ON GUARANTEED ERROR CORRECTION CAPABILITY OF GLDPC CODES

Graduate Students: Shashi Kiran Chilappagari and Dung Viet Nguyen
Advisors: Bane Vasic and Michael W. Marcellin
Dept. of Electrical and Computer Eng.
University of Arizona
Tucson, AZ 85721, USA

Abstract

In this paper, it is shown that generalized LDPC codes can correct a linear fraction of errors under the parallel bit flipping algorithm when the underlying Tanner graph is a good expander. A lower bound on the size of variable node sets which have required expansion is established as a function of the column weight of the code, the girth of the Tanner graph and the error correction capability of the sub-code. It is also shown that the bound on the required expansion cannot be improved when the column weight is even by studying a class of trapping sets. An upper bound on the guaranteed error correction capability is found by investigating the size of smallest possible trapping sets.

I. INTRODUCTION

Tanner [1] studied a class of codes constructed based on bipartite graphs and short error correcting codes. Tanner's work is a generalization of the low-density parity-check (LDPC) codes proposed by Gallager [2] and hence these codes are referred to as generalized LDPC (GLDPC) codes. Tanner proposed code construction techniques, decoding algorithms and complexity and performance analysis to analyze these codes and derived bounds on the rate and minimum distance for these codes. Sipser and Spielman [3] analyzed a special case of GLDPC codes (which they termed as expander codes) using expansion arguments and proposed explicit constructions of asymptotically good codes capable of correcting a fraction of errors. Zemor [4] improved the fraction of correctable errors under a modified decoding algorithm. Barg and Zemor [5] analyzed the error exponents of expander codes and showed that expander codes achieve capacity over the binary symmetric channel (BSC). Janwa and Lal [6] studied GLDPC codes in the most general setting by considering unbalanced bipartite graphs. Miladinovic and Fossorier [7] derived bounds on the guaranteed error correction capability of GLDPC codes for the special case of failures only decoding.

The focus of this paper is to establish lower and upper bounds on the guaranteed error correction capability of GLDPC codes as a function of their column weight, girth and the error correction capability of the sub-code [1]. We also find the expansion required to guarantee correction of a fraction of errors under the parallel bit flipping algorithm. Our approach can be summarized as follows:

Manuscript received May 30, 2008. This work is funded by NSF under Grant CCF-0634969, ECCS-0725405, ITR-0325979 and by the INSIC-EHDR program.

S. K. Chilappagari, D. V. Nguyen, B. Vasic and M. W. Marcellin are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona, 85721 USA. (emails: {shashic, nguyendv, vasic, marcellin}@ece.arizona.edu

An extended version of this work was submitted to the IEEE Transactions on Information Theory

[1] Precise definitions will be given in Section II

(a) to establish lower bounds, we determine the size of variable node sets in a left regular Tanner graph which are guaranteed to have the expansion required by the parallel bit flipping algorithm, based on the Moore bound [8, p.180] and (b) to find upper bounds, we study the size of smallest possible trapping sets [9] in a left regular Tanner graph.

It is well known that a random graph is a good expander with high probability [3]. However, the fraction of nodes having the required expansion is very small and hence the code length to guarantee correction of a fixed number of errors must be large. Moreover, determining the expansion of a given graph is known to be NP hard [10], and spectral gap methods cannot guarantee an expansion factor of more than $1/2$ [3]. On the other hand, code parameters such as column weight and girth can be easily determined or are assumed to be known for the code under consideration. We prove that for a given column weight, the error correction capability grows exponentially in girth. However, we note that since the girth grows logarithmically in the code length, this result does not show that the bit flipping algorithms can correct a linear fraction of errors.

