Two-Bit Message Passing Decoders for LDPC Codes Over the Binary Symmetric Channel

Lucile Sassatelli
LIDS/RLE
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: lucisass@mit.edu

Shashi Kiran Chilappagari, Bane Vasić
Dept. of Electrical and Computer Eng.
University of Arizona
Tucson, AZ 85721, USA
Email: {shashic,vasic}@ece.arizona.edu

David Declercq
ETIS
ENSEA/UCP/CNRS UMR 8051
95014 Cergy-Pontoise, France
Email: declercq@ensea.fr

***Abstract*—A class of two-bit message passing decoders for decoding column-weight-four LDPC codes over the binary symmetric channel is proposed. The thresholds for various decoders in this class are derived using density evolution. For a specific decoder, the sufficient conditions for correcting all error patterns with up to three errors are derived.**

I. Introduction

The performance of various hard decision algorithms for decoding low-density parity-check (LDPC) codes over the binary symmetric channel (BSC), has been studied in great detail. The BSC is a simple yet useful channel model used extensively in areas where decoding speed is a major factor. For this channel model, Gallager [1] proposed two binary message passing algorithms, namely Gallager A and Gallager B algorithms. The message passing algorithms operate on a graphical representation of the code, known as the Tanner graph [2]. Gallager [1] showed that there exist (n, γ, ρ), $\rho > \gamma \geq 3$ regular LDPC codes of length n with column weight γ and row weight ρ, for which the bit error probability approaches zero when we operate below the threshold. Richardson and Urbanke [3] analyzed ensembles of codes under various message passing algorithms and described *density evolution*, a deterministic algorithm to compute the thresholds. Bazzi *et al.* [4] determined exact thresholds for the Gallager A algorithm and outlined methods to analytically determine thresholds of more complex decoders. Burshtein and Miller [5] considered expansion arguments to show that message passing algorithms are capable of correcting a linear fraction of errors (in the code length) when the degree of each variable node is at least six.

In this paper, we consider two-bit decoders for decoding column-weight-four LDPC codes over the BSC. Column-weight-four codes are of special importance as their decoders have low complexity and are interesting for a wide range of applications. The idea of using message alphabets with more than two values for the BSC was first proposed by Richardson and Urbanke in [3]. They proposed a decoding algorithm, known as the Gallager E algorithm, with erasures in the message alphabet. The messages in such a decoder have hence three possible values. They showed that such decoders exhibit thresholds close to the belief propagation algorithm. The class of two-bit decoders that we propose is a generalization of their idea, since we consider a message alphabet with four possible values. We report the thresholds of various decoders in this class. For the sake of simplicity, we consider universal decoders, i.e., the decoders which do not depend on the transition probability of the BSC. Also, we restrict our attention to static decoders, i.e., the decoders in which the update rules do not change with the iterations.

For a specific decoder in the class of two-bit decoders, we derive the sufficient conditions on the Tanner graph of the code to guarantee the correction of all error patterns with up to three errors. The problem of correcting a fixed number of errors assumes significance in the error floor region, where the slope of the frame error rate (FER) curve is determined by the weight of the smallest error pattern uncorrectable by the decoder [6]. For iterative decoding over the binary erasure channel (BEC), it is known that avoiding stopping sets [7] up to size t in the Tanner graph of the code guarantees recovery from t or less erasures. A similar result for decoding over the BSC is still unknown for a large number of cases. For column-weight-three codes, the necessary and sufficient conditions to guarantee the correction of three errors have been derived by Chilappagari *et al.* [8]. For column-weight-four LDPC codes of girth six, the sufficient conditions to correct three errors under the Gallager B algorithm have been derived by Chilappagari *et al.* [9]. The conditions that we derive in this paper are similar to the ones in [9], but impose fewer constraints on the Tanner graph. We note that the decoder that we consider while not necessarily the best possible two-bit decoder is easier to analyze and the methodology in the paper can be extended to other decoders to yield similar results.

The rest of the paper is organized as follows. In Section II, we establish the notation and define a general class of two-bit decoders. For a specific two-bit decoder, the sufficient conditions for correction of three errors are derived in Section III. In Section IV, we report the thresholds for various decoders.

