Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2008, Article ID 598401, 9 pages
doi:10.1155/2008/598401

Research Article

Structured LDPC Codes over Integer Residue Rings

Elisa Mo and Marc A. Armand

Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576

Correspondence should be addressed to Marc A. Armand, eleama@nus.edu.sg

Received 31 October 2007; Revised 31 March 2008; Accepted 3 June 2008

Recommended by Jinhong Yuan

This paper presents a new class of low-density parity-check (LDPC) codes over $\mathbb{Z}_{2^a}$ represented by regular, structured Tanner graphs. These graphs are constructed using Latin squares defined over a multiplicative group of a Galois ring, rather than a finite field. Our approach yields codes for a wide range of code rates and more importantly, codes whose minimum pseudocodeword weights equal their minimum Hamming distances. Simulation studies show that these structured codes, when transmitted using matched signal sets over an additive-white-Gaussian-noise channel, can outperform their random counterparts of similar length and rate.

1. INTRODUCTION

The study of nonbinary LDPC codes over GF(q) was initiated by Davey and Mackay [1]. However, the symbols of a nonbinary code over a finite field cannot be matched to any signal constellation. In other words, it is not possible to obtain a geometrically uniform code (wherein every codeword has the same error probability), from a nonbinary, finite field code. The subject of geometrically uniform codes has been well studied by various authors including Slepian [2] and Forney Jr. [3]. More recently, Sridhara and Fuja [4] introduced geometrically uniform, nonbinary LDPC codes over certain rings, including integer residue rings. Their codes are however unstructured. Structured LDPC codes, which include the family of finite geometry (FG) codes [5] and balanced incomplete block design (BIBD) codes [6], are favored over their random counterparts due to the reduction in storage space for the parity check matrix and the ease in performance analysis they provide, while achieving relatively similar performance. Structured nonbinary LDPC codes that have been proposed thus far however, are constructed over finite fields, for example, [1, 7], and therefore cannot be geometrically uniform.

This paper therefore addresses the problem of designing structured, geometrically uniform, nonbinary LDPC codes over integer residue rings. Motivated by the fact that short nonbinary LDPC codes can outperform their binary counterparts [8–10], we focus our investigations on codes of short codelength. Studies of the so-called pseudocodewords arising from finite covers of a Tanner graph, for example, [11–13], have revealed that while a code's performance under maximum-likelihood (ML) decoding is dictated by its (Hamming) weight distribution, its performance under iterative decoding is dictated by the weight distribution of the pseudocodewords associated with its Tanner graph. More specifically, the presence of pseudocodewords of low weight, particularly those of weight less than the minimum Hamming distance of the code, is detrimental to a code's performance under iterative decoding. We therefore adopt the Latin-squares-based approach of Kelley et al. [14] to construct structured codes, as their method aims at maximizing the minimum pseudocodeword weight of a code. While we maintain the pseudocodeword framework used there, our work nevertheless differs from [14] primarily because our construction relies on an extension of the notion of Latin squares to multiplicative groups of a Galois ring—a key contribution of this paper.

We note that codes based on Latin squares were also studied in [7, 15–17]. However, the authors of these works did not do so in the pseudocodeword framework. Codes constructed using other combinatorial approaches, such as those presented in [6, 18, 19], were similarly not investigated using

the notion of pseudocodewords. Specifically, these related works focused on the optimization of design parameters such as girth, expansion, diameter, and stopping sets. Our work therefore differs from these earlier studies in this regard.

For practical reasons, we only consider linear codes over $\mathbb{Z}_{2^a}$. In the next section, we provide an overview of codes over $\mathbb{Z}_{2^a}$ and their natural mapping to a matched signal constellation, that is, the 2^a-PSK constellation. Section 3 introduces the notion of Latin squares over finite fields, followed by our extension of Latin squares to multiplicative groups of a Galois ring. A method to construct Tanner graphs using Latin squares (over a multiplicative group of a Galois ring) is presented in Section 4. We show that from these graphs, a wide range of code rates may be obtained. We further derive in the same section certain properties of the corresponding codes and, in particular, show that their minimum pseudocodeword weights equal their minimum Hamming distances. This is one of our main results. Finally, Section 5 presents computer simulations which demonstrate that our codes, when mapped to matched signal sets and transmitted over the additive-white-Gaussian-noise (AWGN) channel, outperform their random counterparts of similar length and rate.

2. CODES OVER $\mathbb{Z}_{2^a}$

2.1. An overview

Let $\mathcal{C}$ be a $\mathbb{Z}_{2^a}$-submodule of the free $\mathbb{Z}_{2^a}$-module $\mathbb{Z}_{2^a}^n$. Its $n_{\mathbf{G}} \times n$ generator matrix $\mathbf{G}$ can be expressed in the form [20]

$$\mathbf{G} = \begin{bmatrix} 2^{\lambda_1}\mathbf{g}_1 \\ 2^{\lambda_2}\mathbf{g}_2 \\ \vdots \\ 2^{\lambda_{n_{\mathbf{G}}}}\mathbf{g}_{n_{\mathbf{G}}} \end{bmatrix}, \tag{1}$$

where $0 \le \lambda_i \le a-1$ for $i = 1, 2, \ldots, n_{\mathbf{G}}$ and $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{n_{\mathbf{G}}}\} \subset \mathbb{Z}_{2^a}^n$ is a set of linearly independent elements. The rate of $\mathcal{C}$ is

$$r = \frac{1}{n}\sum_{i=1}^{n_{\mathbf{G}}} \frac{a-\lambda_i}{a} = \frac{n_{\mathbf{G}}}{n} - \frac{\sum_{i=1}^{n_{\mathbf{G}}}\lambda_i}{an}. \tag{2}$$

