Iterative Construction of Regular LDPC Codes from Independent Tree-Based Minimum Distance Bounds

Eric Psota and Lance C. Pérez

***Abstract*—An independent tree-based method for lower bounding the minimum distance of low-density parity-check (LDPC) codes is presented. This lower-bound is then used as the decision criterion during the iterative construction of regular LDPC codes. The new construction algorithm results in LDPC codes with greater girth and improved minimum-distance bounds when compared to regular LDPC codes constructed using the progressive edge-growth (PEG) construction and the approximate cycle extrinsic message degree (ACE)-constrained PEG construction. Simulation results of codes constructed with the new method show improved performance on the additive white Gaussian noise channel at moderate signal-to-noise ratios.**

***Index Terms*—LDPC codes, progressive edge-growth (PEG), girth, computation tree, minimum distance, iterative decoding.**

I. Introduction

SINCE the rediscovery of low-density parity-check (LDPC) codes [1], several methods have been devised for constructing codes that perform well with iterative decoders [2] [3] [4] [5]. One such construction, known as progressive edge-growth (PEG), takes a set of variable node and check node degrees and attempts to maximize the girth of the resultant Tanner graph [2]. The *girth* is the length of the shortest cycle in the Tanner graph. As shown in [2], girth can be used to compute a lower bound on the minimum distance of the code.

In this work, a new lower bound on the minimum distance of regular and irregular LDPC codes is derived. This new independent tree-based (ITB) lower-bound is obtained by constructing independent trees rooted at each variable node in the LDPC code. The notion of independent trees was first developed in earlier work on the extrinsic tree decoder [6] [7]. Unlike computation trees [8], the minimum-weight configuration on each independent tree can be easily determined, and used to compute a lower bound on the minimum distance of the code. This new lower bound is proven to be greater than or equal to the girth-based lower bound on the minimum distance of regular codes.

A new method is also presented for constructing LDPC codes by iteratively improving the independent tree-based lower bound on the minimum distance. Codes constructed in this manner are referred to as independent tree-based (ITB) LDPC codes. The regular ITB LDPC codes have improved girth profiles and minimum distance bounds when compared to both regular PEG LDPC codes and the more recent approximate cycle extrinsic message degree (ACE)-constrained PEG LDPC codes [5]. In addition, simulations show that the ITB LDPC codes achieve lower bit error rates at high SNRs.

Manuscript received November 18, 2010. The associate editor coordinating the review of this letter and approving it for publication was M. Ardakani.

The authors are with the Department of Electrical Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588 (e-mail: epsota24@huskers.unl.edu).

Digital Object Identifier 10.1109/LCOMM.2011.011311.102252

II. Notation and Background

The block length and dimension of LDPC codes will be denoted by N and K, respectively. The parity-check matrix H of an LDPC code has N columns and $M \geq N-K$ rows. The Tanner graph of an LDPC code consists of a set of variable nodes $V = \{v_1, \ldots, v_N\}$ and check nodes $F = \{f_1, \ldots, f_M\}$, such that an edge connects v_i and f_j if and only if $H_{j,i} = 1$. The set of check nodes connected to variable node v_i is denoted $N(v_i)$, and the set of variable nodes connected to check node f_j is denoted $N(f_j)$. The degree of variable node v_i and check node f_j will be denoted d_{v_i} and d_{f_j}, respectively. A (d_V, d_F)-regular LDPC code is one in which $d_{v_i} = d_V$ for all $i = 1, \ldots, N$ and $d_{f_j} = d_F$ for all $j = 1, \ldots, M$. Irregular LDPC codes have at least one pair of variable nodes $\{v_i, v_j\}$ or check nodes $\{f_i, f_j\}$ such that $d_{v_i} \neq d_{v_j}$ or $d_{f_i} \neq d_{f_j}$.

The PEG construction aims to produce LDPC codes with a large girth $\mathcal{G}$ by sequentially determining the connections for each of the variable nodes in the code. After considering each candidate check node connected to a variable node, the one that induces the largest minimum cycle length on the current Tanner graph is chosen. This process is repeated until the desired variable node degree is achieved.

The PEG construction given in [2] does not guarantee that each f_i has degree $d_{f_i} = d_F$. However, check node regularity is desirable for both decoder convergence speed and hardware implementation [9]. A minor modification of PEG construction forces check node regularity. If the minimum check node degree among all candidates at distance ℓ from the variable node is greater than d_F, set $\ell = \ell - 1$, then randomly select an f_i with degree $d_{f_i} < d_F$. The process of reducing ℓ may be done recursively until an f_i with degree $d_{f_i} < d_F$ is found. The PEG codes considered here are constructed using this modification to force regularity, but it has a negligible effect on the girth and performance of the resulting LDPC code.

