 5G LDPC codes belong to QC-LDPC codes and hence can be constructed based on the *superposition* method. Fig. 2 shows the structure of the base matrix of these standard codes. As shown in the figure, the base matrix consists of five submatrices **A**, **D**, **O**, **E**, and **I**:

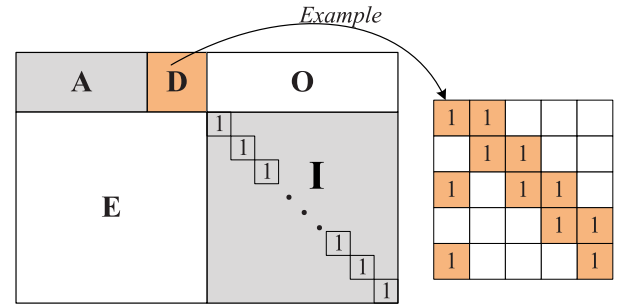


FIGURE 2. Structure diagram of base matrix for standard 5G LDPC codes.

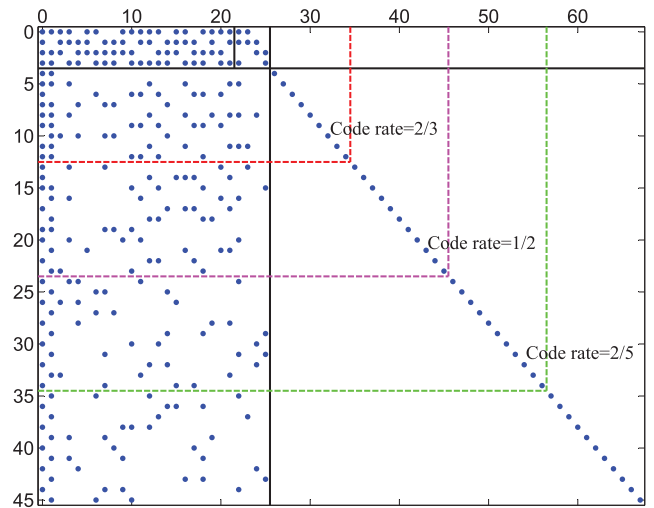


FIGURE 3. Scatter diagram of $B_{5G,1}$ for standard 5G LDPC codes.

- **A** corresponds to systematic bits;
- **D** corresponds to parity bits and is a square matrix with bidiagonal structure:
 - First column is of weight 3;
 - Submatrix composed of other columns after the first column has an upper bidiagonal structure;
- **O** is a zero matrix;
- **I** is an identity matrix.

The combination of **A** and **D** is defined as the *kernel*, and the other parts (**O**, **I**, and **E**) are called *extensions*. Clearly, such code structure is similar to *Raptor-like codes* [34].

Two base matrices, denoted by $B_{5G,1}$ and $B_{5G,2}$, respectively, are adopted for 5G LDPC codes [35]. These two matrices have similar structure, but $B_{5G,1}$ is designed for information lengths up to 8448 and code rates from 1/3 to 8/9, while $B_{5G,2}$ for smaller information lengths no more than 3840 and code rates from 1/5 to 2/3 [36]. Fig. 3 shows the scatter diagram of $B_{5G,1}$. We can see from the figure that $B_{5G,1}$ has 46 rows and 68 columns. VN blocks labeled from 0 to 21 correspond to the information bits and other VN blocks correspond to the parity bits.

C. EXPONENT MATRICES OF STANDARD 5G LDPC CODES

According to the 3GPP requirements [37], the exponent matrices of 5G LDPC codes should support all lifting sizes

TABLE 2. Lifting sizes $Z (= a \cdot 2^j)$ supported by standard 5G LDPC codes.

Z		j							
		0	1	2	3	4	5	6	7
a	2	2	4	8	16	32	64	128	256
	3	3	6	12	24	48	96	192	384
	5	5	10	20	40	80	160	320	
	7	7	14	28	56	112	224		
	9	9	18	36	72	144	288		
	11	11	22	44	88	176	352		
	13	13	26	52	104	208			
	15	15	30	60	120	240			

TABLE 3. Mapping between exponent matrices and lifting size sets of standard 5G LDPC codes.

Exponent Matrix	Lifting Size Set
$\mathbf{P}_1 (a = 2)$	$\{a \times 2^j j = 0, 1, 2, 3, 4, 5, 6, 7\}$
$\mathbf{P}_2 (a = 3)$	$\{a \times 2^j j = 0, 1, 2, 3, 4, 5, 6, 7\}$
$\mathbf{P}_3 (a = 5)$	$\{a \times 2^j j = 0, 1, 2, 3, 4, 5, 6\}$
$\mathbf{P}_4 (a = 7)$	$\{a \times 2^j j = 0, 1, 2, 3, 4, 5\}$
$\mathbf{P}_5 (a = 9)$	$\{a \times 2^j j = 0, 1, 2, 3, 4, 5\}$
$\mathbf{P}_6 (a = 11)$	$\{a \times 2^j j = 0, 1, 2, 3, 4, 5\}$
$\mathbf{P}_7 (a = 13)$	$\{a \times 2^j j = 0, 1, 2, 3, 4\}$
$\mathbf{P}_8 (a = 15)$	$\{a \times 2^j j = 0, 1, 2, 3, 4\}$

in Table 2, where $Z = a \times 2^j$ for $a \in \{2, 3, 5, 7, 9, 11, 13, 15\}$ and $0 \leq j \leq 7$.

Since 3GPP RAN AH NR2 meeting [38], 16 exponent matrices are accepted for 5G LDPC codes and each base matrix possesses 8 exponent matrices. We list various exponent matrices versus their supported lifting size sets in Table 3. As shown in the table, each set is specified by the parameter a .

In order to implement various information lengths and rate adaptation, *shortening* and *puncturing* methods are used for 5G LDPC codes. Puncturing is applied to both the information and parity bits in the codeword, while shortening is just designed by zero padding for the information bits. For both $\mathbf{B}_{5G,1}$ and $\mathbf{B}_{5G,2}$, and for all code rates, code bits corresponding to first two circulant columns are punctured before the transmission. These two punctured blocks have relatively high column weight among all columns and hence are usually called high-weight columns. For rate-compatible usage, the parity bits are punctured from right to left, e.g., for $R = 1/2$, the 24×46 submatrix on the upper left corner of the base/exponent matrix is used, and for $R = 1/3$, the full base/exponent matrix of 46×68 is used.

Assume that $\mathbf{P}_t = [v_{i,j}]_{0 \leq i < m_Z, 0 \leq j < n_Z}$ ($1 \leq t \leq 8$) have been constructed (we will discuss their construction later), $(m_Z, n_Z) = (46, 68)$ for $\mathbf{B}_{5G,1}$, and $(m_Z, n_Z) = (42, 52)$ otherwise. Next we will describe how to achieve the exponent matrix $\mathbf{P} = [p_{i,j}]_{0 \leq i < m_b, 0 \leq j < n_b}$ ($m_b \leq m_Z$ and $n_b \leq n_Z$) for an (N, K) LDPC code with information length K and code rate $R = K/N$. Let k_b denote the number of information circulant columns in $\mathbf{B}_{5G,1}$ or $\mathbf{B}_{5G,2}$. And for $\mathbf{B}_{5G,1}$, $k_b = 22$; for $\mathbf{B}_{5G,2}$, k_b , depending on K , is set to one the following

values $\{10, 9, 8, 6\}$ [39]. The determination of \mathbf{P} includes the following steps:

- 1) Based on **Algorithm 1**, determine the base matrix and k_b for the given K and R ;
- 2) Select Z as the minimum value in Table 2, such that $(k_b \cdot Z) \geq K$;
- 3) $n_b = \lceil k_b/R \rceil + 2$ and $m_b = n_b - k_b$, where $\lceil x \rceil$ denotes the nearest integers greater than or equal to x ;
- 4) Based on Z , determine $\mathbf{P}^* = [v_{i,j}]_{0 \leq i < m_Z, 0 \leq j < n_Z}$ from 8 matrices \mathbf{P}_t ($1 \leq t \leq 8$) in Table 3;
- 5) Calculate $\mathbf{P} = [p_{i,j}]_{0 \leq i < m_b, 0 \leq j < n_b}$, where $p_{i,j} = -1$ if $v_{i,j} = -1$ and $p_{i,j} = v_{i,j} \pmod Z$ otherwise, in which “mod” denotes the modulo arithmetic;
- 6) Disperse each element of \mathbf{P} into binary CPM of size $Z \times Z$ or ZM of the same size, and obtain a parity-check matrix \mathbf{H} of size $m_b Z \times n_b Z$, which will be used for the encoding and decoding of the (N, K) LDPC code.

Algorithm 1 Base Matrix and k_b Determination for 5G LDPC Codes

Input: Information length K , code rate R , base matrices $\mathbf{B}_{5G,1}$ and $\mathbf{B}_{5G,2}$

Output: Base matrix \mathbf{B} and k_b

```

1: if  $K > 3840$  then
2:    $\mathbf{B} = \mathbf{B}_{5G,1}$ 1
3: else if  $K \leq 308$  then
4:    $\mathbf{B} = \mathbf{B}_{5G,2}$ 
5: else
6:   if  $R > (2/3)$  then
7:      $\mathbf{B} = \mathbf{B}_{5G,1}$ 
8:   else
9:      $\mathbf{B} = \mathbf{B}_{5G,2}$ 
10:  end if
11: end if
12: if  $\mathbf{B} = \mathbf{B}_{5G,1}$  then
13:    $k_b = 22$ 
14: else
15:   if  $K > 640$  then
16:      $k_b = 10$ 
17:   else if  $560 < K \leq 640$  then
18:      $k_b = 9$ 
19:   else if  $192 < K \leq 560$  then
20:      $k_b = 8$ 
21:   else
22:      $k_b = 6$ 
23:   end if
24: end if
25: return  $\mathbf{B}$  and  $k_b$ 

```

Based on the above-mentioned steps, we can find that the design and structure of exponent matrices make 5G LDPC codes own better rate-compatible property. Fig. 4 provides an illustration for the encoding process of these codes. Because

¹Assume two matrices \mathbf{B}_1 and \mathbf{B}_2 of the same size, “ $\mathbf{B}_1 = \mathbf{B}_2$ ” denotes the element-wise assignment of \mathbf{B}_2 to \mathbf{B}_1 .