The dual code $\mathcal{C}^{\perp}$ is generated by the $n_{\mathbf{H}} \times n$ parity-check matrix of $\mathcal{C}$, which can be expressed in the form

$$\mathbf{H} = \begin{bmatrix} 2^{\mu_1}\mathbf{h}_1 \\ 2^{\mu_2}\mathbf{h}_2 \\ \vdots \\ 2^{\mu_{n_{\mathbf{H}}}}\mathbf{h}_{n_{\mathbf{H}}} \end{bmatrix}, \tag{3}$$

where $0 \le \mu_i \le a-1$ for $i = 1, 2, \ldots, n_{\mathbf{H}}$ and $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n_{\mathbf{H}}}\} \subset \mathbb{Z}_{2^a}^n$ is a set of linearly independent elements. The rate of $\mathcal{C}$ can also be obtained by

$$r = 1 - \frac{1}{n}\sum_{i=1}^{n_{\mathbf{H}}} \frac{a-\mu_i}{a} = 1 - \frac{n_{\mathbf{H}}}{n} + \frac{\sum_{i=1}^{n_{\mathbf{H}}}\mu_i}{an}. \tag{4}$$

If $\mathbf{G}$ (or $\mathbf{H}$) is not already in the form in (1) (or in (3)), one could perform Gaussian elimination without dividing a row by a zero divisor to obtain the $n_{\mathbf{G}}$ (or $n_{\mathbf{H}}$) linearly independent rows.

Remark 1. $\mathcal{C}$ is a free $\mathbb{Z}$-submodule if $\lambda_i = 0$ for $i = 1, 2, \ldots, n_{\mathbf{G}}$. This also implies that $\mu_i = 0$ for $i = 1, 2, \ldots, n_{\mathbf{H}}$.

2.2. The matched signal set

The 2^a-PSK signal set contains 2^a points that are equidistant from the origin while maximally spread apart on a two-dimensional space. Projecting one dimension on the real axis and the other on the imaginary axis, a symbol $x \in \mathbb{Z}_{2^a}$ is mapped to $s_x = \sqrt{E_s}\exp(j2\pi x/2^a)$ of the signal set, where $\sqrt{E_s}$ is the energy assigned to each symbol [4].

The 2^a-PSK is matched to $\mathbb{Z}_{2^a}$ because for any $x, y \in \mathbb{Z}_{2^a}$,

$$d_E^2(s_x, s_y) = d_E^2(s_{x-y}, s_0), \tag{5}$$

where $d_E^2(s_x, s_y)$ denotes the square Euclidean distance between s_x and s_y [21].

Let $c_x, c_y \in \mathcal{C}$, where $c_x = [x_1, x_2, \ldots, x_n]$ and $c_y = [y_1, y_2, \ldots, y_n]$. They are mapped symbol by symbol to $[s_{x_1}, s_{x_2}, \ldots, s_{x_n}]$ and $[s_{y_1}, s_{y_2}, \ldots, s_{y_n}]$, respectively. The squared Euclidean distance between these two signal vectors is

$$\begin{aligned} &d_E^2([s_{x_1}, s_{x_2}, \ldots, s_{x_n}], [s_{y_1}, s_{y_2}, \ldots, s_{y_n}]) \\ &\quad = \sum_{i=0}^{n} d_E^2(s_{x_i}, s_{y_i}) = \sum_{i=0}^{n} d_E^2(s_{x_i - y_i}, s_0) \\ &\quad = d_E^2([s_{x_1-y_1}, s_{x_2-y_2}, \ldots, s_{x_n-y_n}], [s_0, s_0, \ldots, s_0]). \end{aligned} \tag{6}$$

Observe that the Hamming distance between two codewords is mapped proportionally to the Euclidean distance between their corresponding signal vectors.

3. LATIN SQUARES

3.1. Definition and application to galois fields

The following definition and example are taken from [22, Chapter 17].

Definition 1. A Latin square of order q is denoted as $(R, C, S; L)$, where R, C, and S are sets of cardinality q and L is a mapping $L(i, j) = k$, where $i \in R$, $j \in C$, and $k \in S$, such that given any two of i, j, and k, the third is unique.

A Latin square can be written as a $q \times q$ array for which the cell in row i and column j contains the symbol $L(i, j)$. Two Latin squares with mapping functions L and L' are orthogonal if $(L(i, j), L'(i, j))$ is unique for each pair (i, j). Further, a complete family of $q-1$ mutually orthogonal Latin squares (MOLS) exists for $q = p^s$, where p is prime.

The notion of Latin squares can be easily applied to Galois fields by setting $R = C = S = \mathrm{GF}(p^s)$ and mapping function $L_\beta(i, j) = i + \beta j$ for $\beta \in \mathrm{GF}(p^s) \setminus \{0\}$.

Example 1. Let $R = C = S = \mathrm{GF}(2^2) = \{0, 1, \alpha, \alpha^2\}$. Mapping functions $L_1(i, j) = i + j$, $L_\alpha(i, j) = i + \alpha j$ and $L_{\alpha^2}(i, j) = i + \alpha^2 j$ yield a complete family of three MOLS

$$M_1 = \begin{bmatrix} 0 & 1 & \alpha & \alpha^2 \\ 1 & 0 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 0 & 1 \\ \alpha^2 & \alpha & 1 & 0 \end{bmatrix}, \qquad M_\alpha = \begin{bmatrix} 0 & \alpha & \alpha^2 & 1 \\ 1 & \alpha^2 & \alpha & 0 \\ \alpha & 0 & 1 & \alpha^2 \\ \alpha^2 & 1 & 0 & \alpha \end{bmatrix}$$

$$M_{\alpha^2} = \begin{bmatrix} 0 & \alpha^2 & 1 & \alpha \\ 1 & \alpha & 0 & \alpha^2 \\ \alpha & 1 & \alpha^2 & 0 \\ \alpha^2 & 0 & \alpha & 1 \end{bmatrix}, \tag{7}$$

respectively.

3.2. Extension to multiplicative groups of a Galois ring

Extending the notion of Latin squares over integer residue rings is not trivial. Setting $R = C = S = \mathbb{Z}_{2^s}$ and mapping functions $L_\beta(i, j) = i + \beta j$ for $\beta \in \mathbb{Z}_{2^s} \setminus \{0\}$ do not yield a complete family of $2^s - 1$ MOLS.