II. The class of two-bit decoders

The Tanner graph of a code, whose parity-check matrix $\mathbf{H}$ has size $m \times n$, is a bipartite graph with a set of n variable nodes and a set of m check nodes. Each variable node corresponds to a column of the parity-check matrix, and each check node corresponds to a row. An edge connects a

variable node to a check node if the corresponding element in the parity-check matrix is non-zero. A Tanner graph is said to be γ-left-regular if all variable nodes have degree γ, ρ-right-regular if all check nodes have degree ρ, and (n, γ, ρ) regular if there are n variable nodes, all variable nodes have degree γ and all check nodes have degree ρ. The variable degree and check degree are also referred to as column weight and row weight, respectively.

Message passing algorithms for decoding LDPC codes run iteratively. Every round of message passing (iteration) starts with sending messages from variable nodes to check nodes (first half of the iteration) and ends by sending messages from check nodes to variable nodes (second half of the iteration). Let $\mathbf{r} = (r_1, \ldots, r_n)$, a binary n-tuple be the input to the decoder. Let $\omega_j(v, c)$ denote the message passed by a variable node v to its neighboring check node c in j^{th} iteration and $\varpi_j(c, v)$ denote the message passed by a check node c to its neighboring variable node v. Additionally, let $\omega_j(v, :)$ denote the set of all messages from v, $\omega_j(v, : \backslash c)$ denote the set of messages from v to all its neighbors except to c and $\omega_j(: , c)$ denote the set of all messages to c. The terms $\omega_j(: \backslash v, c), \varpi_j(c, :), \varpi_j(c, : \backslash v), \varpi_j(: , v)$ and $\varpi_j(: \backslash c, v)$ are defined similarly.

Before proceeding to give a formal description of a class of two-bit decoders, we make the following observation. Since, the message alphabet is finite, the message passing update rules can be described using a lookup table and hence only a finite number of two-bit decoders are possible. Also, the Boolean function that represents any particular decoder must be symmetric in the sense that swapping all inputs must imply a swap of the output. In this paper, we focus on a class of two-bit decoders that can be described using simple algebraic rules and illustrate with an example how the lookup table can be constructed from the algebraic description.

Let the message alphabet be denoted by $M = \{-S, -W, W, S\}$ where $-S$ denotes a strong "1", $-W$ denotes a weak "1", W denotes a weak "0" and S denotes a strong "0" and $S, W \in \mathbb{R}^+$. It should be noted that this representation can be mapped onto the alphabet $\{11, 01, 00, 10\}$, but we use the symbols throughout for the sake of convenience. The received value $r_v \in \{0, 1\}$ on the channel of a variable node v is mapped to $R_v \in \{C, -C\}, C \in \mathbb{R}^+$ as follows: $1 \rightarrow -C$ and $0 \rightarrow C$. It can be seen that each message is associated with a value and strength (strength of a message is an indication of its reliability).

Let $\mathcal{N}_1(u)$ denote the set of nodes connected to node u by an edge. Let the quantities $t_j(v, :)$ and $t_j(v)$, $j > 1$ be defined as follows:

$$t_j(v, c) = \sum_{u \in \mathcal{N}_1(v) \backslash c} \varpi_{j-1}(u, v) + R_v$$

and

$$t_j(v) = \sum_{u \in \mathcal{N}_1(v)} \varpi_j(u, v) + R_v \tag{1}$$

Additionally, let

$$sign(\varpi_j(c, v)) = \prod_{u \in \mathcal{N}_1(c) \backslash v} sign(\omega_j(u, c)),$$

where $sign(a) = 1$, if $a \geq 0$ and $sign(a) = -1$, if $a < 0$.

