QC-LDPC Codes From Difference Matrices and Difference Covering Arrays

DIANE M. DONOVAN, ASHA RAO, (Member, IEEE), ELIF ÜSKÜPLÜ, AND E. ŞULE YAZICI

1The Centre for Plant Success in Nature and Agriculture, The University of Queensland, Brisbane, QLD 4072, Australia
2School of Science (Mathematical Sciences), RMIT University, Melbourne, VIC 3000, Australia
3Department of Mathematics, University of Southern California, University Park Campus, Los Angeles, CA 90089 USA
4Mathematics Department, Koç University, 34450 Istanbul, Turkey

Corresponding author: E. Şule Yazici (eyazici@ku.edu.tr)

The work of Diane M. Donovan was supported by the Australian Government through the Australian Research Council Centre of Excellence for Plant Success in Nature and Agriculture under Project CE200100015. The work of Elif Üsküplü was supported in part by NSF under Grant DMS-1902092, in part by the Army Research Office under Grant W911NF-20-1-0075, and in part by the Simons Foundation.

ABSTRACT We give a framework that generalizes LDPC code constructions using transversal designs or related structures such as mutually orthogonal Latin squares. Our constructions offer a broader range of code lengths and codes rates. Similar earlier constructions rely on the existence of finite fields of order a power of a prime, which significantly restricts the functionality of the resulting codes. In contrast, the LDPC codes constructed here are based on difference matrices and difference covering arrays, structures that are available for any order $a$, resulting in LDPC codes across a broader class of parameters, notably length $a(a-1)$, for all even $a$. Such values are not possible with earlier constructions, thus establishing the novelty of these new constructions. Specifically the codes constructed here satisfy the RC constraint and for $a$ odd, have length $a^2$ and rate $1-(4a-3)/a^2$, and for $a$ even, length $a^2-a$ and rate at least $1-(4a-6)/(a^2-a)$. When $3$ does not divide $a$, these LDPC codes have stopping distance at least $8$. When $a$ is odd and both $3$ and $5$ do not divide $a$, our construction delivers an infinite family of QC-LDPC codes with minimum distance at least $10$. We also determine lower bounds for the stopping distance of the code. Further we include simulation results illustrating the performance of our codes. The BER and FER performance of our codes over AWGN (via simulation) is at least equivalent to codes constructed previously.

INDEX TERMS LDPC codes, QC-LDPC codes, combinatorial constructions, difference matrices, difference covering arrays.

I. INTRODUCTION

The roll-out of smart devices for IoT and 5G networks necessitate the development of efficient techniques maximizing the integrity of data sent or received through open channels, where the data may be subject to distortion, attenuation and Gaussian noise. Error correction codes are being developed to meet these needs; codes designed to significantly enhance the reliability and integrity of transmitted data. While turbo codes have been implemented in smart 3G and 4G devices, the current demand for massive machine type communication, with ultra-reliability and low latency, is much higher, with 5G new radio (NR) requirements reaching through-puts of 5Gb/s. To meet this challenge researchers are investigating the use of LDPC (Low Density Parity Check) and polar codes, see [3], [25], [29]. In a 5G network, functionality requirements for control of both information and user data indicate the need for codes that support variable code rates and lengths [3]. In addition, storage and computational power can be restricted in modern smart devices, necessitating the development of codes based on low density or sparse cyclically generated parity-check matrices that can deliver low decoding complexity and enable parallelism in encoding and decoding. LDPC codes have been shown to meet these requirements by delivering effective tools compatible with 5G encoding and decoding, incorporating variable
code lengths and code rates to meet the demands of 5G user data [3].

Randomly constructed LDPC codes were first introduced by Gallager in 1962 [16] with MacKay and Neal later showing that these LDPC codes are able to achieve rates close to channel capacity [21]. However randomly generated LDPC codes can lead to high storage overheads with complex implementation routines. Thus there is a need for LDPC codes having a compact representation with low storage requirements, that also support efficient encoding and decoding algorithms [19]. To address this need, a number of authors, including [7], [20], [22], [28], [30], [32], [33], [39], have proposed constructing quasi-cyclic parity-check matrices for LDPC codes from combinatorial structures such as perfect cyclic difference sets, transversal designs, block designs or similar structures using elements of finite fields. However, the existence of these underlying algebraic and combinatorial structures is generally restricted for example, to orders $a$ of block length $m$ and columns $c$-wise w.r.t. $H$. In this paper, gains are made by developing a construction based on cyclically generated orthogonal Latin squares that works over the cyclic group of order $a$, where the operation is addition modulo $a$, exploiting the fact that cyclic groups exist for all orders $a$. The cyclic nature of the proposed construction provides for reduced storage and enables parallelism in encoding and decoding with increased options for code lengths and rates together with control over other code parameters such as girth and minimum distance.

Further flexibility is obtained by utilising the combinatorial properties of ubiquitous difference (covering) arrays, as opposed to, for example, less prevalent perfect cyclic difference sets [20] or transversal designs. If even greater flexibility is sought, a difference (covering) array may be defined over any abelian group. In addition, we show through simulations that this greater range of code lengths and rates is not at the expense of performance, with the constructed codes performing equal to or better than other codes constructed using similar constructions. It is this increase in functionality that validates the novelty of our new constructions.

We begin with the requisite coding theory definitions and background in the next section, going on to define difference matrices and difference covering arrays and the proposed constructions in Section III. Determination of rates and other properties for the LDPC and QC-LDPC (quasi-cyclic LDPC) codes constructed here are given in Section IV, and a performance analysis given in Section V is followed by the Conclusion.

II. BACKGROUND

We start with the preliminary definitions.

A $(m, w_c, w_r)$-regular binary LDPC code $C$ of block length $m$ is given by the null space of an $x \times m$ sparse $(0, 1)$ parity-check matrix $H = [H(i, j)]$, where both the row weight $w_r$ and column weight $w_c$ are constant, see [16]. Given a parity-check matrix $H = [H(i, j)]$, an $m$-tuple $\nu = (v_0, v_1, v_2, \ldots, v_{m-1})$ is a code word if and only if the syndrome $S$, shown in Equation (1), is the zero vector.

$$S = \begin{bmatrix} \sum_{j=0}^{m-1} H(i, j)v_j \mod 2 \end{bmatrix}, \quad \text{where } 0 \leq i \leq x - 1. \quad (1)$$

Note that since the parity-check matrix is binary, we work over $\mathbb{Z}_2$. Also in this paper the rows and columns of all $x \times m$ matrices will be indexed by the set $\{0, 1, \ldots, x - 1\}$ and $\{0, 1, \ldots, m - 1\}$ respectively.

If, after row reduction, the parity-check matrix can be written in the form

$$H = \begin{bmatrix} p^T | I_{m-k} \end{bmatrix}, \quad (2)$$

then low density manifests as $w_r \ll m$ and $w_c \ll m - k$. The rate of the code is defined to be $\kappa/m$. A parity-check matrix $H$ is said to satisfy the RC-constraint if the inner product of any two distinct rows and any two distinct columns is at most one. The distance of the code is taken to be the minimum Hamming distance between any two distinct code words. Since the code is linear and the zero vector is a code word, the distance of the code is equal to the minimum weight over all the non-zero code words.

In this paper, we seek to construct parity-check matrices that provide good variability in the code length and rate while maintaining the minimum distance of the code to be at least 8 and at least 10 for certain cases. For even $m$, this is achieved by relaxing the regularity condition.

We define an $(m, w_c, \{w_r - 1, w_r\})$-near regular binary LDPC code $C$ of block length $m$ as the null space of a sparse $(0, 1)$ parity-check matrix $H = [H(i, j)]$, with column weight $w_c$ and varying row weights $w_r - 1$ or $w_r$.

Example 1: The following matrix is an example of a parity-check matrix for a $(12, 4, \{4, 3\})$-near regular binary LDPC code. The length of the code is 12, the column weight is $w_c = 4$ and the row weight is $w_r = 3$ or 4. Minimum distance is 8. The rank of $H_{12}$ is 10 giving a code of rate $2/12 = 0.17$. This parity-check matrix satisfies the RC-constraint.

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$
In the above example the code length is relatively small, but the minimum distance is relatively high with respect to the code length $m$.

The construction proposed here follows the principles as set out in Gallager’s 1962 paper, [16], with the parity-check matrices for LDPC codes constructed by combining submatrices, with each column of each submatrix a cyclic shift of the previous. Gallager defined the first submatrix and then applied random permutations to the columns of this submatrix to obtain the remaining submatrices. However, the randomization of the submatrices increases the storage costs resulting in less memory-efficient codes, see [2]. To avoid these storage issues quasi-cyclic LDPC codes, or QC-LDPC codes, have been proposed.

The parity-check matrix $H$ for a QC-LDPC code can be written as a $K \times L$ array of $z \times z$ circulant matrices $H_{(i,j)}$, where each circulant $H_{(i,j)}$ for $1 \leq i \leq K$, $1 \leq j \leq L$, is a square matrix with each row a cyclic shift of the previous. Hence $H_{(i,j)}$ is the zero matrix, a circulant permutation matrix, or the sum of $1 \leq \lambda \leq z$ disjoint circulant permutation matrices. Adhering to the general framework as set out by Gallager in 1962, we will specify the parity-check matrix in terms of related submatrices.

Thus the general structure of the parity-check matrices is

$$
H = \begin{bmatrix}
H_{(1,1)} & H_{(1,2)} & \cdots & H_{(1,L)} \\
H_{(2,1)} & H_{(2,2)} & \cdots & H_{(2,L)} \\
\vdots & \vdots & & \vdots \\
H_{(K,1)} & H_{(K,2)} & \cdots & H_{(K,L)}
\end{bmatrix}
\quad (3)
$$

In some publications, this general format has been termed array-based binary LDPC codes, see, for instance, [13], [35] and the introduction to [40], [41]. Historically, algebraic or combinatorial techniques have been used to specify the submatrices $H_{(i,j)}$ with this compact mathematical representation enhancing the encoding algorithms and minimizing the storage requirements, while maintaining low computational complexity when implemented [20].

In this paper, we will first define our codes to be quasi-cyclic “like” in that, cyclic shifts of any code word within each subblock will also be a code word. Then we will show that some infinite subclasses of these parity-check matrices can be rearranged using row and column permutations to obtain the quasi-cyclic form. The constructed codes with quasi-cyclic structure will be examples of codes with Tanner graphs that are cyclic liftings of fully connected base graphs of size $4 \times a$ with a lifting factor of $a$. Refer to [31] and the references therein for definitions and related results.

To this end, let $a > 3$ be a positive integer. Define $H$ to be a $(4a) \times (a^2)$ matrix

$$
H = \begin{bmatrix}
R_0 & R_1 & \cdots & R_{a-1} \\
\Psi_{0,0} & \Psi_{0,1} & \cdots & \Psi_{0,a-1} \\
\Psi_{1,0} & \Psi_{1,1} & \cdots & \Psi_{1,a-1} \\
\Psi_{2,0} & \Psi_{2,1} & \cdots & \Psi_{2,a-1}
\end{bmatrix}
\quad (4)
$$

where

- $R_v = [R_v(i,j)]$ is taken to be an $a \times a$ square matrix with row $v$ the vector of all one’s and every other row the vector of all zeros.
- For $0 \leq a \leq 2$ and $0 \leq v \leq a-1$, $\Psi_{u,v}$ is taken to be a permutation of the $a \times a$ identity matrix, denoted $I$.

Provided the inner product of any two distinct columns and any two distinct rows of $H$ is at most $1$, $H$ satisfies the RC-constraint and can be taken as a parity-check matrix for an $(a^2, a)$-regular binary LDPC code.

In Section III, we show that for all odd $a > 3$, difference matrices can be used to construct parity-check matrices (as described in Equation (4)) and hence codes satisfying the RC-constraint. The specifications of these parity-check matrices, in terms of circulant submatrices, result in reduced storage requirements. Further, since $w_r = w \ll a^2$ and $w_r = 4 \ll 4a$, these matrices are sparse, leading to reduced decoding complexity.