Example 2. Let $R = C = S = \mathbb{Z}_{2^2} = \{0, 1, 2, 3\}$ and let mapping functions be $L_1(i, j) = i + j$, $L_2(i, j) = i + 2j$ and $L_3(i, j) = i + 3j$,

$$M_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}, \qquad M_2 = \begin{bmatrix} 0 & 2 & 0 & 2 \\ 1 & 3 & 1 & 3 \\ 2 & 0 & 2 & 0 \\ 3 & 1 & 3 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 3 & 2 & 1 \\ 1 & 0 & 3 & 2 \\ 2 & 1 & 0 & 3 \\ 3 & 2 & 1 & 0 \end{bmatrix}, \tag{8}$$

are obtained, respectively. Since the elements in each row of M_2 is not unique, M_2 is not a Latin square. Therefore, we do not have a complete family of three MOLS.

Hence, we propose an alternative way of constructing Latin squares over integer residue rings. Let extension ring $\mathcal{R} = \mathrm{GR}(2^a, s) = \mathbb{Z}_{2^a}[y]/\langle\phi(y)\rangle$, where $\phi(y)$ is a degree s basic irreducible polynomial over $\mathbb{Z}_{2^a}$. Embedded in $\mathcal{R}$ is a multiplicative group G_{2^s-1} of units of order $2^s - 1$. Further, we let $a' < a$ and define $\overline{z} = z \bmod 2^{a'}$, where $z \in \mathcal{R}$, and extend this notation to n-tuples and matrices over $\mathcal{R}$.

Example 3. Let $\mathcal{R} = \mathrm{GR}(2^2, 2) = \mathbb{Z}_4[y]/\langle y^2 + y + 3\rangle$. Embedded in $\mathcal{R}$ is $G_3 = \{1, \alpha, \alpha^2\} = \{1, y + 2, 3y + 1\}$, generated by $\alpha = y + 2$. Let $R = C = G_3 \cup \{0\}$. Mapping functions $L_1(i, j) = i+j$, $L_\alpha(i, j) = i+\alpha j$ and $L_{\alpha^2}(i, j) = i+\alpha^2 j$ yield matrices

$$M_1 = \begin{bmatrix} 0 & 1 & y+2 & 3y+1 \\ 1 & 2 & y+3 & 3y+2 \\ y+2 & y+3 & 2y & 3 \\ 3y+1 & 3y+2 & 3 & 2y+2 \end{bmatrix}$$

$$M_\alpha = \begin{bmatrix} 0 & y+2 & 3y+1 & 1 \\ 1 & y+3 & 3y+2 & 2 \\ y+2 & 2y & 3 & y+3 \\ 3y+1 & 3 & 2y+2 & 3y+2 \end{bmatrix} \tag{9}$$

$$M_{\alpha^2} = \begin{bmatrix} 0 & 3y+1 & 1 & y+2 \\ 1 & 3y+2 & 2 & y+3 \\ y+2 & 3 & y+3 & 2y \\ 3y+1 & 2y+2 & 3y+2 & 3 \end{bmatrix},$$

respectively. Since $G_3 \cup \{0\}$ is not closed under $\mathcal{R}$-addition, $S \subset \mathcal{R}$ so that $|S| \neq |R| = |C| = 2^s$. Thus, all three matrices are not Latin squares.

To overcome this problem, the mapping functions have to be altered slightly such that they map $i \in R$ and $j \in C$ uniquely to $L_\beta(i, j) \in S$ and $|R| = |C| = |S|$.

Definition 2. $L_\beta^{(a)}(i, j) = ((i)^{1/2^{a-1}} + (\beta j)^{1/2^{a-1}})^{2^{a-1}}$, where $i, j \in G_{2^s-1} \cup \{0\}$ and $\beta \in G_{2^s-1}$.

Theorem 1. $L_\beta^{(a)}(i, j) \in G_{2^s-1} \cup \{0\}$.

Proof. It is apparent that $(i)^{1/2^{a-1}}, (\beta j)^{1/2^{a-1}} \in G_{2^s-1} \cup \{0\}$. Since $G_{2^s-1} \cup \{0\}$ is not closed under $\mathcal{R}$-addition, $(i)^{1/2^{a-1}} + (\beta j)^{1/2^{a-1}} = u + 2v$, where $u \in G_{2^s-1} \cup \{0\}$ and $v \in \mathcal{R}$. Using binomial expansion, the mapping function can be expressed as

$$L_\beta^{(a)}(i, j) = (u + 2v)^{2^{a-1}} = \sum_{x=0}^{2^{a-1}} \binom{2^{a-1}}{x} u^{2^{a-1}-x} (2v)^x. \tag{10}$$

Observe that $\binom{2^{a-1}}{x} u^{2^{a-1}-x} (2v)^x = 0 \bmod 2^a$ for $x = 1, 2, \ldots, 2^{a-1}$. Thus, $L_\beta^{(a)}(i, j) = u^{2^{a-1}} \in G_{2^s-1} \cup \{0\}$. □

Theorem 2. *Consider two multiplicative groups* $G_{2^s-1} \subset \mathrm{GR}(2^a, s) = \mathbb{Z}_{2^a}[y]/\langle\phi(y)\rangle$ *and* $G'_{2^s-1} \subset \mathrm{GR}(2^{a'}, s) = \mathbb{Z}_{2^{a'}}[y]/\langle\overline{\phi(y)}\rangle$, *where* $\phi(y)$ *is a degree-s basic irreducible polynomial over* $\mathbb{Z}_{2^a}$. *Let* $i, j \in G_{2^s-1} \cup \{0\}$ *and* $\beta \in G_{2^s-1}$, *then* $\overline{i}, \overline{j} \in G'_{2^s-1} \cup \{0\}$ *and* $\overline{\beta} \in G'_{2^s-1}$. *Then,* $L_{\overline{\beta}}^{(a')}(\overline{i}, \overline{j}) = \overline{L_\beta^{(a)}(i, j)}$.