The message passing update and decision rules can be expressed as follows.

$$\omega_1(v, c) = W \cdot sign(R_v)$$

$$\varpi_j(c, v) = \begin{cases} S \cdot sign(\varpi_j(c, v)), & \text{if } \forall u \in \mathcal{N}_1(c) \backslash v, \\ & |\omega_j(u, c)| = S \\ W \cdot sign(\varpi_j(c, v)), & \text{otherwise} \end{cases}$$

For $j > 1$

$$\omega_j(v, c) = \begin{cases} W \cdot sign(t_j(v, c)), & \text{if } 0 < |t_j(v, c)| < S \\ S \cdot sign(t_j(v, c)), & \text{if } |t_j(v, c)| \geq S \\ W \cdot sign(R_v), & \text{if } t_j(v, c) = 0 \end{cases}$$

Decision: At the end of j^{th} iteration, the estimate r_v^j of a variable node v is given by

$$r_v^j = \begin{cases} 0, & \text{if } t_j(v) > 0 \\ 1, & \text{if } t_j(v) < 0 \\ r_v, & \text{if } t_j(v) = 0 \end{cases}$$

The class of two-bit decoders described above can be interpreted as a voting scheme in the following way: every message has two components namely, the value (0 or 1) and strength (weak or strong). The sign of the message determines the value, whereas the values of W and S denote the number of votes. The received value is associated with C votes. To compute the outgoing message on the variable node side, the total number of votes corresponding to 0 and 1 are summed. The value of the outgoing message is the bit with more number of votes and the strength is determined by the number of votes. In the case of a tie, the outgoing message is set to the received value with a weak strength.

Different decoders in this class can be obtained by varying the values of S, W and C. Hence, we denote a particular decoder by the triplet (C, S, W). Since there are only a finite number of two-bit decoders, different choices for C, S and W might lead to the same decoder. The discussion of the number of unique decoders is beyond the scope of this paper. Table I shows the message passing update rules for $(C, S, W) = (2, 2, 1)$ for $r_v = 0$. The corresponding table for $r_v = 1$ can be similarly obtained. Table II shows the decision rules for $(C, S, W) = (2, 2, 1)$.

III. Conditions to Guarantee the Correction of Three Errors

In this section, we derive the sufficient conditions on the Tanner graph of a column-weight-four LDPC code to guarantee the correction of all error patterns with up to three errors.

TABLE I

UPDATE RULE: NUMBER OF MESSAGES $-S$, $-W$, W AND S GOING INTO THE VARIABLE NODE v LEADING TO DIFFERENT VALUES OF THE MESSAGE $\omega_j(v,c)$ GOING OUT OF v, WHEN THE RECEIVED VALUE IS r_v. THE CODE HAS COLUMN WEIGHT FOUR AND THE $(C,S,W)=(2,2,1)$ TWO-BIT DECODER IS USED.

	$\#\ -S$ mess.	$\#\ -W$ mess.	$\#\ W$ mess.	$\#\ S$ mess.
$r_v=0$	2	1	0	0
$\omega_j(v,c)=-S$	3	0	0	0
	1	2	0	0
$r_v=0$	0	3	0	0
$\omega_j(v,c)=-W$	2	0	1	0
	0	2	1	0
$r_v=0$	1	1	1	0
$\omega_j(v,c)=W$	1	1	0	1
	2	0	0	1
	0	0	0	3
	0	0	1	2
	0	0	2	1
	0	0	3	0
$r_v=0$	0	1	0	2
$\omega_j(v,c)=S$	0	1	1	1
	0	1	2	0
	0	2	0	1
	1	0	0	2
	1	0	1	1
	1	0	2	0

TABLE II

DECISION RULE: NUMBER OF MESSAGES $-S$, $-W$, W AND S GOING INTO A VARIABLE, WHEN THIS VARIABLE NODE IS DECODED AS 0 (RESP. 1) WHEN THE CHANNEL OBSERVATION IS 1 (RESP. 0). THE CODE HAS COLUMN WEIGHT FOUR AND THE $(C,S,W)=(2,2,1)$ TWO-BIT DECODER IS USED.

	$\#\ -S$ mess.	$\#\ -W$ mess.	$\#\ W$ mess.	$\#\ S$ mess.
	0	0	0	4
	0	0	1	3
	0	0	2	2
Received value 1	0	0	3	1
Decoded as 0	0	0	4	0
	0	1	0	3
	0	1	1	2
	0	1	2	1
	1	0	0	3
	1	0	1	2
	0	4	0	0
	1	2	1	0
	1	3	0	0
Received value 0	2	1	0	1
Decoded as 1	2	1	1	0
	2	2	0	0
	3	0	0	1
	3	0	1	0
	3	1	0	0
	4	0	0	0

Since the code is linear and the channel and the decoder are symmetric, we can assume, without loss of generality, that the all-zero-codeword is transmitted over the BSC. We make this assumption throughout the paper. Hence, the variable nodes flipped by the channel are received as "1".