To find an upper bound on the number of correctable errors, we study the size of sets of variable nodes which lead to decoding failures. A decoding failure is said to have occurred if the output of the decoder is not equal to the transmitted codeword [9]. The conditions that lead to decoding failures are well understood for a variety of decoding algorithms such as maximum likelihood decoding, bounded distance decoding and iterative decoding on the binary erasure channel (BEC). However, for iterative decoding on the BSC and the additive white Gaussian noise (AWGN) channel, the understanding is far from complete. Two approaches have been taken in this direction, namely trapping sets [9] and pseudo-codewords [11]. We adopt the trapping set approach in this paper to characterize decoding failures. Richardson [9] introduced the notion of trapping sets to estimate the error floor on the AWGN channel. In [12], trapping sets were used to estimate the frame error rate of column-weigh-three LDPC codes. In this paper, we define trapping sets with the help of fixed points for the bit flipping algorithm. We then find bounds on the size of trapping sets, thereby finding an upper bound on the guaranteed error correction capability. By saying that a code with column weight γ and girth $2g'$ is not guaranteed to correct k errors, we mean that there exists a code with column weight γ and girth $2g'$ that fails to correct k errors.

The rest of the paper is organized as follows. In Section II, we provide a brief introduction to graph theory notation, decoding algorithms and trapping sets [9]. In Section III, we prove that the parallel bit flipping algorithm can correct a fraction of errors if the underlying Tanner graph is a good expander. We establish bounds on the size of variable node sets having the required expansion in Section IV. In Section V, we study trapping sets for GLDPC codes and conclude with a few remarks in Section VI.

II. Preliminaries

In this section, we first establish the notation and then proceed to give a brief introduction to GLDPC codes and hard decision decoding algorithms. We then describe trapping sets for the bit flipping algorithm.

A. Graph Theory Notation

We adopt the standard notation in graph theory (see [13] for example). $G = (U, E)$ denotes a graph with set of nodes U and set of edges E. When there is no ambiguity, we simply denote the graph by G. An edge e is an unordered pair (u_1, u_2) of nodes and is said to be incident on u_1 and u_2.

Two nodes u_1 and u_2 are said to be adjacent (neighbors) if there is an edge $e = (u_1, u_2)$ incident on them. The order of the graph is $|U|$ and the size of the graph is $|E|$. The degree of u, $d(u)$, is the number of its neighbors. A node with degree one is called a leaf or a pendant node. A graph is d-regular if all the nodes have degree d. The average degree $\overline{d}$ of a graph is defined as $\overline{d} = 2|E|/|U|$. The girth $g(G)$ of a graph G, is the length of smallest cycle in G. $H = (V \cup C, E')$ denotes a bipartite graph with two sets of nodes; variable (left) nodes V and check (right) nodes C and edge set E'. Nodes in V have neighbors only in C and vice versa. A bipartite 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 (γ, ρ) regular if all variable nodes have degree γ and all check nodes have degree ρ. The girth of a bipartite graph is even.

B. GLDPC Codes and Decoding Algorithms

Definition 1 (Definition 6, [3]): Let G be a (γ, ρ) regular bipartite graph between n variable nodes $(v_1, v_2, \ldots, v_n)$ and $n\gamma/\rho$ check nodes $(c_1, c_2, \ldots, c_{n\gamma/\rho})$. Let $b(i, j)$ be a function designed so that, for each check node c_i, the variables neighboring c_i are $v_{b(i,1)}, v_{b(i,2)}, \ldots, v_{b(i,\rho)}$. Let $\mathcal{S}$ be an error correcting code of block length ρ. The GLDPC code $\mathcal{C}(G, \mathcal{S})$ is the code of block length n whose codewords are the words $(x_1, x_2, \ldots, x_n)$ such that, for $1 \leq i \leq n\gamma/\rho$, $(x_{b(i,1)}, \ldots, x_{b(i,\rho)})$ is a codeword of $\mathcal{S}$.

Such a graphical representation is called the Tanner graph [1] of the code. The adjacency matrix of G gives a parity check matrix of $\mathcal{C}$. An (n, γ, ρ) regular GLDPC code has a Tanner graph with n variable nodes each of degree γ (column weight) and $n\gamma/\rho$ check nodes each of degree ρ (row weight). The code $\mathcal{S}$ at each check node is sometimes referred to as the sub-code.

Tanner [1] proposed different hard decision decoding algorithms to decode GLDPC codes. We now describe an iterative algorithm known as the parallel bit flipping algorithm originally described in [1], which is employed when the sub-code is capable of correcting t errors.