Proof. Using binomial expansion,

$$\overline{L_\beta^{(a)}(i, j)} = \sum_{x=0}^{2^{a-1}} \binom{2^{a-1}}{x} ((i)^{1/2^{a-1}})^{2^{a-1}-x} ((\beta j)^{1/2^{a-1}})^x \bmod 2^{a'}. \tag{11}$$

Now, observe that

$$\binom{2^{a-1}}{x} \bmod 2^{a'} = \begin{cases} \binom{2^{a'-1}}{y}, & x = y\cdot 2^{a-a'}, \text{ where } y \text{ is an integer,} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Thus,

$$\overline{L_\beta^{(a)}(i,j)} = \sum_{y=0}^{2^{a'-1}} \binom{2^{a'-1}}{y} ((i)^{1/2^{a-1}})^{2^{a-1}-y\cdot 2^{a-a'}} ((\beta j)^{1/2^{a-1}})^{y\cdot 2^{a-a'}} \bmod 2^{a'}$$

$$= \sum_{y=0}^{2^{a'-1}} \binom{2^{a'-1}}{y} ((\bar{i})^{1/2^{a'-1}})^{2^{a'-1}-y} ((\overline{\beta}\,\overline{j})^{1/2^{a'-1}})^{y} = L_{\overline{\beta}}^{(a')}(\bar{i},\bar{j}). \tag{13}$$

□

Remark 2. When $a' = 1$, the mapping function $L_{\overline{\beta}}^{(1)}(\bar{i},\bar{j}) = \bar{i} + \overline{\beta}\,\bar{j}$ coincides with the mapping function applied to the Galois fields. Since $L_{\overline{\beta}}^{(1)}(\bar{i},\bar{j}) = \overline{L_\beta^{(a)}(i,j)}$ (from Theorem 2), $L_\beta^{(a)}(i,j)$ is unique for a given pair (i,j). It follows that two Latin squares constructed by $L_{\beta_0}^{(a)}(i,j)$ and $L_{\beta_1}^{(a)}(i,j)$, where $\beta_0, \beta_1 \in G_{2^s-1}$ and $\beta_0 \neq \beta_1$, are orthogonal.

Let $R = C = S = G_{2^s-1} \cup \{0\}$. A complete family $\{(R, C, S; L_\beta^{(a)}) : \beta \in G_{2^s-1}\}$ of MOLS is obtained by defining $L_\beta^{(a)}(i,j) = ((i)^{1/(2^{a-1})} + (\beta j)^{1/(2^{a-1})})^{2^{a-1}}$.

Example 4. Let $R = C = S = G_3 \cup \{0\} \subset \mathrm{GR}(2^2, 2)$ and mapping functions $L_1^{(2)}(i,j) = ((i)^{1/2} + j^{1/2})^2$, $L_\alpha^{(2)}(i,j) = ((i)^{1/2} + (\alpha j)^{1/2})^2$ and $L_{\alpha^2}^{(2)}(i,j) = ((i)^{1/2} + (\alpha^2 j)^{1/2})^2$. The resultant MOLS are

$$M_1 = \begin{bmatrix} 0 & 1 & \alpha & \alpha^2 \\ 1 & 0 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 0 & 1 \\ \alpha^2 & \alpha & 1 & 0 \end{bmatrix}, \quad M_\alpha = \begin{bmatrix} 0 & \alpha & \alpha^2 & 1 \\ 1 & \alpha^2 & \alpha & 0 \\ \alpha & 0 & 1 & \alpha^2 \\ \alpha^2 & 1 & 0 & \alpha \end{bmatrix},$$

$$M_{\alpha^2} = \begin{bmatrix} 0 & \alpha^2 & 1 & \alpha \\ 1 & \alpha & 0 & \alpha^2 \\ \alpha & 1 & \alpha^2 & 0 \\ \alpha^2 & 0 & \alpha & 1 \end{bmatrix}, \tag{14}$$

respectively. A complete family of three MOLS is obtained. In addition, the mapping function $L_0(i,j) = i$ yields a matrix

$$M_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ \alpha & \alpha & \alpha & \alpha \\ \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 \end{bmatrix} \tag{15}$$

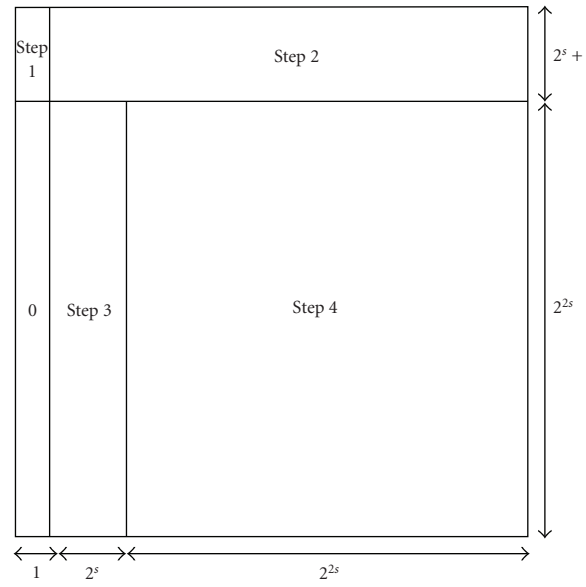


Figure 1: Portion of parity check matrix constructed in each step.

which is orthogonal to each Latin square in the complete family of MOLS.

4. STRUCTURED LDPC CODES OVER $\mathbb{Z}_{2^a}$

4.1. Construction of graphs using latin squares

The construction method proposed in [14, Section IV-A] can be generalized to construct graphs for different values of a and s by altering the mapping functions according to the value of a. The procedure is stated here for easy reference by Theorem 3 that follows. The graph is a tree that has three layers that enumerate from its root; the root is a variable node, the first layer has $2^s + 1$ check nodes, the second layer has $2^s(2^s + 1)$ variable nodes and the third layer has 2^{2s} check nodes. Thus there are $2^{2s} + 2^s + 1$ variable nodes and the same number of check nodes. The connectivity of the nodes are executed in the following steps.

(1) The variable root node is connected to each of the check nodes in the first layer.

(2) Each check node in the first layer is connected to 2^s consecutive variable nodes in the second layer.

(3) Each of the first 2^s variable nodes in the second layer is connected to 2^s consecutive check nodes in the third layer.

(4) For $i, j, k, \beta \in G_{2^s-1} \cup \{0\}$, label the remaining variable nodes in the second layer (β, i) and all check nodes in the third layer (j,k). If $\beta = 0$, variable node $(0,i)$ is connected to check node (j,i). If $\beta \in G_{2^s-1}$, variable node (β,i) is connected to check node $(j, L_\beta^{(a)}(i,j))$. The tree is completed once all possible combinations of (i,j,k,β) are exhausted.