The problem of guaranteed error correction capability assumes significance in the error floor region. Roughly speaking, error floor is the abrupt degradation in the FER performance in the high SNR regime. The error floor phenomenon has been attributed to the presence of a few harmful configurations in the Tanner graph of the code, variously known as stopping sets (for the BEC), near codewords, trapping sets (for iterative decoding on the BSC and the AWGN) and pseudo-codewords (for linear programming decoding). While girth optimized codes have been known to perform well in general, the code length and the degree distribution place a fundamental limit on the best achievable girth. Hence, additional constraints on the Tanner graph are required to ensure better error floor performance.

The guaranteed error correction capability of column-weight-three LDPC codes under the Gallager A algorithm is now completely understood (see [10], [11] for details). For column-weight-four LDPC codes under the Gallager B algorithm, sufficient conditions to guarantee all error patterns with up to three errors have been derived by Chilappagari *et al.*[9]. The conditions derived in [9] impose constraints on the least number of neighboring check nodes for a given set of variable nodes. The conditions that we derive are similar, but impose fewer constraints on the Tanner graph. Due to the close relationship of the conditions to the notion of graph expansion, we refer to such conditions as expansion conditions. If every subset of x variable nodes in the Tanner graph have at least y neighboring check nodes, we say that the condition $x \rightarrow y$ is satisfied.

Before proceeding to the main theorem, we provide some additional definitions and establish the necessary notation.

Definition 1: The neighborhood of depth one of a node u is denoted by $\mathcal{N}_1(u)$ and is composed of all the nodes such that there exists an edge between these nodes and u. Similarly, $\mathcal{N}_d(u)$ denotes the neighborhood of depth d of node u and is composed of all the nodes such that there exists a path of length d between these nodes and u.

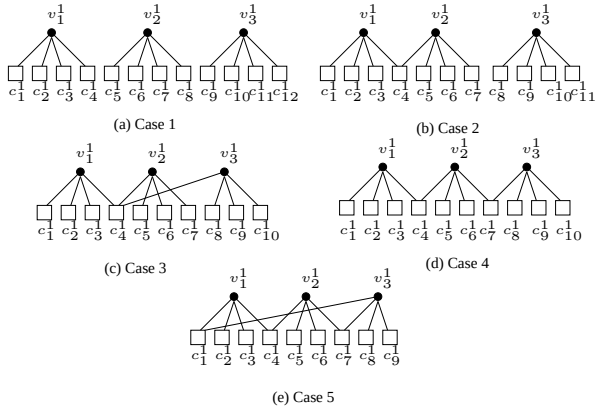

Fig. 1. All possible subgraphs subtended by three erroneous variable nodes.

Let E be a set of nodes, say $E=\cup_i u_i$, then the depth d neighborhood of E is $\mathcal{N}_d(E)=\cup_i \mathcal{N}_d(u_i)$.

Now we state the main theorem[1].

Theorem 1: [Irregular expansion theorem] Let $\mathcal{G}$ be the Tanner graph of a column-weight-four LDPC code with no 4-cycles, satisfying the following expansion conditions: each

[1]The results have been updated after the submission.

variable subset of size 4 has at least 11 neighbors, each one of size 5 at least 12 neighbors, each one of size 6 at least 14 neighbors, each one of size 8 at least 16 neighbors and each one of size 9 at least 18 neighbors. The two-bit decoder, with $C = 2$, $S = 2$ and $W = 1$, can correct up to three errors in the codeword within three iterations, if and only if the above conditions are satisfied.

For ease in notation, each expansion condition will be denoted by "4→11 expansion condition", "5→12 expansion condition" and so on.