In [2], it is also shown that the code's girth can be used to lower bound the minimum distance of the code. Specifically, for variable node degree regular LDPC codes

$$d_{\mathcal{G}\text{-min}} \geq 1 + d_V \left(\sum_{i=1}^{\lfloor \frac{\mathcal{G}-2}{4} \rfloor} (d_V - 1)^{i-1} \right) \tag{1}$$

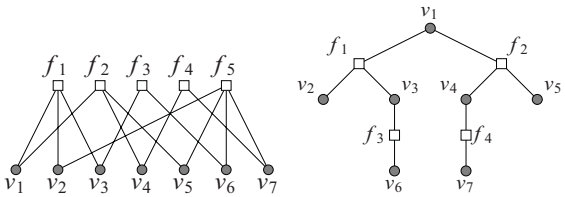
when $\frac{\mathcal{G}}{2}$ is odd, and

$$d_{\mathcal{G}\text{-min}} \geq 1 + d_V \left(\sum_{i=1}^{\lfloor \frac{\mathcal{G}-2}{4} \rfloor} (d_V - 1)^{i-1} \right) + (d_V - 1)^{\lfloor \frac{\mathcal{G}-2}{4} \rfloor} \tag{2}$$

when $\frac{\mathcal{G}}{2}$ is even [2]. In Section III, a new lower bound derived from independent trees, and shown to improve upon the girth-based bounds of (1) and (2).

Independent trees are connected subgraphs of the Tanner graph designed to maximize the distance between the root node and the leaf nodes. Each variable node in an independent

1089-7798/11$25.00 © 2011 IEEE



(a) Tanner Graph (b) Independent Tree rooted at v_1

Fig. 1. Independent tree rooted at variable node v_i for a length $N = 7$, dimension $K = 3$ LDPC code.

tree is connected to a subset of its neighbors in the Tanner graph, and each check node in an independent tree has the same set of neighbors as its copy in the Tanner graph. Thus, any binary assignment to the variable nodes that satisfies each check node in the Tanner graph will also satisfy each check node in the independent trees. The method for constructing independent trees is given in Algorithm 1.

Algorithm 1 (Independent Tree Construction)**.**
for $i = 1, \ldots, N$
  Root tree at variable node v_i, and set level to $\ell = 0$.
  while one or more variable nodes exist in level ℓ.
    for each variable node v_k in level ℓ
      (a) *If check node $f_j \in N(v_k)$ is not the parent node of v_k, connect f_j to v_k, and connect all other variable nodes $v_l \in N(f_j) \backslash v_k$ to f_j at level $\ell + 1$.*
      (b) *If no copies of variable nodes $v_l \in N(f_j) \backslash v_k$ currently exists in the independent tree, keep f_j and its child variable nodes. Otherwise, eliminate f_j and its child variable nodes.*
    end
    $\ell = \ell + 1$.
  end
end

Figure 1(b) shows the independent tree rooted at variable node v_1 constructed using Algorithm 1 for the $N = 7$, $K = 3$ code with Tanner graph shown in Figure 1(a). Note that all codeword configurations on the Tanner graph correspond to valid configurations on the independent tree. Thus, the minimum-weight valid configuration involving v_1 on the independent tree can not exceed the weight of the minimum distance codeword involving v_1.

III. Independent Tree-Based Minimum Distance Bounds and Code Construction

Knowledge of the relationships between configurations on the Tanner graph and configurations on independent trees makes it possible to derive properties of the code from its independent trees [8]. In order to lower bound the minimum-weight codeword that the root node of the independent tree is involved in, it is sufficient to compute the minimum-weight deviation on the corresponding independent tree. A computationally efficient way to determine the minimum-weight deviation on the independent tree is to assign a log-likelihood ratio cost of $+1.0$ to all the variable nodes, and then perform min-sum (MS) decoding [8] operations from the leaf nodes up to the root node. The resulting output of MS at the root node v_i is the weight of the minimum-weight deviation on the independent tree and is denoted $d_{\text{ITB},i}$. Given the set of $d_{\text{ITB},i}$ it is trivial to show the following.

Theorem 2. *The minimum distance of an LDPC code is lower bounded by*

$$d_{\text{min}} \geq d_{\text{ITB-min}} = \min_{i=1,\ldots,N} d_{\text{ITB},i}. \tag{3}$$

Note that Theorem 2 applies to both regular and irregular LDPC codes. We now show the following proposition.