The removal of any of the $(4a) \times a$ submatrices of $H$ does not affect the RC-constraint. Thus, for any $\rho \in \{0, \ldots, a-1\}$, we may define $\overline{H}$ to be a $(4a-1) \times (a^2 - a)$ matrix of the form

$$
\overline{H} = \begin{bmatrix}
R_0 & R_1 & \cdots & R_{\rho-1} & R_{\rho+1} & \cdots & R_{a-1} \\
\Psi_{0,0} & \Psi_{0,1} & \cdots & \Psi_{0,\rho-1} & \Psi_{0,\rho+1} & \cdots & \Psi_{0,a-1} \\
\Psi_{1,0} & \Psi_{1,1} & \cdots & \Psi_{1,\rho-1} & \Psi_{1,\rho+1} & \cdots & \Psi_{1,a-1} \\
\Psi_{2,0} & \Psi_{2,1} & \cdots & \Psi_{2,\rho-1} & \Psi_{2,\rho+1} & \cdots & \Psi_{2,a-1}
\end{bmatrix}
\quad (5)
$$

where the row $\rho$ of all zeros is deleted (Note that $\overline{H}$ does not include submatrices $R_\rho$ and $\Psi_{\rho,\rho}$). Then under the assumption that the inner product of any two distinct columns and any two distinct rows of $\overline{H}$ is at most $1$, $\overline{H}$ can be taken as a parity-check matrix for an $(a^2 - a, a-1, a)$-regular binary LDPC code that satisfies the RC-constraint.

The parity-check matrix $\overline{H}_{12}$ given in Example I provides an example of a parity-check matrix ($w_r = 4$ and $w_r \in [3, 4]$) constructed in this manner. Figure I provides an illustration (blue representing 1, otherwise 0) of the general form of such matrices and is an example of a parity-check matrix for a $(650, 4, [25, 26])$-near regular binary LDPC code. It will also be shown in Section III that for all even $a > 2$, difference covering arrays can be used to construct parity-check matrices (as described in Equation (5)) and codes satisfying the RC-constraint. As before, the specification of sparse parity-check matrices will result in reduced storage requirements and reduced decoding complexity.

Before we give these constructions, it is useful to note that parity-check matrices can be visualised as graphs, with the rows of the parity-check matrix associated with a set,
Let $C = \{c_0, c_1, \ldots, c_{m-k-1}\}$, of parity-check nodes and columns with a set $B = \{b_0, b_1, \ldots, b_{m-1}\}$, of bits or variable nodes. Then, the parity-check matrix $H = [H(i,j)]$ gives the Tanner graph, $G(H)$, with vertex set $C \cup B$ and an edge from $c_j \in C$ to $b_j \in B$ if and only if $H(i,j) = 1$. As stated in [17] and recently in [26], the bit error performance (BER) of LDPC decoding, using the Sum-Product Algorithm (SPA), is affected by cycles of short length in the Tanner graph. It can be shown that a parity-check matrix $H$ satisfies the RC-constraint if and only if all columns in the Tanner graph have length greater than $4$, implying that the girth of the Tanner graph is at least $6$, see [39].

Another factor affecting the performance of a code is its stopping distance. A stopping set, $S$, is a subset of the set of variable nodes $B$ in $G(H)$, such that all neighbors of vertices in $S$ are adjacent to at least two vertices of $S$. In terms of the parity-check matrix $H$, a stopping set $S$ of size $\ell$ is a subset of the columns of $H$ satisfying the property that the induced $(m-k) \times \ell$ submatrix $H$ has row sum either $0$ or at least $2$, for all $m-k$ rows.

The existence of small stopping sets can adversely affect the performance of an LDPC code, with decoding failure caused when certain variable nodes are affected by errors after transmission. Thus the existence of small stopping sets can greatly reduce a code’s error correcting capability. Stopping sets were first described in 2002 by Di et al. [10], when they were researching the average erasure probabilities of bits and blocks over a binary erasure channel (BEC). See [10], [18] for more details on stopping sets. Let $S$ denote the collection of all stopping sets in a Tanner graph, $G(H)$. Define the stopping distance, $s^*$, of $G(H)$ as the size of the smallest, non-empty stopping set in $S$. It is known that the stopping distance of a code aids in the analysis of the code’s error floor (an abrupt change in error rate curves arising from iterative decoding) and that the performance of an LDPC code over the BEC is dominated by the small stopping sets in the Tanner graph [23]. The larger the stopping distance, the lower the error floor of the code. Also if a set of columns of the parity-check matrix is linearly dependent, then the corresponding vertices in the Tanner graph should have even degree in the induced subgraph. Thus the stopping distance provides a lower bound for the minimum distance of the code.

We use difference covering arrays (DCA) and difference matrices (DM) (as defined in Section III) for our constructions of the parity-check matrices. These arrays can also be used to construct orthogonal Latin squares and nearly-orthogonal Latin squares (see [6] for related definitions). Article [18] lays the framework for using a full set of orthogonal Latin squares (equivalently transversal designs) to construct parity-check matrices for binary-LDPC codes. However, this analysis considers only orthogonal Latin squares that are constructed using finite fields, which exist only for a power of a prime. In [14] authors calculate the stopping distance of SA-LDPC codes constructed by inflating transversal designs of prime order, hence only obtaining codes of length a power of a prime. In [12] partially balanced designs obtained from difference covering arrays of order $2n$ were used to construct regular quasi-cyclic codes with constant column weight $3$ and minimum distance $6$ when $n$ is an odd integer.

The novelty and new work in the current paper significantly extends this work by generalizing the ideas to obtain quasi-cyclic-like codes based on structures of both odd and even orders. This fact underlines the versatility of our constructions, in that they have the potential to be applied in the construction of other array-based codes, for instance, the construction of non-Binary LDPC codes as presented in [40], with further ideas developed in [41]. We prove some tight lower bounds for the stopping distance and minimum distance of the constructed codes. We show that all stopping sets of the constructed codes are of size at least $8$ when $a$ is not divisible by $3$. Furthermore, we present examples where our simulation results illustrate that the stopping distance is $10$. More importantly we analyze the minimum distance of the codes constructed here and prove that a large infinite family (more specifically when the smallest prime dividing $a$ is greater than $5$) of these codes have minimum distance $10$ and are quasi-cyclic in structure.

On the other hand, our construction is closely related to other well-known constructions in the literature, that focus on finding codes with large girth. If the $R_i$’s (defined in Equation 4) are removed the resulting parity check matrix will be equivalent to those in QC-LDPC codes that are cyclic liftings of fully connected base graphs of type $(3, a)$ with minimum lifting factor for girth $6$. These codes are classified for small $a$ in [31]. See [12], [24] and [15] for related work and definitions. Article [1] classifies codes that are liftings of fully connected base graphs of type $(4, a)$ with minimum lifting factor for girth $6$ and $5 \leq a \leq 11$. When the smallest prime dividing $a$ is greater than $5$. The new constructions provided here deliver an infinite family of quasi-cyclic codes constructed with girth $6$, minimum distance $10$ and are examples of codes constructed by liftings of fully connected base graphs of type $(4, a)$ with lifting degree $a$ which is the smallest possible lifting degree.

### III. DIFFERENCE MATRICES AND DIFFERENCE COVERING ARRAYS

In this section, the parity-check matrices for regular and near regular binary LDPC codes are constructed using difference matrices and difference covering arrays, respectively.

A difference matrix, $DM(k; a)$, is defined to be an $a \times k$ array $D = [D(i,j)]$, where

- all entries in the first column of $D$ are $0$ and all remaining columns contain each entry $0, \ldots, a-1$ precisely once, and
- for all pairs of distinct columns, $j$ and $j'$, the differences $D(i,j) - D(i,j') \mod a$, for $0 \leq i \leq a-1$, are distinct; that is, $\{D(i,j) - D(i,j') \mod a \mid 0 \leq i \leq a-1\} = \{0, \ldots, a-1\}$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Difference matrices are well studied in the literature, see [6] for constructions. It can be shown that difference matrices with more than 2 columns do not exist for even orders, but that DM(3; a) difference matrices exist for all odd a. (See [6] Section VI.17). Further, for positive integer n, a DM(3; 2n+1) corresponds to an additive permutation, with the numbers of distinct DM(3; 2n + 1) corresponding to the sequence A002047 in Sloane’s encyclopedia [27] and current enumeration putting the number of distinct additive permutations for 2n + 1 = 23 (distinct DM(3; 23)) at 577, 386, 122, 880.

It is clear from the definition, that permuting rows does not change the underlying properties of a difference matrix. Hence we will assume that all difference matrices are in the standard form, namely D = D(1, 1 = i, for all rows 0 ≤ i ≤ a − 1.

Example 2: D5 is an example of a DM(3; 5), whereas D7 is a DM(4; 7). Notice that the property above is satisfied, as, for instance, the set of differences between the second and third columns in D5, is {0, 1, 2, −2, −1} which equals {0, 1, 2, 3, 4} when working modulo 5.

\[
D_5 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 2 \\
0 & 2 & 4 \\
0 & 3 & 1 \\
0 & 4 & 3
\end{bmatrix}, \quad D_7 = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 1 & 2 & 3 \\
0 & 2 & 4 & 6 \\
0 & 3 & 6 & 2 \\
0 & 4 & 1 & 5 \\
0 & 5 & 3 & 1 \\
0 & 6 & 5 & 4
\end{bmatrix}.
\]

Let I denote the a × a identity matrix, and P\ell be a circulant permutation matrix (CPM) obtained from I by cyclically shifting its rows \( i \) positions to the left. We set \( P^1 = P \) and \( P^0 = I \). Note that row \( r \) of \( P^0 \) has precisely one entry equal to 1 at column \( r \mod a \) and all other entries equal to 0. In Construction 1 below, it is demonstrated that, for all odd integers a, a DM(3; a) can be used to construct a matrix \( H = [H(i, j)] \), of the form given in Equation (4), that is a parity check matrix for a \((a^2, 4, a)\)-regular binary LDPC code.

Construction 1: Let a ≥ 3 be an odd positive integer and D = [D(i, j)] be a DM(3; a) in standard form. Construct H as given in Equation (4) where for \( u = 0, 1, 2 \) and \( v = 0, \ldots, a - 1 \), the matrices \( R_u = [R_u(i, j)] \) and \( \Psi_{u,v} \) satisfy

\[
R_u(i, j) = \begin{cases} 
1, & \text{for } i = v \text{ and } j = 0, \ldots, a - 1, \\
0, & \text{otherwise},
\end{cases}
\]

\[
\Psi_{u,v} = \begin{cases} 
R_u(0,0) = P^0 = I, & \text{if } u = 0, \\
R_u(1,1) = P^v, & \text{if } u = 1, \\
R_u(2,2) = P^w, & \text{if } u = 2.
\end{cases}
\]

We then have

\[
H = \begin{bmatrix}
R_0 & R_1 & \ldots & R_{a-1} \\
I & 1 & \ldots & I \\
I & P^1 & \ldots & P^{a-1} \\
P^{0(2)} & P^{D(1,2)} & \ldots & P^{D(a-1,2)}
\end{bmatrix}.
\]

The novelty arises from the simplicity of the construction and the fact that the underlying combinatorial structure is the cyclic group of odd order, and thus, the binary operation is addition modulo \( a \). These two facts allow us to both verify that the RC-constraint is satisfied as well as analytically determine bounds for the minimum distance of the code, the rate of the code and the size of the minimum stopping set, as shown below. In addition, with regards to storage requirements, it is only necessary to store the DM(3; a). The entries in row \( v \) of this array DM(3; a) then determine the non-zero entries in the first column of each of the \( a \times a \) submatrices \( R_v \) and \( \Psi_{u,v} \), for \( v = 0, \ldots, a - 1 \), (more precisely, the non-zero entries of \( \Psi_{1,v} \) and \( \Psi_{2,v} \)), with all remaining columns of \( \Psi_{1,v} \) and \( \Psi_{2,v} \) taken as cyclic shifts of the first column.

Furthermore, for \( a \geq 5 \) we give a family of DM(3; a), resulting in a parity-check matrix in quasi-cyclic form after row and column permutations. These features greatly enhance applicability of the resulting \((a^2, 4, a)\)-regular binary LDPC codes.

