# Structured LDPC Codes from Permutation Matrices Free of Small Trapping Sets 

Dung Viet Nguyen, Bane Vasić, Michael Marcellin<br>Department of Electrical and Computer Engineering<br>University of Arizona, Tucson, AZ 85721<br>Email: \{nguyendv, vasic, marcellin\} @ece.arizona.edu

Shashi Kiran Chilappagari<br>Marvell Semiconductor Inc,<br>5488 Marvell Lane, Santa Clara, CA 95054<br>Email: shashickiran@gmail.com


#### Abstract

This paper introduces a class of structured lowdensity parity-check (LDPC) codes whose parity check matrices are arrays of permutation matrices. The permutation matrices are obtained from Latin squares and form a finite field under some matrix operations. They are chosen so that the Tanner graphs do not contain subgraphs harmful to iterative decoding algorithms. The construction of column-weight-three codes is presented. Although the codes are optimized for the Gallager A/B algorithm over the binary symmetric channel (BSC), their error performance is very good on the additive white Gaussian noise channel (AWGNC) as well.


## I. Introduction

It is now well established that the error floor phenomenon, an abrupt degradation in the error performance of low-density parity-check (LDPC) codes [1] in the high signal-to-noise-ratio (SNR) region, is due to the presence of certain structures in the Tanner graph that lead to decoder failures. For iterative decoding, these structures are known as trapping sets.

Ideally, LDPC codes should be designed so that their Tanner graphs do not contain most harmful trapping sets, but unfortunately, except for the binary erasure channel, trapping sets for other channels such as the BSC or the AWGNC, are only partially understood. Consequently, many existing methods of constructing LDPC codes only attempt to maximize the girth of the Tanner graphs [2]-[4] or avoid subgraphs that are believed to be harmful [5]. In the latter approach, the subgraphs are identified either by computer simulation or hardware emulation, or are conveniently defined to make the search easier. The problem in these approaches lies in the underlying assumption about harmfulness, which is not proven or is restricted to specific cases.

In this paper, LDPC codes are constructed so that their Tanner graphs do not contain trapping sets of the Gallager A/B algorithm on the BSC. The code construction utilizes the Trapping Set Ontology (TSO) given by Vasic et. al. in [6]. This database contains trapping sets for the Gallager A/B algorithms which are organized based on their topological relations. Our approach relies greatly on the relative harmfulnes. ${ }^{1}$ of different trapping sets to determine which trapping

[^0]sets should not be present in the Tanner graphs of codes. The choice of which trapping sets to avoid is critical to the error performance and the code rate.

Although the codes are optimized for the Gallager A/B algorithm over the BSC, experimental results indicate that their error performance under other iterative algorithms such as the sum product algorithm (SPA) are also extremely good. The explanation for this is based on the observation by Chilappagari et. al. in [8] that the decoding failures for various decoding algorithms and channels are closely related and subgraphs responsible for these failures share some common underlying topological structures. These structures are either trapping sets for iterative decoding algorithms on the BSC or larger subgraphs containing these trapping sets.

The above approach can be incorporated into many existing classes of LDPC codes to result in codes with good error performance (e.g., [9], [10]), including the new class of structured LDPC codes that we propose in this paper.

To be efficiently encodable and decodable, LDPC codes must be structured. An important class of structured LDPC codes are quasi-cyclic ( QC ) codes. In the past few years, numerous QC constructions have been proposed. They can be broadly classified as algebraic [9], [11]-[13] and combinatorial [14], [15]. In this paper, we give a class of structured LDPC codes whose parity check matrices are arrays of permutation matrices. Our design is motivated by the work of Lan et. al. [9]. In [9], the authors give a general algebraic construction of QC-LDPC codes based on a one-to-one correspondence between an element of the multiplicative group of $\operatorname{GF}(q)$ and a circulant permutation matrix of size $(q-1) \times(q-1)$. In our construction, the set of permutation matrices together with some matrix operations (introduced later in this paper) form a field isomorphic to $\mathrm{GF}(q)$. Our permutation matrices are similar to circulants in the sense that the set of $q-1$ nonidentity permutation matrices form a cyclic group, but they are more general as the circulant property holds on indices understood as elements of $\mathrm{GF}(q)$. More specifically, the permutation corresponding to $\alpha^{t}$ sends the indices $\left(0,1, \alpha, \ldots, \alpha^{q-2}\right)$ to $\left(0+\alpha^{t}, 1+\alpha^{t}, \alpha+\alpha^{t}, \ldots, \alpha^{q-2}+\alpha^{t}\right)$.