Proposition 3. *Given a parity-check matrix H of a variable node degree regular LDPC code with girth $\mathcal{G}$, the independent tree-based lower bound $d_{\text{ITB},i}$ for any variable node v_i is greater than or equal to $d_{\mathcal{G}\text{-min}}$.*

Proof: Assume that the $\frac{\mathcal{G}}{2}$ is odd, and $d_{\text{ITB},i}$ is less than the bound obtained from (1). This implies that a variable node appears more than once in the independent tree during the construction of some level $\ell \leq \frac{\mathcal{G}-2}{4}$. Since the length of any path from one node to another in the independent tree after $\ell \leq \frac{\mathcal{G}-2}{4}$ levels is less than or equal to $\mathcal{G}-2$, no variable nodes could have been eliminated, because they would appear on the independent tree for the first time. Thus, $d_{\text{ITB},i}$ is greater than or equal to the bound given by (1) when $\frac{\mathcal{G}}{2}$ is odd.

Now assume that the $\frac{\mathcal{G}}{2}$ is even, and $d_{\text{ITB},i}$ is less than the bound obtained from (2). This implies that one node appears more than once before or during the consideration of the first $((d_F - 1)(d_V - 1))^{\lfloor \frac{\mathcal{G}-2}{4} \rfloor}$ check nodes of level $\ell = \lfloor \frac{\mathcal{G}-2}{4} \rfloor + 1$ in the independent tree. This requires that there is a path from a variable node to a copy of itself of length $\mathcal{G} - 2$ or less, which is not possible since that implies that the girth is less than $\mathcal{G}$. Thus, $d_{\text{ITB},i}$ is also greater than or equal to the bound given by (2) when $\frac{\mathcal{G}}{2}$ is even.

■

By combining the bound of (2) and Proposition 3, it is easy to see that $d_{\text{ITB-min}}$ is also greater than or equal to the girth-based bounds of (1) and (2).

Rather than attempting to maximize girth, the ITB LDPC code construction attempts to maximize $d_{\text{ITB},i}$ for all $v_i \in V$.

Algorithm 4 (Regular ITB LDPC Code Construction)**.** *Begin with a randomly constructed LDPC code that has the desired number of variable nodes N, check nodes M, d_F, and d_V.*
while at least one edge has been switched.
  for all pairs of edges $e_i, e_j \in E$ such that their connected variable nodes are not adjacent to any common check nodes.
    $-$*Exchange only the check nodes connections of e_i and e_j.*
    $-$*Using the construction method of Algorithm 1, build N independent trees and find $d_{\text{ITB,i}}$ for all $i = 1, \ldots, N$.*
    if $d_{\text{ITB-min}}$ has decreased.
      $-$*Keep the switch.*
    else if $d_{\text{ITB-min}}$ is the same and the number of i with $d_{\text{ITB,i}} = d_{\text{ITB-min}}$ has decreased.
      $-$*Keep the switch.*
    else if $d_{\text{ITB-min}}$ is the same, the number of i with $d_{\text{ITB,i}} = d_{\text{ITB-min}}$ is the same, and the mean of $d_{\text{ITB,i}}$ over all i has increased.
      $-$*Keep the switch.*
    else
      $-$*Undo the switch.*
    end
  end
end

TABLE I
GIRTH AND INDEPENDENT TREE-BASED LOWER BOUND PARAMETERS FOR $(3,6)$-REGULAR LDPC CODES.

(N,K)	Type	$\mathcal{G}$	Mean $\mathcal{G}_i$	$d_{\mathcal{G}\text{-min}}$	$d_{\text{ITB-min}}$	Mean $d_{\text{ITB},i}$
$(504,252)$	Random	4	5.84	2	2	5.27
	PEG	8	8.00	6	6	7.50
	ACE-PEG	8	8.04	6	6	8.08
	ITB	8	8.01	6	7	8.07
$(1008,504)$	Random	4	6.77	2	3	6.69
	PEG	8	9.34	6	7	9.58
	ACE-PEG	8	9.56	6	8	9.76
	ITB	10	10.00	10	10	10.00
$(1512,756)$	Random	4	7.09	2	4	7.25
	PEG	8	9.92	6	7	9.97
	ACE-PEG	8	9.96	6	8	10.15
	ITB	10	10.00	10	10	10.36

It should be noted that the ITB LDPC code construction has a much higher computational complexity than PEG. While PEG fixes an edge after determining the minimum cycle length that it induces on the Tanner graph, the ITB LDPC construction evaluates the global effect of each possible edge switch using the more precise criterion of $d_{\text{ITB},i}$. In practice, the complexity of ITB LDPC code construction can be mitigated by parallelizing the construction of the N independent trees.