In addition, Construction 1 can be generalised and the existence of a DM(k; a) used to construct a \((a^2, k + 1, a)\)-regular binary LDPC code, but the existence of such DM(k; a)'s is not known for all admissible a. In particular, as stated above, difference matrices do not exit for even order a. However, for even order we are able to adapt the above construction using the next best structure, namely difference covering arrays where we cover as many differences as possible. It is this adaption that forms one of the main innovations of this paper. It delivers a new construction with novelty validated by increased variability in code lengths and rates, lengths and rates that were not possible before. In the second construction, the resulting parity-check matrix takes a similar form, namely the form given in Equation (5) where \( \overline{H} \) is similar to \( H \) (Equation (8)) except that a \( (4a) \times a \) submatrix has been removed, as well as a row of all zeros.

We start with verifying the properties necessary to show that, when \( a \) is odd, the \((a^2, 4, a)\)-regular binary LDPC code satisfies the RC-constraint. This argument is then extended to \( a \) even, and Construction 2 (page 52147) is used to obtain an \((a^2 - a, 4, (a - 1, a))\)-near regular binary LDPC code that satisfies the RC-constraint.

In what follows, let \( a \geq 3 \) be odd and \( D = [D(i, j)] \) be a DM(3; a) in standard form. Take \( H \) to be a (0, 1) matrix constructed as in Construction 1 using a DM(3; a). First, we give some straightforward observations that will come in useful for later proofs.

Lemma 1: For any \( b \in \{0, \ldots, a^2 - 1\} \), column \( b \) has sum 4. Further, given \( x < y < z < t \) such that \( H(x, b) = H(y, b) = H(z, b) = H(t, b) = 1 \), then \( 0 \leq x \leq a - 1, a \leq y \leq 2a - 1, 2a \leq z \leq 3a - 1 \) and \( 3a \leq t \leq 4a - 1 \), with \( y \equiv q \mod a, z = q + D(x, 1) \equiv q + t \mod a \) and \( t \equiv q + D(x, 2) \mod a, \) for some \( 0 \leq q \leq a - 1 \).

Below are some notations that will be used in later proofs. The rows of \( H \) are partitioned into four subsets denoted \( V_{\overline{1}} = \{0, \ldots, a - 1\} \), \( V_0 = \{a, \ldots, 2a - 1\} \), \( V_1 = \{2a, \ldots, 3a - 1\} \) and \( V_2 = \{3a, \ldots, 4a - 1\} \), that is, respectively, the rows of \( R_v \) and \( \Psi_{u,v}, u = 0, 1, 2 \). Further for each...
column $b \in \{0, \ldots, a^2 - 1\}$, define the set of rows

$$C_b = \{ x, y, z, t \}$$

$$\mid H(x, b) = H(y, b) = H(z, b) = H(t, b) = 1 \}. \quad (9)$$

Thus $C_b$ gives the set of rows of $H$ with entry 1 in column $b$.

Consequently,

$$C_{x+a+q} = \{ x, q + a, (q + D(x, 1) \mod a) + 2a, (q + D(x, 2) \mod a) + 3a \} = \{ x, z, t \}.$$ 

for $0 \leq x, q \leq a - 1$.

**Lemma 2:** The inner product of any two distinct rows of $H$ is at most one.

**Proof:** Since $P_{x,v}$ is a permutation of the identity matrix for $u = 0, 1, 2$ and $v = 0, \ldots, a - 1$, it follows immediately that

(a) The inner product of any two distinct rows of $R_v$ or $P_{u,v}$ is zero. Hence, the inner product of any two distinct rows of $H$ in the same subset $V_i$ is 0 for all $-1 \leq i \leq 2$.

(b) The inner product of any row of $P_{u,v}$ with any row of $R_v$ is at most one. Furthermore, for any row in $V_{-1}$, there is precisely one $v$ such that row $r$ of $R_v$ is not the zero vector. Hence, the inner product of any row of $V_{-1}$ with any row of $V_i$ is exactly one for $0 \leq i \leq 2$.

(c) Finally, assume that there exists rows $r$ and $r'$ and distinct $v$ and $v'$ such that the inner product of rows $r$ and $r'$ of $P_{u,v}$ and $P_{u,v'}$ is equal to one or is the inner product of rows $r$ and $r'$ of $P_{i,v}$ and $P_{j,v'}$, for some $0 \leq i < j \leq 2$. This implies that

$$r - D(v, i) \mod a = r' - D(v, j) \mod a$$

Rearranging the difference of these equations gives

$$D(v, i) - D(v, j) = D(v', i) - D(v', j) \mod a,$$

which contradicts the definition of a $DM(3; a)$. Hence the inner product of any two rows $r \in V_i$ and $r' \in V_j$, for $0 \leq i < j \leq 2$, is at most one. On the other hand $r$ and $r'$ both contain one in the same column when $r = r + D(v, i) - D(v, j) \mod a$. Therefore the inner product of any two rows $r \in V_i$ and $r' \in V_j$, for $0 \leq i < j \leq 2$, is exactly 1.

**Lemma 3:** Let $a \geq 3$ be odd. Then $H$ is a parity-check matrix for an $(a^2, 4, a)$--regular binary LDPC code that satisfies the RC-constraint.

**Proof:** The fact that the inner product of any two distinct rows of $H$ is at most one follows from Lemma 2.

Now consider any two distinct columns of the parity check matrix, $b = va + q$ and $b' = v'a + q'$, where $0 \leq v, v', q, q' \leq a - 1$ and $b' > b$.

By definition the set of rows that contain 1 in the columns $b$ and $b'$ is given by $C_b$ and $C_{b'}$, respectively. The inner product of columns $b$ and $b'$ is given by $|C_b \cap C_{b'}|$. By Equation (10), $C_b = C_{a+q} = \{ q + a, (q + D(v, 1) \mod a) + 2a, (q + D(v, 2) \mod a) + 3a \}$ and $C_{b'} = C_{a'+q'} = \{ q' + a, (q' + D(v', 1) \mod a) + 2a, (q' + D(v', 2) \mod a) + 3a \}$.

We then have the following cases:

Case 1: If $v = v'$, since $b$ and $b'$ are distinct columns, $q \neq q'$. We have:

$$q + a \neq q' + a,$$

$$q + D(v, 1) \neq q' + D(v', 1) \mod a$$

$$q + D(v, 2) \neq q' + D(v', 2) \mod a.$$

Hence the inner product of columns $b$ and $b'$ is 1.

Case 2: If $v \neq v'$ then $D(v, 1) \neq D(v', 1)$ and $D(v, 2) \neq D(v', 2)$.

Case 2.1 If $q = q'$ then $a + q = a + q'$, $q + D(v, 1) \neq q' + D(v', 1) \mod a$ and $q + D(v, 2) \neq q' + D(v', 2) \mod a$ and the inner product of columns $b$ and $b'$ is 1.

Case 2.2 If $q \neq q'$ then $a + q = a + q'$. Now assume

$$(q + D(v, 1) \mod a) + 2a = (q' + D(v', 1) \mod a) + 2a,$$

$$(q + D(v, 2) \mod a) + 3a = (q' + D(v', 2) \mod a) + 3a.$$ 

This implies $q - q' = (D(v', 1) - D(v, 1) = D(v', 2) - D(v, 2)$. Hence $D(v, 1) - D(v, 2) = D(v', 1) - D(v', 2)$ which contradicts the definition of a $DM(3; a)$. Therefore the inner product of columns $b$ and $b'$ is at most 1.

Thus the inner product of any pair of distinct columns of $H$, is at most one, as required. Hence $H$ satisfies the RC-constraint.

Let $n$ be a positive integer. We now present a second construction replacing the DM(3; 2n + 1) with a difference covering array DCA(3; 2n) that exists for all $n \geq 2$.

A difference covering array, DCA(k; 2n), is defined to be a $2n \times k$ array $D = \{D(i, j)\}$, where

- all entries in the first column of $D$ are 0 and the remaining columns contain each entry 0, 1, $\ldots$, $2n - 1$ precisely once, and

- for all pairs of distinct non-zero columns, $j$ and $j'$, the differences $D(i, j) - D(i, j') \mod 2n$, for $0 \leq i \leq 2n - 1$, are non-zero and cover the set $\{1, 2, \ldots, 2n - 1\}$.

Similar to difference matrices we will assume that all difference covering arrays studied here are in standard form with $D(i, 1) = i$, for all $0 \leq i \leq 2n - 1$.

**Example 3:** $D_4$ is an example of a DCA(3; 4), whereas $D_6$ is a DCA(4; 6). Notice that the first property above is satisfied and, for instance, the set of differences between the second and third columns in $D_4$ is $\{-1, -2, 2, 1\}$ which equals $\{3, 2, 2, 1\}$ when working modulo 4.

$$D_4 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix}, \quad D_6 = \begin{bmatrix} 0 & 0 & 1 & 3 \\ 0 & 1 & 3 & 0 \\ 0 & 2 & 5 & 4 \\ 0 & 3 & 0 & 1 \\ 0 & 4 & 2 & 5 \\ 0 & 5 & 4 & 2 \end{bmatrix}.$$

In a difference covering array there are $2n$ rows and, since the $2n$ differences $D(i, j) - D(i, j')$ are non-zero, it can be shown that for any pair of distinct non-zero columns $j$ and $j'$ there exists two rows $r_0$ and $r_1$ such that $D(r_0, j) - D(r_0, j') = n = D(r_1, j) - D(r_1, j')$. That is, the repeated difference is $n$ (see [8]
for a proof). While not a lot is known for general $k$, when $k = 3$ it is known that as $n$ grows the number of distinct DCA($3; 2n$) grows significantly, see [9]. For further results on difference covering arrays see [38].

Construction 2: Let $a \geq 4$ be an even positive integer, $D = [D(i, j)]$ be a DCA($3; a$), where $r_0$ represents precisely one of the two rows where $D(r_0, 2) - D(r_0, 1) = n$. Then construct $\overline{P}$ as given in Equation (5) where, for $u = 0, 1, 2, v = 0, \ldots, a - 1$ and $v \neq r_0$, the matrices $R_u = [R_u(i, j)]$ and $\overline{P}_{u,v}$ are as given below and the row $r_0$ of all zeros has been removed.

$$R_v(i, j) = \begin{cases} 1, & \text{for } i = v \text{ and } j = 0, \ldots, a - 1, \\ 0, & \text{otherwise}, \end{cases}$$

(13)

$$\overline{P}_{u,v} = \begin{cases} P_{D(v,0)} = I, & \text{if } u = 0, \\ P_{D(v,1)} = P^v, & \text{if } u = 1, \\ P_{D(v,2)} & \text{if } u = 2. \end{cases}$$

(14)

Example 4: Take $D_4$ as set out in Example 3 with $a = 4$ and $r_0 = 2$. Then Construction 2 gives the parity-check matrix displayed in Example 1.

Lemma 4: Let $a \geq 4$ be even, then $\overline{P}$ given in Construction 2 is a parity-check matrix for an $(a^2 - a, 4, [a - 1, a])$–near regular binary LDPC code that satisfies the RC-constraint.

The proof, for $a$ even, follows as in the proof for Lemma 3 (a odd) where the row and column ranges have been relabeled appropriately.

We now examine the properties of the above codes and get exact bounds for the rate of the code as well as the minimum distance of a particular class of these codes. Further we note that our simulations show that the rank of $\overline{P}$ is exactly $4a - 6$ for all even $a \leq 200$.

Next, recalling that for any matrix $A$ with $m$ columns the rank($A$) + nullity($A$) = $m$, when $a$ is odd, the dimension of the code with parity check matrix $H$ equals $a^2 - 4a + 3$, while, for $a$ even, the dimension of the code with parity check matrix $\overline{P}$ is greater than or equal to $a^2 - 5a + 6$. Thus we verify that the rate of the $(a^2, 4, a)$–regular binary LDPC code in Construction 1 is equal to Identity (15) (below) while the rate of the $(a^2 - a, 4, [a - 1, a])$–near regular binary LDPC code in Construction 2 is greater than or equal to Identity (16):

$$\text{Rate} = 1 - \frac{(4a - 3)}{a^2}, \quad \text{for } a \text{ odd } (H),$$

(15)

$$\text{Rate} = 1 - \frac{(4a - 6)}{a^2 - a}, \quad \text{for } a \text{ even } (\overline{P}).$$

(16)

Lemma 6: Let $a \geq 3$ be odd, then the rank of the matrix $H$ is exactly $4a - 3$.