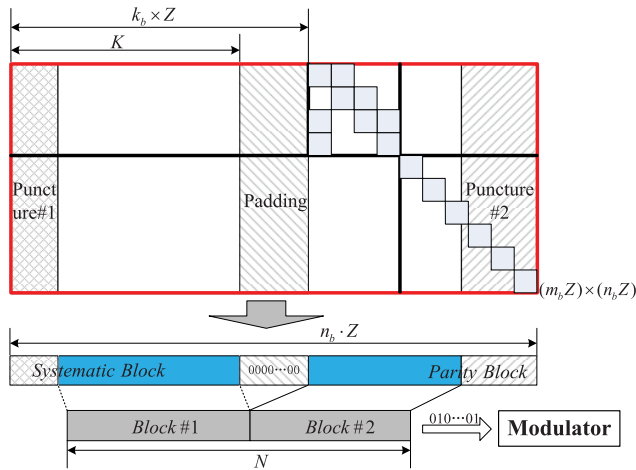


FIGURE 4. Shortening by zero padding and puncturing of standard 5G LDPC codes.

punctured bits involve two parts, first two circulant columns and block(s) consisting of partial parity bits from right to left, respectively. Then we can easily calculate the number of punctured and shortened bits as follows:

- 1) Number of punctured bits composed of first two circulant columns $N_{\text{punc1}} = 2Z$;
- 2) Number of shortened information bits by zero padding $N_{\text{padd}} = k_b \times Z - K$;
- 3) Number of punctured parity bits from right to left $N_{\text{punc2}} = n_b \times Z - 2Z - N - N_{\text{padd}}$.

Thus, as shown in Fig. 4, the number of coded bits that really enter the **Modulator** is not $n_b Z$ but N .

IV. ALGEBRA-ASSISTED CONSTRUCTION OF QC-LDPC CODES WITH MULTIPLE LIFTING SIZES

From the discussion above, we can see that the construction of the exponent matrix is a key point for the QC-LDPC codes. For an LDPC code decoded with iterative algorithms, the performance is closely related to the structure of its Tanner graph and relevant cycles. Hence, cycle structure has been utilized by many publications as a crucial measure of the error performance of an LDPC code, and with some consensus that short cycles should be avoided in the Tanner graph of a code for better performance.

Inspired by standard 5G LDPC codes, we propose a feasible metric from the perspective of cycle structure for the shift value selection of one class of QC-LDPC codes, whose exponent matrices can support multiple lifting sizes. Based on this metric, we develop a practical algorithm for constructing the QC-LDPC codes with multiple lifting sizes. By taking 5G LDPC codes as an example, we extend the proposed metric and algorithm to design the exponent matrices for $\mathbf{B}_{5G,1}$ and $\mathbf{B}_{5G,2}$. Note that, standard 5G LDPC codes have not yet been discussed in-depth in theoretical analysis. It is the important role that cycles play in the error performance of LDPC codes and the deficiency of theoretical research on such codes motivate the cycle analysis in this section.

In particular, the analysis can simplify our design of these codes.

A. CYCLE METRIC FOR QC-LDPC CODES WITH MULTIPLE LIFTING SIZES

From [17], for QC-LDPC codes with exponent matrix $\mathbf{P} = [p_{i,j}]_{0 \leq i < m_b, 0 \leq j < n_b}$, cycles of length $2d$ ($g \leq 2d \leq 2g - 2$, where g denotes the girth) exist if and only if there are $2d$ nonnegative shift values $p_{i,j}$ satisfying

$$\sum_{z=0}^{d-1} (p_{i_z, j_z} - p_{i_{z+1}, j_{z+1}}) = 0 \pmod{Z} \quad (5)$$

where $0 \leq i_z < m_b, 0 \leq j_z < n_b, i_z \neq i_{z+1}, j_z \neq j_{z+1}, i_0 = i_d, \text{ and } j_0 = j_d$.

Proposition 1: The group of shift values involved in (5) will result in M cycles of length $2d$, where M is a factor of Z . The analysis on M can be found in [41].

Proposition 2: For integers Z_1 and Z_2 , where $Z_2 > Z_1 > 0$, consider $2d$ nonnegative integers that are less than Z_2 and can consist of an ordered sequence $p_{i_0, j_0}, p_{i_1, j_1}, p_{i_2, j_2}, \dots, p_{i_{d-1}, j_{d-1}}, p_{i_0, j_0}$, then

$$\Pr \left\{ \sum_{z=0}^{d-1} (p_{i_z, j_z} - p_{i_{z+1}, j_{z+1}}) = 0 \pmod{Z_1} \right\} \geq \Pr \left\{ \sum_{z=0}^{d-1} (p_{i_z, j_z} - p_{i_{z+1}, j_{z+1}}) = 0 \pmod{Z_2} \right\}. \quad (6)$$

For QC-LDPC codes, in order to adapt various information lengths, multiple lifting sizes are generally recommended for a single exponent matrix, e.g., standard 5G LDPC codes. Assume that the base matrix \mathbf{B} and lifting size set $\mathcal{S}_Z = \{Z_1, Z_2, \dots, Z_{N_Z}\}$ are given, we propose a metric, referred to as *weighted average number of cycles* (WANC), to select shift values for relevant exponent matrix \mathbf{P} that can support all lifting sizes in \mathcal{S}_Z . The proposed WANC metric is defined by

$$\mathcal{W}(\mathbf{P}) = \sum_{i=1}^{N_Z} \mathcal{W}_Z(i) = \sum_{i=1}^{N_Z} \omega_Z(i) \sum_{j=1}^{N_c} \omega_c(j) N(i, j) \quad (7)$$

where $N_c = (L_{\max} - 2)/2$ with L_{\max} equal to the maximum length of tested cycles, $\omega_Z(1) \geq \omega_Z(2) \geq \dots \geq \omega_Z(N_Z)$, and $\omega_c(1) > \omega_c(2) > \dots > \omega_c(N_c)$. For $1 \leq i \leq N_Z$ and $1 \leq j \leq N_c$, $\omega_Z(i)$ is the weight for $Z_i \in \mathcal{S}_Z$, $\omega_c(j)$ is the weight for those cycles of length $(2j + 2)$, and $N(i, j)^2$ denotes the group number of shift values that satisfy (5) while Z_i and cycles of length $(2j + 2)$ are considered. For a given index i and lifting size $Z_i \in \mathcal{S}_Z$, $\mathcal{W}_Z(i)$ is the weighted sum of all $N(i, j)$ for $1 \leq j \leq N_c$, and can be equivalent to *the weighted sum of the number of various-length cycles for a single lifting size*. We can see that $\mathcal{W}(\mathbf{P})$ actually indicates the weighted average of the number of various-length cycles normalized by the corresponding lifting size. Following proposition shows that WANC metric is finite for a given base matrix and thus

can provide a good measure in designing the QC-LDPC codes with multiple lifting sizes.

Proposition 3: $\mathcal{W}(\mathbf{P})$ is lower- and upper-bounded by

$$0 \leq \mathcal{W}(\mathbf{P}) \leq \mathcal{W}(\mathbf{B} - \mathbf{1}). \quad (8)$$

where $\mathbf{1}$ is the all-one matrix of the same size as \mathbf{B} .

Proof: The upper bound of $\mathcal{W}(\mathbf{P})$ originates from *Proposition 1*, and can be determined based on such an exponent matrix, which has the same size as \mathbf{P} and all the same nonnegative shift values. The simplest case is $(\mathbf{B} - \mathbf{1})$, i.e., all nonnegative shift values are fixed to zero. ■

The WANC metric $\mathcal{W}(\mathbf{P})$ seems reasonable from the perspective of cycle structure. According to (7), the contribution of each lifting size to $\mathcal{W}(\mathbf{P})$ depends on not the number of cycles of various lengths but $N(i, j)$, and hence weakens the impact of the value of each lifting size as far as possible. And for a given lifting size, shorter cycles will more degrade the performance, thus larger $\omega_c(j)$ is assigned for the shorter cycles. In addition, for a given group of shift values, cycles easily occur in the case of smaller lifting size (*Proposition 2*), thus we introduce the weight $w_Z(i)$ for each lifting size in \mathcal{S}_Z . And the smaller the lifting size is, the larger the relevant weight is. Note that, for $1 \leq i, j \leq N_Z$, the ratio $\omega_Z(i)/\omega_Z(j)$ is not necessarily equal to Z_j/Z_i .

For a single lifting size, the WANC metric reduces to

$$\mathcal{W}(\mathbf{P}) = \mathcal{W}_Z(1) = \sum_{j=1}^{N_c} \omega_c(j)N(1, j), \quad (9)$$

which still meets the general metric defined as the weighted sum of the number of various-length cycles.

B. CONSTRUCTION OF QC-LDPC CODES WITH MULTIPLE LIFTING SIZES

Based on the WANC metric proposed in last subsection, we now discuss how to construct QC-LDPC codes whose exponent matrix can support multiple lifting sizes. We summarize our construction as the following steps:

- 1) Based on the actual requirements (e.g., rate-compatible), determine the structure of parity-check matrix;
- 2) According to the given design targets (e.g., degree distributions and threshold), construct the base matrix \mathbf{B} for the desired exponent matrix \mathbf{P} ;
- 3) Simplify the design with some alternative analysis methods (e.g., cycle analysis): Determine the $m_s \times n_s$ submatrix \mathbf{P}_{sm} with shift values to be specified in \mathbf{P} and relevant base matrix \mathbf{B}_{sm} of the same size (The subscript “sm” stands for “submatrix”);
- 4) Based on algebraic methods, construct a *circulant coefficient table* \mathbf{T}_{cc} of size $N_t \times n_s$ ($N_t > m_s$), which must satisfy at least: *all the entries in a row (a column) are distinct*;

²Consider lifting size $Z = Z_1$ and cycles of length 4, if we find t group of shift values satisfying (5), then $N(1, 1) = t$ and there are tZ_1 resulting cycles of length 4.

- 5) Select m_s rows from \mathbf{T}_{cc} to form \mathbf{P}_{sm}^* and construct $\mathbf{P}_{sm} = \mathbf{B}_{sm} \otimes \mathbf{P}_{sm}^* + \mathbf{B}_{sm} - \mathbf{1}$ subject to minimizing $\mathcal{W}(\mathbf{P}_{sm})$, where the operation \otimes is defined as the element-wise product, i.e.,

$$\mathbf{B}_{sm} \otimes \mathbf{F} = [b_{i,j}f_{i,j}]_{0 \leq i < m_s, 0 \leq j < n_s} \quad (10)$$

in which matrices \mathbf{F} and \mathbf{B}_{sm} have the same size $m_s \times n_s$;

- 6) Combine \mathbf{P}_{sm} with other specific parts and obtain \mathbf{P} .