Let $\mathcal{T}(a,s)$ denote the resultant tree constructed using the complete family of MOLS derived from $G_{2^s-1} \cup \{0\} \subseteq \mathcal{R}$. $\mathcal{T}(a,s)$ is a degree-2^s+1 regular tree. By reading the variable (check) nodes as columns (rows) of a matrix $\mathbf{H}(a,s) \in \mathbb{Z}_{2^a}^{(2^{2s}+2^s+1)\times(2^{2s}+2^s+1)}$ in top-bottom, left-right manner while setting the edge weights to be randomly chosen units from $\mathbb{Z}_{2^a}$, the portion of $\mathbf{H}(a,s)$ constructed at each step is illustrated in Figure 1. The null space of $\mathbf{H}(a,s)$ yields an LDPC code $\mathcal{C}(a,s)$ over $\mathbb{Z}_{2^a}$.

Example 5. Let $a = 2$ and $s = 2$. The Latin squares are shown in Example 4. Steps (1)–(3) are illustrated in Figure 2(a). As observed, this can be perceived as the nonrandom portion of the parity-check matrix. Step 4, on the other hand, executes the pseudorandom portion of the parity-check matrix that is commonly seen in most LDPC parity-check matrices. The resultant tree is shown in Figure 2(b).

4.2. Properties of $\mathcal{C}(a,s)$

$\mathcal{C}(a,s)$ is a length $n(s) = 2^{2s} + 2^s + 1$ regular LDPC code represented by $\mathbf{H}(a,s)$ (or $\mathcal{T}(a,s)$). We denote the minimum distance of $\mathcal{C}(a,s)$ as $d_{\min}(a,s)$. Following the definition given in [14], $w_{\min}(a,s)$ denotes the minimum pseudocodeword weight that arises from the Tanner graph of $\mathcal{C}(a,s)$ for the 2^a-ary symmetric channel.

Theorem 3. *Let $\overline{\mathcal{T}(a,s)}$ denote the graph resulting from reducing* mod $2^{a'}$*, all edge weights of $\mathcal{T}(a,s)$. $\mathcal{T}(a',s) = \overline{\mathcal{T}(a,s)}$, that is, $\mathbf{H}(a',s) = \overline{\mathbf{H}(a,s)}$.*

Proof. First, the connection procedure is regardless of a in steps (1)–(3), and similarly for $\beta = 0$ in step (4). Since $L_{\overline{\beta}}^{(a')}(\overline{i},\overline{j}) = \overline{L_{\beta}^{(a)}(i,j)}$ (from Theorem 2), the edge $((\beta,i),(j,L_{\beta}^{(a)}(i,j)))$ in $\mathcal{T}(a,s)$ is equivalent to the edge $((\overline{\beta},\overline{i}),(\overline{j},L_{\overline{\beta}}^{(a')}(\overline{i},\overline{j})))$ in $\mathcal{T}(a',s)$. □

Remark 3. The graphs constructed by setting $a = 1$ yield binary codes that are the same as those in [14, Section IV-A]. Further, it has also been shown that these codes are the binary projective geometry (PG) LDPC codes introduced in [5]. Thus, it is known that $d_{\min}(1,s) = 2^s + 2$.

Before deriving $d_{\min}(a,s)$, we state two relationships between the codewords in $\mathcal{C}(a,s)$ and $\mathcal{C}(a',s)$.

Corollary 1. (i) *If $\mathbf{c} \in \mathcal{C}(a,s)$, then $\overline{\mathbf{c}} \in \mathcal{C}(a',s)$.*

(ii) *If $\mathbf{c} \in \mathcal{C}(a,s)$ can be expressed as $\mathbf{c} = 2^{a-a'}\mathbf{c}'$, where $\mathbf{C} \in \mathbb{Z}_{2^{a'}}^n$, then $\mathbf{c}' \in \mathcal{C}(a',s)$ and is unique.*

Proof. Corollary 1(i) is a simple consequence of Theorem 3; while for Corollary 1(ii),

$$\begin{aligned} 2^{a-a'}\mathbf{c}'\mathbf{H}^T(a,s) &= 0 \mod 2^a \\ &\Longrightarrow \mathbf{c}'\mathbf{H}^T(a,s) = 0 \mod 2^{a'} \\ &\Longrightarrow \mathbf{c}'\mathbf{H}^T(a',s) = 0 \mod 2^{a'} \text{ (from Theorem 3).} \end{aligned} \tag{16}$$

The uniqueness of $\mathbf{c}'$ follows from the natural group embedding, $\mathrm{GR}(2^{a'},s) \to \mathcal{R} : r \mapsto 2^{a-a'}r$. □

Theorem 4. $d_{\min}(a,s) = d_{\min}(1,s)$.

Proof. Let $d_{\mathbf{c}}$ be the Hamming weight of $\mathbf{c} \in \mathcal{C}(a,s) \setminus \{\mathbf{0}\}$.

Case 1. $\mathbf{c}$ contains at least one unit. From Corollary 1(i), when $a' = 1$, $\overline{\mathbf{c}} \in \mathcal{C}(1,s)$. Further, $d_{\mathbf{c}} \geq d_{\overline{\mathbf{c}}}$. If $d_{\overline{\mathbf{c}}} = d_{\min}(1,s)$, $d_{\mathbf{c}} \geq d_{\min}(1,s)$.

Case 2.1. $\mathbf{c}$ can be expressed as $\mathbf{c} = 2^{a-a'}\mathbf{c}'$, where $\mathbf{c}'$ contains at least one unit of $\mathbb{Z}_{2^{a'}}$. From Corollary 1(ii), $\mathbf{c}' \in \mathcal{C}(a',s)$. Further, $d_{\mathbf{c}} = d_{\mathbf{c}'}$ and from Case 1, $d_{\mathbf{c}'} \geq d_{\overline{\mathbf{c}'}}$. When $a' = 1$, $\mathbf{c} = 2^{a-1}\mathbf{c}'$, and $\mathbf{c}' \in \mathcal{C}(1,s)$. If $d_{\mathbf{c}'} = d_{\min}(1,s)$, $d_{\mathbf{c}} = d_{\min}(1,s)$.