Proof: While this is an important result, the proof involves basic but tedious computation. Consequently, the proof has been moved to Appendix A.

Lemma 7: Let $a \geq 4$ be even, then the rank of the matrix $\overline{P}$ is at most $4a - 6$.

### Table 1. The rates of some LDPC codes constructed in this paper for \(a \geq 12\).

| \(a\) odd | Code Length \(m\) | Code Dimension | Code Rate |
|-----------|------------------|----------------|-----------|
| 13        | 169              | 120            | 0.71      |
| 15        | 225              | 168            | 0.75      |
| 17        | 289              | 224            | 0.78      |
| 19        | 361              | 288            | 0.80      |
| 21        | 441              | 360            | 0.82      |
| 23        | 529              | 440            | 0.83      |
| 25        | 625              | 528            | 0.84      |
| 27        | 729              | 624            | 0.86      |
| 29        | 841              | 728            | 0.87      |
| 31        | 1521             | 1368           | 0.90      |

| \(a\) even | Code Length \(m\) | Lower Bound Code Dimension | Lower Bound Code Rate |
|------------|-------------------|---------------------------|----------------------|
| 12         | 132              | 90                        | 0.68                 |
| 14         | 182              | 132                       | 0.72                 |
| 16         | 240              | 182                       | 0.76                 |
| 18         | 306              | 240                       | 0.78                 |
| 20         | 380              | 306                       | 0.81                 |
| 22         | 462              | 380                       | 0.82                 |
| 24         | 552              | 462                       | 0.84                 |
| 26         | 650              | 552                       | 0.85                 |
| 28         | 756              | 650                       | 0.86                 |
| 30         | 870              | 756                       | 0.87                 |
Proof: While this is an important result, the proof involves basic but tedious computation. Consequently, this proof has been moved to Appendix A.

The above results lead directly to the bound on the rate of the codes as stated below.

Lemma 8: For odd a, the \((a^2, 4, a)\)-regular binary LDPC code has rate 1 \(- (4a - 3)/a^2\) and, for even a, the \((a^2 - a, 4, (a - 1, a))\)-near regular binary LDPC code has rate at least 1 \(- (4a - 6)/(a^2 - a)\).

Proof: Recall that the nullity(\(H\)) \(= m - \text{rank}(H)\). Hence, Lemma 6 and 7 imply that for odd a with \(m = a^2\), the rate of the code is exactly \(\frac{a^2 - 4a + 3}{a^2} = 1 - \frac{4a - 3}{a^2}\) and, for even a with \(m = a^2 - a\), at least \(\frac{a^2 - 5a + 6}{a^2 - a} = 1 - \frac{4a - 6}{a^2 - a}\), respectively.

To investigate possible stopping sets for our codes we exploit the facts that the parity-check matrix \(H\) has column weight 4, satisfies the RC-constraint, and that for each column \(b_i\) we may define sets \(C_{b_i} = \{x_1, y_1, z_1, t_i\}\) as in Equation (10) where \(x_i \in X, y_i \in Y, z_i \in Z\) and \(t_i \in T\) for disjoint sets \(X, Y, Z, T\).

Furthermore, an exhaustive computer search of the possible sets of \(C_i\)'s satisfying the above conditions shows that, up to isomorphism [18] there are no possible stopping sets of size 7. Analysing size 6 it can be shown that there only two possible stopping sets, \(S_1 = \{b_1, b_2, b_3, b_4, b_5, b_6\}\) and \(S_2 = \{b_1, b_2, b_3, b_4, b_5, b_6\}\), occurring as subsets of columns of parity-check matrices with the above properties. Next, it can be shown analytically that there exist only 2 non-isomorphic cases involving only 2 different \(x_i\)'s (case \(S_1\)) and 3 different \(x_i\)'s (case \(S_2\)). The \(y_i\)'s, \(z_i\)'s and \(t_i\)'s can then be uniquely determined up-to-isomorphism. This analysis shows \(S_1\) and \(S_2\) take the following forms:

\[
S_1 = \{C_{b_1} = \{x_1, y_1, z_1, t_1\}, C_{b_2} = \{x_1, y_2, z_2, t_2\}, C_{b_3} = \{x_2, y_1, z_1, t_3\}, C_{b_4} = \{x_2, y_2, z_2, t_3\}, C_{b_5} = \{x_3, y_1, z_2, t_1\}, C_{b_6} = \{x_3, y_2, z_1, t_2\}\},
\]

\[
S_2 = \{C_{b_1} = \{x_1, y_1, z_1, t_1\}, C_{b_2} = \{x_1, y_2, z_2, t_2\}, C_{b_3} = \{x_2, y_1, z_2, t_3\}, C_{b_4} = \{x_2, y_2, z_1, t_3\}, C_{b_5} = \{x_3, y_2, z_2, t_1\}, C_{b_6} = \{x_3, y_3, z_1, t_2\}\}.
\]

(17)

The above simplifications allow us to provide a relatively simple proof to establish a minimum stopping distance of at least 8, leading to bounds on the minimum distance.

Lemma 9: Suppose 3 \(\nmid a\), then the constructed \((a^2, 4, a)\)-regular and \((a^2 - a, 4, (a - 1, a))\)-near regular binary LDPC codes have stopping distance at least 8.

Proof: We need to demonstrate that \(S_1\) and \(S_2\) given above do not occur in the corresponding LDPC code.

First assume that there exist columns \(b_1, b_2, b_3, b_4, b_5, b_6\) of \(H\) such that \(C_{b_1}, C_{b_2}, C_{b_3}, C_{b_4}, C_{b_5}, C_{b_6}\) take the form given in \(S_1\). Then, by Lemma 1, \(D(x_1, 1) + y_1 \equiv z_1 mod a\) for \(1 \leq i \leq 3\), \(D(x_2, 1) + y_1 \equiv z_2 mod a\), \(D(x_1, 1) + y_2 \equiv z_3 mod a\) and \(D(x_2, 1) + y_3 \equiv z_1 mod a\). Summing these equivalences implies \(3D(x_1, 1) \equiv 3D(x_2, 1) mod a\). If \(3 \nmid a\) then \(D(x_1, 1) = D(x_2, 1)\), leading to a contradiction.

Next, assume that there exists columns \(b_1, b_2, b_3, b_4, b_5, b_6\) of \(H\) such that \(C_{b_1}, C_{b_2}, C_{b_3}, C_{b_4}, C_{b_5}, C_{b_6}\) take the form given in \(S_2\). Then, by Lemma 1, \(D(x_1, 1) + y_1 \equiv D(x_1, 1) + y_2 mod a\), \(D(x_2, 1) + y_1 \equiv D(x_1, 1) + y_2 mod a\), \(D(x_3, 1) + y_1 \equiv D(x_2, 1) + y_3 mod a\). Summing these equivalences we obtain \(2y_1 \equiv 2y_2 mod a\). Similarly we have \(t_1 - y_1 \equiv t_2 - y_2 mod a\), \(t_3 - y_1 \equiv t_1 - y_3 mod a\) and \(t_2 - y_3 \equiv t_3 - y_2 mod a\). Summing these equivalences we obtain \(2y_1 \equiv 2y_2 \equiv 2y_3 mod a\), giving us \(2y_1 \equiv 2y_2 \equiv 2y_3 mod a\). This is a contradiction since it implies \(y_1 = y_2, y_1 = y_3 \) or \(y_2 = y_3\).

\[\square\]

**Corollary 1:** The constructed \((a^2, 4, a)\)-regular and the \((a^2 - a, 4, (a - 1, a))\)-near regular binary LDPC codes have minimum distance at least 8.

Proof: By definition, every set of linearly dependent columns sums to zero modulo 2, leading to this set of columns intersecting any row in an even number of ones (possibly zero ones). Hence, any set of linearly dependent columns forms a stopping set, with the stopping distance of these LDPC codes being a lower bound for the minimum distance. If \(3 \nmid a\), then the stopping distance of these LDPC codes is at least 8 by Lemma 9, implying the minimum distance of the LDPC code is at least 8. If \(3 \mid a\) then, by the proof of Lemma 9, \(S_2\) cannot be a subset of the columns of the parity-check matrix of the LDPC codes. Furthermore, \(S_1\) cannot define a linearly independent set as \(S_1\) intersects rows \(x_1\) and \(x_2\) an odd number of times. Hence the minimum distance of the code is also at least 8 when \(3 \mid a\).

\[\square\]

A computer search shows that, for \(a \leq 26\), all classified non-isomorphic DCA(3; a) constructed in [8] produce \((a^2 - a, 4, (a - 1, a))\)-near regular binary LDPC codes that have minimum distance 8.

It is also possible to prove that the infinite family of LDPC codes constructed from the DCA(3; a) given by Equation (18), have minimum distance 8.

Let \(a\) be even. It is known that the matrix \(D = [D(j, g)]\), where

\[
D(j, g) = \begin{cases} 
0, & \text{if } g = 0, \\
1, & \text{if } g = 1, \\
2j + 1, & \text{for } 0 \leq j \leq \frac{a}{2} - 1, \\
2(j - \frac{a}{2}), & \text{for } \frac{a}{2} \leq j \leq a - 1, 
\end{cases}
\]

forms a DCA(3; a).

Lemma 10: Suppose \(D = [D(i, j)]\) is the DCA(3; a) given by Equation (18). Then the \((a^2 - a, 4, (a - 1, a))\)-near regular binary LDPC codes with parity-check matrices \(\overline{H}\) has minimum distance 8.

Proof: We know by Corollary 1 that the minimum distance is at least 8. It is also easy to see that columns of \(\overline{H}\) corresponding to the blocks given below sum to zero mod 2 and thus form a linearly dependent set of columns.

\[C_{b_1} = \{0, a + 1, 2a + 1, 3a + 2\},\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
$C_{b_2} = \{0, 3a/2 - 2, 5a/2 - 2, 7a/2 - 1\}$,
$C_{b_1} = \{1, 3a/2 - 2, 5a/2 - 1, 7a/2 + 1\}$,
$C_{b_6} = \{1, 2a - 1, 2a, 3a + 2\}$,
$C_{b_5} = \{a/2 - 2, a + 1, 5a/2 - 1, 4a - 2\}$,
$C_{b_4} = \{a/2 - 2, 3a/2 + 2, 2a, 7a/2 - 1\}$,
$C_{b_3} = \{a/2 - 1, 3a/2 + 2, 2a + 1, 7a/2 + 1\}$,
$C_{b_2} = \{a/2 - 1, 2a - 1, 5a/2 - 2, 4a - 2\}$,

where $b_1 = 1$, $b_2 = \frac{a}{2} - 2$, $b_3 = \frac{3a}{2} - 2$, $b_4 = 2a - 1$, $b_5 = \frac{a}{2} - 1 + \alpha$, $b_6 = \frac{a}{2} - 2 + \frac{\alpha}{2} + 2$, $b_7 = \frac{a}{2} - 1 + \frac{\alpha}{2} + 2$, $b_8 = \frac{a}{2} - 1 + a - 1$. Hence, there exists a code word of weight 8. □

Thus we have established the following theorem:

**Theorem 1:** Let $H$ be the parity-check matrix based on the DCA($3; a$) given by Equation (18). Then $H$ is the parity-check matrix of an $(a^2 - a, 4, \{a - 1, a\})$-- near regular binary LDPC code of length $a^2 - a$, girth at least 6, rate at least $1 - (4a - 6)/(a^2 - a)$ and minimum distance 8.