The construction allows for a systematic reduction of error floors. We present a construction algorithm which recursively builds the parity check matrix by adding permutation matrices. The algorithm ensures that after each step, the corresponding

Tanner graph does not contain certain trapping sets defined in the TSO.

The rest of the paper is organized as follows. In Section [I]. we provide the background related to LDPC codes and the necessary preliminaries for the construction method. In Section III, we give the general definition of the class of structured LDPC codes from permutation matrices. In Section IV. we present the construction of codes based on Latin squares obtained from the additive group of a Galois field. In Section V , we describe the construction algorithm and present the construction of several column-weight-three codes. Finally, we conclude the paper in Section VI.

## II. Preliminaries

In this section, we introduce the definitions and notation used throughout the paper.

## A. LDPC Codes and Trapping Sets

Let $\mathcal{C}$ denote an $(n, k)$ LDPC code over the binary field $\mathrm{GF}(2) . \mathcal{C}$ is defined by the null space of $H$, an $m \times n$ parity check matrix of $\mathcal{C} . H$ is the bi-adjacency matrix of $G$, a Tanner graph representation of $\mathcal{C} . G$ is a bipartite graph with two sets of nodes: $n$ variable (bit) nodes $V=\{1,2, \ldots, n\}$ and $m$ check (constraint) nodes $C=\{1,2, \ldots, m\}$. The length of the shortest cycle in the Tanner graph $G$ is called the girth $g$ of $G$.

A trapping set for an iterative decoding algorithm is defined as a non-empty set of variable nodes in $G$ that are not eventually corrected by the decoder [16]. A trapping set $\mathcal{T}$ is called an $(a, b)$ trapping set if it contains $a$ variable nodes and the subgraph induced by these variable nodes has $b$ odd degree check nodes.

## B. Permutation Matrices from Latin Squares

A permutation matrix is a square binary matrix that has exactly one entry 1 in each row and each column and 0's elsewhere. Our construction makes use of permutation matrices that do not have 1's in common positions. These sets of permutation matrices can be obtained conveniently from Latin squares.

A Latin square of size $q$ (or order $q$ ) is a $q \times q$ array in which each cell contains a single symbol from a $q$-set $S$, such that each symbol occurs exactly once in each row and exactly once in each column. A Latin square of size $q$ is equivalent to the Cayley table of a quasigroup $\mathcal{Q}$ on $q$ elements (see [17, pp. 135-152] for details).

For mathematical convenience, we use elements of $\mathcal{Q}$ to index the rows and columns of Latin squares and permutation matrices. Let $\mathcal{L}=\left[l_{i, j}\right]_{i, j \in \mathcal{Q}}$ denote the Latin square defined on the Cayley table of a quasigroup $(\mathcal{Q}, \oplus)$ of order $q$. Define $f$, an injective map from $\mathcal{Q}$ to $\operatorname{Mat}(q, q, \operatorname{GF}(2))$, the set of matrices of size $q \times q$ over $\mathrm{GF}(2)$, as follows:

$$
\begin{aligned}
f: \mathcal{Q} & \rightarrow \operatorname{Mat}(q, q, \operatorname{GF}(2)) \\
\alpha & \mapsto f(\alpha)=\left[m_{i, j}\right]_{i, j \in \mathcal{Q}}
\end{aligned}
$$

such that:

$$
m_{i, j}= \begin{cases}1 & \text { if } l_{i, j}=\alpha \\ 0 & \text { if } l_{i, j} \neq \alpha\end{cases}
$$

It follows from the above definition that the images of elements of $\mathcal{Q}$ under $f$ give a set of $q$ permutation matrices that do not have 1's in common positions.