Proof of sufficiency:

Remark: The proof can be followed more easily by looking at Tables II and I. Let $V^1 = \{v_1^1, v_2^1, v_3^1\}$ and $C^1 = \mathcal{N}_1(V^1)$. For more easily readable notation, let $\mathcal{N}_2(V^1)\backslash V^1$ be denoted by V^2 and $\mathcal{N}_1(V^2)\backslash C^1$ by C^2. Also, we say that a variable node is of type T_p^q when it has p connections to C^1 and q connection to C^2. The union of order d neighborhoods of all the T_p^q variable nodes is denoted by $\mathcal{N}_d(T_p^q)$.

We consider all the subgraphs induced by three erroneous variable nodes in a graph and prove that, in each case, the errors are corrected. The possible subgraphs are shown in Figure 1. As shown, five cases arise. In the reminder, we assume that the all-zero codeword has been sent. For lack of space, we provide the proof for Case 2. The proofs for necessity and other cases can be found in the longer version of the paper [12].

Case 2: Consider the error configuration shown in Figure 1(b).

In the second half of the first iteration, we have:

$$\begin{aligned}
\varpi_1(c_4^1, v) &= -W, \quad v \in \{v_1^1, v_2^1\} \\
\varpi_1(c, v) &= -W, \quad v \in V^2, \quad c \in C^1\backslash c_4^1 \\
\varpi_1(c, v) &= W, \quad \text{otherwise}
\end{aligned}$$

In the first half of the second iteration, according to Table I no $-S$ messages can be sent by variables neither in $V\backslash V^1$ because no $-S$ message propagate in the first iteration, nor variables in V^1 because they all receive at least three W messages:

$$\begin{aligned}
\omega_2(v, c) &= -W, \quad v \in \{v_1^1, v_2^1\}, \quad c \in C^1\backslash c_4^1 \\
\omega_2(v, c_4^1) &= W, \quad v \in \{v_1^1, v_2^1\} \\
\omega_2(v_3^1, c) &= W, \quad c \in C^1 \\
\omega_2(v, c) &= -W, \quad v \in \mathcal{N}_0(T_3^1), \quad c \in C^2 \\
\omega_2(v, c) &= W, \quad v \in \mathcal{N}_0(T_2^2), \quad c \in C^2 \\
\omega_2(v, c) &= W, \quad v \in \mathcal{N}_0(T_3^1), \quad c \in C^1 \\
\omega_2(v, c) &= S, \quad \text{otherwise}
\end{aligned}$$

In the second half of the second iteration, the messages going out of certain check nodes depend on the connection degree of these check nodes. However, we do not want that the proof be dependent on the degree of connection of check nodes. Hence, we consider in the following the "worst" case, that is the configuration where each message has the smallest possible value. In that case, the messages along the edges in the second half of the second iteration are such that:

$$\begin{aligned}
\varpi_2(c, v) &= -W, \quad v \in V^2 \cap \mathcal{N}_2(\{v_1^1, v_2^1\}), \quad c \in C^1\backslash c_4^1 \\
\varpi_2(c_4^1, :) &= W \\
\varpi_2(c, :\backslash v) &= -W, \quad v \in \mathcal{N}_0(T_3^1), \quad c \in C^2 \cap \mathcal{N}_1(T_3^1) \\
\varpi_2(c, v) &= W, \quad v \in V^2, \quad c \in \{c_8^1, c_9^1, c_{10}^1, c_{11}^1\} \\
\varpi_2(c, :) &= W, \quad c \in C^1 \cap \mathcal{N}_1(T_3^1) \\
\varpi_2(c, :) &= W, \quad c \in C^2 \cap \mathcal{N}_1(T_2^2) \\
\varpi_2(c, v) &= S, \quad \text{otherwise}
\end{aligned}$$

At the end of the second iteration, all $v \in V^1$ receive all correct messages W or S. According to Table II, all variables in V^1 are hence corrected at the end of the second iteration. For variables in V^2, since no $-S$ messages propagate in the second half of the second iteration, we see on Table II that variables in V^2, which are not received in error, are decoded as 1 if and only if they receive four $-W$ messages. The following lemma prove that this is not possible.

Lemma 1: No variable node receives four incorrect $-W$ messages at the end of second iteration.