Let $a$ be odd. It can be shown that for $k = 3$ and $\alpha$ satisfying $\gcd(\alpha, a) = 1$ and $\gcd(\alpha - 1, a) = 1$, the array $D = [D(j, g)]$, where

$$D(j, g) = \begin{cases} 0, & \text{if } g = 0, \\ j, & \text{if } g = 1, \\ \alpha j, & \text{if } g = 2, \end{cases} \tag{19}$$

forms a DM($3; a$) in standard form. Choosing $\alpha = 2$ will give a DM($3; a$) for all odd $a$. But for better performance, we will assume that $3 \not| a, 5 \not| a$ and choose $\alpha = (a - 1)/2$. When $3 \not| a, 5 \not| a$ and $\alpha = (a - 1)/2$ the constructed codes have improved stopping distance and minimum distance.

In what follows, let $a > 3$ be an odd integer and $H$ the parity-check matrix based on the DCA($3; a$) given by Equation (19) with $\alpha = \frac{a - 1}{2}$ where $\gcd(a, 3) = 1$, and $\gcd(a, 5) = 1$.

**Lemma 11:** The $(a^2, 4, a)$--regular binary LDPC code with parity-check matrix $H$ has minimum distance 10.

**Proof:** We show that the minimum distance of this code is at most 10, by finding a set of 10 columns that are linearly dependent. Note that, the minimum distance of a code equals the smallest number of linearly dependent columns.

Let $\alpha = (a - 1)/2$ and $\alpha^{-1}$ be the multiplicative inverse of $\alpha$ mod $a$ such that $\alpha \alpha^{-1} \equiv 1 \mod a$. Observe that $\alpha^{-1} \not\equiv 1 \mod a$ and $\alpha^{-1} \not\equiv 2 \mod a$. Now it can be seen that the columns of $H$ corresponding to the blocks given below sum to zero mod 2 and thus form a linearly dependent set of columns.

$C_{b_6} = \{\alpha^{-1}, a + a + 1, 2a + \alpha^{-1} + a + 1, 3a + a + 1\}$,
$C_{b_5} = \{\alpha^{-1} + 1, a, 2a + \alpha^{-1} + 1, 3a + \alpha + 1\}$,
$C_{b_4} = \{\alpha^{-1} + 1, a + a + 1, 2a + \alpha^{-1} + 1 + 3a\}$,

$C_{b_3} = \{\alpha^{-1}, a + a + 1, 2a + \alpha^{-1} + a + 1, 3a + 2\}$,

where $b_1 = 2, b_2 = b_3 = a + 2, b_4 = a + a, b_5 = 2a, b_6 = 2a + 1, b_7 = \alpha^{-a} + a + 1, b_8 = \alpha^{-a} + a + 1, b_9 = (a^{-1} + 1)a$ and $b_{10} = (a^{-1} + 1)a + \alpha$. Hence, there exists a code word of weight 10. Refer to Equations 10 and 19 to calculate the sets $C_i$.

The rest of the proof which shows the minimum distance is at least 10 is given in Appendix A □

**Lemma 12:** There exists $H^* \subset H$ obtained from $H$ by row and column permutations that is the parity-check matrix of a $(a^2, 4, a)$--regular binary QC-LDPC code.

**Proof:** The proof is given in Appendix A. □

Combining these results we obtain the following lemma.

**Lemma 13:** $H$ is the parity-check matrix of a $(a^2, 4, a)$--regular binary QC-LDPC code with minimum distance 10.

The above Lemmas can be combined to prove the following theorem.

**Theorem 2:** Assume $a$ is an odd positive integer satisfying $\gcd(a, 3) = 1$ and $\gcd(a, 5) = 1$. Let $H$ be the parity-check matrix based on the DCA($3; a$) given by Equation (19), where $\alpha = (a - 1)/2$. Then $H$ is the parity-check matrix of an $(a^2, 4, a)$--regular binary QC-LDPC code of length $a^2$, girth at least 6, rate equal to $1 - \frac{4a - 3}{a^2}$ and minimum distance 10.

The matrix $H^*$ given below, is the general quasi-cyclic form of the parity check matrix for the code obtained in the above theorem.

$$H^* = \begin{bmatrix}
I & I & I & \cdots & I \\
I & p^1 & p^2 & \cdots & p^{a-1} \\
I & p^{2a-1} & p^{2a-1} & \cdots & p^{(a-1)s-1} \\
I & p^{(a+1)} & p^{2(a+1)-1} & \cdots & p^{(a-1)s(a+1)-1}
\end{bmatrix}. \tag{20}$$

**V. PERFORMANCE ANALYSIS**

In this section we present the error correcting performance of the proposed LDPC codes via simulations. We then compare the performance of the codes constructed in this paper with a number of codes found in the literature, including from [12], [13], [18], [33], [35]. Simulations can provide another indicator for the performance of an LDPC code, in addition to the code rate and minimum distance. We conduct simulations over two different channels, AWGN and BEC. For clarity, we label the codes compared using the quadruple $[m, R, w_c, w_r]$ consisting of code length $m$, code rate $R$, column weight $w_c$ and row weight $w_r$, with the last column of Table 2 given the source of the codes from the literature. For irregular or near-regular LDPC codes, the row weight is given as $a - 1$, or taken as an average and indicated by $\sim$. Also to reduce rebiase we refer to an $(a^2, 4, a)$--regular
TABLE 2. LDPC codes that were compared for BER and FER performance.

| No. | LDPC Code Type | [n_v, k_v, w_v, w_v] | Min. Distance | Reference |
|-----|---------------|---------------------|---------------|-----------|
| 1   | PREV(3, 34)   | [1122, 0, 91, 3, 33] | 6             | [12]      |
| 2   | PREV(3, 44)   | [1892, 0, 95, 3, 43] | 4             | [12]      |
| 3   | DM(3; 43)     | [1892, 0, 91, 4, 43] | 10            | *         |
| 4   | DCA(3, 44)    | [1892, 0, 91, 4, 43] | 8             | *         |
| 5   | Array         | [1849, 0, 91, 4, 43] | 10 ≤ d ≤ 12   | [13]      |
| 6   | TD-LDPC       | [1849, 0, 91, 4, 43] | d ≥ 8         | [18]      |
| 7   | Lattice       | [1849, 0, 91, 4, 43] | 8 ≤ d ≤ 12    | [33]      |
| 8   | SCB-LDPC      | [1849, 0, 91, 4, 43] | 10 ≤ d ≤ 12   | [35]      |

* = Current paper

binary QC-LDPC code constructed in Construction 1 using a DM(3; a) as a DM(3; a) code, and an (a² - a, 4, {a - 1, a})-near regular binary LDPC code constructed in Construction 2 using a DCA(3; a) as a DCA(3; a) code. Table 2 gives the list of codes that are used for comparison purposes, for both AWGN and BEC channels. Figures 6-12 in Appendix B, illustrate the parity-check matrices.

The codes listed in Table 2 were chosen as they all have similar parameters, namely column weight 4, with rates in the range 0.91 to 0.93. In addition, this presented the opportunity to compare codes within a very narrow range of lengths. The only exceptions are codes (1) and (2) in Table 2, which are constructed in [12] also using DCA’s. These codes have column weight 3, and for comparison purposes, we chose two examples, the first example (1) with similar code rate as code (3), while the other (2) with similar code length as code (4). Similar to the codes proposed in this paper, the transversal design code (termed TD-LDPC) as constructed in [18] and the code constructed using lattices (termed Lattice) as in [33] (codes (6) and (7) in Table 2), are structured LDPC codes, namely, they are constructed using certain combinatorial designs, finite geometries, etc. We also simulated other structured LDPC codes, including the array-based LDPC codes (termed Array) presented in [13], as well as an enhanced version of an array-based code (termed SCB-LDPC) described in [35]. The latter code utilizes a specific row-selection function, [0,1,3,4], to prevent certain absorbing sets [11] and to enhance the code’s performance, as detailed on page 2309 of [35]. The simulations were performed via an open-source library called AFF3CT [4], a toolbox dedicated to forward error correction, written in C++.

A. THE ADDITIVE WHITE GAUSSIAN NOISE CHANNEL (AWGN)

The analysis was performed through the transmission of the zero vector using binary-phase shift-key (BPSK) modulation over varying signal-to-noise ratios (SNRs, Eb/No) assuming transmission over AWGN channel. Since only binary messages are being transmitted, we chose the zero vector, which allows errors to be added randomly across the entire vector. We used the belief propagation (BP) based decoding algorithm given in [36] with the sum-product algorithm (SPA) implementation. The procedure is iterated until the zero vector is obtained or a maximum number of iterations (100) is reached. Also, at each SNR level, we monitor the analysis until it reaches 50 wrongly decoded vectors. Figures 2 and 3 illustrate the decoding BER and FER performance of the codes in Table 2.

As can be observed from Figures 2 and 3, the performance of the DM(3; 43) code is better than the performance of the DCA(3; 44) code at higher Eb/No values. Theorem 1 and 2 ensure that the minimum distances of the DM(3; 43) code and the DCA(3; 44) code are 10 and 8, respectively. Therefore, such a difference in their performance can be expected.

The difference in performance between the old design (codes (1) and (2)) and the new design (codes (3) and (4)) is noticeably big. Given that the minimum distances of the PREV(3; 34) code and the PREV(3; 44) code are 6 and 4, respectively, it would be fair to say that this new design is one step ahead.

The other four structured codes, namely Array, TD-LDPC, Lattice, and SCB-LDPC, have exactly the same parameters as the DM(3; 43) code, and Figures 7, 9, 10, 11, and 12 show that they are similar in the sparsity pattern of their matrices. Thus, they perform similarly as expected. As proposed in [35], an improvement in array-based codes can boost their performance at higher Eb/No values, which is corroborated by our results. Table 3 provides details on the number of decoded...
TABLE 3. Comparison of the five [1849, 0.91, 4, 43] codes at Eb/No = 5.5.

| LDPC Code Type | Number of decoded vectors | Number of wrong bits |
|----------------|---------------------------|----------------------|
| DM(3; 43)      | 3.48e + 08                | 415                  |
| Array          | 3.54e + 08                | 427                  |
| TD-LDPC        | 3.62e + 08                | 419                  |
| Lattice        | 3.52e + 08                | 427                  |
| SCB-LDPC       | 4.29e + 08                | 397                  |

FIGURE 4. BER comparison of the codes in Table 2 over BEC.

FIGURE 5. FER comparison of the codes in Table 2 over BEC.

TABLE 4. Comparison of the five [1849, 0.91, 4, 43] codes at EP = 0.04.

| LDPC Code Type | Number of decoded vectors | Number of wrong bits |
|----------------|---------------------------|----------------------|
| DM(3; 43)      | 2.06e + 09                | 1683                 |
| Array          | 1.65e + 09                | 1682                 |
| TD-LDPC        | 1.74e + 09                | 1616                 |
| Lattice        | 1.66e + 09                | 1682                 |
| SCB-LDPC       | 2.02e + 09                | 1611                 |

vectors at Eb/No = 5.5 required to reach 50 incorrectly decoded vectors and the number of erroneously decoded bits for these five codes. As anticipated, SCB-LDPC code outperforms the others, albeit marginally. However, our proposed codes still have the advantage of their algebraic properties which results in better flexibility in the code lengths. Additionally, our proposed code’s error-floor can be enhanced by analyzing absorbing sets, as demonstrated in [11] and [35].

B. BINARY ERASURE CHANNEL (BEC)

Additionally, an analysis was conducted through the transmission of the zero vector using on-off keying (OOK) modulation over the BEC under varying error probabilities. For the decoder, we used the belief propagation (BP) based decoding algorithm given in [36] with the normalized minimum-sum (NMS) implementation [5]. As in the previous case, the maximum number of iterations is 100, and the maximum number of block errors 50. Figures 4 and 5 illustrate the decoding BER and FER performance of the codes in Table 2.

The results are almost the same as those obtained over AWGN channel. Again, the new proposed codes perform better than the codes constructed in [12]. While the DM(3; 43) code and the other [1849, 0.91, 4, 43] codes perform similarly, they outperform the DCA(3; 44) code. Table 4 provides details on the number of decoded vectors at EP = 0.04 required to reach 50 incorrectly decoded vectors and the number of erroneously decoded bits for these five codes. Based on the similarity, it might be concluded that the proposed codes have similar stopping set size/distribution as the other two structured codes. As in [18], an additional advantage is possible in that the proposed codes have the potential for larger stopping sets. Such improvements can be investigated by discarding some rows or columns in a DM(3; a) or DCA(3; a), as done in Construction 2.