## III. LDPC Codes as Array of Permutation Matrices

In this section, we give the general definition of LDPC codes whose parity check matrices are arrays of permutation matrices. Let $\mathcal{W}=\left[w_{i, j}\right]_{1 \leq i \leq \mu, 1 \leq j \leq \eta}$ be an $\mu \times \eta$ matrix over a quasigroup $\mathcal{Q}$,

$$
\mathcal{W}=\left[\begin{array}{cccc}
w_{1,1} & w_{1,2} & \cdots & w_{1, \eta}  \tag{1}\\
w_{2,1} & w_{1,2} & \cdots & w_{2, \eta} \\
\vdots & \vdots & \ddots & \vdots \\
w_{\mu, 1} & w_{\mu, 2} & \cdots & w_{\mu, \eta}
\end{array}\right]
$$

With some abuse of notation, let $\mathcal{H}=f(\mathcal{W})=\left[f\left(w_{i, j}\right)\right]$ be an array of permutation matrices, obtained by replacing elements of $\mathcal{W}$ with their images under $f$, i.e.,

$$
\mathcal{H}=\left[\begin{array}{cccc}
f\left(w_{1,1}\right) & f\left(w_{1,2}\right) & \cdots & f\left(w_{1, \eta}\right)  \tag{2}\\
f\left(w_{2,1}\right) & f\left(w_{1,2}\right) & \cdots & f\left(w_{2, \eta}\right) \\
\vdots & \vdots & \ddots & \vdots \\
f\left(w_{\mu, 1}\right) & f\left(w_{\mu, 2}\right) & \cdots & f\left(w_{\mu, \eta}\right)
\end{array}\right]
$$

Then $\mathcal{H}$ is a binary matrix of size $\mu q \times \eta q$. The null space of $\mathcal{H}$ gives an LDPC code of length $\eta q$. The column weight and row weight of $\mathcal{H}$ are $\mu$ and $\eta$, respectively.

## IV. Structured LDPC Codes from Galois Fields of Permutation Matrices

## A. Galois Fields of Permutation Matrices

Consider the Galois field $\mathrm{GF}(q)$, where $q$ is a power of a prime. Let $\alpha$ be a primitive element of $\operatorname{GF}(q)$. The powers of $\alpha, \alpha^{-\infty} \triangleq 0, \alpha^{0}=1, \alpha, \alpha^{2}, \ldots, \alpha^{q-2}$, give all the $q$ elements of $\operatorname{GF}(q)$ and $\alpha^{q-1}=1$. Let $\mathcal{L}=\left[l_{i, j}\right]_{i, j \in \mathrm{GF}(q)}$ denote the Latin square defined by the Cayley table of the quasigroup given by the set $\left\{0,1, \alpha, \ldots, \alpha^{q-2}\right\}$ together with the subtractive operation of $\mathrm{GF}(q)$, i.e., $l_{i, j}=i-j$. Let $\mathcal{M}=\left\{M_{-\infty}, M_{0}, M_{1}, \ldots, M_{q-2}\right\}$ be the set of images of elements of $\mathrm{GF}(q)$ under $f$, i.e., $M_{t}=\left[m_{i, j}^{(t)}\right]_{i, j \in \mathrm{GF}(q)}=f\left(\alpha^{t}\right)$. It is easy to see that $M_{-\infty}=I$, the $q \times q$ identity matrix. To show that $\mathcal{M}$ forms a field isomorphic to $\operatorname{GF}(q)$ under the matrix operations defined below, we give the following propositions. Due to page limitations, the proofs are omitted.