Parallel bit flipping algorithm: Each decoding round consists of the following steps.

- A variable node sends its current estimate to check nodes.
- A check node performs decoding on incoming messages and finds the nearest codeword. For all variable nodes which differ from the codeword, the check node sends a flip message. If the check node does not find a unique codeword, it does not send any flip messages.
- A variable node flips if it receives more than $\gamma/2$ flip messages.

The set of variable nodes which differ from their correct value are known as corrupt variables. The rest of the variable nodes are referred to as correct variables. Following the algorithms, we have the following definition adopted from [3]:

Definition 2: A check node is said to be *confused* if it sends flip messages to correct variable nodes, or if it does not send flip message to corrupt variable nodes, or both. Otherwise, a check node is said to be *helpful*.

Remarks:

1) For the parallel bit flipping decoding algorithm, a check node with sub-code of minimum distance at least $d_{min} = 2t + 1$ can be confused only if it is connected to more than t corrupt variable nodes.
2) The parallel bit flipping algorithm is different from the algorithm presented by Sipser and Spielman in [3] for expander codes, but is similar to the algorithm proposed by Zemor in [4].

However, we note that the codes considered in [4] are based on d-regular bipartite graphs and are a special case of doubly generalized LDPC codes, where each variable node is also associated with an error correcting code.

3) Apart from helpful checks and confused checks, Sipser and Spielman defined unhelpful checks. However, our definition of confused checks includes unhelpful checks as well.
4) Miladinovic and Fossorier in [7] considered a decoding algorithm where the decoding at every check either results in correct decoding or a failure but not miscorrection. While this assumption is reasonable when the sub-code is a long code, it is not true in general. We however, point out that the methodology we adopt can be applied to this case as well.
5) The work by Sipser and Spielman [3], Zemor [4], Barg and Zemor [5] and Janwa and Lal [6] focused on asymptotic results and explicit construction of expander codes. The proofs and constructions are based on spectral gap and as noted earlier, such methods cannot guarantee expansion factor of more than 1/2. Our proofs require a greater expansion factor.

C. Decoding Failures and Trapping Sets

We now characterize failures of the iterative decoders using fixed points and trapping sets. Some of the following discussion appears in [14], [12], [15] and we include it for sake of completeness.

Consider a GLDPC code of length n and let $\mathbf{x} = (x_1 x_2 \dots x_n)$ be the binary vector which is the input to the iterative decoder. Let $S(\mathbf{x})$ be the support of $\mathbf{x}$. The support of $\mathbf{x}$ is defined as the set of all positions i where $x_i \neq 0$.

Definition 3: [14] A decoder failure is said to have occurred if the output of the decoder is not equal to the transmitted codeword.

Definition 4: $\mathbf{x}$ is a fixed point of the bit flipping algorithm if the set of corrupt variables remains unchanged after one round of decoding.

Definition 5: [12] The support of a fixed point is known as a trapping set. A (V, C) trapping set $\mathcal{T}$ is a set of V variable nodes whose induced subgraph has C odd degree checks.

It is clear that if the variable nodes corresponding to a trapping set are in error, then a decoder failure occurs.

III. Expansion and Error Correction Capability

We now prove that the above described algorithm can correct a fraction of errors if the underlying Tanner graph is a good expander. We start with the following definitions from [3].

Definition 6: Let $G = (U, E)$ with $|U| = n_1$. Then *every set of at most m_1 nodes expands by a factor of δ* if, for all sets $S \subset U$

$$|S| \leq m_1 \Rightarrow |\{y : \exists x \in S \text{ such that } (x, y) \in E\}| > \delta |S|.$$

We consider bipartite graphs and expansion of variable nodes only.

Definition 7: A graph is a $(\gamma, \rho, \alpha, \delta)$ expander if it is a (γ, ρ) regular bipartite graph in which every subset of at most α fraction of the variable nodes expands by a factor of at least δ.

We now have the following theorem.