VI. CONCLUSION

With explosion in the number of smart devices, there is need for a new generation of error correcting codes to meet the resultant demand for ultra-reliable and low latency smart object communication. Furthermore, 5G networks need codes that support variable code rates and lengths. Both LDPC and polar codes promise the requisite functionality and are currently being widely researched.

In this paper we presented two new constructions of LDPC codes developed from difference matrices and difference covering arrays. When compared to previous constructions, the constructions presented here leverage the advantage of the underlying algebraic structure, which is the cyclic group with binary operation addition modulo an integer. These algebraic structures, difference matrices and difference covering arrays, exist for all orders \( a \), allowing construction of an infinite family of LDPC codes and theoretical verification of the properties of the codes. In particular, for \( a \) even, we presented LDPC codes with lengths at least \( a^2 - a \) and a rate at least \( 1 - \frac{4a-6}{a^2} \). Similarly, for \( a \) odd, we constructed LDPC codes with length \( a^2 \) and rate \( 1 - \frac{4a-3}{a^2} \). Furthermore, for \( a \) odd, we showed that the constructed codes are quasi-cyclic and provided \( a \) is not divisible by 3 or 5, the codes have minimum distance at least 10. The simulation results presented in this paper, using standard decoding algorithms, showed that these LDPC codes perform well enough when compared to previous constructions of LDPC codes similar to ours.
APPENDIX A
PROPERTIES OF THE PARITY-CHECK MATRICES

In the following proofs we rely on the results in Lemma 3 and 4 where it was verified that for both $H$ and $\overline{H}$ the inner product of any two distinct columns is less than or equal to one. In addition, it should be noted that since we are working with $(0,1)$ matrices the linear dependence of rows is calculated bitwise over the binary field $\mathbb{Z}_2$.

Lemma 14: Let $a \geq 3$ be odd, then the rank of the matrix $H$ is exactly $4a - 3$.

Proof: We will establish this result by showing that there is a set of 3 rows in the row space of $H$ such that each row may be written as linear combinations of the remaining $4a - 3$ rows. Further we will establish that there exists a set of $4a - 3$ rows that are linearly independent.

Lemma 1 implies that, for each column $b$, there exists $x \in V_0$, $y \in V_1$ and $t \in V_2$ with $1 = H(x,b) = H(y,b) = H(t,b)$, where $V_i = \{(i+1)a, \ldots , (i+2)a-1\}$, $i = -1, 0, 1, 2$. Thus when the row space of $H$ is restricted to the rows given by any two of these sets, $V_i \cap V_j = \emptyset$, $-1 \leq i < j \leq 2$, the bitwise sum of the corresponding rows is 0 modulo 2, implying the rows corresponding to $V_{i,j}$, $-1 \leq i < j \leq 2$, are linearly dependent. Thus at least 1 row from 3 distinct $V_i$’s needs to be removed to obtain a subset of linearly independent rows of $H$, or equivalently, the size of any linearly independent subset is at most $4a - 3$. Without loss of generality we will eliminate the rows $2a - 1, 3a - 1$ and $4a - 1$.

Now we claim that the set of rows corresponding to $V = V_{-1} \cup V_0 \cup V_1 \cup V_2 \setminus \{2a - 1, 3a - 1, 4a - 1\}$ is a linearly independent set.

Assume that this is not the case and that there exists $L \subseteq V$ such that the corresponding rows of $H$ give a linearly dependent set. Note, this implies when $H$ is restricted to the rows of $L$ then the sum of the entries in columns of $H$ is 0 mod 2

For any $i \in \{-1, 0, 1, 2\}$ and any $r \in V_i$, the row sum of row $r$ is $a$, so let $(b_{r,1}, \ldots , b_{r,a})$ denote the set of a column where $H(r,b_{r,j}) = 1$ for $j = 1, \ldots , a$. The proof of Lemma 2 implies that for distinct $r, r' \in V_i$ if $H(r,b_{r,j}) = 1$, then $H(r',b_{r,j}) \neq 1$. Further, for distinct $i,i' \in V_i$ and $r' \in V_i$, there exists a unique $b_{r,j} \in \{b_{r,1}, \ldots , b_{r,a}\}$ such that $H(r,b_{r,j}) = 1$ and $H(r',b_{r,j}) = 1$.

Next we proceed by assuming $r \in L \cap V_i$, for some $i \in \{-1, 0, 1, 2\}$ and let $H(L, (b_{r,1}, \ldots , b_{r,a}))$ denote the restriction of the matrix $H$ to rows in $L$ and columns $b_{r,1}, \ldots , b_{r,a}$. Since $L$ corresponds to a linearly dependent set, summing over all rows of $H(L, (b_{r,1}, \ldots , b_{r,a}))$ mod 2 gives the 0 vector of length $a$. Hence the number of non-zero entries in $H(L, (b_{r,1}, \ldots , b_{r,a}))$ is even, say $2\ell$ for some $\ell \in \mathbb{Z}$. But now the argument above implies that $2\ell = a + |L \setminus V_i|$ (the number of rows in $L$ but not in $V_i$). Now as $a$ is assumed to be odd, we have $|L \setminus V_i|$ is also odd.

Next assume $L \cap V_i \neq \emptyset$ and $r' \in V_i$ but $r' \not\in L$. Again let $(b_{r',1}, \ldots , b_{r',a})$ denote the set of columns such that $H(r',b_{r',j}) = 1$, $j = 1, \ldots , a$. As above let $H(L, (b_{r',1}, \ldots , b_{r',a}))$ be the restriction of the matrix $H$ to rows in $L$ and columns in $(b_{r',1}, \ldots , b_{r',a})$. Again the number of entries in $H(L, (b_{r',1}, \ldots , b_{r',a}))$ is even, say $2\ell'$, for some $\ell' \in \mathbb{Z}$. We have $2\ell' = |L \setminus V_i|$ implying $|L \setminus V_i|$ is even. Thus we have a contradiction, and no such $r'$ exists.

So if $L \cap V_i \neq \emptyset$ then $V_i \subseteq L \subseteq V = V_{-1} \cup V_0 \cup V_1 \cup V_2 \setminus \{2a - 1, 3a - 1, 4a - 1\}$ implying $L = V_{-1}$, but as the sum of each column restricted to $V_{-1}$ is 1, again giving a contradiction.

Hence $V$ is linearly independent and the rank of $H$ is at least $4a - 3$ for odd $a$, implying the rank of $H$ is exactly $4a - 3$ for odd $a$.

Lemma 15: Let $a \geq 4$ be even, then the rank of the matrix $\overline{H}$ is at most $4a - 6$.

Proof: In this proof we establish the bound on the rank by showing that there is a set of 5 rows in the row space of $\overline{H}$ (i.e. the set of vectors corresponding to rows of $\overline{H}$) that can be written as linear combinations of the remaining rows. To aid understanding we will prove the result for a $(4a) \times (a^2 - a)$ matrix $\overline{H}_{r_0}$ that agrees with $\overline{H}$ in rows 0 to $r_0 - 1$, with the next row $r_0$ having all entries equal to 0, and then followed by rows $r_0 + 1$ to $4a - 2$ of $\overline{H}$. Thus we have reinstated the row of all zeros to $\overline{H}$. This will allow us to simplify the arguments, while the introduction of a zero row will not change the calculation of the rank of $\overline{H}$.

We proceed by using Construction 2 to deduce the following properties of $\overline{H}_{r_0}$.

Firstly, note that as in the case for odd $a$, when the row space of $\overline{H}_{r_0}$ is restricted to rows given by any two of the sets, $V_i \cap V_j = \emptyset$, $-1 \leq i < j \leq 2$, the component-wise sum of the corresponding vectors is 0 modulo 2, implying the vectors corresponding to $V_{i,j}$, $-1 \leq i < j \leq 2$, are linearly dependent.

To obtain a subset of linearly independent vectors of the row space of $\overline{H}_{r_0}$ at least 4 rows need to be removed, or equivalently, the size of any linearly independent subset is at most $4a - 4$. Without loss of generality we will eliminate the vectors corresponding to rows $r_0, 2a - 1, 3a - 1$ and $4a - 1$.

Now consider the following sets $L_3, L_4, L_5$ of rows of $\overline{H}_{r_0}$. For $0 \leq i \leq \frac{a^2 - 2}{2}$

$L_3 = \{x, a + 2i, 2a + 2i \mid D(x, 1) \equiv 1 \mod 2\}$

$L_4 = \{x, a + 2i, 3a + 2i \mid D(x, 2) \equiv 1 \mod 2\}$

$L_5 = \{x, 2a + 2i, 3a + 2i \mid D(x, 1) \not\equiv D(x, 2) \mod 2\}$

We claim that the vectors corresponding to each of these sets are linearly dependent.

To see that each of $L_3$ and $L_4$ corresponds to a linearly dependent set of vectors observe that for any column $b$ and $C_b = \{x, y, z, t\}$, Lemma 1 implies if $D(x, 1) \equiv 0$ then $y \equiv z \mod 2$ and if $D(x, 1) \equiv 1$ then $y \equiv z \mod 2$.

Similarly if $D(x, 2) \equiv 0$ then $y \equiv t \mod 2$ and if $D(x, 2) \equiv 1$ then $y \not\equiv t \mod 2$. Furthermore, to see that $L_5$ corresponds to a linearly dependent set of vectors observe that, if $D(x, 1) \equiv D(x, 2) \mod 2$ then $z \equiv t \mod 2$ and if $D(x, 1) \not\equiv D(x, 2) \mod 2$ then $z \not\equiv t \mod 2$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Note that there are no rows that are common to all three sets and $L_3$, $L_4$, $L_5 \subseteq \{ V = V_{a,1} \cup V_0 \cup V_1 \cup V_2 \setminus \{ r_0, 2a-1, 3a-1, 4a-1 \}$. So as $L_3$, $L_4$ and $L_5$ are linearly dependent sets of rows then to obtain a subset of linearly independent vectors of the row space of $H_{\alpha}$, without loss of generality we can further eliminate the vectors corresponding to rows $2a - 2$ and $3a - 2$.

Hence we can eliminate the rows $r_0, 2a - 2, 2a - 1, 3a - 2, 3a - 1, 4a - 1$ without changing the rank of $H_{\alpha}$. Thus the rank of $H$ is at most $4a - 6$ for even $a$.

**Lemma 16:** Let $a > 3$ and $H$ be the parity-check matrix based on the DM(3; a) given by Equation (19) with $a = \frac{a-1}{2}$ where gcd(a, 3) = 1 and gcd(a, 5) = 1. Then $H$ is the parity-check matrix of a $(a^2, 4, a)$-regular binary LDPC code with minimum distance 10.

**Proof:** To reduce excessive notation in this proof, all equality will be assumed to be equivalences modulo $a$.

We verified in Section IV that the minimum distance is at most 10 by specifying a set of 10 linearly dependent columns. Now we will show that minimum distance of the constructed code is at least 10. First observe that the minimum distance cannot be odd. Each column contains exactly one 1 in the first $a$ rows and each row should have an even number of 1’s in these columns so the number of columns in the linearly dependent set of columns should be even. If we can show that there are no 8 columns that are linearly dependent then this would imply the minimum distance is at least 10.

Assume $a$ is odd and the parity-check matrix $H$ contains 8 columns that are linearly dependent, where the bitwise sum over the columns is taken modulo 2. Then by Lemma 1 these columns take the form

$$\{x_1, y_1, z_1, t_1\}, \ {x_5, y_5, z_5, t_5}, \ {x_2, y_2, z_2, t_2}, \ {x_6, y_6, z_6, t_6}, \ {x_3, y_3, z_3, t_3}, \ {x_7, y_7, z_7, t_7}, \ {x_4, y_4, z_4, t_4}, \ {x_8, y_8, z_8, t_8},$$

where $x_1 + y_1 = z_1 \mod a$ and $\alpha x_1 + y_1 = t_1 \mod a$ for $i = 1, \ldots, 8$ and $a = \frac{a-1}{2}$. Note that, as $a$ is odd, gcd($a, a-1$) = 1; since gcd($a, 3$) = 1 we have gcd($\alpha$ - 1, $a$) = 1 and as gcd($a, 5$) = 1 we have gcd($\alpha$ - 2, $a$) = 1. Furthermore, note that since $a$ is odd, $2k = 2l \mod a$ implies $k = l$ modulo $a$ in general.