Case 2.2. $\mathbf{c}$ can be expressed as $\mathbf{c} = 2^{a-a'}\mathbf{c}'$, where $\mathbf{c}'$ does not contain any unit of $\mathbb{Z}_{2^{a'}}$. Similarly, from Corollary 1(ii), $\mathbf{c}' \in \mathcal{C}(a',s)$. Therefore, $d_{\mathbf{c}} = d_{\mathbf{c}'}$ and the bounds on $d_{\mathbf{c}'}$ follow Case 2.1.

Thus, $d_{\min}(a,s) = d_{\min}(1,s)$. □

It has already been shown in [14, Section IV-A] that $w_{\min}(1,s) = d_{\min}(1,s)$. The following theorem states the relationship between $w_{\min}(a,s)$ and $d_{\min}(a,s)$.

Theorem 5. $w_{\min}(a,s) = d_{\min}(a,s)$.

Proof. Since $\mathcal{T}(1,s) = \overline{\mathcal{T}(a,s)}$ when $a' = 1$ (from Theorem 3) and all edge weights in $\mathcal{T}(a,s)$ are units of $\mathbb{Z}_{2^a}$, $w_{\min}(a,s)$ and $w_{\min}(1,s)$ share the same tree bound [14], that is, $w_{\min}(a,s) \geq 2^s + 2$, for all a. Further, $d_{\min}(a,s) = d_{\min}(1,s) = 2^s + 2$ (from Theorem 4). Thus,

$$\begin{aligned} 2^s + 2 \leq w_{\min}(a,s) \leq d_{\min}(a,s) &= 2^s + 2 \\ \Longrightarrow w_{\min}(a,s) = d_{\min}(a,s) &= 2^s + 2. \end{aligned} \tag{17}$$

□

The code rate $r(a,s)$ has to be computed first by reducing $\mathbf{H}(a,s)$ to the form as discussed in Section 2.1. $r(a,s)$ is bounded by

$$\frac{2^{2s} + 2^s - 3^s}{a(2^{2s} + 2^s + 1)} \leq r(a,s) \leq \frac{2^{2s} + 2^s - 3^s}{2^{2s} + 2^s + 1}, \tag{18}$$

where the upper bound corresponds to the code rates of the binary PG-LDPC codes [5]. We observe that by setting the edge weights of $\mathcal{T}(a,s)$ as randomly chosen units from $\mathbb{Z}_{2^a}$, $r(a,s)$ tends to the lower bound which results in codes suitable for low-rate applications. On the other hand, by setting all edge weights to be unity, $r(a,s)$ increases significantly. The corresponding codes can thus be deployed in moderate-rate applications. Table 1 compiles the properties of $\mathcal{C}(a,s)$ for various values of a and s.

5. SIMULATION RESULTS

Figures 3 and 4 show the bit error rate (BER) and symbol error rate (SER) performance of our structured codes over