Theorem 1: Let $\mathcal{C}(G, \mathcal{S})$ be a GLDPC code with a γ-left regular Tanner graph G. Assume that the sub-code $\mathcal{S}$ has minimum distance at least $d_{min} = 2t + 1$ and is capable of correcting t errors. Let G be a $(\gamma, \rho, \alpha, \beta\gamma)$ expander where

$$1 > \beta > \frac{t+2}{2(t+1)}.$$

Then the parallel bit flipping decoding algorithm will correct any $\alpha_0 \leq \alpha$ fraction of errors.

Proof: Let n be the number of variable nodes in $\mathcal{C}$. Let V be the set of corrupt variables at the beginning of a decoding round. Assume that $|V| \leq \alpha n$. We will show that after the decoding round, the number of corrupt variables is strictly less than $|V|$.

Let F be the set of corrupt variables that fail to flip in one decoding round, and let C be the set of variables that were originally uncorrupt, but which become corrupt after one decoding round. After one decoding round, the set of corrupt variables is $F \cup C$. In the worst case scenario, a confused check sends t flip messages to the uncorrupt variables and no flip message to the corrupt variables. We now have the following lemma:

Lemma 1: Let C_k be the set of confused checks, then

$$|C_k| < \frac{(1-\beta)\gamma|V|}{t}. \tag{1}$$

Proof: The total number of edges connected to the corrupt variables is $\gamma|V|$. Each confused check must have at least $t+1$ neighbors in V. Let S be the set of helpful checks that have at least one neighbor in V. Then,

$$\gamma|V| \geq |C_k|(t+1) + |S|. \tag{2}$$

By expansion,

$$|S| + |C_k| > \beta\gamma|V|. \tag{3}$$

By (2) and (3), we obtain

$$|C_k| < \frac{(1-\beta)\gamma|V|}{t}.$$

■

We now prove that $|F \cup C| < |V|$. The proof is by contradiction. Assume that $|F \cup C| \geq |V|$. Then there exists a subset $C' \subset C$ such that $|F \cup C'| = |V|$. We observe that a variable node in F can have at most $\lfloor \gamma/2 \rfloor$ neighbors that are not in C_k. Also, a variable node in C' must have at least $\lfloor \gamma/2 \rfloor + 1$ neighbors in C_k, and hence can have at most $\lceil \gamma/2 \rceil - 1$ neighbors that are not in C_k. Let $N(F \cup C')$ be the set of neighbors of $F \cup C'$. Then,

$$\begin{aligned} N(F \cup C') &\leq |C_k| + \lfloor \frac{\gamma}{2} \rfloor |F| + \left(\lceil \frac{\gamma}{2} \rceil - 1 \right) |C'| \\ &< |C_k| + \frac{\gamma}{2}|F| + \frac{\gamma}{2}|C'| = |C_k| + \frac{\gamma}{2}|V|. \end{aligned} \tag{4}$$

Substituting (1) into (4), we obtain

$$N(F \cup C') < \left(\frac{1-\beta}{t} + \frac{1}{2} \right) \gamma|V|.$$

Now

$$\begin{aligned} & \beta > \frac{t+2}{2(t+1)} \\ \Rightarrow \quad & \frac{1-\beta}{t} < \frac{2\beta-1}{2} \\ \Rightarrow \quad & \frac{1-\beta}{t} + \frac{1}{2} < \beta \\ \Rightarrow \quad & N(F \cup C') < \beta\gamma|V| \end{aligned}$$

which is a contradiction. ■

Remark: The above theorem proves that the parallel bit flipping algorithm can correct a fraction of errors in linear number of rounds (in code length). However, if we assume an expansion of $(\beta + \epsilon)\gamma$, it can be shown that the number of errors decreases by a constant factor with every iteration resulting in convergence in logarithmic number of rounds.

IV. Column Weight, Girth and Expansion

In this section, we find lower bounds on the size of variable node sets that are guaranteed to have the required expansion as a function of the column weight, girth and error correction capability of the sub-code. We begin with some definitions.