We formulate the steps 5 and 6 of above processes as *Algorithm 2*.³

Algorithm 2 Construction of Exponent Matrix With Multiple Lifting Sizes

Input: Circulant coefficient table \mathbf{T}_{cc} , base matrix \mathbf{B} of size $m_b \times n_b$, lifting size set $\mathcal{S}_Z = \{Z_1, Z_2, \dots, Z_{N_Z}\}$, number of searches N_{search}

Output: Exponent matrix \mathbf{P} of size $m_b \times n_b$

Initialization: Determine relevant \mathbf{B}_{sm} of \mathbf{P}_{sm} from \mathbf{B} , set $\mathbf{P}_{sm} = \mathbf{B}_{sm} - \mathbf{1}$ and $count = 0$

- 1: Calculate $\mathcal{W}^* = \mathcal{W}(\mathbf{P}_{sm})$ based on (7)
 - 2: **repeat**
 - 3: Assign $r_0, r_1, \dots, r_{m_s-1}$ to distinct nonnegative integers less than N_t
 - 4: **for** $i = 0 : m_s - 1$ **do**
 - 5: $\mathbf{M}[i, :] = \mathbf{B}_{sm}[i, :] \otimes \mathbf{T}_{cc}[r_i, :] + \mathbf{B}_{sm}[i, :] - \mathbf{1}$
 - 6: **end for**
 - 7: Calculate $\mathcal{W}(\mathbf{M})$ based on (7)
 - 8: **if** $\mathcal{W}(\mathbf{M}) < \mathcal{W}^*$ **then**
 - 9: $\mathbf{P}_{sm} = \mathbf{M}, \mathcal{W}^* = \mathcal{W}(\mathbf{M})$
 - 10: **end if**
 - 11: $count = count + 1$
 - 12: **until** $count > N_{search}$
 - 13: Obtain \mathbf{P} by combining \mathbf{P}_{sm} with other specific parts
 - 14: **return** \mathbf{P}
-

In *Algorithm 2*, the base matrix \mathbf{B} is required beforehand. \mathbf{B} is generally designed with good decoding threshold by using computer-aided EXIT chart-based search methods, e.g., protograph EXIT (P-EXIT) in [42] and [43], or algebraic method [24], [44]. In general, good threshold of decoding convergence can provide good waterfall performance, but not necessarily low error-floor [11]. The PEG algorithm [6] can also be used to help design the base matrix. The circulant coefficient table \mathbf{T}_{cc} can be designed to be a *Latin square*. In addition, the algebraic methods generalized in [16] and *cyclic difference family* can also be used to construct \mathbf{T}_{cc} .

Here we provide several methods to construct \mathbf{T}_{cc} by following [16]. Let \mathbb{F}_q be a finite field with q elements, where q is a prime or a power of a prime (q is fixed to prime in the rest of this paper). Let α be a primitive element of \mathbb{F}_q . Then, the powers of $\alpha, \alpha^{-\infty} = 0, \alpha^0 = 1, \alpha, \dots, \alpha^{q-2}$, give all q elements of \mathbb{F}_q .

³Assume matrix $\mathbf{B}, \mathbf{B}[i, :]$ and $\mathbf{B}[:, j]$ denote the i -th row and j -th column of \mathbf{B} , respectively.

For $0 \leq n_s, N_t < q$, let $\mathbf{S}_{\text{row}} = \{\alpha^{i_0}, \alpha^{i_1}, \dots, \alpha^{i_{N_t-1}}\}$ and $\mathbf{S}_{\text{col}} = \{\alpha^{j_0}, \alpha^{j_1}, \dots, \alpha^{j_{n_s-1}}\}$ be two arbitrary subsets of elements in \mathbb{F}_q with $i_l \in \{-\infty, 0, 1, \dots, q-2\}$ for $0 \leq l < N_t$, $j_l \in \{-\infty, 0, 1, \dots, q-2\}$ for $0 \leq l < n_s$, $i_0 < i_1 < \dots < i_{N_t-1}$, and $j_0 < j_1 < \dots < j_{n_s-1}$. Let η be a nonzero element in \mathbb{F}_q . \mathbf{T}_{cc} can be obtained by one of the following methods:

$$\mathbf{T}_{\text{cc}} = \left[\alpha^{i_r} + \eta \alpha^{j_c} \right]_{0 \leq r < N_t, 0 \leq c < n_s} \quad (11)$$

$$\mathbf{T}_{\text{cc}} = \left[\alpha^{i_r} - \eta \alpha^{j_c} \right]_{0 \leq r < N_t, 0 \leq c < n_s} \quad (12)$$

$$\mathbf{T}_{\text{cc}} = \left[\eta \alpha^{i_r} + \alpha^{j_c} \right]_{0 \leq r < N_t, 0 \leq c < n_s} \quad (13)$$

and

$$\mathbf{T}_{\text{cc}} = \left[\log_{\alpha}(\alpha^{i_r} + \eta \alpha^{j_c}) \right]_{0 \leq r < N_t, 0 \leq c < n_s} \quad (14)$$

$$\mathbf{T}_{\text{cc}} = \left[\log_{\alpha}(\alpha^{i_r} - \eta \alpha^{j_c}) \right]_{0 \leq r < N_t, 0 \leq c < n_s} \quad (15)$$

$$\mathbf{T}_{\text{cc}} = \left[\log_{\alpha}(\eta \alpha^{i_r} + \alpha^{j_c}) \right]_{0 \leq r < N_t, 0 \leq c < n_s} \quad (16)$$

where $\log_y(x)$ computes the base y logarithm of x . With this method, any two elements in the same row/column of \mathbf{T}_{cc} are different, which can be verified by following [16].

As an example, **Algorithm 2** will be extended to design the exponent matrix for $\mathbf{B}_{5G,1}$ and $\mathbf{B}_{5G,2}$ of standard 5G LDPC codes in the next subsection.

C. EXPONENT MATRIX CONSTRUCTION FOR 5G LDPC CODES

By corresponding to Fig. 2, the exponent matrix of standard 5G LDPC codes can be divided into five submatrices \mathbf{A}_b , \mathbf{D}_b , \mathbf{O}_b , \mathbf{E}_b , and \mathbf{I}_b . Based on (5), possible cycles can be classified into the following seven categories:

- 1) Cycles generated only from \mathbf{A}_b ;
- 2) Cycles generated only from \mathbf{D}_b ;
- 3) Cycles generated only from \mathbf{E}_b ;
- 4) Cycles generated between \mathbf{A}_b and \mathbf{D}_b ;
- 5) Cycles generated between \mathbf{A}_b and the first k_b columns of \mathbf{E}_b ;
- 6) Cycles generated between \mathbf{D}_b and the last 4 columns of \mathbf{E}_b ;
- 7) Cycles generated between \mathbf{A}_b , \mathbf{E}_b , and \mathbf{D}_b .

In order to further reduce the complexity of encoder implementation, for a given base matrix ($\mathbf{B}_{5G,1}$ or $\mathbf{B}_{5G,2}$), same bidiagonal submatrix \mathbf{D}_b is used for most matrices in Table 3. For example, for $\mathbf{B}_{5G,1}$, \mathbf{D}_b is fixed as

$$\mathbf{D}_b = \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 \\ 1 & -1 & -1 & 0 \end{bmatrix} \quad (17)$$

for 7 exponent matrices except \mathbf{P}_7 . Actually, it only requires that top and last elements in the first column of \mathbf{D}_b are same.

For convenience, we limit the cycle analysis to the case of $\mathbf{B}_{5G,1}$. Similar analysis can be made for $\mathbf{B}_{5G,2}$.

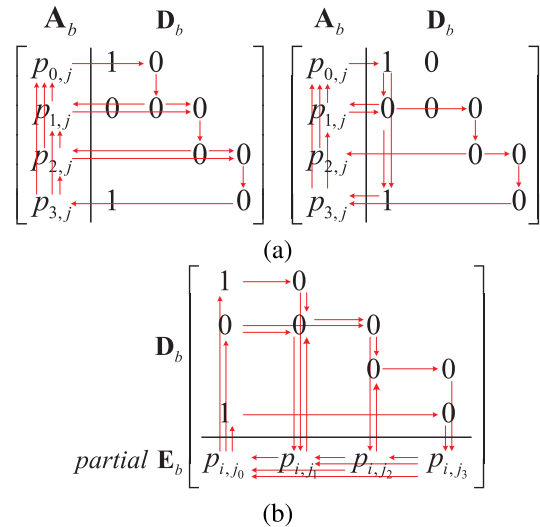


FIGURE 5. (a) Cycles generated between \mathbf{A}_b and \mathbf{D}_b for $\mathbf{B}_{5G,1}$, where $0 \leq j < k_b$; (b) Cycles generated between \mathbf{D}_b and last 4 columns of \mathbf{E}_b for $\mathbf{B}_{5G,1}$, where $j_l = k_b + l$ with $0 \leq l \leq 3$.

Now consider the fourth kind of cycles and assume that such cycles exist. As shown in Fig. 5 (a), we summarize such cycles as follows:

- 1) Cycles of length 4 exist if and only if

$$\begin{cases} p_{i,j} - p_{i+1,j} = 0 & (\text{mod } Z), \quad i = 0, 1, 2 \\ p_{i,j} - p_{1,j} - 1 = 0 & (\text{mod } Z), \quad i = 0, 3 \\ p_{0,j} - p_{3,j} = 0 & (\text{mod } Z) \end{cases} \quad (18)$$

- 2) Cycles of length 6 exist if and only if

$$\begin{cases} p_{i,j} - p_{i+2,j} = 0 & (\text{mod } Z), \quad i = 0, 1 \\ p_{0,j} - p_{2,j} - 1 = 0 & (\text{mod } Z) \end{cases} \quad (19)$$

- 3) Cycles of length 8 result in

$$\begin{cases} p_{0,j} - p_{3,j} = 0 & (\text{mod } Z) \\ p_{0,j} - p_{3,j} - 1 = 0 & (\text{mod } Z) \end{cases} \quad (20)$$

where $0 \leq j < k_b$. Therefore, to avoid such cycles, the following **column constraints** should be imposed on \mathbf{A}_b :

- (1) Any two nonnegative elements in the same column are different;
- (2) For $0 \leq j < k_b$, $p_{0,j} \neq p_{u,j} + 1$ if both $p_{0,j}$ and $p_{u,j}$ are nonnegative, where $1 \leq u \leq 3$, and $p_{3,j} \neq p_{1,j} + 1$ if both $p_{1,j}$ and $p_{3,j}$ are nonnegative.