Proof: Let v be such a variable. Then the four neighboring checks of v must belong to $\{c_1^1, c_2^1, c_3^1, c_5^1, c_6^1, c_7^1\} \cup \left(C^2 \cap \mathcal{N}_1(T_3^1)\right)$. Note that only two neighbors of v can belong to $\{c_1^1, c_2^1, c_3^1, c_5^1, c_6^1, c_7^1\}$ without introducing a 4-cycle. This implies that there are only three cases:

- v has two neighboring checks, say c_1^2 and c_2^2, in $C^2 \cap \mathcal{N}_1(T_3^1)$, and two checks in $\{c_1^1, c_2^1, c_3^1, c_5^1, c_6^1, c_7^1\}$. Let v_1^2 and v_2^2 be the T_3^1 variables connected to c_1^2 and c_2^2. It results that the set of variables $\{v_1^1, v_2^1, v_1^2, v_2^2, v\}$ is connected to only 11 checks, which contradicts the 5→12 expansion condition. This case is hence not possible.
- v has one neighbor in $\{c_1^1, c_2^1, c_3^1, c_5^1, c_6^1, c_7^1\}$ and three neighbors in $C^2 \cap \mathcal{N}_1(T_3^1)$, say c_1^2, c_2^2 and c_3^2. Let v_1^2, v_2^2 and v_3^2 be the T_3^1 variables connected to c_1^2, c_2^2 and c_3^1. It results that the set of variables $\{v_1^1, v_2^1, v_1^2, v_2^2, v_3^2, v\}$ is connected to only 13 checks, which contradicts the 6→14 expansion condition. This case is hence not possible.
- v has four neighbors in $C^2 \cap \mathcal{N}_1(T_3^1)$, say c_1^2, c_2^2, c_3^2 and c_4^2. Let v_1^2, v_2^2, v_3^2 and v_4^2 be the T_3^1 variables connected to c_1^2, c_2^2, c_3^1 and c_4^1. It results that the set of variables $\{v_1^1, v_2^1, v_3^1, v_1^2, v_2^2, v_3^2, v_4^2, v\}$ is connected to only 15 checks, which contradicts the 8→16 expansion condition. This case is hence not possible.

■

Hence, the decoder converges at the end of the second iteration.

■

Note that similar conditions for a column-weight-four LDPC code of girth six to correct any weight-three error pattern within four iterations, when it is decoded with Gallager B algorithm, has been found by Chilappagari *et al.* [9]. The conditions are that each variable subset of size 4 has at least 11 neighbors, each one of size 5 at least 12 neighbors, each one of size 6 at least 14 neighbors, each one of size 7 at least

TABLE III
THRESHOLDS (GIVEN IN PROBABILITY OF CROSSOVER ON THE BSC) OF COLUMN-WEIGHT-FOUR CODES WITH ROW DEGREE ρ. ALGORITHM E IS PRESENTED IN [3]. FOR THE TWO-BIT DECODERS, THE SET (C,S,W) IS GIVEN.

ρ	Rate	Gallager A	Gallager B	Algorithm E
8	0.5	0.0474	0.0516	0.0583
16	0.75	0.0175	0.0175	0.0240
32	0.875	0.00585	0.00585	0.00935
ρ	Rate	(1,1,1)	(1,2,1)	(1,3,1)
8	0.5	0.0467	0.0509	0.0552
16	0.75	0.0175	0.0165	0.0175
32	0.875	0.00585	0.00562	0.00486
ρ	Rate	(2,2,1)	(2,3,1)	(3,3,1)
8	0.5	0.0567	0.0532	0.0655
16	0.75	0.0177	0.0168	0.0222
32	0.875	0.00587	0.00568	0.00754
ρ	Rate	Dynamic two-bit decoder with $S = 2$ and $W = 1$		
8	0.5	0.0638		
16	0.75	0.0249		
32	0.875	0.00953		

16 neighbors and each one of size 8 at least 18 neighbors. These conditions are stronger than the ones of Theorem 1. The higher the rate of the code, the more difficult for the Tanner graph of the code to satisfy the expansion conditions, since the variable nodes tend to be less and less connected when the code rate increases. Hence, it is likely that weaker expansion conditions, obtained for the two-bit decoder, make possible the construction of higher rate codes, with weight-three error correction capability, than expansion conditions required by the one-bit Gallager B decoder do. However, determining analytically the highest achievable rate for a given set of expansion conditions is a problem which may be very hard to solve, and which is out of the scope of this paper.