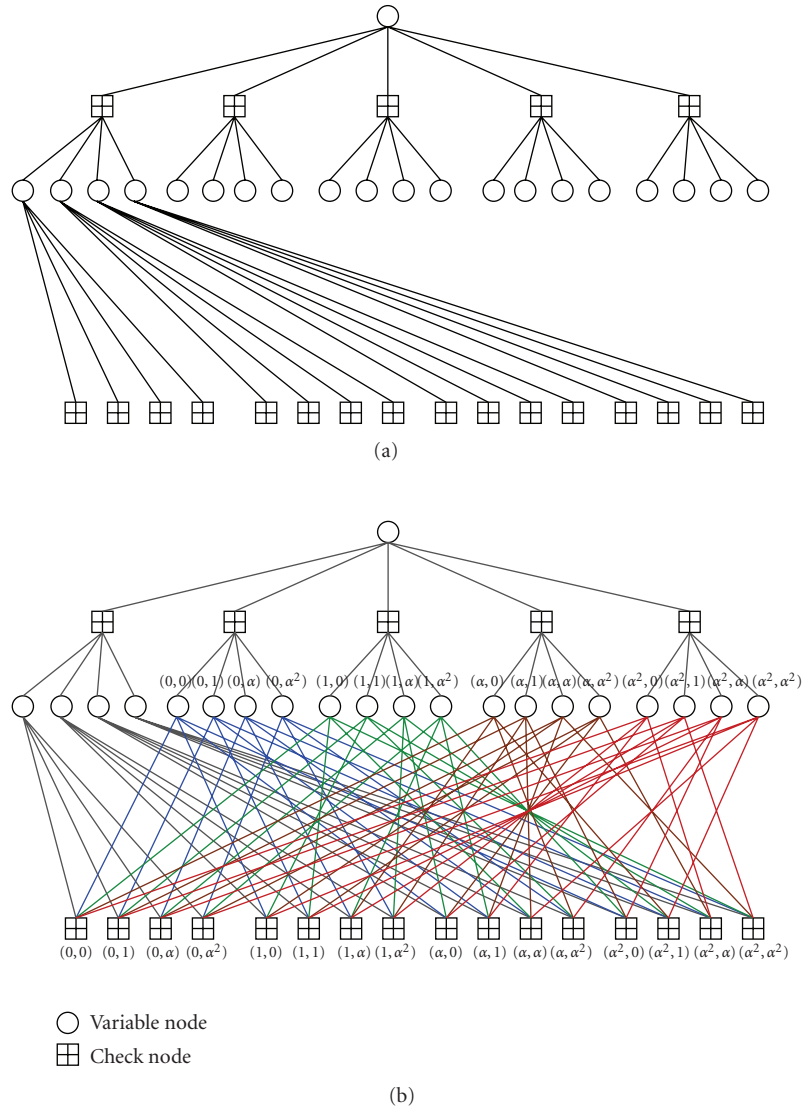


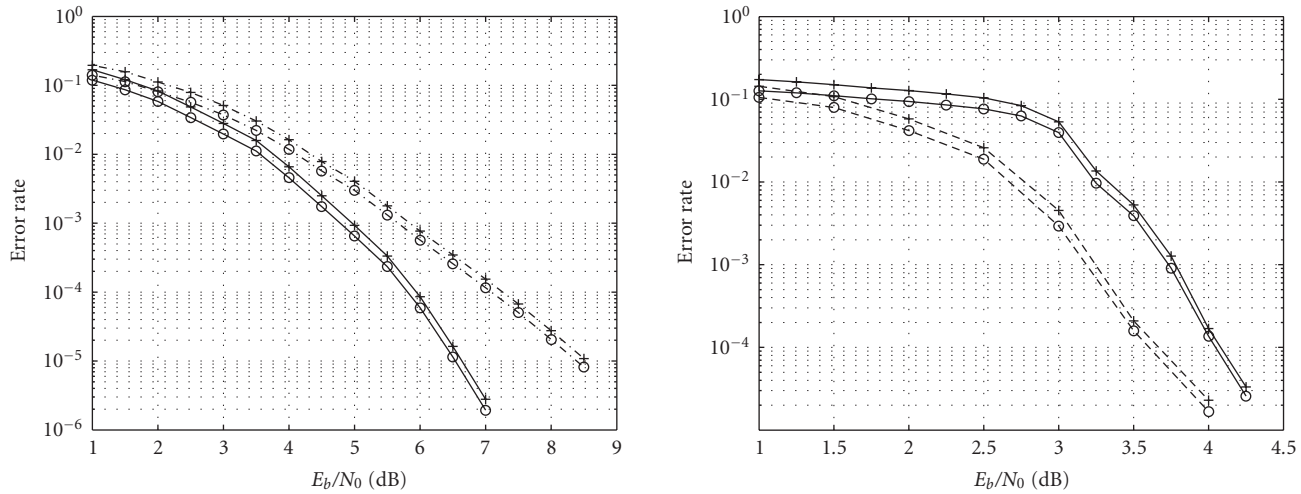
FIGURE 2: Tree constructed for $a = 2$, $s = 2$ after (a) steps (1)–(3), and (b) step (4) (the final structure).

the AWGN channel. In Figure 3(a), the corresponding edge weights of the codes simulated are randomly chosen units of $\mathbb{Z}_4$, while those in Figures 3(b) and 4 are set to unity. The codewords are transmitted using the matched signals discussed in Section 2.2. The received signals are decoded using the sum-product algorithm. The performance of random, near-regular LDPC codes with constant variable node degree of 3, is also shown. These codes have similar codelengths and rates to that of the structured codes. For each data point, 10^4 error bits are obtained for a maximum of 100 iterations allowed for decoding each received signal vector.

Figure 3(a) shows our structured $\mathbb{Z}_4$ code outperforming the random code when the codelength is small, that is, 42 bits. On the other hand, Figure 3(b) shows our structured code performing worse than its random counterpart when the codelength is much larger, specifically, 2114 bits. At a glance, it therefore appears that our structured codes are only better than random codes for short codelengths. To get a clearer picture as to how our codes fair in comparison to their random counterparts, we turn to Figures 4(a) and 4(b) which summarize the BER performance of random and structured codes over $\mathbb{Z}_4$, respectively, $\mathbb{Z}_8$, for increasing codelengths of 21, 146, and 546 bits, respectively, 63, 219, and 819 bits. From these empirical results, we conclude that our codes significantly outperform their random counterparts over a wide BER range for very small codelengths, that is, less than 100 bits. On the other hand, for larger codelengths, random codes perform better in the higher BER region while our structured codes are superior at lower BERs, specifically, 10^{-4} and below for codelengths close to 1000 bits and 10^{-6} and below for larger codelengths, exceeding 2000 bits. This phenomenon may be attributed to the fact that the minimum distance of our codes grow linearly with the square root of

Table 1: Properties of $\mathcal{C}(a, s)$.

a	s	$n(s)$	Degree of $\mathcal{T}(a, s)$	$d_{\min}(a, s) = w_{\min}(a, s)$	$r(a, s)$ (Lower bound)	$r(a, s)$ (Unity edge weights)
1					0.5238	0.5238
2	2	21	5	6	0.2619	0.4762
3					0.1746	0.3175
4					0.1309	0.2381
1					0.6164	0.6164
2	3	73	9	10	0.3082	0.5548
3					0.2055	0.4932
4					0.1541	0.3699
1					0.6996	0.6996
2	4	273	17	18	0.3498	0.6337
3					0.2332	0.5653
4					0.1749	0.4982
1					0.7692	0.7692
2	5	1057	33	34	0.3846	0.7053
3					0.2564	0.6367
4					0.1923	0.5669



