Towards Improved LDPC Code Designs Using Absorbing Set Spectrum Properties

Lara Dolecek, Jiadong Wang
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, CA, 90095
Email: {dolecek, wjd}@ee.ucla.edu

Zhengya Zhang
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI, 48109
Email: zhengya@eecs.umich.edu

***Abstract*—This paper focuses on methods for a systematic modification of the parity check matrix of regular LDPC codes for improved performance in the low BER region (i.e., the error floor). A judicious elimination of dominant absorbing sets strictly improves the absorbing set spectrum and thereby improves the code performance. This absorbing set elimination is accomplished without compromising code properties and parameters such as the girth, node degree, and the structure of the parity check matrix. For a representative class of practical codes we substantiate theoretical analysis with experimental results obtained in the low BER region. Our results demonstrate at least an order of magnitude improvement of the error floor relative to the original code designs. Given that the conventional code parameters remain intact, the new code can easily be implemented on the existing software or hardware platforms employing high-throughput, compact architectures. As such, the proposed approach provides a step towards the improved code design that is compatible with practical implementation constraints.**

I. Introduction

Low-density parity-check (LDPC) codes are defined on sparse graphs, and are known to perform extremely well in the moderate bit-error-rate (BER) region. In the lower BER region, the signal-to-noise ratio (SNR) to BER curve changes its slope resulting in a so-called error-floor. This performance degradation is attributed to the suboptimality of iterative decoding algorithms on graphs with cycles. Typically, the decoder enters a non-codeword steady state from which it cannot escape.

Several important works studied such non-codeword objects, including stopping sets [3], near-codewords [14], trapping sets [19], and absorbing sets [28]. In particular, stopping sets are well suited for the analysis on the binary erasure channels [17]. Absorbing sets can be viewed as combinatorial counterparts of the stopping sets for channels with additive noise. Specifically, an absorbing set is a special type of a near-codeword or a trapping set that is guaranteed to be stable under a bit-flipping decoder.

Previous work [4], [20] studied structural properties of dominant absorbing sets for a representative class of practical regular LDPC codes. Concurrently, absorbing sets were also experimentally verified on a hardware emulator [28], [30] to govern the low BER region performance for several representative LDPC code families. Recent results in [1], [10], [18], [24] provided insights into the trapping set spectrum of other important classes of codes.

One promising direction for improving the performance and alleviating the error floors of finite-length sparse graph codes is to focus on practical aspects of iterative decoding algorithms. Various implementation components of the decoding algorithm have already been successfully analyzed and improved, including more efficient quantization of messages [26], [30], better message-exchange scheduling algorithms [2], [9], better iteration averaging schemes [11], [22], and the automation of postprocessing for escaping convergent absorbing sets [29].

The convergent non-codewords (i.e., absorbing sets) are in fact a structural property of the parity-check matrix of the code. This paper focuses on the systematic improvement of the parity-check matrix rather than the decoding algorithm. The resulting improved parity-check matrices may be seamlessly combined with the improved decoding algorithms discussed above.

A recent line of work [6], [8], [12] investigates techniques to improve the absorbing set (or trapping set) spectrum by introducing redundant or independent checks. An approach of selecting circulant submatrices for better performance via girth increase was proposed in [15]. In contrast, this paper focuses on the combinatorial properties of the absorbing sets and the provable improvement of the absorbing set spectrum via a systematic code modification that is valid over the whole code family. A distinguishing feature of the proposed approach is that it preserves original code properties such as the structure, rate, girth, and node degrees. Preservation of these properties facilitates practical realizations of high-throughput compact hardware architectures of LDPC codes and their decoding algorithms.

Section II presents the relevant background. Section III studies a representative structured code family. By analyzing the combinatorial properties of the smallest (lowest-weight) absorbing sets, we propose a systematic method for eliminating the smallest absorbing sets, which often dominate the error–floor performance. Elimination of the smallest absorbing sets induces benefit across the absorbing-set spectrum since the smallest absorbing sets are also components of many larger-weight absorbing sets that themselves cannot be easily eliminated by improving the decoding algorithm alone.