Proposition 1: For all $t_{1}, t_{2} \in \mathbb{Z}, f\left(\alpha^{t_{1}}+\alpha^{t_{2}}\right)=M_{t_{1}} M_{t_{2}}$.
Proposition 2: For all $t \geq 0, M_{t+1}=P M_{t} Q$, where $P \in$ $\mathcal{M}_{q \times q}[\mathrm{GF}(2)]$ is given as

$$
P=\left[\begin{array}{cccccc}
1 & 0 & 0 & \cdots & 0 & 0  \tag{3}\\
0 & 0 & 0 & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0
\end{array}\right]
$$

and $Q$ is the transpose of $P$.
Define the addition $\boxplus$ and the multiplication $\square$ on $\mathcal{M}$ as:

$$
\begin{aligned}
M_{t_{1}} \boxplus M_{t_{2}} & =M_{t_{1}} M_{t_{2}} \\
M_{t_{1}} \boxtimes M_{t_{2}} & =P^{\left(t_{2}-t_{1}\right)} M_{t_{1}} Q^{\left(t_{2}-t_{1}\right)} \\
& =P^{\left(t_{1}-t_{2}\right)} M_{t_{2}} Q^{\left(t_{1}-t_{2}\right)}
\end{aligned}
$$

then it can be shown that $\mathcal{M}$ together with $\boxplus$ and $\square$ form a field isomorphic to $\operatorname{GF}(q)$.

## B. LDPC Codes from Galois Fields of Permutation Matrices

Define $\mathcal{W}$ and $\mathcal{H}$ as in (1) and (2), where $\mathcal{Q}$ is the set $\left\{0,1, \alpha, \ldots, \alpha^{q-2}\right\}$ together with the subtractive operation of $\mathrm{GF}(q)$. The following theorem gives the necessary and sufficient condition on $\mathcal{W}$, such that the Tanner graph corresponding to $\mathcal{H}$ has girth at least 6 .

Theorem 1 (Cross-addition Constraint): The Tanner graph corresponding to $\mathcal{H}$ contains no cycle of length four iff $w_{i_{1}, j_{1}}+$ $w_{i_{2}, j_{2}} \neq w_{i_{1}, j_{2}}+w_{i_{2}, j_{1}}$ for any $1 \leq i_{1}, i_{2} \leq \mu, 1 \leq j_{1}, j_{2} \leq \eta$, $i_{1} \neq i_{2}, j_{1} \neq j_{2}$.

Proof: The proof is omitted due to page limitations.
It can be seen that the construction of LDPC codes with girth at least 6 from a Galois field of permutation matrices reduces to the finding of a matrix $\mathcal{W}$ that satisfies the crossaddition constraint. One form of $\mathcal{W}$ that satisfies the crossaddition constraint is given by

$$
\mathcal{W}=\left[\begin{array}{ccccc}
0 & 0 & 0 & \cdots & 0  \tag{4}\\
0 & 1 & \alpha & \cdots & \alpha^{q-2} \\
0 & \alpha & \alpha^{2} & \cdots & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \alpha^{q-2} & 1 & \cdots & \alpha^{q-3}
\end{array}\right]
$$

Let $\mathcal{H}=f(\mathcal{W})$. From Proposition 2, it follows that $\mathcal{H}$ has the following structure:

$$
\mathcal{H}=\left[\begin{array}{ccccc}
I & I & I & \cdots & I  \tag{5}\\
I & M_{0} & M_{1} & \cdots & M_{q-2} \\
I & M_{1} & M_{2} & \cdots & M_{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
I & M_{q-2} & M_{0} & \cdots & M_{q-3}
\end{array}\right]
$$

where $M_{t}=P^{t} M_{0} Q^{t}$ and $I$ is the $q \times q$ identity matrix. $\mathcal{H}$ is an array of permutation matrices from $\mathcal{M}$ and is a $q^{2} \times q^{2}$ matrix over $\operatorname{GF}(q)$ with both row and column weights $q$. Since $\mathcal{W}$ satisfies the cross-addition constraint, the Tanner graph corresponding to $\mathcal{H}$ contains no cycle of length 4 .

For any pair $(\gamma, \rho)$ of positive integers with $1 \leq \gamma, \rho \leq q$, let $H$ be a $\gamma \times \rho$ subarray of $\mathcal{H}$. Then $H$ is a $\gamma q \times \rho q$ matrix over $\operatorname{GF}(2)$ which is also free of cycles of length $4 . H$ has constant column $\gamma$ and row weight $\rho$. The null space of $H$ gives a regular structured LDPC code $\mathcal{C}$ of length $\rho q$ with rate at least $(\rho-\gamma) / \rho[1]$.