Next, consider the sixth kind of cycles generated between \mathbf{D}_b and the last 4 columns of \mathbf{E}_b , and assume that these cycles exist. Then according to (5) and Fig. 5 (b), such cycles of various lengths can be enumerated as follows:

- 1) Cycles of length 4 exist if and only if

$$\begin{cases} p_{i,j_l} - p_{i,j_{l+1}} = 0 & (\text{mod } Z), \quad l = 0, 1, 2 \\ p_{i,j_0} - p_{i,j_l} - 1 = 0 & (\text{mod } Z), \quad l = 1, 3 \\ p_{i,j_0} - p_{i,j_2} = 0 & (\text{mod } Z) \end{cases} \quad (21)$$

2) Cycles of length 6 exist if and only if

$$\begin{cases} p_{i,j_l} - p_{i,j_3} = 0 & (\text{mod } Z), l = 0, 1 \\ p_{i,j_0} - p_{i,j_2} - 1 = 0 & (\text{mod } Z) \end{cases} \quad (22)$$

3) Cycles of length 8 exist result in

$$p_{i,j_0} - p_{i,j_3} - 1 = 0 \quad (\text{mod } Z) \quad (23)$$

where $4 \leq i < 46$. Thus, to make desirable exponent matrix free of such cycles, we have following **row constraints** for the last 4 columns of \mathbf{E}_b :

- 1) Any two nonnegative elements in the same row are different;
- 2) For $4 \leq i < 46$, $p_{i,j_0} \neq p_{i,j_l} + 1$ if both p_{i,j_0} and p_{i,j_l} are nonnegative, where $1 \leq l \leq 3$.

There are many nonnegative elements in both \mathbf{A}_b and the first k_b columns of \mathbf{E}_b , which mean that it will be very difficult to analyze and especially remove the fifth kind of cycles generated between them. But the above analysis on the fourth and sixth kinds of cycles, as well as the equation (5) are enough to guide us to design better exponent matrices with the structure as shown in Fig. 2.

It is important to point out that the cycle analysis for 5G LDPC codes in this subsection can apply only to design the exponent matrix for a single lifting size. But we know from Section III that each exponent matrix of 5G LDPC codes is required to support multiple lifting sizes. Thus, the cycle analysis mentioned above cannot be directly used for designing the exponent matrices of 5G LDPC codes, but shows us that the code design can be simplified as constructing the first $(k_b + 4)$ columns of the exponent matrices independently, especially submatrices \mathbf{A}_b and \mathbf{E}_b . And, as mentioned in the last subsection, \mathbf{T}_{cc} with the forms given by (11)–(16) can fulfill the first one of **row/column constraints** for larger lifting sizes. Furthermore, minimizing the proposed WANC is actually equivalent to asymptotically meeting these constraints of our analysis.

By generalizing **Algorithm 2** and applying aforementioned cycle analysis, we present an algorithm to design the exponent matrix for the base matrices of 5G LDPC codes. The design is divided into following three steps:

- 1) Construct \mathbf{A}_b : Globally design all rows of \mathbf{A}_b with the constraint of \mathbf{D}_b ;
- 2) Construct \mathbf{E}_b : With the constraint of $[\mathbf{A}_b \ \mathbf{D}_b]$, design \mathbf{E}_b in the row-by-row way from top to bottom;
- 3) Combine \mathbf{A}_b and \mathbf{E}_b with other specific parts (i.e., \mathbf{D}_b , \mathbf{O}_b , and \mathbf{I}_b), and obtain \mathbf{P} .

We formulate the above processes as **Algorithm 3**.⁴

Design of 5G LDPC codes also needs to follow that of general QC-LDPC codes. However, we mainly focus on the design of exponent matrix, since design of a single exponent matrix for multiple lifting sizes is one major challenge of 5G

⁴Assume matrix \mathbf{B} : $\mathbf{B}[\text{end}, :]$ and $\mathbf{B}[:, \text{end}]$ denote the last row and last column of \mathbf{B} , respectively; $\mathbf{B}[i_1 : i_2, j_1 : j_2]$ denotes the submatrix of \mathbf{B} with elements from rows i_1, \dots, i_2 and columns j_1, \dots, j_2 .

Algorithm 3 Exponent Matrix Construction for 5G LDPC Codes

Input: Circulant coefficient table \mathbf{T}_{cc} , base matrix \mathbf{B} , \mathbf{D}_b , lifting size set $\mathcal{S}_Z = \{Z_1, Z_2, \dots, Z_{N_Z}\}$, number of searches N_{search}

Output: Exponent matrix \mathbf{P}

Initialization: $\mathbf{B}_{\text{left}} = \mathbf{B}[:, 0 : k_b + 3]$, $\mathbf{P}_{\text{left}} = \mathbf{B}_{\text{left}} - \mathbf{1}$,

$\mathbf{P}_{\text{left}}[0 : 3, k_b : k_b + 3] = \mathbf{D}_b$, $\mathbf{M} = \mathbf{P}_{\text{left}}$, $\text{count} = 0$

```

1: /***** Shift Value Selection for  $\mathbf{A}_b$  *****/
2: Calculate  $\mathcal{W}^* = \mathcal{W}(\mathbf{P}_{\text{left}}[0 : 3, :])$  based on (7)
3: repeat
4:   Assign  $r_0, r_1, r_2, r_3$  to distinct nonnegative integers less than  $N_t$ 
5:   for  $i = 0 : 3$  do
6:      $\mathbf{J} = \mathbf{B}_{\text{left}}[i, 0 : k_b - 1] \otimes \mathbf{T}_{cc}[r_i, 0 : k_b - 1]$ 
7:      $\mathbf{M}[i, 0 : k_b - 1] = \mathbf{J} + \mathbf{B}_{\text{left}}[i, 0 : k_b - 1] - \mathbf{1}$ 
8:   end for
9:   Calculate  $\mathcal{W}(\mathbf{M}[0 : 3, :])$  based on (7)
10:  if  $\mathcal{W}(\mathbf{M}[0 : 3, :]) < \mathcal{W}^*$  then
11:     $\mathbf{P}_{\text{left}}[0 : 3, 0 : k_b - 1] = \mathbf{M}[0 : 3, 0 : k_b - 1]$ 
12:     $\mathcal{W}^* = \mathcal{W}(\mathbf{M}[0 : 3, :])$ 
13:  end if
14:   $\text{count} = \text{count} + 1$ 
15: until  $\text{count} > N_{\text{search}}$ 
16:  $\mathbf{M}[0 : 3, 0 : k_b - 1] = \mathbf{P}_{\text{left}}[0 : 3, 0 : k_b - 1]$ 
17: /***** Shift Value Selection for  $\mathbf{E}_b$  *****/
18: for  $r = 4 : m_Z - 1$  do
19:   Calculate  $\mathcal{W}^* = \mathcal{W}(\mathbf{P}_{\text{left}}[0 : r, :])$  based on (7)
20:   for  $\text{row} = 0 : N_t - 1$  do
21:      $\mathbf{M}[\text{row}, :] = \mathbf{B}_{\text{left}}[\text{row}, :] \otimes \mathbf{T}_{cc}[\text{row}, :] + \mathbf{B}_{\text{left}}[\text{row}, :] - \mathbf{1}$ 
22:     Calculate  $\mathcal{W}(\mathbf{M}[0 : r, :])$  based on (7)
23:     if  $\mathcal{W}(\mathbf{M}[0 : r, :]) < \mathcal{W}^*$  then
24:        $\mathbf{P}_{\text{left}}[\text{row}, :] = \mathbf{M}[\text{row}, :]$ ,  $\mathcal{W}^* = \mathcal{W}(\mathbf{M}[0 : r, :])$ 
25:     end if
26:   end for
27: end for
28:  $\mathbf{B}_{\text{right}} = \mathbf{B}[:, k_b + 4 : \text{end}]$ 
29:  $\mathbf{P}_{\text{right}} = \mathbf{B}_{\text{right}} - \mathbf{1}$ 
30:  $\mathbf{P} = [\mathbf{P}_{\text{left}} \ \mathbf{P}_{\text{right}}]$ 
31: return  $\mathbf{P}$ 

```

LDPC code design. For simplicity, \mathbf{B} in **Algorithm 3** is selected as either $\mathbf{B}_{5G,1}$ or $\mathbf{B}_{5G,2}$. The design philosophy of $\mathbf{B}_{5G,1}$ and $\mathbf{B}_{5G,2}$ can be found in [45]. Based on previous cycle analysis, the construction is simplified as designing the submatrix composed of the first $(k_b + 4)$ columns. Then, \mathbf{T}_{cc} of size $N_t \times (k_b + 4)$ is required, where N_t needs to be large relative to m_Z . The submatrix \mathbf{D}_b depends on \mathbf{B} , i.e., \mathbf{D}_b is chosen as in (17) for $\mathbf{B}_{5G,1}$, while for $\mathbf{B}_{5G,2}$

$$\mathbf{D}_b = \begin{bmatrix} u_0 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 \\ u_1 & -1 & 0 & 0 \\ u_0 & -1 & -1 & 0 \end{bmatrix}$$

where (u_0, u_1) is set to $(0, 1)/(1, 0)$ for simplicity.

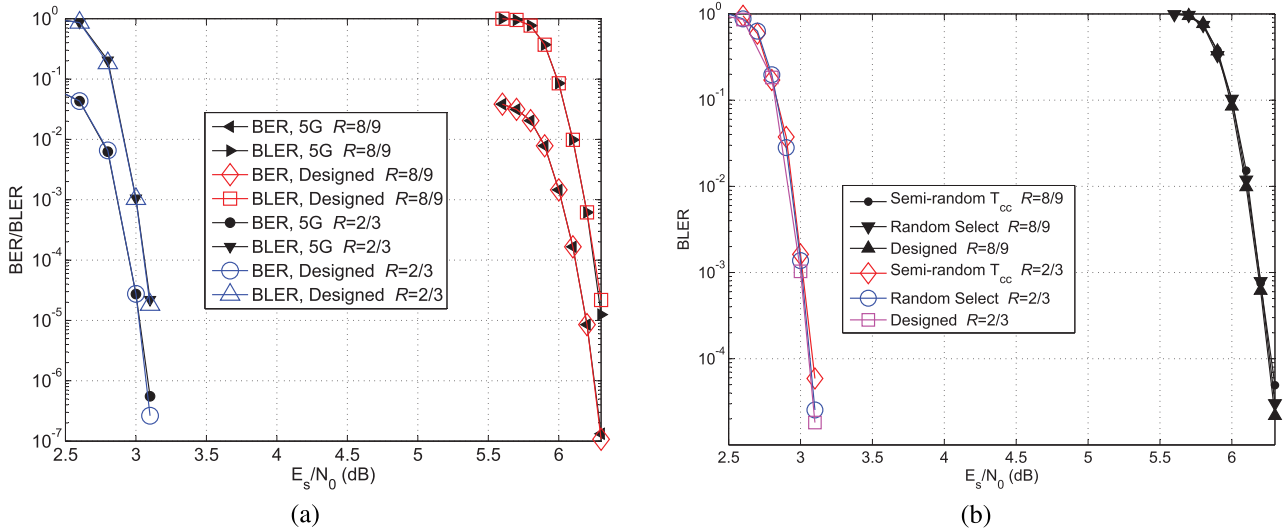


FIGURE 6. Information length K is set to 8192 and relevant lifting size is 384. (a) BLER performance of standard 5G LDPC codes and designed codes. (b) BLER performance of designed codes and other codes from simplistic constructions.