(a) $a = 2, s = 2$, random edge weights

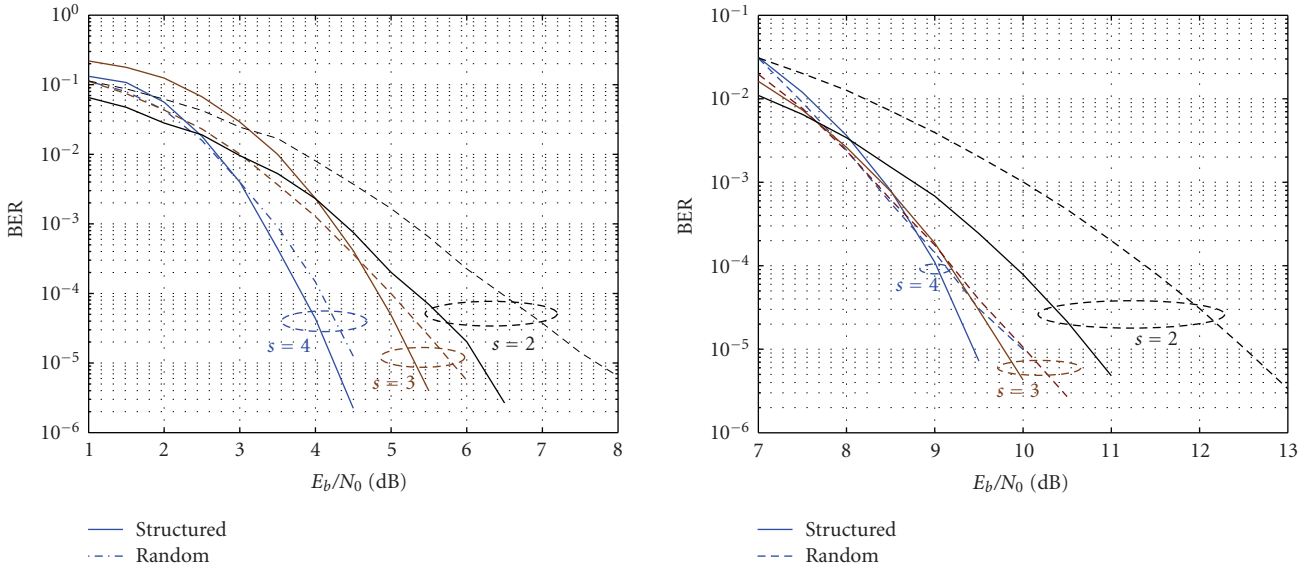
(b) $a = 2, s = 5$, unity edge weights

Figure 3: Performance of structured and random LDPC codes over $\mathbb{Z}_4$ with QPSK signaling over the AWGN channel.

their codelength. On the other hand, from [23, Theorem 26], we have that the minimum distance of a random, regular LDPC code with constant variable node degree of 3 grows linearly with its codelength with high probability. As the random codes considered here are near regular, we believe that they have superior minimum distances compared to our structured codes.

6. CONCLUSION

To summarize, we have extended the notion of Latin squares to multiplicative groups of a Galois ring. Using the generalized mapping function, we have constructed Tanner graphs representing a family of structured LDPC codes over $\mathbb{Z}_{2^a}$ spanning a wide range of code rates. In addition, we



(a) $a = 2$, unity edge weights, transmitted using QPSK signaling

(b) $a = 3$, unity edge weights, transmitted using 8-PSK signaling

Figure 4: Performance of structured and random LDPC codes transmitted using matched signals over the AWGN channel.

have shown that the minimum pseudocodeword weight of these codes are equal to their minimum Hamming distance—a desirable attribute under iterative decoding. Finally, our simulation results show that these codes, when transmitted by matched signal sets over the AWGN channel, can significantly outperform their random counterparts of similar length and rate, at BERs of practical interest.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments which led to significant improvements in Sections 1 and 5 of this paper. The authors also gratefully acknowledge financial support from the Ministry of Education ACRF Tier 1 Research Grant no. R-263-000-361-112.

REFERENCES

[1] M. C. Davey and D. J. C Mackay, "Low density parity check codes over GF(q)," *IEEE Communications Letters*, vol. 2, no. 5, pp. 159–166, 1998.

[2] D. Slepian, "Group codes for the Gaussian channel," *Bell System Technical Journal*, vol. 47, pp. 575–602, 1968.

[3] G. D. Forney Jr., "Geometrically uniform codes," *IEEE Transactions on Information Theory*, vol. 37, no. 5, pp. 1241–1260, 1991.

[4] D. Sridhara and T. E. Fuja, "LDPC codes over rings for PSK modulation," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3209–3220, 2005.

[5] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2711–2736, 2001.

[6] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1156–1176, 2004.

[7] I. B. Djordjevic and B. Vasic, "Nonbinary LDPC codes for optical communication systems," *IEEE Photonics Technology Letters*, vol. 17, no. 10, pp. 2224–2226, 2005.

[8] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 549–583, 2006.

[9] X.-Y. Hu and E. Eleftheriou, "Binary representation of cycle Tanner-graph GF(2^b) codes," in *Proceedings of IEEE International Conference on Communications (ICC '04)*, vol. 1, pp. 528–532, Paris, France, June 2004.

[10] C. Poulliat, M. Fossorier, and D. Declercq, "Using binary image of nonbinary LDPC codes to improve overall performance," in *Proceedings of IEEE International Symposium on Turbo Codes*, Munich, Germany, April 2006.

[11] C. A. Kelley, D. Sridhara, and J. Rosenthal, "Pseudocodeword weights for non-binary LDPC codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1379–1383, Seattle, Wash, USA, July 2006.

[12] R. Koetter and P. O. Vontobel, "Graph-covers and iterative decoding of finite length codes," in *Proceedings of the 3rd IEEE International Symposium on Turbo Codes and Applications*, pp. 75–82, Brest, France, September 2003.

[13] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. thesis, Linköping University, Linköping, Sweden, 1996.

[14] C. A. Kelley, D. Sridhara, and J. Rosenthal, "Tree-based construction of LDPC codes having good pseudocodeword weights," *IEEE Transactions on Information Theory*, vol. 53, no. 4, pp. 1460–1478, 2007.

[15] I. B. Djordjevic and B. Vasic, "MacNeish-Mann theorem based iteratively decodable codes for optical communication systems," *IEEE Communications Letters*, vol. 8, no. 8, pp. 538–540, 2004.

[16] O. Milenkovic and S. Laendner, "Analysis of the cycle-structure of LDPC codes based on Latin squares," in *Proceedings of IEEE International Conference on Communications (ICC '04)*, vol. 2, pp. 777–781, Paris, France, June 2004.

[17] B. Vasic, I. B. Djordjevic, and R. K. Kostuk, "Low-density parity check codes and iterative decoding for long-haul optical communication systems," *Journal of Lightwave Technology*, vol. 21, no. 2, pp. 438–446, 2003.

[18] I. B. Djordjevic and B. Vasic, "Iteratively decodable codes from orthogonal arrays for optical communication systems," *IEEE Communications Letters*, vol. 9, no. 10, pp. 924–926, 2005.

[19] O. Milenkovic, N. Kashyap, and D. Leyba, "Shortened array codes of large girth," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3707–3722, 2006.

[20] G. Caire and E. M. Biglieri, "Linear block codes over cyclic groups," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1246–1256, 1995.

[21] H. A. Loeliger, "Signal sets matched to groups," *IEEE Transactions on Information Theory*, vol. 37, no. 6, pp. 1675–1682, 1991.

[22] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, Cambridge, UK, 2nd edition, 2001.

[23] G. Como an F. Fagnani, "Average spectra and minimum distances of low density parity check codes over cyclic groups," http://calvino.polito.it/~fagnani/groupcodes/ldpcgroupcodes.pdf.