IV. COMPARISON OF ITB AND PEG LDPC CODES

The ITB LDPC codes are now compared to random, PEG, and ACE-constrained PEG (ACE-PEG) codes in terms of code parameters and simulated performance. Table I shows girth and independent tree-based minimum distance parameters for each variable node in random, PEG, ACE-PEG, and ITB $(3,6)$-regular LDPC codes with block length $N = 504, 1008$ and 1512. Note that $\mathcal{G}_i$ refers to the minimum length of all cycles that involve v_i, and "Mean $\mathcal{G}_i$" is the average minimum cycles length over all $\{v_1, \ldots, v_N\}$. For length $N = 504$, the ITB code achieves roughly the same girth properties as the PEG code and the ACE-PEG code, while for lengths $N = 1008$ and $N = 1512$, the girth of the ITB codes is greater than that of random, PEG, and ACE-PEG codes. Although ITB LDPC code construction does not explicitly attempt to maximize girth, the process of minimizing the ITB lower bounds results in codes with excellent girth.

With respect to the independent tree-based lower bound, the ITB codes have the largest lower bound for all three block lengths. This is as expected, since the ITB construction makes explicit use of the ITB bound. The ACE-PEG codes also show noticeable improvement over PEG codes in their ITB properties. This is likely due to the fact that the ACE-PEG construction attempts to minimize the multiplicity of short cycles, resulting in independent trees with more variable nodes and larger $d_{\text{ITB}-\min}$ and mean $d_{\text{ITB},i}$.

Simulation results given in Figure 2 show the bit error rate performance using belief propagation (BP) decoding for a maximum of 80 iterations. The ITB codes perform better than all other codes as the signal-to-noise ratio (SNR) increases, and the bit error rates drop below $P_b = 10^{-5}$. This is not surprising because girth and minimum distance have the most significant impact on performance at moderate to high SNRs.

The simulation results for each block length are consistent with the ITB properties given in Table I, as higher ITB lower bounds result in lower bit error rates. The length $N = 1008$ ITB LDPC code improves the most upon the PEG and ACE-PEG codes in terms of girth properties, ITB properties, and bit error rate. Such large improvements are likely due to limitations in sequential construction methods like PEG and ACE-PEG for constructing codes with these particular parameters. These same limitations do not apply to the global approach taken by ITB LDPC construction, where the effect of all edge connections is computed over the complete Tanner graph.

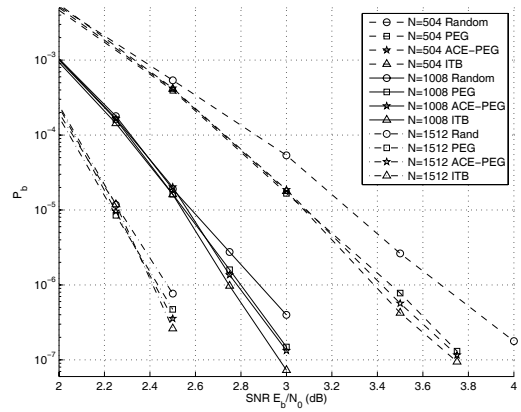


Fig. 2. Probability of bit error for random, PEG, ACE-PEG, and ITB $(3,6)$-regular LDPC codes with block lengths of $N = 504, 1008$, and 1512 with BP decoding for a maximum of 80 iterations.

V. CONCLUSION

A new independent tree-based method for lower bounding the minimum distance of regular and irregular LDPC codes was presented. The ITB lower bound was shown to be greater than the girth-based lower bound on the minimum distance of variable node degree regular LDPC codes. The ITB lower bound was then used to construct regular LDPC codes with larger girth, better minimum distance properties, and lower bit error rates than regular PEG and ACE-constrained PEG codes.

REFERENCES

[1] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity check codes," *IEE Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.

[2] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. 2001 Global Telecommunications Conference*, vol. 2, pp. 995–1001, Nov. 2001.

[3] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[4] H. Xiao and A. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 8, pp. 715–717, Dec. 2004.

[5] D. Vukobratovic and V. Senk, "Generalized ACE constrained progressive edge-growth LDPC code design," *IEEE Commun. Lett.*, vol. 12, pp. 32–34, Jan. 2008.

[6] E. Psota and L. C. Pérez, "Extrinsic tree decoding," in *Proc. 2009 Conference on Information Sciences and Systems*, Mar. 2009.

[7] E. Psota and L. C. Pérez, "LDPC decoding and code design on extrinsic trees," in *Proc. 2009 Int'l Symposium on Information Theory*, June 2009.

[8] N. Wiberg, "Codes and decoding on general graphs," Ph.D. thesis, Linköping University, Linköping, Sweden, 1996.

[9] H. Chen and Z. Cao, "A modified PEG algorithm for construction of LDPC codes with strictly concentrated check-node degree distributions," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2007)*, pp. 564–568, Mar. 2007.