Structured LDPC Codes from Permutation Matrices Free of Small Trapping Sets

Dung Viet Nguyen, Bane Vasić, Michael Marcellin and Shashi Kiran Chilappagari
Department of Electrical and Computer Engineering
University of Arizona, Tucson, Arizona 85721
Email: {nguyendv, vasic, marcellin, shashic}@ece.arizona.edu

***Abstract*—This paper introduces a class of structured low-density 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 of is presented. Although the codes are optimized for the Gallager A/B algorithm over the binary symmetric channel (BSC), they 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 later 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 harmfulness*[1] of different trapping sets to determine which trapping sets should not be presented 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, the experimental results indicate that their error performances 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 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 by 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 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 GF(q). Our permutation matrices are similar to circulants in a sense that the set of $q-1$ non-identity permutation matrices form a cyclic group, but they are more general as the circulant property holds on indices understood as elements of GF(q). More specifically, the permutation corresponding to α^t sends the indices $(0,1,\alpha,\ldots,\alpha^{q-2})$ to $(0+\alpha^t,1+\alpha^t,\alpha+\alpha^t,\ldots,\alpha^{q-2}+\alpha^t)$.

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.

[1] The relative harmfulness of a trapping set given in the TSO for the Gallager A/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 SPA over BSC is currently being studied and will be discussed in the journal version of this paper.

The rest of the paper is organized as follows. In Section II, we provide the background related to LDPC codes and the necessary preliminaries for the methods of construction. 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 LDPC code over the binary field 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 a Latin square.

A *Latin square* of *size* q (or *order* q) is an $q \times q$ array in which each cell contains a single symbol from an 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 on q elements (see [17, pp. 135–152] for details).

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

$$\begin{array}{rcl} f: Q & \rightarrow & \text{Mat}(q, q, \text{GF}(2)) \\ \alpha & \mapsto & f(\alpha) = [m_{i,j}]_{i,j \in Q} \end{array}$$

such that:

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

It follows from the above definition that the images of elements of 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 $\mathcal{W} = [w_{i,j}]_{1 \leq i \leq m, 1 \leq j \leq n}$ be an $m \times n$ matrix over a quasigroup Q,

$$\mathcal{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{1,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{bmatrix} . \tag{1}$$

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

$$\mathcal{H} = \begin{bmatrix} f(w_{1,1}) & f(w_{1,2}) & \cdots & f(w_{1,n}) \\ f(w_{2,1}) & f(w_{1,2}) & \cdots & f(w_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ f(w_{m,1}) & f(w_{m,2}) & \cdots & f(w_{m,n}) \end{bmatrix} , \tag{2}$$

then $\mathcal{H}$ is a binary matrix of size $mq \times nq$. The null space of $\mathcal{H}$ gives an LDPC code of length nq. The column weight and row weight of $\mathcal{H}$ are m and n, respectively.

IV. Structured LDPC Codes from Galois Fields of Permutation Matrices

A. Galois Fields of Permutation Matrices

Consider the Galois field GF(q), where q is a power of a prime. Let α be a primitive element of GF(q). Let $\mathcal{L} = [l_{i,j}]_{i,j \in \text{GF}(q)}$ denote the Latin square defined by the Cayley table of the quasigroup given by the set $\{0, 1, \alpha, \ldots, \alpha^{q-2}\}$ together with the subtractive operation of GF(q), i.e. $l_{i,j} = i - j$. Let $\mathcal{M} = \{M_{-\infty}, M_0, M_1, \ldots, M_{q-2}\}$ be the set of images of elements of GF(q) under f, i.e. $M_t = [m_{i,j}^{(t)}]_{i,j \in \text{GF}(q)} = f(\alpha^t)$. It is easy to see that $M_{-\infty} = I$, the $q \times q$ identity matrix. To show that $\mathcal{M}$ forms a Galois field isomorphic to 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 $t_1, t_2 \in \mathbb{Z}$, $f(\alpha^{t_1} + \alpha^{t_2}) = M_{t_1} M_{t_2}$.

Proposition 2: For all $t \geq 0$, $M_{t+1} = P M_t Q$, where $P \in \mathcal{M}_{q \times q}[\text{GF}(2)]$ is given as

$$P = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} , \tag{3}$$

and Q is the transpose of P.