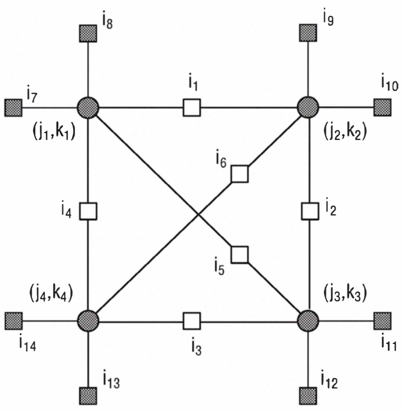


Fig. 1. Depiction of a $(4, 8)$ absorbing set. Black circles are bit nodes in the absorbing set, white squares are their satisfied checks, and black squares are their unsatisfied checks.

This process is accomplished while keeping desirable code properties intact. Section IV provides experimental results demonstrating consistent performance improvement for several choices of practical decoding algorithms and code instances. Section V delivers the conclusions.

II. Background

This section reviews absorbing sets, [4], [28] and describes a class of high-performance LDPC codes that serves as our illustrative case study.

Absorbing sets: Using the standard notation, let $G_H = (V, F, E)$ be the bipartite graph (Tanner graph) describing the parity-check matrix H, such that the set V corresponds to the columns of H, the set F corresponds to the rows of H, and $E = \{e(i, j) | H(j, i) = 1\}$.

For a subset D of V, let $O(D)$ be the set of neighboring vertices of D in F with odd degree with respect to D. For the integer pair (a, b), an (a, b) absorbing set is a subset D of V of size a, with $O(D)$ of size b, and with the property that each element of D has strictly fewer neighbors in $O(D)$ than in $F \setminus O(D)$.

Moreover, if all variable nodes in $V \setminus D$ have strictly fewer neighbors in $O(D)$ than in $F \setminus O(D)$, an (a, b) absorbing set is called an (a, b) fully absorbing set. An important property of fully absorbing sets is that they are stable under the bit-flipping operations. A $(4, 8)$ absorbing set is illustrated in Figure 1. (Here, the node labels are in the context of a particular code family, as explained next.)

Circulant-based LDPC codes: LDPC codes built from circulant matrices are particularly amenable for high-throughput applications and compact hardware design [23], [30]. These codes can be described as follows:

Given integers r and c, a mapping rule $f(i, j) : (i, j) \to \mathbb{N}$ and a $p \times p$ permutation matrix σ of the form

$$\sigma = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \tag{1}$$

we form the $(rp) \times (cp)$ parity check matrix $H^{r,c}_{p,f}$ as

$$H^{r,c}_{p,f} = \begin{bmatrix} \sigma^{f(0,0)} & \sigma^{f(0,1)} & \sigma^{f(0,2)} & \dots & \sigma^{f(0,c-1)} \\ \sigma^{f(1,0)} & \sigma^{f(1,1)} & \sigma^{f(1,2)} & \dots & \sigma^{f(1,c-1)} \\ \sigma^{f(2,0)} & \sigma^{f(2,1)} & \sigma^{f(2,2)} & \dots & \sigma^{f(2,c-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \sigma^{f(r-1,0)} & \sigma^{f(r-1,1)} & \sigma^{f(r-1,2)} & \dots & \sigma^{f(r-1,c-1)} \end{bmatrix}. \tag{2}$$

We assume that $(f(i_1, j_1) - f(i_1, j_2) + f(i_2, j_2) - f(i_2, j_1)) \neq 0 \mod p$ for $0 \leq i_1, i_2 \leq r-1$, $i_1 \neq i_2$ and $0 \leq j_1, j_2 \leq c-1$, $j_1 \neq j_2$ so that the girth of the code is at least 6.

Examples of codes that have the above descriptions include array-based LDPC codes [7] (with $f(i, j) = i \cdot j$, and c is set to p, the size of σ), quasi-cyclic (QC) LDPC codes [21] (with $f(i, j) = a^j b^i$ where a and b are nonzero elements of $GF(p)$ having multiplicative orders $o(a) = c$ and $o(b) = r$), and constructions discussed in [13].

It is convenient to label bit nodes and check nodes as follows: label each bit node in G_H with the unique label (j, k) that describes the corresponding column of $H^{r,c}_{p,f}$, and label each check node in G_H with a label i if the corresponding row of $H^{r,c}_{p,f}$ belongs to the row group i. For the illustration of the labeling, see Figure 1.