By using the proposed algebra-assisted method and following **Algorithm 3**, our designed exponent matrix for $\mathbf{B}_{5G,2}$ and lifting size set $\mathcal{S}_Z = \{3, 6, 12, 24, 48, 96, 192, 384\}$ in Table 3 has been accepted by 3GPP for 5G LDPC codes [38], [40]. In the next section, we will present an example for utilizing our method to design an exponent matrix for $\mathbf{B}_{5G,2}$ and same lifting size set, and compare the performance with the standard codes corresponding to same base matrix and lifting size set.

V. DESIGN EXAMPLES AND NUMERICAL RESULTS

In this section, in order to verify the effectiveness of the proposed WANC metric and algorithms in the last section, we give two numerical examples of code design, and provide simulation results for the resultant codes. In all simulations, the additive white Gaussian noise (AWGN) channel with quadrature phase shift keying (QPSK) modulation and the SPA with 50 iterations are assumed.

A. EXAMPLE 1: 5G LDPC CODES

By following **Algorithm 3**, we take $\mathbf{B}_{5G,1}$ of standard 5G LDPC codes as an example and try to design an exponent matrix, denoted by $\mathbf{P}_{2,des}$, for the second lifting size set. Based on Table 3, $N_Z = 8$ and $\mathcal{S}_Z = \{3, 6, 12, 24, 48, 96, 192, 384\}$. Assume the prime field \mathbb{F}_{769} and take the primitive element $\alpha = 11$. Then we construct the circulant coefficient table \mathbf{T}_{cc} of size 741×26 in the form of (15) with $\eta = 1$, $\mathcal{S}_{col} = \{\alpha^{11}, \alpha^{22}, \dots, \alpha^{286}\}$ and $\mathcal{S}_{row} = \{\alpha, \alpha^2, \dots, \alpha^{767}\} \setminus \mathcal{S}_{col}$. Cycle detection for $\mathbf{B}_{5G,1}$ shows that, for a given exponent matrix, there exist cycles of length 6 for all lifting sizes and cycles of length 4 for smaller lifting sizes. Thus, we set $\mathcal{N}_c = 3$. For $1 \leq i \leq 8$ and $1 \leq j \leq 3$, weights $\omega_Z(i)$ and $\omega_c(j)$ are assigned from experience as follows: $\omega_Z(1) = \omega_Z(2) = 4$, $\omega_Z(3) = \omega_Z(4) = \omega_Z(5) = 2$, $\omega_Z(6) = \omega_Z(7) = \omega_Z(8) = 1$; $\omega_c(1) = 10000$,

$\omega_c(2) = 100$, $\omega_c(3) = 1$. By applying **Algorithm 3**, we obtain $\mathbf{P}_{2,des}$ as shown in the Appendix. The subscript “des” stands for “designed”.

Fig. 6 shows the BER and BLER performance of standard 5G LDPC codes and our designed codes for $K = 8192$, and code rates $R = 2/3$ and $8/9$. Because information length K is fixed to 8192, lifting size is calculated as 384 by following the encoding method presented in Section III-C. We can see from the figure that performance curves of 5G standard codes and our designed codes almost overlap each other for various code rates down to BLER of 10^{-4} . To further show the feasibility and effectiveness of our method, we simulate another case of $K = 4096$, relevant lifting size $Z = 192$, and add another two scenarios of $R = 1/2$ and $R = 1/3$. As shown in Fig. 7, for the same code rates, the performance curves of the 5G standard codes and designed codes still almost overlap each other down to BLER of 10^{-4} . In Figs. 6 and 7, we also include two cases of simplistic constructions: 1) “Semi-random T_{cc} ” means that T_{cc} is constructed semi-randomly (i.e., the performance of largest lifting size is not too bad), which can be equivalent to generating the exponent matrix directly; 2) “Random Select” denotes that 46 rows are selected at random from T_{cc} to construct the exponent matrix. As shown in the figures, our designed codes have better performance than that of two simplistic constructions for various information lengths and code rates. These numerical results demonstrate that our proposed WANC metric and relevant algorithms are feasible. Moreover, performance comparison of two simplistic constructions shows that the utilized construction method for T_{cc} is effective.

To further verify the above observation, we follow the performance evaluation method of 3GPP and provide more simulations to compare the required signal-to-noise ratio (SNR) E_s/N_0 at BLER = 10^{-2} and BLER = 10^{-4} , respectively, for various information lengths and code rates.

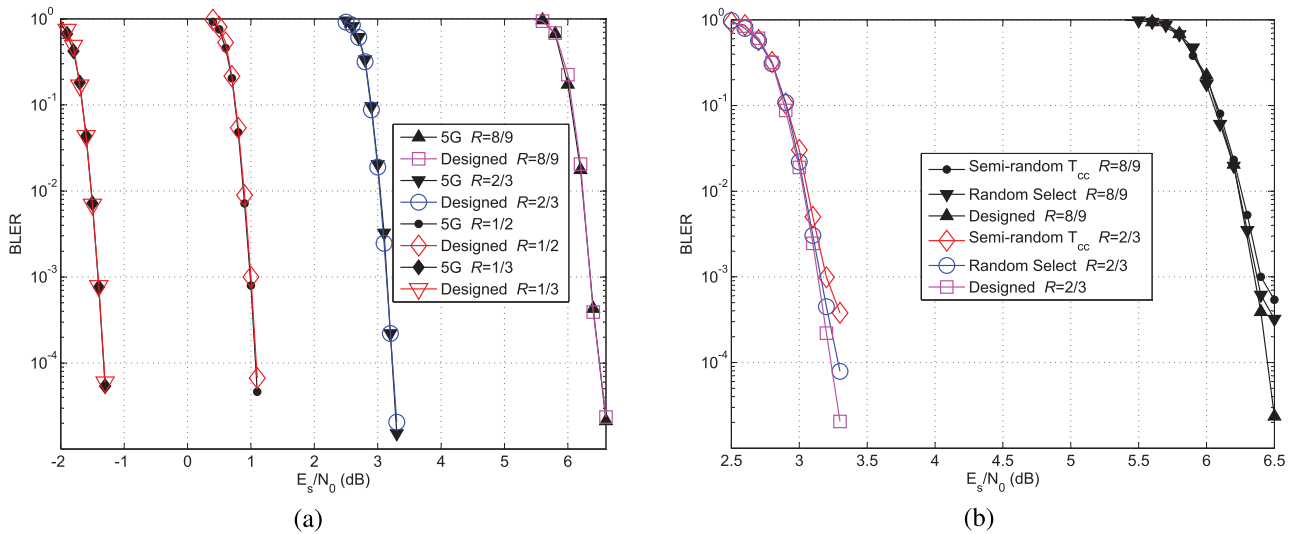


FIGURE 7. Information length K is set to 4096 and relevant lifting size is 192. (a) BLER performance of standard 5G LDPC codes and designed codes. (b) BLER performance of designed codes and other codes from simplistic constructions.

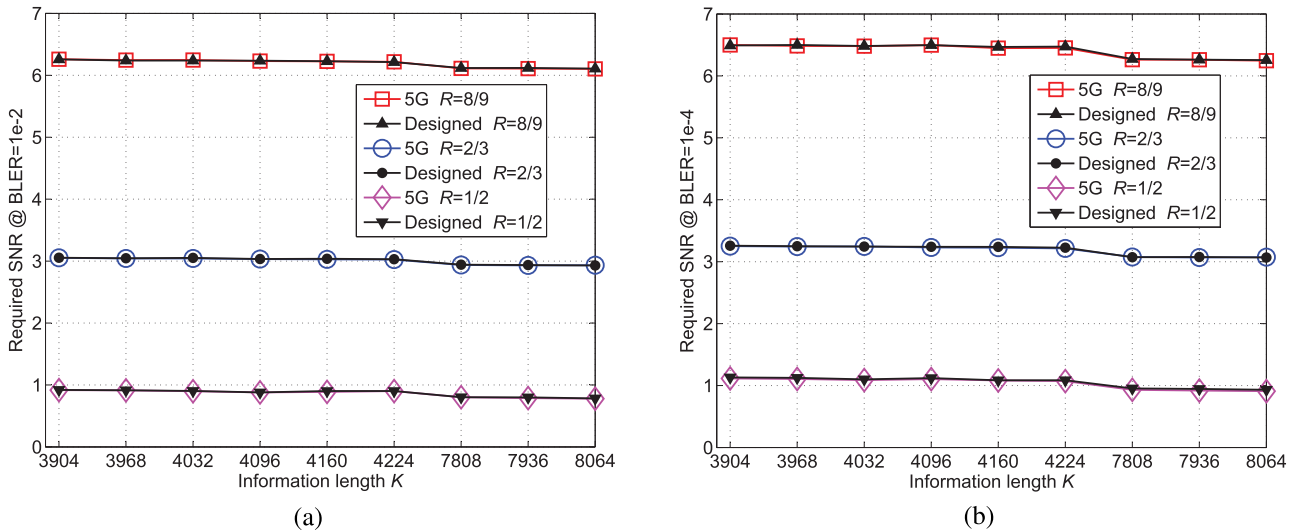


FIGURE 8. Required SNR at BLER = $10^{-2}/10^{-4}$ for various information lengths at code rates $R = 1/2, 2/3,$ and $8/9$. Lifting size for $K \in \{3904, 3968, 4032, 4096, 4160, 4224\}$ is 192 while that for $K \in \{7808, 7936, 8064\}$ is 384.

The results are depicted in Fig. 8. As shown in the figure, our designed codes have similar performance to standard 5G LDPC codes for different information lengths and code rates. This further confirms the feasibility and effectiveness of our proposed WANC metric and relevant algorithm.

It is worth mentioning that the search process does not need to run through all rows of the circulant coefficient table. For example, the time complexity of **Algorithm 3** mainly lies in the design of first 4 rows, i.e., the kernel. For $0 \leq i < 4$, r_i may not range from 0 to $N_t - 1$, which proves viable. Actually, we have dramatically narrowed the search ranges for the kernel while designing $\mathbf{P}_{2,des}$ (i.e., $0 \leq r_0, r_1 < 100, 100 \leq r_2 < 200,$ and $r_3 < N_t$), thus N_{search} in **Algorithm 3** is at most $100^3 \times N_t$, but we still can obtain a

considerable exponent matrix. If designed codes are required to support rate-compatible property, e.g., 5G LDPC codes, design complexity will be reduced significantly by utilizing row-by-row search as in **Algorithm 3**.