Define the addition $\boxplus$ and the multiplication $\boxdot$ on $\mathcal{M}$ as:

$$\begin{aligned} M_{t_1} \boxplus M_{t_2} &= M_{t_1} M_{t_2}, \\ M_{t_1} \boxdot M_{t_2} &= P^{(t_2-t_1)} M_{t_1} Q^{(t_2-t_1)} \\ &= P^{(t_1-t_2)} M_{t_2} Q^{(t_1-t_2)} \end{aligned}$$

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

B. LDPC Codes from Galois Fields of Permutation Matrices

Define $\mathcal{W}$ and $\mathcal{H}$ as in (1) and (2), where Q is the set $\{0, 1, \alpha, \ldots, \alpha^{q-2}\}$ together with the subtractive operation of 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 $w_{i_1,j_1} + w_{i_2,j_2} \neq w_{i_2,j_1} + w_{i_1,j_2}$ for any $1 \leq i_1, i_2 \leq m, 1 \leq j_1, j_2 \leq n$, $i_1 \neq i_2, j_1 \neq j_2$.

Proof: The proof is omitted due to page limitation. ■

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 cross-addition constraint. One form of $\mathcal{W}$ that satisfies the cross-addition constraint is given by

$$\mathcal{W} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \alpha & \cdots & \alpha^{q-2} \\ 0 & \alpha & \alpha^2 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \alpha^{q-2} & 1 & \cdots & \alpha^{q-3} \end{bmatrix}. \quad (4)$$

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

$$\mathcal{H} = \begin{bmatrix} I & I & I & \cdots & I \\ I & M_0 & M_1 & \cdots & M_{q-2} \\ I & M_1 & M_2 & \cdots & M_0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I & M_{q-2} & M_0 & \cdots & M_{q-3} \end{bmatrix}, \quad (5)$$

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 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 (γ, ρ) 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 GF(2) which is also free cycles of length 4. H has constant column and row weight γ and ρ, respectively. The null space of H gives a regular structured LDPC codes $\mathcal{C}$ of length ρq 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 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 $f(\mathcal{W}_p)$, where $\mathcal{W}_p$ is obtained from 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 later become array codes). In [19], permutation matrices of size $q \times q$, where $q = p^k$, 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 GF(q), then circulant permutation matrices of size $(q-1) \times (q-1)$ are obtained as images of elements of 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 *cross-multiplication constraint*. This condition can obtained from the cross-addition 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 form the matrix $\mathcal{W}$ in (1). The algorithm is based on a check and select-or-disregard procedure. Let τ specify which graphical structures should not be presented in the Tanner graph G. For example, Figure 1 shows the subgraphs induced by some small trapping sets. τ may specify the girth of G and may also specify the minimum distance of the code. For column-weight-three 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 ρ stages, where ρ is the row weight of the parity check matrix. Usually, ρ 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 on the

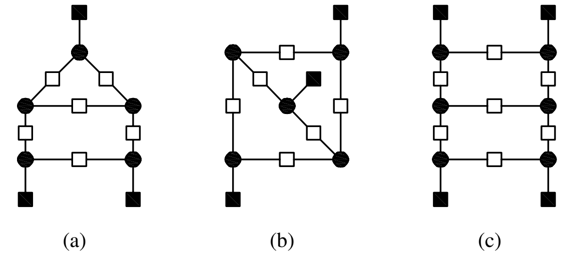

Fig. 1. (a) $(5,3)$ trapping set of girth 6, (b) $(5,3)$ trapping set of girth 8 and (c) $(6,4)$ trapping set. We use ● to represent variable nodes, ■ to represent odd degree check nodes and □ to represent even degree check nodes.

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 GF(q). An elements of 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 τ. If the condition τ 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 τ. It can be seen that the algorithm is a combination of the progressive edge grow 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 τ 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-the-art computers. For example, the construction of a $(1111, 808)$ code which has girth 8, minimum distance at least 10 and contain 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 $\mathcal{C}_8^{(n)}$ denote the obtained from the greedy progressive block grow algorithm when τ is defined as a condition that the corresponding Tanner graph of $\mathcal{C}_8^{(n)}$ has girth $g \geq 8$. $\mathcal{C}_8^{(n)}$ is an (530, 371) LDPC code with rate $R = 0.7$. Let $\mathcal{C}_8^{(o)}$ denote the (530, 371) shortened array codes (or integer lattice code described in [18]). $\mathcal{C}_8^{(o)}$ is obtained by extensive computer search and has the maximum possible rate of $R = 0.7$. Denote by $\mathcal{C}_{\text{d10}}$ the code obtained when τ is such that the minimum distance of $\mathcal{C}_{\text{d10}}$ is at least 10. $\mathcal{C}_{\text{d10}}$ 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 $g \geq 6$). $\mathcal{C}_{\text{d10}}$ is an (795, 636) LDPC code with rate $R = 0.8$ and girth $g = 6$. The error performance of $\mathcal{C}_8^{(n)}$, $\mathcal{C}_8^{(o)}$ and $\mathcal{C}_{\text{d10}}$ under the SPA a maximum of 50 iterations over the AWGNC is shown in Figure 2. It can be seen that the error performance of $\mathcal{C}_8^{(n)}$ is better than that of $\mathcal{C}_8^{(o)}$. One possible explanation for this observation is that the Tanner graph of $\mathcal{C}_8^{(o)}$ contain subgraphs induced by the codeword of weight 6 while the minimum distance of $\mathcal{C}_8^{(o)}$ is 10. Allowing the Tanner graph of $\mathcal{C}_{\text{d10}}$ to have girth 6 but requiring that the minimum distance is at least 10 results in higher rate than the rate of $\mathcal{C}_8^{(n)}$, while maintaining the good error performance. This example clearly demonstrates that larger girth alone does not necessarily lead to better performance.