A. Definitions

Definition 8: The *reduced graph* $H_r = (V \cup C_r, E'_r)$ of $H = (V \cup C, E')$ is a graph with vertex set $V \cup C_r$ and edge set E'_r given by

$$\begin{aligned} C_r &= C \setminus C_p,\ C_p = \{c \in C : \text{c is a pendant node}\} \\ E'_r &= E' \setminus E'_p,\ E'_p = \{(v_i, c_j) \in E : c_j \in C_p\}. \end{aligned}$$

Definition 9: Let $H = (V \cup C, E')$ be such that $\forall v \in V, d(v) \leq \gamma$. The γ *augmented graph* $H_\gamma = (V \cup C_\gamma, E'_\gamma)$ is a graph with vertex set $V \cup C_\gamma$ and edge set E'_γ given by

$$\begin{aligned} C_\gamma &= C \cup C_a, \text{ where } C_a = \bigcup_{i=1}^{|V|} C_a^i \text{ and} \\ C_a^i &= \{c_1^i, \ldots, c_{\gamma - d(v_i)}^i\}; \\ E'_\gamma &= E' \cup E'_a, \text{ where } E'_a = \bigcup_{i=1}^{|V|} E'^i_a \text{ and} \\ E'^i_a &= \{(v_i, c_j) \in V \times C_a : c_j \in C_a^i\}. \end{aligned}$$

Definition 10: [3, Definition 4] The *edge-vertex incidence graph* $G_{ev} = (U \cup E, E_{ev})$ of $G = (U, E)$ is the bipartite graph with vertex set $U \cup E$ and edge set

$$E_{ev} = \{(e, u) \in E \times U : u \text{ is an endpoint of e}\}.$$

Notes:

1) The edge-vertex incidence graph is right regular with degree two.

2) $|E_{ev}| = 2|E|$.
3) $g(G_{ev}) = 2g(G)$.

Definition 11: An *inverse edge-vertex incidence graph* $H_{iev} = (V, E'_{iev})$ of $H = (V \cup C, E')$ is a graph with vertex set V and edge set E'_{iev} which is obtained as follows. For $c \in C_r$, let $N(c)$ denote the set of neighbors of c. Label one node $v_i \in N(c)$ as a root node. Then

$$\begin{aligned} E'_{iev} &= \{(v_i, v_j) \in V \times V : v_i \in N(c), v_j \in N(c), \\ &\quad i \neq j,\ v_i \text{ is a root node, for some } c \in C_r\}. \end{aligned}$$

Notes:

1) Given a graph, the inverse edge-vertex incidence graph is not unique.
2) $g(H_{iev}) \geq g(H)/2$, $|E'_{iev}| = |E'_r| - |C_r|$ and $|C_r| \leq |E'_r|/2$.
3) $|E'_{iev}| \geq |E'_r|/2$ with equality only if all checks in C_r have degree two.
4) The term inverse edge-vertex incidence is used for the following reason. Suppose all checks in H have degree two. Then the edge-vertex incidence graph of H_{iev} is H.

The *Moore bound* [8, p.180] denoted by $n_0(d, g)$ is a lower bound on the least number of vertices in a d-regular graph with girth g. It is given by

$$\begin{aligned} n_0(d, g) = n_0(d, 2r+1) &= 1 + d\sum_{i=0}^{r-1}(d-1)^i,\ g \text{ odd} \\ n_0(d, g) = n_0(d, 2r) &= 2\sum_{i=0}^{r-1}(d-1)^i,\ g \text{ even}. \end{aligned}$$

In [16], it was shown that a similar bound holds for irregular graphs.

Theorem 2: [16] The number of nodes $n(\overline{d}, g)$ in a graph of girth g and average degree at least $\overline{d} \geq 2$ satisfies

$$n(\overline{d}, g) \geq n_0(\overline{d}, g).$$

Note that $\overline{d}$ need not be an integer in the above theorem.

The following theorem establishes a lower bound on the number of nodes in a left regular graph which expand by a factor required by the parallel bit flipping algorithm.