## Remarks:

- The matrix $\mathcal{W}$ in (4) is obtained by adding a row and a column of all zeros to $\mathcal{L}$, where $\mathcal{L}$ is the Latin square obtained from the Cayley table of the multiplicative group of $\mathrm{GF}(q)$.
- The codes given in this paper can be alternatively defined on integer latices. Since array LDPC codes introduced by Fan in [10], can be defined on integer lattices as shown in [18], they are special cases of the codes given in this paper. If $q$ is a prime, then the parity check matrices of array LDPC codes are subarrays of $f\left(\mathcal{W}_{p}\right)$, where $\mathcal{W}_{p}$ is obtained by permuting rows and columns of $\mathcal{W}$ (in (4)). The codes by Lan et. al. [9] based on the additive groups of prime fields are also array LDPC codes.
- Our class of codes is also different from codes given by Gabidulin et. al in [19] (except for codes based on prime fields, for which the latter become array codes). In [19], permutation matrices of size $q \times q$, where $q$ is a power of a prime, are obtained from the Tensor product of circulant matrices of size $p \times p$, thus are different from the permutation matrices in (5).
- If $\mathcal{L}$ is defined by the Cayley table of the multiplicative group of $\mathrm{GF}(q)$, then circulant permutation matrices of size $(q-1) \times(q-1)$ are obtained as images of elements of $\operatorname{GF}(q) \backslash\{0\}$. In such case, the necessary and sufficient condition on $\mathcal{W}$ such that the Tanner graph corresponding to $\mathcal{H}$ has girth at least 6 is called the cross-multiplication constraint. This condition can be obtained from the crossaddition constraint by replacing addition with multiplication. This gives an alternative description for the codes described in [9].


## V. Construction of Codes Free of Small Trapping SETS

The description of the class of LDPC codes given in the previous sections along with Theorem 1 allow us to construct codes by progressively building the Tanner graphs. The construction is performed by an algorithm which forms the matrix $\mathcal{W}$ in (1). The algorithm is based on a check and select-or-disregard procedure. Let $\tau$ specify which graphical structures should not be present in the Tanner graph $G$. For example, Figure 1 shows the subgraphs induced by some small trapping sets. $\tau$ may specify the girth of $G$ and may also specify the minimum distance of the code. For column-weightthree codes, all possible codewords of even weight less than 12 are known and their induced subgraphs are listed in the TSO. It is simple to check the Tanner graph for cycles of length four thanks to Theorem 1. Finding girth of the Tanner graph can be done in polynomial time using the Dijkstra or Bellman-Ford algorithm, while enumerating cycles of a given length using a standard tree-based algorithm has linear complexity in the code length [6]. An efficient search of the Tanner graph for trapping sets relies on the topological relations among them and carefully analyzing the induced subgraphs. Details on the graph searching techniques will be given in the journal version of this paper.

The Tanner graph of a code is built progressively in $\rho$ stages, where $\rho$ is the row weight of the parity check matrix. Usually, $\rho$ is not pre-specified, and codes are constructed to have rate as high as possible. At each stage, a set of $q$ variable nodes are introduced, initially not connected to check nodes of the


Fig. 1. (a) $(5,3)$ trapping set of girth $6,(\mathrm{~b})(5,3)$ trapping set of girth 8 and (c) ( 6,4 ) trapping set. We use o represent variable nodes, $\square$ to represent odd degree check nodes and $\square$ to represent even degree check nodes.

Tanner graph. Blocks of edges are then added to connect the new variable nodes and the check nodes. Each block of edges corresponds to a permutation matrix and hence corresponds to an element of $\operatorname{GF}(q)$. An element of $\operatorname{GF}(q)$ may be chosen randomly, or it may be chosen in a predetermined order. After a block of edges is added, the Tanner graph is checked for condition $\tau$. If the condition $\tau$ is violated, then that block of edges is removed and replaced by a different block. The algorithm proceeds until no block of edges can be added without violating condition $\tau$. It can be seen that the algorithm is a combination of the progressive edge growth algorithm for constructing random LDPC codes [20] and the method in [18].