B. EXAMPLE 2: CCSDS LDPC CODES

Algorithm 3 just takes the base matrix of standard 5G LDPC codes as an example to show the feasibility and effectiveness of the proposed algebra-assisted method. In fact, our method is not limited to specific base matrix and can be used ingeniously for constructing other QC-LDPC codes to achieve improvement in some aspects, e.g., storage complexity. In this subsection, we take standard CCSDS LDPC codes as an example.

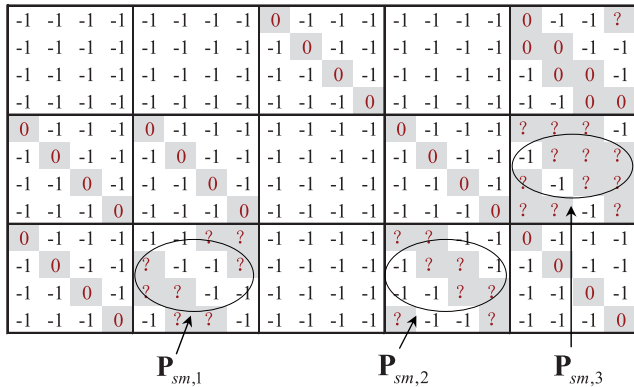


FIGURE 9. Exponent matrix of standard CCSDS LDPC codes for code rate $R = 1/2$. Non-specified shift values marked with “?” depend on the information lengths and are given in Table 4 for $K = 1024$ and $K = 4096$, respectively, more details can be found in [5]. The parity bits corresponding to last 4 circulant columns are punctured.

TABLE 4. Shift values in the cells marked with “?” from left to right in each row of exponent matrix in Fig. 9 for $K = 1024$ and $K = 4096$.

	$K = 1024$	$K = 4096$
0-th row	16	108
4-th row	103, 105, 0	126, 238, 481
5-th row	53, 74, 45	375, 436, 350
6-th row	89, 8, 119	263, 219, 16
7-th row	97, 112, 35	503, 388, 312
8-th row	50, 29, 115, 30	96, 28, 59, 225
9-th row	0, 47, 59, 102	84, 260, 318, 382
10-th row	31, 122, 1, 69	415, 403, 184, 279
11-th row	64, 93, 94, 99	48, 7, 328, 185

QC-LDPC codes with code rates $\{1/2, 2/3, 4/5\}$ and information lengths $\{1024, 4096, 16384\}$ are recommended by CCSDS for the deep space communication [5]. For these standard codes, each pair of parameters (K, R) is assigned with an exponent matrix, i.e., in total 9 exponent matrices are required. But such codes possess an amazing property: The codes with different information lengths but same code rate share a single base matrix. For example, Fig. 9 shows the exponent matrix of size 12×20 for $R = 1/2$, where the parity bits corresponding to last 4 circulant columns are punctured. Non-specified shift values marked with “?” in the figure depend on K and are given in Table 4 for two cases, i.e., $K = 1024$ and $K = 4096$. As shown in Fig. 9, we divide this matrix into 15 4×4 subblocks by bold line and find: shift values that vary with K are mainly distributed in three submatrices, denoted by $\mathbf{P}_{sm,1}$, $\mathbf{P}_{sm,2}$, and $\mathbf{P}_{sm,3}$, respectively. In the following, we will take the case of $R = 1/2$ as the example and desire to design an exponent matrix that can simultaneously fulfill ($K = 1024, Z = 128$) and ($K = 4096, Z = 512$). Before that, we first re-recognize the exponent matrix in Fig. 9 from the perspective of cycle structure:

1) No cycles of length 4 that are generated between $\mathbf{P}_{sm,1}$, $\mathbf{P}_{sm,2}$, and other two diagonal submatrices since $\mathbf{P}_{sm,1}$ and $\mathbf{P}_{sm,2}$ are disjoint;

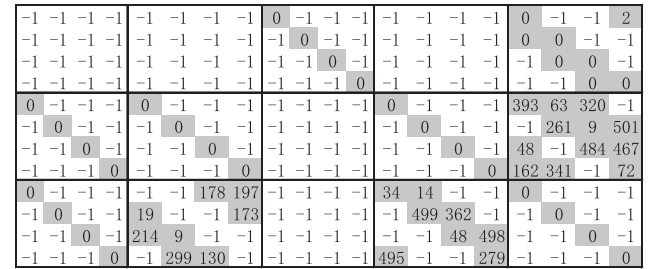


FIGURE 10. Designed exponent matrix $\mathbf{P}_{0.5,des}$ for the base matrix of standard CCSDS LDPC codes with $R = 1/2$ and $\mathcal{S}_Z = \{128, 512\}$.

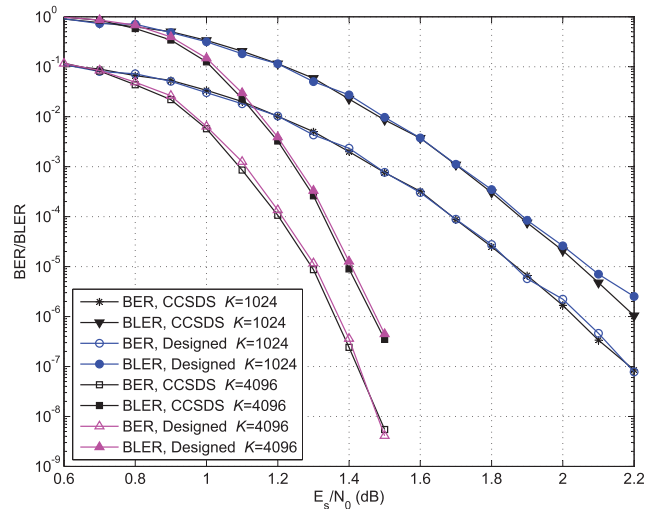


FIGURE 11. BER and BLER performance of standard CCSDS LDPC codes [5] and designed codes with $R = 1/2$ in case of $K = 1024$ and $K = 4096$. Lifting size for $K = 1024$ is 128 while that for $K = 4096$ is 512.

2) No cycles of length 4 that are generated between $\mathbf{P}_{sm,t} = [u_{i,j}]_{0 \leq i,j < 4}$ ($t = 1, 2$), $\mathbf{P}_{sm,3} = [p_{i,j}]_{0 \leq i,j < 4}$, and other two diagonal submatrices if and only if $u_{i,j} \neq p_{i,j}$ when both $u_{i,j}$ and $p_{i,j}$ are nonnegative;

3) No cycles of various lengths that are generated between the zero shift values of submatrix in the top right corner and $\mathbf{P}_{sm,3}$ if and only if all the nonnegative entries in the same row of $\mathbf{P}_{sm,3}$ are distinct;

4) **Constraint $\mathcal{C}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$:** for lifting size set \mathcal{S}_Z and $0 < q < \max(\mathcal{S}_Z)$, we say that q satisfies the constraint $\mathcal{C}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$ if from top to bottom

$$q \neq \begin{cases} \mathbf{P}_{sm,3}[u, 3] - \mathbf{P}_{sm,3}[u, 0], & u = 2, 3 \\ \mathbf{P}_{sm,3}[u, 3] - \mathbf{P}_{sm,3}[u, 1], & u = 1, 3 \pmod{Z \in \mathcal{S}_Z} \\ \mathbf{P}_{sm,3}[u, 3] - \mathbf{P}_{sm,3}[u, 2], & u = 1, 2 \end{cases}$$

We define $\mathbb{P}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$ as the set consisting of all values that satisfy $\mathcal{C}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$. Denote the non-specified shift value in the top right corner by $p_{0,end}$. If $p_{0,end} \in \mathbb{P}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$, then the cycles of various length generated between $p_{0,end}$ and $\mathbf{P}_{sm,3}$ can be avoided. We divide our design into three steps:

- 1) Jointly design $\mathbf{P}_{sm,1}$ and $\mathbf{P}_{sm,2}$;
- 2) Design $\mathbf{P}_{sm,3}$ with the constraint of $\mathbf{P}_{sm,1}$ and $\mathbf{P}_{sm,2}$;

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	65	66	67			
0	315	282	228	12	-1	155	133	-1	-1	278	81	76	125	31	-1	127	281	-1	289	335	359	356	1	0	-1	-1	-1	-1	-1	-1	-1	-1			
1	231	-1	110	123	119	348	-1	35	45	33	-1	337	225	-1	135	302	193	146	-1	122	-1	84	0	0	0	-1	-1	-1	-1	-1	-1	-1			
2	125	178	269	-1	354	275	260	357	152	113	278	-1	-1	146	192	236	-1	331	223	262	105	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1			
3	33	379	-1	219	164	-1	107	140	321	-1	246	311	355	20	296	-1	232	352	84	-1	266	112	1	-1	-1	0	-1	-1	-1	-1	-1	-1			
4	55	196	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	
5	345	259	-1	226	-1	-1	-1	-1	-1	-1	-1	-1	224	-1	-1	-1	118	-1	-1	-1	-1	111	125	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1	
6	208	-1	-1	-1	-1	372	-1	-1	-1	-1	125	40	-1	362	-1	-1	-1	20	74	-1	242	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	-1
7	5	372	-1	-1	223	-1	-1	159	7	-1	-1	-1	-1	225	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
8	35	297	-1	66	-1	-1	-1	-1	-1	-1	-1	-1	52	-1	-1	-1	53	-1	-1	265	-1	117	269	-1	145	-1	-1	-1	-1	-1	-1	-1	-1		
9	10	269	-1	-1	-1	-1	-1	-1	-1	-1	265	274	-1	200	-1	-1	-1	16	321	-1	365	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
10	-1	7	92	-1	367	-1	-1	136	372	-1	-1	-1	-1	94	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
11	92	358	-1	-1	-1	-1	-1	-1	-1	-1	-1	15	-1	-1	-1	365	-1	-1	-1	-1	215	231	55	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
12	219	34	-1	-1	-1	-1	-1	-1	-1	-1	16	139	-1	301	-1	-1	-1	287	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
13	69	-1	-1	353	-1	-1	-1	45	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	30	-1	-1	267	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
14	123	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	32	-1	-1	117	357	333	-1	-1	148	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
15	100	78	-1	-1	-1	-1	-1	-1	-1	-1	72	-1	-1	234	-1	-1	-1	92	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
16	-1	159	-1	39	-1	-1	-1	-1	-1	-1	321	-1	-1	-1	-1	-1	-1	-1	-1	301	-1	134	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
17	169	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	126	-1	-1	192	150	-1	-1	257	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
18	-1	344	-1	-1	-1	-1	-1	-1	-1	-1	-1	38	161	-1	-1	-1	-1	256	287	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
19	65	96	-1	-1	-1	-1	20	148	-1	271	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
20	247	-1	-1	236	-1	-1	-1	-1	-1	351	-1	335	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	36	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
21	-1	277	-1	-1	-1	117	-1	-1	-1	-1	-1	-1	-1	-1	-1	61	-1	-1	-1	353	44	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
22	194	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	157	196	-1	-1	-1	139	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
23	-1	25	190	-1	-1	-1	-1	-1	-1	135	-1	-1	-1	-1	-1	-1	-1	314	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
24	124	-1	-1	372	37	-1	-1	-1	-1	-1	247	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	96	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
25	-1	187	-1	-1	-1	-1	16	226	-1	-1	-1	-1	-1	6	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
26	137	-1	229	-1	128	-1	-1	-1	-1	-1	-1	-1	-1	-1	5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
27	-1	216	-1	-1	-1	23	-1	305	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
28	205	-1	-1	-1	89	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	253	-1	39	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
29	-1	77	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	140	-1	-1	-1	246	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
30	258	-1	-1	-1	-1	-1	-1	-1	184	-1	-1	296	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
31	-1	32	-1	-1	-1	-1	95	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	199	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
32	364	-1	-1	-1	-1	-1	-1	-1	-1	-1	87	-1	42	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
33	-1	343	185	-1	-1	-1	-1	-1	-1	223	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	362	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
34	374	-1	-1	-1	-1	-1	375	-1	-1	-1	-1	-1	214	-1	-1	60	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
35	-1	240	-1	-1	-1	44	-1	-1	-1	-1	90	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	35	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
36	344	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	35	44	-1	-1	268	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
37	-1	10	-1	-1	-1	-1	-1	-1	-1	-1	-1	123	-1	-1	-1	-1	-1	-1	-1	-1	-1	158	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
38	262	-1	-1	-1	-1	-1	-1	138	204	-1	77	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
39	-1	245	-1	288	-1	-1	321	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	258	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
40	304	-1	-1	-1	-1	-1	90	-1	-1	-1	-1	-1	-1	-1	-1	336	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
41	-1	270	-1	171	-1	-1	-1	276	-1	-1	-1	-1	-1	-1	-1	-1	65	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
42	185	-1	-1	117	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
43	-1	149	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	43	-1	251	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1			
44	293	-1	-1	-1	-1	-1	170	-1	80	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	34	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1		
45	-1	138	-1	-1	-1	-1	360	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0		