Theorem 3: Let G be a γ-left regular Tanner graph G with $g(G) = 2g'$. Then for all $k < n_0(\gamma t/(t+1), g')$, any set of k variable nodes in G expands by a factor of at least $\beta\gamma$ where,

$$\beta = \frac{t+2}{2(t+1)}.$$

Proof: Let $G^k = (V^k \cup C^k, E^k)$ denote the subgraph induced by a set of k variable nodes V^k. Since G is γ-left regular, $|E^k| = \gamma k$. Let $G^k_r = (V^k \cup C^k_r, E^k_r)$ be the reduced graph. We have

$$\begin{aligned} |C^k| &= |C^k_r| + |C^k_p| \\ |E^k| &= |E^k_p| + |E^k_r| \\ |E^k_p| &= |C^k_p| \\ |C^k_p| &= \gamma k - |E^k_r|. \end{aligned}$$

We need to prove that $|C^k| > \beta\gamma k$.

Let $f(k, g')$ denote the maximum number of edges in an arbitrary graph of order k and girth g'. By Theorem 2, for all $k < n_0(\gamma t/(t+1), g')$, the average degree of a graph with k nodes and girth g' is less than $\gamma t/(t+1)$. Hence, $f(k, g') < \gamma tk/(2(t+1))$. We now have the following lemma.

Lemma 2: The number of edges in G_r^k cannot exceed $2f(k, g')$ i.e.,

$$|E_r^k| \leq 2f(k, g').$$

Proof: The proof is by contradiction. Assume that $|E_r^k| > 2f(k, g')$. Consider $G_{iev}^k = (V^k, E_{iev}^k)$, an inverse edge vertex incidence graph of G^k. We have

$$|E_{iev}^k| > f(k, g').$$

This is a contradiction as G_{eiv}^k is a graph of order k and girth at least g'. ■

We now find a lower bound on $|C^k|$ in terms of $f(k, g')$. We have the following lemma.

Lemma 3: $|C^k| \geq \gamma k - f(k, g')$.

Proof: Let $|E_r^k| = 2f(k, g') - j$ for some integer $j \geq 0$. Then $|E_p^k| = \gamma k - 2f(k, g') + j$. We claim that $|C_r^k| \geq f(k, g') + j$. To see this, we note that

$$\begin{aligned} |E_{iev}^k| &= |E_r^k| - |C_r^k|, \text{ or} \\ |C_r^k| &= |E_r^k| - |E_{iev}^k|. \end{aligned}$$

But

$$\begin{aligned} |E_{iev}^k| &\leq f(k, g') \\ \Rightarrow |C_r^k| &\geq 2f(k, g') - j - f(k, g') \\ \Rightarrow |C_r^k| &\geq f(k, g') - j. \end{aligned}$$

Hence we have,

$$\begin{aligned} |C^k| &= |C_r^k| + |C_p^k| \\ \Rightarrow |C^k| &\geq f(k, g') - j + \gamma k - 2f(k, g') + j \\ \Rightarrow |C^k| &\geq \gamma k - f(k, g'). \end{aligned}$$

■

The theorem now follows as

$$f(k, g') < \frac{k\gamma t}{2(t+1)}.$$

and therefore

$$|C^k| > \frac{t+2}{2(t+1)}\gamma k.$$

■

Note that the above theorem holds when $\gamma t/(t+1) \geq 2$.

Corollary 1: Let $\mathcal{C}(G, \mathcal{S})$ be a GLDPC code with a γ-left regular Tanner graph G and $g(G) = 2g'$. Assume that the sub-code $\mathcal{S}$ has minimum distance at least $d_{min} = 2t + 1$ and is capable of correcting t errors. Then the parallel bit flipping algorithm can correct any error pattern of weight less than $n_0(\gamma t/(t+1), g')$.

V. TRAPPING SETS OF GLDPC CODES

We now exhibit a trapping set for the parallel bit flipping algorithm. By examining the expansion of the trapping set, we show that the bound given in Theorem 1 cannot be improved when γ is even.