The complexity of the algorithm grows exponentially with the column weight. The complexity also depends greatly on how condition $\tau$ is checked on a Tanner graph. However, for small column weights, say 3 or 4 , and small to moderate code lengths, the algorithm is well handled by state-of-theart computers. For example, the construction of a $(1111,808)$ code which has girth 8, minimum distance at least 10 and which contains no $(5,3)$ trapping set given in Figure 1|(b) takes less than 2 minutes on a 2.4 GHz computer.

We continue this section by providing two examples of construction of column-weight-three codes whose Tanner graphs do not contain small trapping sets described in the TSO.

Example 1: Let $q=53$ and let $\mathcal{C}_{8}^{(\text {new })}$ denote the code obtained when $\tau$ is defined as a condition that the corresponding Tanner graph of $\mathcal{C}_{8}^{(\text {new })}$ has girth $g \geq 8 . \mathcal{C}_{8}^{(\text {new })}$ is a $(530,371)$ LDPC code with rate $R=0.7$. Let $\mathcal{C}_{8}^{\text {(old) }}$ denote the (530, 371) shortened array code (or integer lattice code described in [18]). $\mathcal{C}_{8}^{\text {(old) }}$ is obtained by extensive computer search and has the maximum possible rate of $R=0.7$. Denote by $\mathcal{C}_{\mathrm{d}_{10}}$ the code obtained when $\tau$ is such that the minimum distance of $\mathcal{C}_{\mathrm{d}_{10}}$ is at least 10. $\mathcal{C}_{\mathrm{d}_{10}}$ is constructed by avoiding codewords of weight 6 and 8 in the Tanner graph (the TSO lists two possible codewords of weight 6 and five possible codewords of weight 8 for codes with $g \geq 6) . \mathcal{C}_{\mathrm{d}_{10}}$ is a $(795,636)$ LDPC code with rate $R=0.8$ and girth $g=6$. The error performance of $\mathcal{C}_{8}^{\text {(new) }}, \mathcal{C}_{8}^{\text {(old) }}$ and $\mathcal{C}_{\mathrm{d} 10}$ under the SPA with a maximum of 50 iterations over the AWGNC is shown in Figure 2. It can be seen that the error performance of $\mathcal{C}_{8}^{\text {(new) }}$ is better than that of $\mathcal{C}_{8}^{(\text {old })}$. One possible explanation for this observation is that the Tanner graph of $\mathcal{C}_{8}^{(\text {old })}$ contains


Fig. 2. Performances of the codes given in Example 1 over the AWGNC.
subgraphs induced by the codeword of weight 6 while the minimum distance of $\mathcal{C}_{8}^{(\text {new })}$ is 10 . Allowing the Tanner graph of $\mathcal{C}_{\mathrm{d}_{10}}$ to have girth 6 but requiring that the minimum distance is at least 10 results in higher rate than the rate of $\mathcal{C}_{8}^{(\text {new })}$, while maintaining the good error performance. This example clearly demonstrates that larger girth alone does not necessarily lead to better performance. We also remark that although minimum distance is used as the design parameter to construct $\mathcal{C}_{\mathrm{d}_{10}}$ with good error performance, in general this may not be sufficient to guarantee low error floors, since codes with high minimum distance may still contain trapping sets.