The following convenient properties follow from the fact that the parity check matrix $H^{r,c}_{p,f}$ is a two-dimensional array of permutation matrices, and they play a key role in establishing combinatorial properties necessarily satisfied by the absorbing sets of interest.

Bit Consistency: For a bit node, all its incident check nodes, labeled i_{s_1} through i_{s_γ}, must have distinct labels, i.e., these check nodes are in distinct row groups. ■

Check Consistency: All bit nodes, say (j_1, k_1) through (j_p, k_p), participating in the same check node must have distinct j_ℓ values, i.e., they are all in distinct column groups. ■

III. Case study: Array-based code

We now consider the $H^{r,p}_{p,i\cdot j}$ array-based LDPC codes [7] as an instance of high-performance practical LDPC codes with hardware-friendly structure. As a concrete example, we focus on the bit degree $r = 5$ (the analysis for other choices of r follows similarly, and is in fact somewhat simpler for $r < 5$).

A previous result [5] proves that $(4, 8)$ absorbing sets are the smallest possible for a general $r = 5$ code family. We first show in Lemma 1 that $(4, 8)$ absorbing sets indeed exist for this $r = 5$ array-based code family. This theoretical result is also consistent with previous experimental results of a sum-product decoding algorithm built in software and on a hardware emulator [30] for which it was shown that decoding errors due to $(4, 8)$ absorbing sets dominate the low

BER region. This error floor region is determined by non-codewords (absorbing sets) whose weight is strictly smaller than the minimum distance of the code [27]. We then propose a method for the systematic elimination of these absorbing sets that also reshapes the absorbing set spectrum for better performance.

Lemma 1: $(4,8)$ (fully) absorbing sets exist in the Tanner code described by the parity check matrix $H^{5,p}_{p,ij}$, and their number scales as $\Theta(p^3)$.

Proof: The proof follows techniques developed in [4], so we only outline the main steps.

Due to the cardinality of the absorbing set, bit nodes and check nodes must necessarily be arranged as in Figure 1, and we may then label the bit nodes with labels (j_1,k_1) through (j_4,k_4), their satisfied checks with labels i_1 through i_6 and their unsatisfied checks with labels i_7 through i_{14}, as in the figure.

It then follows that under the bit consistency constraints applied to the bit nodes in this candidate $(4,8)$ absorbing set, there are only two possible non-isomorphic check labelings: one using only four distinct labels (out of five available) and one using all five distinct labels for the satisfied checks i_1 through i_6. Without loss of generality, we may assign $(i_1,i_2,i_3,i_4,i_5,i_6)$ to be either (x,y,x,y,z,w) for $\{x,y,z,w\}\subset\{0,1,2,3,4\}$ (assignment 1) or (x,t,w,y,z,z) for $\{x,y,z,w,t\}=\{0,1,2,3,4\}$ (assignment 2).

Setting up a system of equations relating the bit node labels and the check node labels in a candidate $(4,8)$ absorbing set, after some calculations utilizing bit and check consistency constraints, the following necessary conditions for the existence of a $(4,8)$ absorbing set result:

$$(z-x)(w-y)+(z-y)(w-x)\equiv 0 \mod p\ , \quad (3)$$

for assignment 1, and

$$(z-w)(x-t)(y-z)-(y-w)(x-z)(z-t)\equiv 0 \mod p\, , \quad (4)$$

for assignment 2.

In the former case, in fact one can show there are no solution sets for prime p large enough ($p>17$). For the latter case there are 8 solution sets $(x,y,z,w,t)\in\{(4,3,2,0,1),(4,1,2,0,3),(3,4,2,1,0),(3,0,2,1,4),$ $(1,4,2,3,0),(1,0,2,3,4),(0,3,2,4,1),(0,1,2,4,3)\}$ that always evaluate to zero on the left-hand side of equation (4). These numerical solutions are in fact symmetric so that once the labels of the check nodes are selected (cf. Figure 1), the bit node labels (pairs (j_1,k_1) through (j_4,k_4)) can be selected in $\Theta(p^3)$ ways, thereby completely characterizing the absorbing set of interest.