Theorem 4: Let $\mathcal{C}(G,\mathcal{S})$ be a GLDPC code with γ-left regular Tanner graph G and a t error correcting sub-code $\mathcal{S}$. Let $\mathcal{T}$ be a set consisting of V variable nodes with induced subgraph $\mathcal{I}$ with the following properties: (a) The degree of each check in $\mathcal{I}$ is either 1 or $t+1$; (b) Each variable node in V is connected to $\lceil\gamma/2\rceil$ checks of degree $t+1$ and $\lfloor\gamma/2\rfloor$ checks of degree 1; and (c) No $\lfloor\gamma/2\rfloor+1$ checks of degree $t+1$ share a variable node outside $\mathcal{I}$. Then, $\mathcal{T}$ is a trapping set.

Proof: Observe that all the checks of degree $t+1$ in $\mathcal{I}$ are confused. Further, each confused check does not send flip messages to variable nodes in V. Since any variable node in V is connected to $\lceil\gamma/2\rceil$ confused checks, it remains corrupt. Also, no variable node outside $\mathcal{I}$ can receive more than $\lfloor\gamma/2\rfloor$ flip messages. Hence, no variable node which is originally correct can get corrupted. By definition, $\mathcal{T}$ is a trapping set.

It can be seen that the total number of checks in $\mathcal{I}$ is equal to $|V|(\lfloor\gamma/2\rfloor+\lceil\gamma/2\rceil/(t+1))$. Hence, the set of variable nodes V expands by a factor of $\gamma(t+2)/(2(t+1))$ when γ is even. Hence, the bound given in Theorem 1 cannot be improved in this case. ■

To determine whether a given set of variables is a trapping set, it is necessary to not only know the induced subgraph but also the neighbors of the odd degree checks. However, in order to establish general bounds on the sizes of trapping sets given only the column weight and the girth, we consider only conditions (a) and (b) of Theorem 4 which are necessary conditions. A set of variable nodes satisfying conditions (a) and (b) is known as a *potential trapping set*. For a set of variable nodes to be a trapping set, it is necessary that every variable node in the set is connected to at least $\lceil\gamma/2\rceil$ confused checks. This observation leads to the following bound on the size of trapping sets.

Theorem 5: Let $\mathcal{C}(G,\mathcal{S})$ be a GLDPC code with γ-left regular Tanner graph G and a t error correcting sub-code $\mathcal{S}$. Let $n_c(d_l,d_r,2g')$ denote the number of left vertices in a (d_l,d_r) regular bipartite graph of girth $2g'$. Then the size of the smallest possible trapping set $\mathcal{T}(\gamma,\mathcal{S},2g')$ of $\mathcal{C}(G,\mathcal{S})$ is $n_c(\lceil\gamma/2\rceil,t+1,2g')$.

Proof: We first prove the following lemma and then exhibit a potential trapping set of size $n_c(\lceil\gamma/2\rceil,t+1,2g')$.

Lemma 4: $|\mathcal{T}(\gamma,\mathcal{S},2g')|\geq n_c(\lceil\gamma/2\rceil,t+1,g')$.

Proof: Let $\mathcal{T}_1$ be a trapping set with $|\mathcal{T}_1|<n_c(\lceil\gamma/2\rceil,t+1,2g')$ and let G_1 denote the induced subgraph of $\mathcal{T}_1$. We can construct a $(\lceil\gamma/2\rceil,t+1)$ regular bipartite graph with girth $(g''\geq 2g')$ with $|\mathcal{T}_1|<n_c(\lceil\gamma/2\rceil,t+1,2g')$ nodes by removing edges and check nodes (if necessary) from G_1 which is a contradiction. ■

We now exhibit a potential trapping set of size $n_c(\lceil\gamma/2\rceil,t+1,2g')$. Let $G(\lceil\gamma/2\rceil,t+1,2g')$ be a $(\lceil\gamma/2\rceil,t+1)$ regular bipartite graph with girth $2g'$. Note that in $G(\lceil\gamma/2\rceil,g')$, all the variable nodes have degree $\lceil\gamma/2\rceil$ and all checks have degree $t+1$. Now consider $G_\gamma(\lceil\gamma/2\rceil,t+1,2g')$, the γ augmented graph of $G(\lceil\gamma/2\rceil,t+1,2g')$. It can be seen that $G_\gamma(\lceil\gamma/2\rceil,g')$ is a potential trapping set. ■

Corollary 2: Let $\mathcal{C}(G,\mathcal{S})$ be a GLDPC code with a γ-left regular Tanner graph G and $g(G)=2g'$.