Example 2: Let $q=19^{2}=361$. The codes obtained when conditions $\tau_{1}, \tau_{2}, \tau_{3}$ and $\tau_{4}$ are imposed are denoted with $\mathcal{C}_{1}$, $\mathcal{C}_{2}, \mathcal{C}_{3}$ and $\mathcal{C}_{4}$, respectively. Conditions $\tau_{1}, \tau_{2}, \tau_{3}$ and $\tau_{4}$ are defined as

- $\tau_{1}: G$ has girth $g \geq 10$.
- $\tau_{2}: G$ has girth $g \geq 8 ; G$ does not contain the $(5,3)$ trapping set of girth 8 shown in Figure 11(b), and $G$ does not contain the $(6,4)$ trapping set shown in Figure $11(\mathrm{c})$
- $\tau_{3}: G$ has girth $g \geq 8 ; G$ does not contain the $(5,3)$ trapping set of girth 8 and an eight cycle in $G$ can share 2 variable nodes with at most one other eight cycle.
- $\tau_{4}$ : $G$ has girth $g \geq 6 ; G$ does not contain the $(5,3)$ trapping set of girth 6 shown in Figure [1|(a), $G$ does not contain the $(5,3)$ trapping set of girth 8 ; and an eight cycle in $G$ can share 2 variable nodes with at most one other eight cycle.

The Tanner graphs of these codes contain 361 check nodes. $\mathcal{C}_{1}, \mathcal{C}_{2}, \mathcal{C}_{3}$ and $\mathcal{C}_{4}$ have lengths $n_{1}=2888, n_{2}=3249, n_{3}=$ $3610, n_{4}=3971$ and rates $R_{1}=0.63, R_{2}=0.67, R_{3}=$ $0.70, R_{4}=0.73$. The error performance of $\mathcal{C}_{1}, \mathcal{C}_{2}, \mathcal{C}_{3}$ and $\mathcal{C}_{4}$ under the SPA with a maximum of 50 iterations over the


Fig. 3. Performances of the codes given in Example 2 over the AWGNC.

## AWGNC is shown in Figure 3 .

It can be seen that the conditions $\tau_{1}, \tau_{2}, \tau_{3}$ and $\tau_{4}$ are successively weaker. Since stronger conditions usually lead to codes with lower rates, we can observe in this example that $R_{1}<R_{2}<R_{3}<R_{4}$. From the simulation results, we see no loss in the error performance of codes with weaker conditions. This example emphasizes the importance of properly identifying the trapping sets to be avoided in the Tanner graph since it is crucial for the rate and the error performance of the code.

Remark: Conditions $\tau_{3}$ and $\tau_{4}$ permit an eight cycle in $G$ to share 2 variable nodes with at most one other eight cycle. Consequently, $(6,4)$ trapping sets can be present in the Tanner graphs but their variable nodes can be involved in at most two eight cycles. Therefore many children of the $(6,4)$ trapping set are avoided (see [6] for more details).

## VI. DISCUSSION AND CONCLUSIONS

We have introduced a class of structured LDPC codes with a wide range of rates and lengths. The code description is based on Latin squares, hence they can be explained both algebraically or combinatorially. Moreover, the description allows a code construction by progressively building the Tanner graph. The Tanner graph is built so that it does not contain a predefined set of trapping sets of iterative decoding algorithms. In this paper, we rely on the TSO - a database of trapping sets for the Gallager A/B algorithm on the BSC. Our conjecture is that trapping sets for other iterative decoding algorithms such as the SPA must contain trapping sets for the Gallager $\mathrm{A} / \mathrm{B}$ algorithm. By eliminating trapping sets listed in the TSO, the codes have good error performance when decoded by other iterative decoding algorithms on the BSC or AWGNC. Although we could not provide enough experimental results for comparison with existing codes due to page limitations, to the best of our knowledge, our codes outperform the best
known short length structured LDPC codes. Our current and future work includes identifying trapping sets for various decoding algorithms over the BSC and AWGNC, with the TSO as a starting point.

## VII. Acknowledgment

This work was funded by NSF under the grants IHCS0725403, CCF-0634969, CCF-0830245.