IV. THRESHOLDS OF TWO-BIT DECODERS

In this section, we report the thresholds of two-bit decoders for column-weight-four codes for different values of (C, S, W) as well as for different code rates. The details of the density evolution equations are given in [12]. Table III provides the thresholds for the various two-bit decoders as well as for the Gallager A, B and E algorithms.

From the table it is clear that there exist two-bit decoders that have better thresholds than one-bit decoders Gallager A and B algorithms. However, it should be noted that the class of two-bit decoders considered in this paper need not necessarily contain the best possible two-bit decoder. Nevertheless, our approach can be applied to any decoder to obtain similar results.

We have also derived the threshold for a dynamic two-bit decoder i.e., a decoder in which the values of C, S and W can vary with iterations. This decoder has the best thresholds among all the decoders presented. In the dynamic decoder we consider, the value of C is optimized for every iteration to achieve a very good threshold. The underlying idea is similar to the optimization performed in the threshold calculation of the Gallager E algorithm (see [3] for details). The better thresholds of the presented dynamic two-bit decoder over Algorithm E indicates that the potential of using multiple bits in the message passing alphabet even if the channel observation is still one bit.

V. DISCUSSION

We have considered a class of two-bit decoders for decoding column-weight-four LDPC codes over the BSC. Codes satisfying the conditions derived in this paper can be constructed by a modified version of the progressive edge growth (PEG) [13] algorithm (see [9] for illustration). The question whether all the valid two-bit decoders can be expressed using the simple algebraic rules presented in this paper remains open. It is also of interest to derive bounds on the achievable rate at a given code length. Future work includes investigation of the above problems as well as extending the analysis to derive sufficient conditions to guarantee correction of higher number of errors.

ACKNOWLEDGMENT

This work has been done while L. Sassatelli was with ETIS lab, and funded by the French Armament Procurement Agency (DGA). B. Vasic and S. K. Chilappagari would like to acknowledge the financial support of the NSF (Grants CCF-0634969 and IHCS-0725405).

REFERENCES

[1] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[3] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[4] L. Bazzi, T. Richardson, and R. Urbanke, "Exact thresholds and optimal codes for the binary symmetric channel and Gallager's decoding algorithm A," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2010–2021, 2004.

[5] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 782–790, 2001.

[6] M. Ivkovic, S. K. Chilappagari, and B. Vasic, "Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3763–3768, 2008. [Online]. Available: http://dx.doi.org/10.1109/TIT.2008.926319

[7] 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. Inform. Theory*, vol. 48, pp. 1570–1579, 2002.

[8] S. K. Chilappagari, A. R. Krishnan, and B. Vasic, "LDPC codes which can correct three errors under iterative decoding," in *Proc. IEEE Inform. on Theory Workshop*, 2008.

[9] S. K. Chilappagari, A. R. Krishnan, B. Vasic, and M. W. Marcellin, "Low-density parity-check codes which can correct three errors under iterative decoding," 2008, submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://arxiv.org/abs/0810.1105

[10] S. K. Chilappagari and B. Vasic, "Error correction capability of column-weight-three LDPC codes," *IEEE Trans. Inform. Theory*, accepted for publication. [Online]. Available: http://arxiv.org/abs/0710.3427

[11] S. K. Chilappagari, D. V. Nguyen, B. Vasic, and M. W. Marcellin, "Error correction capability of column-weight-three LDPC codes: Part II," July 2008, submitted to *IEEE Trans. Inform. Theory*. [Online]. Available: http://arxiv.org/abs/0807.3582

[12] L. Sassatelli, S. K. Chilappagari, B. Vasic, and D. Declercq, "Two-bit message passing decoders for ldpc codes over the binary symmetric channel," 2009, submitted to *IEEE Trans. Comm.* [Online]. Available: http://arxiv.org/PS_cache/arxiv/pdf/0901/0901.2090v3.pdf

[13] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, 2005.