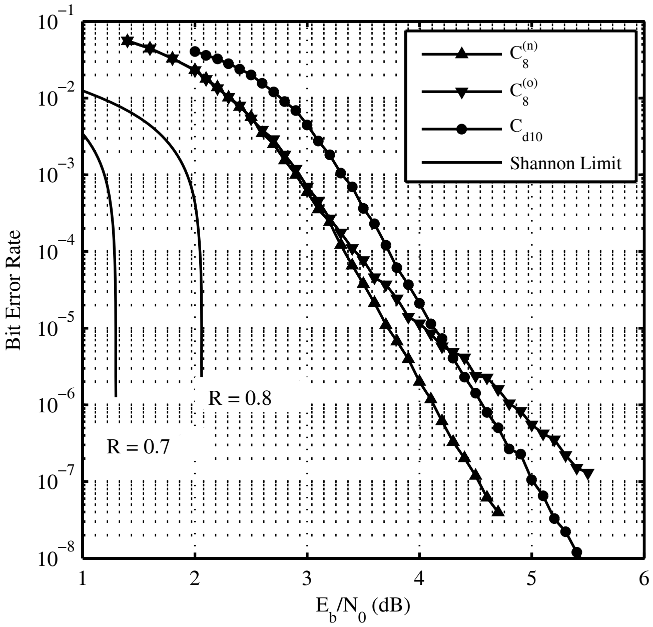

Fig. 2. Performances of the codes given in Example 1 over the AWGNC.

Example 2: Let $q = 19^2 = 361$. The codes obtained when conditions τ_1, τ_2, τ_3 and τ_4 are imposed are denoted with $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$, respectively. τ_1, τ_2, τ_3 and τ_4 are defined as

- τ_1: G has girth $g \geq 10$.
- τ_2: G has girth $g \geq 8$; G does not contain the (5,3) trapping set of girth 8 shown in Figure 1(b); and G does not contain the (6,4) trapping set shown in Figure 1(c).
- τ_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 another eight cycle.
- τ_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 another 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 AWGNC is shown in Figure 3.

It can be seen that the condition τ_1, τ_2, τ_3 and τ_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

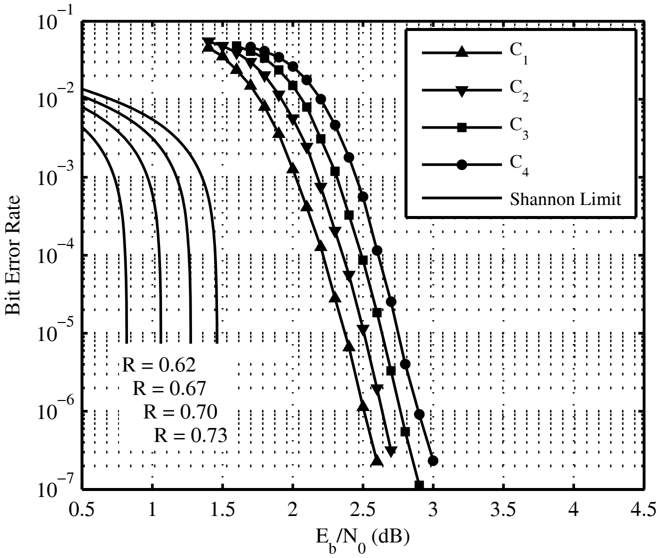

Fig. 3. Performances of the codes given in Example 2 over the AWGNC.

weaker conditions. This example emphasizes the importance of properly identifying the trapping sets to be avoided in the Tanner graph since it is crucial to the rate and the error performance of the code.

Remark: Condition τ_3 and τ_4 permits an eight cycle in G to share 2 variable nodes with at most one another 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 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 the 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 A/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, our codes outperform the best known short length structured LDPC codes. Our current and future works include 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 IHCS-0725403, 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 QC-LDPC 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 Commun.*, 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 low-density 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 quasi-cyclic 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 low-density 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 low-density 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.