## REFERENCES

[1] R. G. Gallager, Low Density Parity Check Codes. Cambridge, MA: M.I.T. Press, 1963.
[2] O. Milenkovic, N. Kashyap, and D. Leyba, "Shortened array codes of large girth," IEEE Trans. Inf. Theory, vol. 52, no. 8, pp. 3707-3722, Aug. 2006.
[3] Y. Wang, J. Yedidia, and S. Draper, "Construction of high-girth QCLDPC codes," in Proc. 5th Int. Symp. on Turbo Codes and Related Topics, Sept. 2008, pp. 180-185.
[4] S. Kim, J.-S. No, H. Chung, and D.-J. Shin, "Quasi-cyclic low-density parity-check codes with girth larger than 12," IEEE Int. Symp. Inf. Theory, vol. 53, no. 8, pp. 2885-2891, Aug. 2007.
[5] L. Dolecek, Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic, "Analysis of absorbing sets for array-based LDPC codes," in Proc. Int. Conf. on Commun., Galsgow, Scotland, June 2007, pp. 6261-6268.
[6] B. Vasic, S. Chilappagari, D. Nguyen, and S. Planjery, "Trapping set ontology," in Proc. 47th Annual Allerton Conf. on Commun., Control and Computing, Sept. 2009, pp. 1-7.
[7] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error floors of LDPC codes on the binary symmetric channel," in Proc. Int. Conf. on Comтии., vol. 3, 2006, pp. 1089-1094.
[8] S. K. Chilappagari, M. Chertkov, M. G. Stepanov, and B. Vasic, "Instanton-based techniques for analysis and reduction of error floors of LDPC codes," IEEE JSAC on Capacity Approaching Codes, vol. 27, no. 6, pp. 855-865, Aug. 2009.
[9] L. Lan, L. Zeng, Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: a finite field approach," IEEE Trans. Inf. Theory, vol. 53, no. 7, pp. 2429-2458, Jul. 2007.
[10] J. L. Fan, "Array codes as low-density parity-check codes," in Proc. 2nd Int. Symp. Turbo Codes and Related topics, Sept. 2000, pp. 543-546.
[11] R. M. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," in Proc. ISTA, 2001.
[12] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, "A class of lowdensity parity-check codes constructed based on Reed-Solomon codes with two information symbols," IEEE Commun. Lett., vol. 7, no. 7, pp. 317-319, Jul. 2003.
[13] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-Shannon-limit quasicyclic low-density parity-check codes," IEEE Trans. Commun., vol. 52, no. 7, pp. 1038-1042, Jul. 2004.
[14] Y. Kou, S. Lin, and M. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," IEEE Trans. Inf. Theory, vol. 47, no. 7, pp. 2711 -2736, Nov. 2001.
[15] B. Vasic and O. Milenkovic, "Combinatorial constructions of lowdensity parity-check codes for iterative decoding," IEEE Trans. Inf. Theory, vol. 50, no. 6, pp. 1156-1176, Jun. 2004.
[16] T. J. Richardson, "Error floors of LDPC codes," in Proc. 41st Annual Allerton Conf. on Commun., Control and Computing, Sept. 2003, pp. 1426-1435. [Online]. Available: http://www.hpl.hp.com/personal/ Pascal_Vontobel/pseudocodewords/papers
[17] C. J. Colbourn and J. H. Dinitz, Handbook of combinatorial designs, second edition (Discrete mathematics and its applications). Chapman \& Hall/CRC, 2006.
[18] B. Vasic, K. Pedagani, and M. Ivkovic, "High-rate girth-eight lowdensity parity-check codes on rectangular integer lattices," IEEE Trans. Commun, vol. 52, no. 8, pp. 1248-1252, Aug. 2004.
[19] E. Gabidulin, A. Moinian, and B. Honary, "Generalized construction of quasi-cyclic regular LDPC codes based on permutation matrices," in Proc. IEEE Int. Symp. Inf. Theory, Jul. 2006, pp. 679-683.
[20] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," IEEE Trans. Inf. Theory, vol. 51, no. 1, pp. 386-398, Jan. 2005.


[^0]:    ${ }^{1}$ The relative harmfulness of a trapping set given in the TSO for the Gallager $\mathrm{A} / \mathrm{B}$ algorithm over the BSC is determined by its critical number and strength [7]. The relative harmfulness of a trapping set given in the TSO for the sum product algorithm over the BSC is currently being studied and will be discussed in the journal version of this paper.