FIGURE 12. Designed exponent matrix $\mathbf{P}_{2,des}$ in case of $\mathbf{B}_{5G,1}$. Same as \mathbf{P}_2 of $\mathbf{B}_{5G,1}$ of standard 5G LDPC codes in [38], for $25 < j < 68$, the j -th column in $\mathbf{P}_{2,des}$ has single "0" at position $(j - 22)$.

- 3) Assign $p_{0,end}$ with any one of elements in $\mathbb{P}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$, where $\mathcal{S}_Z = \{128, 512\}$.

We formulate these steps as Algorithm 4.5

Two circulant coefficient tables, $\mathbf{T}_{cc,1}$ of size $N_{t,1} \times 8$ and $\mathbf{T}_{cc,2}$ of size $N_{t,2} \times 4$, are required. $\mathbf{T}_{cc,1}$ is used to jointly design $\mathbf{P}_{sm,1}$ and $\mathbf{P}_{sm,2}$ while $\mathbf{T}_{cc,2}$ is only used for designing $\mathbf{P}_{sm,3}$. $\mathbf{B}_{ccsds,0.5}$ and $\mathbf{B}_{sm,t}$ ($1 \leq t \leq 3$) represent the base matrices of the exponent matrix in Fig. 9 and submatrix $\mathbf{P}_{sm,t}$ ($1 \leq t \leq 3$), respectively. Let $\mathbf{B}_{sms,12} = [\mathbf{B}_{sm,1} \ \mathbf{B}_{sm,2}]$ and $\mathbf{M}_{over,r3} = \mathbf{B}_{sm,r} \otimes \mathbf{B}_{sm,3}$, where $r = 1$ and 2. For convenience, let $\mathcal{P}_q \triangleq \{\text{all primes in } \mathbb{F}_q\}$. Suppose the finite field \mathbb{F}_{509} with primitive element $\alpha = 2$ is used. We construct $\mathbf{T}_{cc,1}$ and $\mathbf{T}_{cc,2}$ as follows:

- 1) $\mathbf{T}_{cc,1}$ of size 88×8 in the form of (14) with $\eta_1 = \alpha^{119}$, $\mathcal{S}_{col,1} = \{\alpha^{24}, \alpha^{280}, \alpha^{355}, \alpha^{357}, \alpha^{394}, \alpha^{400}, \alpha^{405}, \alpha^{438}\}$, and $\mathcal{S}_{row,1} = \mathcal{P}_{509} \setminus \mathcal{S}_{col,1}$;

⁵In the algorithm, the subscripts "sms" and "over" stand for "submatrices" and "overlap", respectively.

- 2) $\mathbf{T}_{cc,2}$ of size 92×4 in the form of (14) with $\eta_2 = \alpha^9$, $\mathcal{S}_{col,2} = \{\alpha^{86}, \alpha^{236}, \alpha^{358}, \alpha^{405}\}$, and $\mathcal{S}_{row,2} = \mathcal{P}_{509} \setminus \mathcal{S}_{col,2}$.

We just test cycles of lengths 4, 6, and 8. Relevant weights are assigned as follows: $\omega_Z(1) = 2, \omega_Z(2) = 1; \omega_c(1) = 10000, \omega_c(2) = 100, \omega_c(3) = 1$. By applying Algorithm 4, we obtain $\mathbf{P}_{0.5,des}$ as shown in Fig. 10. Parity-check matrix for $K = 4096$ can be obtained directly by dispersing $\mathbf{P}_{0.5,des} = [p_{i,j}]_{0 \leq i < 12, 0 \leq j < 20}$ with lifting size 512 while that for $K = 1024$ is given by 128-fold CPM-dispersion of $\mathbf{P}'_{0.5,des} = [p'_{i,j}]_{0 \leq i < 12, 0 \leq j < 20}$, where $p'_{i,j} = -1$ if $p_{i,j} = -1$, and $p'_{i,j} = p_{i,j} \pmod{128}$ otherwise.

Algorithm 4 Exponent Matrix Construction for CCSDS LDPC Codes

Input: Circulant coefficient tables $\mathbf{T}_{cc,1}$ and $\mathbf{T}_{cc,2}$, base matrix $\mathbf{B}_{ccds,0.5}$, $\mathbf{B}_{sm,3}$, $\mathbf{B}_{sms,12}$, $\mathbf{M}_{over,13}$, $\mathbf{M}_{over,23}$, lifting size set $\mathcal{S}_Z = \{128, 512\}$, number of searches $N_{search,1}$ and $N_{search,2}$

Output: Exponent matrix $\mathbf{P}_{0.5,des}$ of size 12×20

Initialization: $\mathbf{P}_{0.5,des} = \mathbf{B}_{0.5,ccds} - \mathbf{1}$, $\mathbf{P}_{sms,12} = \mathbf{B}_{sms,12} - \mathbf{1}$, $\mathbf{M}_{12} = \mathbf{P}_{sms,12}$, $\mathbf{P}_{sm,3} = \mathbf{B}_{sm,3} - \mathbf{1}$, $\mathbf{M}_3 = \mathbf{P}_{sm,3}$

- 1: ****** Shift Value Selection for $\mathbf{P}_{sm,1}$ and $\mathbf{P}_{sm,2}$ ******
- 2: Calculate $\mathcal{W}^* = \mathcal{W}(\mathbf{P}_{sms,12})$ based on (7)
- 3: $count = 0$
- 4: **repeat**
- 5: Assign r_0, r_1, r_2, r_3 to distinct nonnegative integers less than $N_{t,1}$
- 6: **for** $i = 0 : 3$ **do**
- 7: $\mathbf{M}_{12}[i, :] = \mathbf{B}_{sms,12}[i, :] \otimes \mathbf{T}_{cc,1}[r_i, :] + \mathbf{B}_{sms,12}[i, :] - \mathbf{1}$
- 8: **end for**
- 9: Calculate $\mathcal{W}(\mathbf{M}_{12})$ based on (7)
- 10: **if** $\mathcal{W}(\mathbf{M}_{12}) < \mathcal{W}^*$ **then**
- 11: $\mathbf{P}_{sms,12} = \mathbf{M}_{12}$, $\mathcal{W}^* = \mathcal{W}(\mathbf{M}_{12})$
- 12: **end if**
- 13: $count = count + 1$
- 14: **until** $count > N_{search,1}$
- 15: $\mathbf{P}_{sm,1} = \mathbf{P}_{sms,12}[:, 0 : 3]$ and $\mathbf{P}_{sm,2} = \mathbf{P}_{sms,12}[:, 4 : 7]$
- 16: ****** Shift Value Selection for $\mathbf{P}_{sm,3}$ ******
- 17: $\mathbf{I}_1 = find(\mathbf{M}_{over,13} == 1)$, $\mathbf{I}_2 = find(\mathbf{M}_{over,23} == 1)$
- 18: Calculate $\mathcal{W}^* = \mathcal{W}(\mathbf{P}_{sm,3})$ based on (7)
- 19: $count = 0$
- 20: **repeat**
- 21: Assign r_0, r_1, r_2, r_3 to distinct nonnegative integers less than $N_{t,2}$
- 22: **for** $i = 0 : 3$ **do**
- 23: $\mathbf{M}_3[i, :] = \mathbf{B}_{sm,3}[i, :] \otimes \mathbf{T}_{cc,2}[r_i, :] + \mathbf{B}_{sm,3}[i, :] - \mathbf{1}$
- 24: **end for**
- 25: **if** $\mathbf{M}_3(\mathbf{I}_1) \neq \mathbf{P}_{sm,1}(\mathbf{I}_1)$ and $\mathbf{M}_3(\mathbf{I}_2) \neq \mathbf{P}_{sm,2}(\mathbf{I}_2)$ **then**
- 26: Calculate $\mathcal{W}(\mathbf{M}_3)$ based on (7)
- 27: **if** $\mathcal{W}(\mathbf{M}_3) < \mathcal{W}^*$ **then**
- 28: $\mathbf{P}_{sm,3} = \mathbf{M}_3$, $\mathcal{W}^* = \mathcal{W}(\mathbf{M}_3)$
- 29: **end if**
- 30: **end if**
- 31: $count = count + 1$
- 32: **until** $count > N_{search,2}$
- 33: ****** Shift Value Selection for $\mathbf{P}_{0.5,des}[0, end]$ ******
- 34: $\mathbf{P}_{0.5,des}[0, end]$ is randomly selected from $\mathbb{P}(\mathcal{S}_Z, \mathbf{P}_{sm,3})$
- 35: $\mathbf{P}_{0.5,des}[4 : 7, 16 : end] = \mathbf{P}_{sm,3}$, $\mathbf{P}_{0.5,des}[8 : end, 4 : 7] = \mathbf{P}_{sm,2}$, $\mathbf{P}_{0.5,des}[8 : end, 9 : 15] = \mathbf{P}_{sm,1}$
- 36: **return** $\mathbf{P}_{0.5,des}$

and 512 for $K = 4096$, respectively. As shown in the figure, our designed codes share similar performance to CCSDS LDPC codes for different scenarios down to BLER of

about 10^{-6} . Note that, for a given code rate of CCSDS LDPC codes, each information length owns one exponent matrix. But our design can simultaneously satisfy two information lengths (1024 and 4096) with a single exponent matrix. That is to say, our method is of significance at least in terms of storage complexity.