Moreover, such an $(4,8)$ absorbing set is always a $(4,8)$ fully absorbing set since otherwise there would exist a bit node (j_5,k_5) outside the absorbing set incident to at least three of the checks labeled i_7 through i_{14}. Such a configuration would either violate the girth constraint, or it would imply the existence of a new configuration spanning four bit nodes (these being the node (j_5,k_5) and three bit nodes from the starting $(4,8)$ absorbing set). These four bit nodes would necessarily be connected such that their common constraint is given in (3), previously shown to not hold for large enough p. ■

The following lemma discusses how a different choice of the mapping rule $f(i,j)$ can be made such that neither of the necessary conditions (3) and (4) for the existence of $(4,8)$ absorbing sets is fulfilled.

Lemma 2: There are no $(4,8)$ absorbing sets in the Tanner code described by the parity check matrix $H^{5,p}_{p,f(i,j)}$, for prime p large enough and $f(i,j)=a(i)\cdot j$ for $(i,a(i))\in\{(0,0),(1,1),(2,3),(3,8),(4,19)\}$.

Proof: It is sufficient to assign values to $a(i)$ such that the selected labels for the check nodes i_1 through i_6 do not satisfy equations (3) and (4). Let $b(i)$ be the set of all pairwise differences of the elements of $a(i)$. First, by construction they are all distinct. Second, all of the products taken $\mod p$ of two elements of $b(i)$ are distinct as are all of the products taken $\mod p$ of three elements of $b(i)$. These conditions are sufficient to ensure that the congruential constraints (3) and (4) do not hold for prime p large enough [1]. ■

Remark 1: In fact, there are other choices for $a(i)$ that guarantee that no $(4,8)$ absorbing sets exist in the Tanner code described by the parity check matrix $H^{5,p}_{p,f(i,j)}$ with $f(i,j)=a(i)\cdot j$ by ensuring that the equations (3) and (4) do not hold. Another such example for $f(i,j)=a(i)\cdot j$ is for $(i,a(i))\in\{(0,0),(1,1),(2,2),(3,4),(4,6)\}$, where it is sufficient for the prime p to be greater than 23. While not all pairwise differences and their products of the elements of $a(i)$ are distinct as in the previous case, one can verify that the necessary subset of such combinations is, thereby ensuring the non-existence of $(4,8)$ absorbing sets. ■

We will return to the topic of choices for $a(i)$ in Section IV where we compare the performance of the original and the modified codes.

The following result shows how the elimination of the smallest term in the absorbing set spectrum also helps reduce the cardinality of other contributing terms in the absorbing set spectrum.

Lemma 3: The transformation of $f(i,j)=i\cdot j$ into $f(i,j)=a(i)\cdot j$ for $(i,a(i))\in\{(0,0),(1,1),(2,3),(3,8),(4,19)\}$ strictly reduces the number of $(6,8)$ absorbing sets for p large enough.

Proof outline: We provide an outline of the proof. More details can be found in [25]. We first consider the case where there exists a satisfied check (unsatisfied check) incident to bit nodes in the absorbing set of degree at least 4 (at least 3). By the girth constraint the only possibility is if 3 out of 6 bit nodes share an unsatisfied check (itself of degree 3) and the other 3 out of 6 bit nodes also share an unsatisfied check (also itself of degree 3). Each bit node has an additional unsatisfied check. Using combinatorial arguments and the bit consistency and check consistency conditions (cf. Section II) it follows that this configuration is in fact not possible for either $f(i,j)$ for p large enough.

[1] In fact, p can be any prime larger than 23 except for the set $\{37,41,59,61,67,73,89,101,163,223,263,271,277,317,337,1361\}$, itself easily identified by numerical substitutions for labels in (3) and (4).

There are five non-isomorphic candidate $(6,8)$ absorbing sets, in which all satisfied checks (unsatisfied checks) incident to bit nodes in the absorbing set have degree 2 (degree 1). Out of these five configurations, with some analytical work, three can be eliminated for p large enough using combinatorial arguments and the bit consistency and check consistency conditions applied to either choice of $f(i,j)$. The remaining two configurations have the cardinality $\Theta(p^3)$ in the original code, and interestingly, both contain a $(4,8)$ absorbing set as a subset. These are therefore completely eliminated by the transformation on $f(i,j)$.