Under the assumption that these columns are linearly dependent, it follows that all elements $x_1, y_1, z_1, t_1, \ldots, x_8, y_8, z_8, t_8$, for $1 \leq i \leq 8$, occur an even number of times, with the RC-constraint implying these elements each occur either 2 or 4 times.

Assume, without loss of generality (wlog), that $x_1$ occurs 4 times. Then the RC-constraint implies there are $y_1, y_2, y_3$ and $y_4$ all distinct, each occurring exactly twice, similarly for $z_1, z_2, z_3, z_4$ and $t_1, t_2, t_3, t_4$. Note that since $a$ is odd, the equations $x_1 + y_1 = z_1, x_1 + y_2 = z_2, x_2 + y_2 = z_1$ and $x_2 + y_1 = z_2$ together results in a contradiction.

Thus there are two possibilities:

(i) Either it may be assumed that there exists $x_2$ and $x_3$ not necessarily distinct such that $x_2 + y_2 = z_1, x_2 + y_3 = z_2, x_3 + y_4 = z_3$ and $x_3 + y_1 = z_4$. Consequently $z_2 - z_1 = z_3 - z_2$ and $z_4 - z_3 = z_1 - z_4$, which gives $z_2 = z_3 + z_4 = 2z_4$, contradicting the fact that $z_2$ and $z_4$ are distinct modulo $a$.

(ii) Or it may be assumed that there exists distinct $x_2$ and $x_3$ such that $x_2 + y_1 = z_1, x_2 + y_2 = z_4, x_3 + y_3 = z_1$ and $x_3 + y_4 = z_2$. This case is a special case of Case 1-a) below where we set $x_1 = x_4$ and it results in a contradiction.

A similar argument will show that it is not possible for any $y_i$ to occur 4 times.

Next assume that, for all $1 \leq i \leq 4$, each $x_i$ and $y_i$ occurs exactly twice, leading to two non-isomorphic subcases.

**Case 1-)** or **Case 2-)**

$$\{x_1, y_1, z_1, t_1\}, \ {x_1, y_1, z_1, t_1}, \ {x_1, y_2, z_2, t_2}, \ {x_1, y_2, z_2, t_2}, \ {x_2, y_1, z_3, t_3}, \ {x_2, y_2, z_3, t_3}, \ {x_2, y_2, z_4, t_4}, \ {x_2, y_3, z_4, t_4},$$

(21)

**CASE 1-)**: First observe that, wlog, $x_1 + y_1 = z_1, x_1 + y_2 = z_2, x_2 + y_1 = z_2$ and $x_2 + y_2 = z_1$ will imply $z_1 - z_2 = y_1 - y_2 = z_2 - z_1$, a contradiction. Hence we may assume that $|z_1, z_2, z_3, z_4|, |z_5, z_6, z_7, z_8| \geq 3$, and similarly $|\{t_1, t_2, t_3, t_4\}, |t_5, t_6, t_7, t_8| \geq 3$.

Furthermore the description given above implies

$$\alpha(x_2 - x_1) = t_3 - t_1 = t_4 - t_2.$$

(22)

$$\alpha(x_4 - x_3) = t_7 - t_5 = t_8 - t_6.$$

(23)

$$z_1 + z_4 = z_2 + z_3, \quad z_5 + z_8 = z_6 + z_7.$$  

(24)

Now assume, wlog, $|\{z_1, z_2, z_3, z_4\}| = 3$ and $z_1 = z_4$. Equation (26) gives $z_2 = z_3 + z_4$. We may deduce $|z_5, z_6, z_7, z_8| = 3$ and, wlog, $z_5 = z_8$ and $z_6 = z_2$ so $z_7 = z_3$. Equation (26) implies $z_2 = z_3 + z_4$. Now combining this information gives $2(z_1 - z_5) = 0$ and so $z_1 = z_4 = z_5 = z_8$, leading to $t_1, t_4$, $t_5, t_8$ being distinct. Now, assuming that $t_5 = t_2$ and $t_8 = t_3$ then either $(t_6, t_7) = (t_1, t_4)$ or $(t_6, t_7) = (t_4, t_1)$.

The former implies $z_1 - z_3 = (x_1 - x_2) = y_2 - y_1 = t_2 - t_1 = y_3 - y_4 = (x_4 - x_3) = z_3 - z_1$ a contradiction.

The latter implies $(x_2 - x_1) = z_3 - z_1 = (x_4 - x_3)$, so $t_4 - t_2 = t_1 - t_2$ which leads to a contradiction since $t_1$ and $t_4$ are distinct.

The case $t_5 = t_3$ and $t_8 = t_2$ follows similarly.

Also similarly, $|\{t_1, t_2, t_3, t_4\}| = 3$ is not possible.

So $|z_1, z_2, z_3, z_4| = |z_5, z_6, z_7, z_8| = |t_1, t_2, t_3, t_4| = |t_5, t_6, t_7, t_8| = 4$.

Hence in Case 1-) we may assume WLOG $z_5 = z_1$.

**Case 1-)a-)**: Assume $z_6 = z_2$.

If $z_7 = z_4$, then $z_8 = z_3$ and so Equation (26) gives $z_1 + z_4 = z_2 + z_3$ and $z_1 + z_3 = z_2 + z_4$, implying
\[ z_1 + z_3 - z_4 = z_2 + z_4 - z_3 \] and leading to the contradiction \( z_3 = z_4 \) since \( a \) is odd. Thus \( z_7 = z_3 \) and \( z_8 = z_4 \).

Then we have

\[
t_2 - t_1 = y_2 - y_1 = z_2 - z_1 = y_4 - y_3 = z_4 - z_3 = t_4 - t_3 = b_6 - t_5 = t_8 - t_7.
\]  
(28)

Consider the case where \((t_5, t_8) = (t_1, t_1)\), for \( i \in \{2, 3\} \) then \((t_6, t_7) = (t_1, t_4)\) or \((t_6, t_7) = (t_4, t_1)\) where \( j \in \{2, 3\} \setminus \{i\}\).

Equation (27) implies \( t_5 + t_1 = t_2 + t_4 \) and \( t_5 + t_4 = t_2 + t_3 \). Combining these equations gives \( t_4 - t_1 = t_5 - t_4 \), a contradiction since \( t_4 \) and \( t_1 \) are distinct.

Thus, we have the following possibilities.

i-) \((t_5, t_8) = (t_4, t_1)\) and \((t_6, t_7) = (t_3, t_2)\). By Equation (28)

\[ t_2 - t_1 = t_2 - t_2 \] a contradiction.

ii-) \((t_5, t_8) = (t_2, t_3)\) and \((t_6, t_7) = (t_1, t_4)\). By Equation (28), \( t_2 - t_2 = t_1 - t_1 \), a contradiction.

iii-) \((t_5, t_8) = (t_2, t_5)\) and \((t_6, t_7) = (t_4, t_1)\). By Equation (28), \( t_2 - t_4 = t_5 - t_2 \) implying \( 2t_2 = t_1 + t_4 \) combining with \( t_1 + t_4 = t_2 + t_3 \) we have \( t_2 = t_3 \), a contradiction.

iv-) \((t_5, t_8) = (t_3, t_2)\) and \((t_6, t_7) = (t_1, t_4)\). Then by Equation (28), \( t_2 - t_1 = t_2 - t_1 \) implying \( 2t_2 = t_3 + t_3 \) with \( t_1 + t_4 = t_2 + t_3 \) we have \( t_1 = t_4 \), a contradiction.

v-) \((t_5, t_8) = (t_3, t_2)\) and \((t_6, t_7) = (t_4, t_1)\). Then by Equations (22) and (23), we have \( x_2 - x_1 = z_3 - z_1 = z_7 - z_5 = (x_4 - x_3) \). Hence by Equations (24) and (25),

\[ t_3 - t_1 = \alpha(x_2 - x_1) = \alpha(x_4 - x_3) = t_7 - t_5 = t_1 - t_3, \] a contradiction.

Case 1-\(i\)-b): Assume \( z_6 = z_3 \).

If \( z_7 = z_2, z_8 = z_3 \) then Equation (26) gives \( z_1 + z_3 = z_2 + z_4 \) and \( z_1 + z_4 = z_3 + z_2 \) hence \( 2z_4 = 2z_3 \), a contradiction.

If \( z_7 = z_3, z_8 = z_2 \) then Equation (26) gives \( z_4 + z_4 = z_3 + z_2 \) and \( z_1 + z_2 = z_3 + z_4 \) hence \( 2z_2 = 2z_4 \), a contradiction.

CASE 2-\(i\): Considering the list of entries \( z_1, \ldots, z_8 \) in Case 2-\(i\), as given in Equation (21), these entries can be written as a cyclic list, that is, \((z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8)\). Any even shift will be isomorphic to this list in nature and an odd shifts will interchange \( x_i \)'s with \( y_i \)'s in the equations. There exists 4 sets of pairs \( i, j \), \( 1 \leq i < j \leq 8 \) such that \( z_i = z_j \) when we may say that \( z_i \) and \( z_j \) are distance \(| j - i | \) apart. The cyclic nature of the list implies that for all pairs \( i, j \) such that \( z_i = z_j \) we may take \(| j - i | \leq 4 \). Assume \( z_1 \) has the smallest distance among the \( z_i \)'s.

We address the possibilities with the following subcases:

a-) Assume \( z_j = z_1 \) where \( j = 1 = 2 \) and \( z' = z_2 \) where \( f' = 2 = 2 \), implying cyclic list \((z_1, z_2, z_1, z_2, z_3, z_4, z_5)\). Then \( y_2 - y_1 = y_3 - y_2 \) and \( y_4 - y_3 = y_1 - y_4 \) implying \( 2y_2 = 2y_4 \) a contradiction.

Similarly an odd shift will give a list isomorphic to \((z_2, z_1, z_2, z_3, z_4, z_5, z_6, z_7)\). Then we have \( x_1 - x_4 = x_1 - x_4 \) and \( x_3 - x_4 = x_4 - x_3 \) implying the contradiction \( 2x_1 = 2x_3 \).

b-) Assume \( z_j = z_1 \) where \( j = 1 = 2 \) and \( z' = z_2 \) where \( f' = 2 = 3 \), with cyclic list \((z_1, z_2, z_1, z_2, z_3, z_4, z_5, z_4)\). Then \( x_1 + x_1 = x_2 + y_2, x_1 + y_1 = x_3 + y_3, x_2 + x_3 = x_4 + y_4, x_3 + y_4 = x_4 + y_1, y_1 + y_2 = (x_2 + x_2) + y_2 + y_3, (x_2 + x_3) + y_3 - y_4 = y_4 - y_1 \). Then \( 2y_4 - y_3 = y_1 - 2y_2 + y_3 = y_1 - 2y_2 + y_3 = y_1 - 2y_2 + y_3 = 2(y_1 + y_2) = 2(y_3 + y_1), \) so \( y_4 + y_3 = y_4 + y_1 \). On the other hand, we have \( x_1 + x_2 + y_1 = x_3 + x_3 = x_4 + y_2 + y_4, \) where \( x_1 = x_4 \), a contradiction.