VI. CONCLUSION

QC-LDPC codes have been selected as the standard codes for 5G eMBB data channel recently. These codes have a fantastic property that each exponent matrix can support multiple lifting sizes, with which 5G LDPC codes can adapt various information lengths and code rates well while simultaneously utilizing shortening and puncturing techniques.

In this paper we consider the design of LDPC codes with these properties and have proposed construction methods. With the cycle analysis of such codes from the perspective of cycle structure, we proposed a metric and referred to it as *weighted average number of cycles* (WANC). Based on the proposed WANC metric, we developed a simple and practical algorithm to help select shift values for the exponent matrix of QC-LDPC codes that can support more than one lifting size and especially rate-compatible requirement. We finally provided several design examples and numerical results for 5G LDPC and CCSDS LDPC codes to demonstrate the feasibility and validity of our proposed WANC metric and designed algorithm.

By applying **Algorithm 3**, we design an exponent matrix, denoted as $\mathbf{P}_{2,des}$, for the second lifting size set of $\mathbf{B}_{5G,1}$ and show the resultant matrix in Fig. 12.

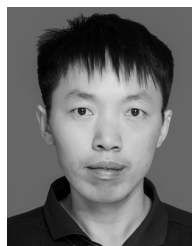
ACKNOWLEDGMENT

The authors would like to thank Professor Shu Lin for encouragement and helpful discussions on the algebraic construction of LDPC codes, and Dr. Jiaqing Wang and Dr. Shaohui Sun of CATT for their support and helpful discussions on 5G LDPC codes.

REFERENCES

- [1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [3] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. New York, NY, USA: Cambridge Univ. Press, 2009.
- [4] *Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications*, European Standard ETSI EN 302–307 V1.1.2, 2006.
- [5] *Low Density Parity Check Codes for Use in Near-Earth and Deep Space Applications*, document CCSDS 131.1-O-2, CCSDS, Sep. 2007. [Online]. Available: <http://public.ccsds.org/publications/SilverBooks.aspx>
- [6] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [7] L. Zeng, L. Lan, Y. Y. Tai, S. Song, S. Lin, and K. Abdel-Ghaffar, "Transactions papers—Constructions of nonbinary quasi-cyclic LDPC codes: A finite field approach," *IEEE Trans. Commun.*, vol. 56, no. 4, pp. 545–554, Apr. 2008.

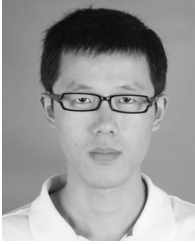
- [8] L. Zeng, L. Lan, Y. Y. Tai, B. Zhou, S. Lin, and K. A. S. Abdel-Ghaffar, "Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: A finite geometry approach," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 378–387, Mar. 2008.
- [9] B. Zhou, J. Kang, S. Song, S. Lin, K. Abdel-Ghaffar, and M. Xu, "Construction of non-binary quasi-cyclic LDPC codes by arrays and array dispersions," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1652–1662, Jun. 2009.
- [10] J. Li, K. Liu, S. Lin, and K. Abdel-Ghaffar, "A matrix-theoretic approach to the construction of non-binary quasi-cyclic LDPC codes," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1057–1068, Apr. 2015.
- [11] J. Li, S. Lin, K. Abdel-Ghaffar, W. Ryan, and D. J. Costello, Jr., *LDPC Code Designs, Constructions, and Unification*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [12] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.
- [13] I. E. Bocharova, B. D. Kudryashov, and R. Johannesson, "Searching for binary and nonbinary block and convolutional LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 163–183, Jan. 2016.
- [14] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
- [15] Q. Huang, Q. Diao, S. Lin, and K. Abdel-Ghaffar, "Cyclic and quasi-cyclic LDPC codes on constrained parity-check matrices and their trapping sets," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2648–2671, May 2012.
- [16] J. Li, K. Liu, S. Lin, and K. Abdel-Ghaffar, "Algebraic quasi-cyclic LDPC codes: Construction, low error-floor, large girth and a reduced-complexity decoding scheme," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 2626–2637, Aug. 2014.
- [17] H. Xu, D. Feng, R. Luo, and B. Bai, "Construction of quasi-cyclic LDPC codes via masking with successive cycle elimination," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2370–2373, Dec. 2016.
- [18] H. Xu and B. Bai, "Superposition construction of Q -ary LDPC codes by jointly optimizing girth and number of shortest cycles," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1285–1288, Jul. 2016.
- [19] X. Mu, C. Shen, and B. Bai, "A combined algebraic- and graph-based method for constructing structured RC-LDPC codes," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1273–1276, Jul. 2016.
- [20] X. Mu, G. Chen, and B. Bai, "An expurgating-based method for constructing multi-rate non-binary quasi-cyclic LDPC codes," *IEEE Commun. Lett.*, vol. 21, no. 8, pp. 1699–1702, Aug. 2017.
- [21] *Chairman's Notes of Agenda Item 7.1.5 Channel Coding and Modulation*, document R1-1613710, 3GPP, Nov. 2016. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_87/Docs/R1-1613710.zip
- [22] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.
- [23] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.
- [24] J. Li, K. Liu, S. Lin, K. Abdel-Ghaffar, and W. E. Ryan, "An unnoticed strong connection between algebraic-based and protograph-based LDPC codes, Part I: Binary case and interpretation," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Feb. 2015, pp. 36–45.
- [25] *Air Interface for Fixed Broadband Wireless Access Systems*, IEEE Standard P802.16e/D12, Oct. 2005.
- [26] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
- [27] *Study on Scenarios and Requirements for Next Generation Access Technologies*, document TR 38.913, 3GPP, Mar. 2016.
- [28] *Final Report of 3GPP TSG RAN1 #84bis V1.0.0*, document R1-163961, 3GPP, May 2016. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_85/Docs/R1-163961.zip
- [29] *Minimum Requirements Related to Technical Performance for IMT-2020 Radio Interface(s)*, document ITU-R M.[IMT-2020.TECH PERF REQ], Oct. 2016.
- [30] *Chairman's Notes of Agenda Item 8.1.4 Channel Coding*, document R1-1703980, 3GPP, Feb. 2017. [Online]. Available: http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_88/Docs/R1-1703980.zip
- [31] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.
- [32] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [33] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
- [34] T. Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May 2015.
- [35] *Chairman's Notes of Agenda Item 7.1.4 Channel Coding*, document R1-1709681, 3GPP, May 2017. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_89/Docs/R1-1709681.zip
- [36] *Chairman's Notes of Agenda Item 5.1.4 Channel Coding*, document R1-1711917, 3GPP, June 2017. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_AH/NR_AH_1706/Docs/R1-1711917.zip
- [37] *Chairman's Notes of Agenda Item 8.1.4 Channel Coding*, document R1-1706377, 3GPP, Apr. 2017. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_88b/Docs/R1-1706377.zip
- [38] *WF on LDPC Parity Check Matrices*, document R1-1711982, 3GPP, June 2017. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_AH/NR_AH_1706/Docs/R1-1711982.zip
- [39] *Padding for LDPC Codes*, document R1-1712254, 3GPP, Aug. 2017. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_90/Docs/R1-1712254.zip
- [40] *LDPC Design for eMBB Data*, document R1-1711711, 3GPP, June 2017. [Online]. Available: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1_AH/NR_AH_1706/Docs/R1-1711711.zip
- [41] C. Sun, H. Xu, D. Feng, and B. Bai, "(3, L) quasi-cyclic LDPC codes: Simplified exhaustive search and designs," in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process.*, Sep. 2016, pp. 271–275.
- [42] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 2007, pp. 3250–3254.
- [43] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.
- [44] J. Xu, L. Chen, L. Zeng, L. Lan, and S. Lin, "Construction of low-density parity-check codes by superposition," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 243–251, Feb. 2005.
- [45] T. J. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, Mar. 2018.



HUAN LI received the B.S. degree in electrical and information engineering from the Harbin Institute of Technology, Harbin, China, in 2013, and the M.S. degree in signal and information processing from Lanzhou University, Lanzhou, China, in 2016. He is currently pursuing the Ph.D. degree in communication and information system with the State Key Laboratory of Integrated Services Networks, Xidian University, China. His research interests include information and coding theory, and wireless communications.



BAOMING BAI (S'98–M'00) received the B.S. degree from the Northwest Telecommunications Engineering Institute, China, in 1987, and the M.S. and Ph.D. degrees in communication engineering from Xidian University, China, in 1990 and 2000, respectively. From 2000 to 2003, he was a Senior Research Assistant with the Department of Electronic Engineering, City University of Hong Kong. Since 2003, he has been with the State Key Laboratory of Integrated Services Networks, School of Telecommunication Engineering, Xidian University, where he is currently a Professor. In 2005, he was with the University of California at Davis, Davis, as a Visiting Scholar. His research interests include information theory and channel coding, wireless communication, and quantum communication.



XIJIN MU received the B.S., M.S., and Ph.D. degrees from Xidian University, China, in 2011, 2014, and 2017, respectively. He is currently a Post-Doctoral Fellow with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include information and coding theory, and wireless communications.



HENGZHOU XU received the B.S. and M.S. degrees from the School of Mathematics and Statistics, Zhengzhou University, China, in 2009 and 2013, respectively, and the Ph.D. degree in communication and information system from Xidian University, China, in 2017. He is currently a Lecturer with the School of Network Engineering, Zhoukou Normal University, Zhoukou, China. His research interests include information theory, channel coding, and combinatorial designs.

• • •



JI ZHANG received the B.S. and M.S. degrees from the School of Information Engineering, Henan University of Science and Technology, China, in 2005 and 2010, respectively. He is currently pursuing the Ph.D. degree in communication and information system with the State Key Laboratory of Integrated Services Networks, Xidian University, China. His research interests include information and coding theory, and wireless communications.