Assume that the sub-code $\mathcal{S}$ has minimum distance at least $d_{min} = 2t + 1$ and is capable of correcting t errors. Then the parallel bit flipping algorithm cannot be guaranteed to correct all error patterns of weight greater than or equal to $n_c(\lceil \gamma/2 \rceil, t + 1, 2g')$.

VI. Concluding Remarks

We derived lower bounds on the guaranteed error correction capability of GLDPC codes by finding bounds on the number of nodes that have the required expansion. The bounds depend on three important code parameters namely: column weight, girth and error correction capability of the sub-code. Since the relations between rate, column weight, girth and code length are well explored in the literature (see [2], [1] for example), bounds on the code length needed to achieve certain error correction capability can be derived for different column weights and sub-codes (for GLDPC codes). The bounds presented in the paper serve as guidelines in choosing code parameters in practical scenarios.

The lower bounds derived in this paper are weak. However, extremal graphs avoiding three, four and five cycles have been studied in great detail (see [17], [18]) and these results can be used to derive tighter bounds when the girth is eight, ten or twelve. The results can be extended to message passing algorithms as well. Results similar to the ones reported by Miladinovic and Fossorier [7] based on the size of generalized stopping sets can also be derived.

References

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

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

[3] M. Sipser and D. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[4] G. Zemor, "On expander codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 835–837, Feb. 2001.

[5] A. Barg and G. Zemor, "Error exponents of expander codes," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1725–1729, Jun. 2002.

[6] H. Janwa and A. K. Lal, "On tanner codes: minimum distance and decoding," *Appl. Algebra Eng. Commun. Comput.*, vol. 13, no. 5, pp. 335–347, 2003.

[7] N. Miladinovic and M. Fossorier, "Generalized ldpc codes with reed-solomon and bch codes as component codes for binary channels," vol. 3, 28 Nov. -2 Dec. 2005, pp. 6–10.

[8] N. Biggs, *Algebraic graph theory*. Cambridge: Cambridge University Press, 1993.

[9] T. J. Richardson, "Error floors of LDPC codes," in *Proc. of 41st Annual Allerton Conf. on Communications, Control and Computing*, 2003, pp. 1426–1435.

[10] N. Alon, "Spectral techniques in graph algorithms," in *LATIN '98: Proceedings of the Third Latin American Symposium on Theoretical Informatics*. London, UK: Springer-Verlag, 1998, pp. 206–215.

[11] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," May 2007, accepted for IEEE Transactions on Information Theory. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0512078

[12] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error floors of LDPC codes on the binary symmetric channel," in *Proc. of IEEE International Conference on Communications*, vol. 3, June 11-15 2006, pp. 1089–1094.

[13] B. Bollobas, *Extremal graph theory*. London: Academic Press Inc., 1978.

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

[15] S. Sankaranarayanan, S. K. Chilappagari, R. Radhakrishnan, and B. Vasic, "Failures of the Gallager B decoder: Analysis and applications," in *Proc. of UCSD Center for Information Theory and its Applications Inaugural Workshop*, Feb 6-9 2006. [Online]. Available: htpp//ita.5i.net/papers/160.pdf

[16] N. Alon, S. Hoory, and M. Linial, "The moore bound for irregular graphs," *Graphs and Combinatorics*, vol. 18, no. 1, pp. 53–57, 2002.

[17] D. K. Garnick, Y. H. H. Kwong, and F. Lazebnik, "Extremal graphs without three-cycles or four-cycles," *J. Graph Theory*, vol. 17, no. 5, pp. 633–645, 1993.

[18] Y. Yuansheng, L. Xiaohui, D. Guocheng, and Z. Yongxiang, "Extremal graphs without three-cycles, four-cycles or five-cycles," *Utilitas Mathematica*, vol. 66, pp. 249–266, 2004.