Since no new $(6,8)$ absorbing sets are introduced, and since the number of existing $(6,8)$ absorbing sets is reduced (and in fact entirely eliminated for p large enough), the transformation strictly improves this component in the abs. set spectrum. ■

Remark 2: The same conclusion as in Lemma 3 holds for the performance improvement under the mapping $f(i,j) = a(i) \cdot j$ where $(i, a(i)) \in \{(0,0),(1,1),(2,2),(3,4),(4,6)\}$, given in Remark 1. The details are also in [25]. ■

Observe that the proposed modification preserves the girth of the code. That is, there exists a length-6 cycle passing through checks in the first three row groups in both the original and the modified code, and no length-4 cycles are introduced during the transformation. Therefore, short cycles that are the underlying building blocks of the (dominant) absorbing sets are still present. What we have in fact achieved is breaking apart the troublesome superpositions of such short cycles rather than breaking apart all of the short cycles themselves. This enables us to maintain desirable structural symmetry of the party check matrix.

IV. Experimental Results

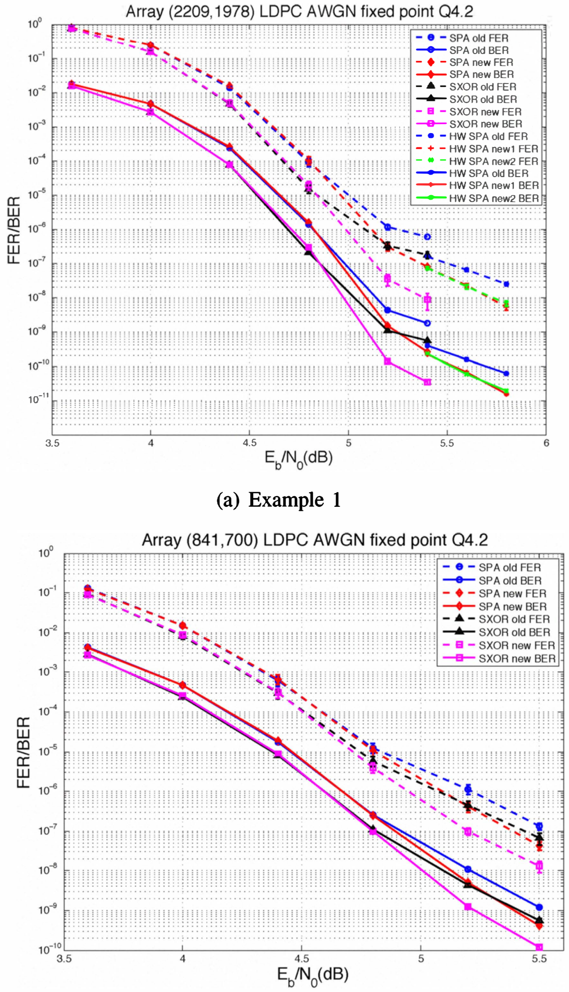
In this section we experimentally demonstrate performance improvement with the proposed modifications that is *consistently valid across various choices of decoding algorithms and implementations*.

In simulations, we use 200 iterations and a $Q4.2$ fixed-point quantization, 4 bits to the left of the radix point to represent integer values and 2 bits to the right of the radix point to represent fractional values. We simulate sum-product algorithm [30] and soft-xor algorithm [16] for different codes.

In Figure 2(a) we compare the performance of the $(2209, 1978)$ original code and the modified code. Both codes fit the description of the parity check matrix as in (2) with the following parameters: check node degree = 47, bit node degree = 5, and $f(i,j) = a(i) \cdot j$. The original code uses $a(i) = i$ and the modified code uses $a(i)$ proposed in Lemma 2.

For all choices of simulated decoding algorithms, the following is observed: a significant fraction of decoding errors under the original code is due to the $(4,8)$ absorbing sets, whereas under the modified codes there are no such errors. Moreover, the total number of errors decreases as the overall number of higher-order error terms in the absorbing set spectrum decreases.

For illustration purposes, another pair of codes are contrasted in Figure 2(b), and with the modified mapping from



(a) Example 1

(b) Example 2

Fig. 2. Performance comparison of the original and modified LDPC codes.

Remark 2. Again, the performance improvement (albeit less than in the previous example as the considered code is shorter) is achieved by the systematic elimination of the $(4,8)$ absorbing sets that in turn also reduces other contributing elements in the absorbing set spectrum.