Similarly an odd shift will give a list isomorphic to \((z_2, z_1, z_3, z_2, z_4, z_3, z_4, z_1)\). Then we have \( z_3 - z_2 = y_2 - y_3 = x_3 - x_1 = y_4 - y_1 \) and \( z_4 - z_1 = y_1 - y_4 = x_3 - x_1 + y_3 - y_2 \) or equivalently \( 2y_2 = 2y_4 \) a contradiction.

c-) Assume \( z_j = z_1 \) where \( j = 1 = 2 \) and \( z' = z_2 \) where \( f' = 2 = 4 \), with cyclic list \((z_1, z_2, z_1, z_3, z_4, z_2, z_4, z_3)\). Then \( x_1 + y_1 = x_2 + y_2, x_1 + y_1 = x_3 + y_4, x_2 + x_3 = x_4 + y_1, x_3 + y_3 = x_4 + y_4, y_1 + y_2 = (x_2 + x_3) + y_2 + y_4, (x_2 + x_3) + y_3 - y_4 = y_4 - y_1 \). Then \( 2y_4 - y_3 = y_1 - 2y_2 + y_3 = 2(y_1 + y_2) = 2(y_3 + y_1), \) so \( y_4 + y_3 = y_4 + y_1 \). On the other hand, we have \( x_1 + x_2 + y_1 = x_3 + x_3 = x_4 + y_2 + y_4, \) where \( x_1 = x_4 \), a contradiction.

d-) Assume \( z_j = z_1 \) where \( j = 1 = 2 \) and \( z' = z_2 \) where \( f' = 2 = 4 \), with cyclic list \((z_1, z_2, z_1, z_3, z_4, z_2, z_3, z_4)\). Then
So $y_1 = x_2 + y_2$, $x_1 + y_2 = x_3 + y_4$, $x_2 + y_3 = x_4 + y_4$, $x_3 + y_3 = x_4 + y_1$. So $y_1 = y_2 = (x_2 - x_3) + y_2 - y_4$ and $(x_2 - x_3) = y_4 - y_1$. Then $y_1 - y_2 = y_4 - y_1 + y_2 - y_4$ or equivalently $y_1 - y_2 = y_2 - y_1$, a contradiction.

Similarly an odd shift will give a list isomorphic to $(z_2, z_3, z_5, z_7, z_8, z_9, z_1)$. Then we have $z_2 - z_1 = y_1 - y_2 = x_3 - x_4 + y_3 - y_1$ and $z_7 - z_2 = y_3 - y_2 = x_4 - x_3$ or equivalently $2y_2 = 2y_1$ a contradiction.

e) Assume $z_j = z_1$ where $j - 1 = 3$ and $z_j = z_2$ where $j' = 2 = 4$, with cyclic list $(z_1, z_2, z_3, z_4, z_5, z_6, z_7)$. Then $x_1 + y_1 = x_2 + y_3, x_1 + y_2 = x_3 + y_4, x_2 + y_2 = x_4 + y_4, x_3 + y_3 = x_4 + y_1$, implying $y_1 - y_2 = (x_2 - x_3) + y_3 - y_4$, $y_4 - y_1 = (x_2 - x_3) + y_3 - y_4$. Then $y_4 - y_1 = y_1 - y_2 + y_3 - y_3 = 2y_2 = 2y_1$, leading to a contradiction.

Similarly an odd shift will give a list isomorphic to $(z_2, z_3, z_5, z_7, z_8, z_9, z_1)$. Then we have $z_2 - z_3 = y_1 - y_2 = y_3 - y_4$ and $z_4 - z_1 = y_4 - y_1 = y_3 - y_2$ or equivalently $2y_3 = 2y_2$ a contradiction.

f) Assume $z_j = z_1$ where $j - 1 = 4$, giving the cyclic list $(z_1, z_2, z_3, z_4, z_5, z_6, z_7)$. Then $x_1 + y_1 = x_3 + y_3, x_1 + y_2 = x_3 + y_4, x_2 + y_2 = x_4 + y_4, x_3 + y_3 = x_4 + y_1$, implying $y_1 - y_2 = y_3 - y_4$ and $y_2 - y_3 = y_4 - y_1$. Then $y_1 - y_3 = y_3 - y_1$, a contradiction.

Similarly an odd shift will give a list isomorphic to $(z_2, z_3, z_5, z_7, z_8, z_9, z_1)$. Then we have $z_2 - z_3 = y_1 - y_2 = y_3 - y_4$ and $z_4 - z_1 = y_4 - y_1 = y_3 - y_2$ or equivalently $2y_3 = 2y_2$ a contradiction.

g) Assume $z_j = z_1$ where $j - 1 = 3$ and $z_j = z_2$ where $j' = 2 = 3$, with cyclic list $(z_1, z_2, z_3, z_4, z_5, z_6, z_7)$. Observe that all $z_i$ have distance 3 in the cyclic list. Then $x_1 + y_1 = x_2 + y_3, x_1 + y_2 = x_3 + y_4, x_2 + y_2 = x_4 + y_4$, $x_3 + y_3 = x_4 + y_1$ implying $y_1 - y_2 = y_3 - y_4$ and $y_2 - y_3 = y_4 - y_1$. Then $y_1 - y_3 = y_3 - y_1$, a contradiction.

Finally consider the list of entries $t_1, \ldots, t_6$ as given in Equation (21). Again, these entries can be written as a cyclic list, that is, $(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)$. The above arguments can be repeated for this cyclic list where any occurrence of $x_i$ in an equation is replaced by $\alpha x_i$. Thus, it can be argued that the distinct entries $t_1, t_2, t_3, t_4$ have distance 3 in the above cyclic list. Now considering the RC-constraint we have $t_6 = t_1$. Then $t_2 = t_3 = t_5 = t_0$. Hence we have the list $(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8)$.

But then we have $\alpha x_1 + y_1 = k x_3 + y_3, \alpha x_1 + y_2 = k x_3 + y_3$, implying $y_1 - y_2 = y_4 - y_3$. Hence $2(y_1 - y_2) = (x_2 - x_4)$ and combining with $\alpha x_2 + y_2 = \alpha x_3 + y_1$ we have $2\alpha (x_2 - x_4) = (x_2 - x_4)$. Hence $(2\alpha - 1)(x_2 - x_4) = (a - 2)(x_2 - x_4) = 0$, a contradiction since $\gcd(a, a - 2) = 1$.

Similarly an odd shift will give a list isomorphic to $(z_2, z_3, z_5, z_7, z_8, z_9, z_1)$. And the list of $t_i$ as $(t_2, t_3, t_4, t_5, t_1, t_2, t_3, t_4, t_1, t_2, t_3, t_4, t_1, t_2, t_3, t_4, t_1)$. Then we have $z_3 - z_2 = y_1 - y_2 = y_4 - y_3$ and $z_2 - z_3 = y_1 - y_2 = \alpha (x_2 - x_4) + y_3 - y_4$ implying $2(y_1 - y_2) = \alpha (x_2 - x_4)$. Now combining with $z_1 = x_2 + y_2 = x_3 + y_1$ we have $\alpha (y_1 - y_2) = 2(y_1 - y_2)$ or equivalently $(\alpha - 2)(y_1 - y_2) = 0$. But as $\alpha - 2 = \frac{a + 1}{a - 1}$ and $\gcd(a, 5) = 1$ we have $y_1 = y_2$ a contradiction.

The proof of this Lemma can be readily generalized for any $\alpha$ with required properties:
which is

\[
H^* = \begin{bmatrix}
I & I & I & I & I \\
I & p_1 & p_2 & p_3 & p_4 \\
I & p_2^{-1} & p_2+2^{-1} & p_3+2^{-1} & p_4+2^{-1} \\
I & p_3^{-1} & p_2+3^{-1} & p_3+3^{-1} & p_4+3^{-1}
\end{bmatrix}.
\]

Note that \(3^{-1} = (\alpha + 1)^{-1}\).

APPENDIX B
ILLUSTRATIONS OF RELEVANT PARITY CHECK MATRICES
OF THE CODES LISTED IN TABLE 2

FIGURE 6. The parity-check matrix of the proposed code PREV(3;34)
[1122, 0.91, 3, 33].

FIGURE 7. The parity-check matrix of the proposed code DM(3;43)
[1849, 0.91, 4, 43].

FIGURE 8. The parity-check matrix of the proposed code DCA(3;44)
[1892, 0.91, 4, 43].

FIGURE 9. The parity-check matrix of Array [1849, 0.91, 4, 43].

FIGURE 10. The parity-check matrix of TD-LDPC [1849, 0.91, 4, 43].

FIGURE 11. The parity-check matrix of Lattice [1849, 0.91, 4, 43].

FIGURE 12. The parity-check matrix of SCB-LDPC [1849, 0.91, 4, 43].

ACKNOWLEDGMENT

Yazici would like to thank RMIT University for travel support. This research was carried out during her visit to RMIT University. Üsküplü was affiliated with Koç University as a master’s student when she started working on this paper.
S. Zhao, X. Huang, and X. Ma, “Structural analysis of array-based non-binary LDPC codes for 5G new radio,” IEEE Commun. Mag., vol. 56, no. 3, pp. 28–34, Mar. 2018, doi: 10.1109/MCOM.2018.1700839.

M. Sarvaghad-Moghadam, W. Ullah, D. N. K. Jayakody, and S. Affes, “A new construction of high performance LDPC matrices for mobile networks,” Sensors, vol. 20, no. 8, p. 2300, Apr. 2020.

N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences. Accessed: May 1, 2023. [Online]. Available: https://oeis.org/

R. Smarandache and D. G. M. Mitchell, “Necessary and sufficient girth conditions for Tanner graphs of quasi-cyclic LDPC codes,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2021, pp. 380–385.

M. Stark, G. Bauch, L. Wang, and R. D. Wesel, “Information bottleneck decoding of rate-compatible 5G-LDPC codes,” in Proc. IEEE Int. Conf. Commun. (ICC), Dublin, Ireland, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149304.

R. Sun, Y. Tian, and J. Liu, “Construction of QC-LDPC codes based on generalized RS codes with girth larger than 6,” in Proc. IEEE Int. Conf. Commun. Syst. (ICCS), Shenzhen, China, Dec. 2016, pp. 1–6.

A. Tasdighi, A. H. Baniasghemi, and M. Sadeghi, “Efficient search of girth-optimal QC-LDPC codes,” IEEE Trans. Inf. Theory, vol. 62, no. 4, pp. 1552–1564, Apr. 2016, doi: 10.1109/TIT.2016.2523979.

A. Tasdighi, A. H. Baniasghemi, and M. R. Sadeghi, “Symmetrical constructions for regular girth-8 QC-LDPC codes,” IEEE Trans. Commun., vol. 65, no. 1, pp. 22–34, Jan. 2017.

B. Vasic and O. Milenkovic, “Combinatorial constructions of low-density parity-check codes for iterative decoding,” IEEE Trans. Inf. Theory, vol. 50, no. 6, pp. 1156–1176, Jun. 2004, doi: 10.1109/TIT.2004.828066.

U. Vasily. (2020). Progressive Edge Growth for LDPC Code Construction C++ and MATLAB PEG+ACE Implementations. [Online]. Available: https://github.com/Lcrypto/classic-PEG-

J. Wang, L. Dolecek, and R. D. Wesel, “The cycle consistency matrix approach to absorbing sets in separable circulant-based LDPC codes,” IEEE Trans. Inf. Theory, vol. 59, no. 4, pp. 2293–2314, Apr. 2013, doi: 10.1109/TIT.2012.2235122.

E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, “High throughput low-density parity-check decoder architectures,” in Proc. IEEE Global Telecommun. Conf. (GLOBECOM), vol. 5, Nov. 2001, pp. 3019–3024, doi: 10.1109/GLOCOM.2001.965981.

J. Yin, “Constructions of difference covering arrays,” J. Combinat. Theory, A, vol. 104, no. 2, pp. 327–339, 2003.

J. Yin, “Cyclic difference packing and covering arrays,” Des., Codes Cryptogr., vol. 37, no. 2, pp. 281–292, Nov. 2005.

L. Zhang, Q. Huang, S. Lin, K. Abdel-Ghaffar, and I. F. Blake, “Quasi-cyclic LDPC codes: An algebraic construction, rank analysis, and codes on Latin squares,” IEEE Trans. Commun., vol. 59, no. 11, pp. 3126–3139, Nov. 2010, doi: 10.1109/TCOMM.2010.091710.090721.

S. Zhao, X. Huang, and X. Ma, “Structural analysis of array-based non-binary LDPC codes,” IEEE Trans. Commun., vol. 64, no. 12, pp. 4910–4922, Dec. 2016, doi: 10.1109/TCOMM.2016.2609906.

S. Zhao and X. Ma, “Extended construction of array-based non-binary LDPC codes,” Electron. Lett., vol. 55, no. 4, pp. 196–198, Feb. 2019.