For a more convincing experimental proof, we programmed a sum-product decoder on an FPGA platform to decode the original $(2209, 1978)$ array-based LDPC code as well as its two optimized versions, discussed in Lemma 1 and Remark 1, respectively. Since the structure of the array code is preserved in the modified versions, the decoder is reused with no added complexity. A substantial simulation speedup via the FPGA platform allows us to extend the BER curve down to 10^{-11}, as shown in Figure 2(a).

The error profiles are shown in Tables I, II, and III for comparison. Consistent with the software-based simulation and the theoretical analysis, $(4,8)$ absorbing sets dominate the error floor of the original array code. In both modified array codes, having virtually same performance (see Fig. 2), the

SNR	no.runs	a.s	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(8,6)	(8,8)	(8,10)
5.6dB	4.2E9	272	97	27	64	10	16	17	11	5
5.8dB	8.5E9	211	121	20	34	9	3	3	4	3

TABLE I

SNR	no.runs	a.s	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(8,6)	(8,8)	(8,10)
5.6dB	8.5E9	197	0	30	36	27	25	3	17	13
5.8dB	1.7E10	90	0	25	16	22	16	3	10	1

TABLE II

SNR	no.runs	a.s.	(4,8)	(5,9)	(6,8)	(6,10)	(7,9)	(8,6)	(8,8)	(8,10)
5.6dB	8.5E9	179	0	30	28	29	22	5	9	10
5.8dB	1.7E10	116	0	25	16	22	16	3	10	1

TABLE III

Hardware error profiles for the original (2209, 1978) code (Table I), and the two modified codes (Table II and III). More input frames were emulated for higher SNR levels. The number of errors collected is divided by the total number of frames (no. runs) to produce Figure 2 data. Total number of collected absorbing set errors is denoted a.s.

$(4,8)$ absorbing sets are completely eliminated and the BER curve is lowered by one order of magnitude Another noticeable effect is the reduction of the number of $(6,8)$ absorbing set errors after the code modification, which is attributed to the elimination of the $(4,8)$ substructures.

V. Conclusion

In this paper we studied methods to improve regular LDPC code designs under iterative decoding. By taking into account the combinatorial characterization of absorbing sets, as dominant fixed points of practical iterative decoding algorithms, we provided methods to improve the absorbing set spectrum of the code while keeping other desirable code properties (such as the node degrees and the parity check matrix structure) intact. Theoretical findings were substantiated by experimental results showing good improvement in the low BER region for a class of structured LDPC codes. A useful feature of the proposed approach from the implementation standpoint is that it can be easily combined with better decoding algorithms for the maximum improvement of the overall performance. Future work involves extending the analysis to irregular codes.

Acknowledgement

The authors gratefully acknowledge valuable comments provided by Prof. Rick Wesel during the course of this work.

References

[1] S. Abu-Surra, D. DeClerq, D. Divsalar, and W. Ryan, "Trapping set enumerators for specific LDPC codes," in *Proc. Info. Theory and Applications (ITA) Workshop*, San Diego, CA, Feb. 2010.

[2] A. I. V. Casado, M. Griot and R. D. Wesel, "Improving LDPC decoders via informed dynamic scheduling," in *Proc. IEEE Info. Theory Workshop (ITW) 2007*, Lake Tahoe, CA, Sep. 2007.

[3] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke, "Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. on Info. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[4] L. Dolecek, Z. Zhang, M. J. Wainwright, V. Anantharam, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. on Info. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[5] L. Dolecek, "On absorbing sets of structured sparse graph codes," in *Proc. Info. Theory and Applications (ITA) Workshop*, San Diego, CA, Feb. 2010.

[6] O. Fainzilber, E. Sharon and S. Litsyn, "Decreasing error floor in LDPC codes by parity-check matrix extensions," in *Proc. IEEE Int. Symp. on Info. Theory (ISIT)*, Seoul, Korea, Jun. - Jul. 2009.

[7] J. L. Fan, "Array-codes as low-density parity-check codes," in *Proc. of Second Int. Symp. on Turbo Codes*, Brest, France, Sep. 2000.

[8] M. Ivkovic, S. K. Chilappagari, and B. Vasic, "Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers," *IEEE Trans. on Info. Theory*, vol. 54, no. 8, pp. 3763–3768, Aug. 2008.

[9] S. Kim, K. Ko, J. Heo, and J. Kim, "Two-staged informed dynamic scheduling for sequential belief propagation decoding of LDPC codes," *IEEE Comm. Letters*, vol. 13, no. 3, pp. 193 - 195, Mar. 2009.

[10] C. Koller, A. Graell i Amat, J. Kliewer, and D. J. Costello, "Trapping set enumerators for repeat multiple accumulate code ensembles," in *Proc. IEEE Int. Symp. on Info. Theory (ISIT)*, Seoul, Korea, Jun. - Jul. 2009.

[11] S. Laendner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proc. of Wireless Comm.*, Honolulu, HI, Jun. 2005.

[12] S. Laendner, T. Hehn, O. Milenkovic, and J. Huber, "When does one redundant parity-check equation matter?," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, San Francisco, CA, Nov. 2006.

[13] S. Lin and D. J. Costello, Chapter 17 of *Error Control Coding*, Prentice Hall, 2004.

[14] D. MacKay and M. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," *Electronic Notes in Theo. Computer Science*, vol. 74, 2003.

[15] O. Milenkovic, N. Kashyap and D. Leyba, "Shortened array codes of large girth," ,*IEEE Trans. on Info. Theory*, vol. 53, no. 8, pp. 3707-3722, Aug 2006.

[16] M.M. Mansour and N.R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. on VLSI Systems* vol. 11, no. 6, pp. 976 – 996, Dec. 2003.

[17] A. Orlitsky, K. Viswanathan, J. Zhang, "Stopping set distribution of LDPC code ensembles," *IEEE Trans. on Info. Theory*, vol.5̄1, no. 3, pp. 929–953, Mar. 2005.

[18] A. E. Pusane, D. J. Costello, and D. G. M. Mitchell, "Trapping set analysis of protograph-based LDPC convolutional codes," in *Proc. IEEE Int. Symp. on Info. Theory (ISIT)*, Seoul, Korea, Jun. - Jul. 2009.

[19] T. Richardson, "Error-floors of LDPC codes," in *Proc. 41st Annual Allerton Conf.*, Monticello, IL, Oct. 2003.

[20] C. Schlegel and S. Zhang, "On the dynamics of the error floor behavior in (regular) LDPC codes," submitted to *IEEE Trans. on Info. Theory* , 2009.

[21] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. on Info. Theory*, vol. 50, no. 12, pp. 2966 –2984, Dec. 2004.

[22] N. Varnica and M. Fossorier, "Improvements in belief-propagation decoding based on averaging information from decoder and correction of clusters of nodes," *IEEE Comm. Letters*, vol. 10, no. 12, pp. 846 – 848, Dec. 2006.

[23] B. Vasic and E. Kurtas, *Coding and Signal Processing for Magnetic Recording Systems*, CRC press, 2005.

[24] B. Vasic, S. K. Chilappagari, D. V. Nguyen, and S. K. Planjery, "Trapping set ontology," in *Proc. 47th Annual Allerton Conf.*, Monticello, IL, Sept. 2009.

[25] J. Wang and L. Dolecek, "Combinatorial analysis of $(6,8)$ absorbing sets for array-based LDPC codes," Available at http://www.ee.ucla.edu/ dolecek/myPubs/TurboAppend.pdf

[26] Z. Wang and Z. Cui, "Low-complexity high-speed decoder design for quasi-cyclic LDPC code," *IEEE Trans. on VLSI Systems* vol. 15, no. 1, pp. 104 – 114, Jan. 2007.

[27] K. Yang and T. Helleseth, "On the minimum distance of array codes as LDPC codes," *IEEE Trans. on Info. Theory*, vol. 49, no.12, pp. 3268–3271, Dec. 2003.

[28] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, M. Wainwright, "Investigation of error floors of structured low-density parity-check codes by hardware emulation, in *Proc. IEEE Global Comm. Conf (GLOBECOM)*, San Francisco, CA, Nov. 2006.

[29] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, M.J. Wainwright, "Lowering LDPC error floors by postprocessing," in *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, New Orleans, LA, Nov. 2008.

[30] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright, "Design of LDPC decoders for improved low error rate performance: quantization and algorithm choices," *IEEE Trans. on Comm.*, vol. 57, no. 11, pp. 3258–3268, Nov. 2009.