2003 4th IEEE Workshop on Signal Processing
Advances in Wireless Communications

LARGE-GIRTH LDPC CODES BASED ON GRAPHICAL MODELS

Haotian Zhang and José M. F. Moura

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213
{haotian, moura}@ece.cmu.edu

ABSTRACT

In this paper we construct two classes of LDPC codes with girth 16 and 20, respectively, based on graphical models. These codes are well-structured, regular, and with column weight $j = 2$, which greatly simplifies their implementation. The codes with girth $g = 16$ have code rate $r = 1/2$. And the codes with girth $g = 20$ have code rate $r = 1/3$. Simulation results compare their bit-error-rate (BER) decoding performance in AWGN channels with randomly constructed LDPC codes.

1. INTRODUCTION

LDPC codes, i.e., Gallager codes [1], can perform very close to Shannon capacity limit in additive white Gaussian noise (AWGN) channels [2] when iteratively decoded by the sum-product algorithm [3].

We address in this paper the design of structured regular LDPC codes with large girths. Their structure and regularity simplify their implementation.

Cycles in the Tanner graphs of LDPC codes prevent the sum-product algorithm from converging [4, 5]. Further, cycles, especially short cycles, degrade the performance of LDPC decoders, because they affect the independence of the extrinsic information exchanged in the iterative decoding. Hence, LDPC codes with large girth are desired. The girth of a Tanner graph is the length of the shortest cycle in the graph.

Gallager proved in his original work [1] that LDPC codes with column weight $j \geq 3$ have a minimum distance that grows linearly with block length n for given j and row weight k, and that the minimum distance of an LDPC code with $j = 2$ can only grow logarithmically with n. Therefore, most of the previous research on the design of LDPC codes has focused on the LDPC codes with column weight $j \geq 3$. However, Song, Liu, and Kumar [6] pointed out recently that LDPC codes with column weight $j = 2$ have less computational complexity and better block error statistics properties than those with $j \geq 3$. Our work in this paper focus on the design of LDPC codes with $j = 2$ and large girth g.

Research supported by the Data Storage Systems Center (DSSC) at Carnegie Mellon University, and by NSF grant # ECS-0225449

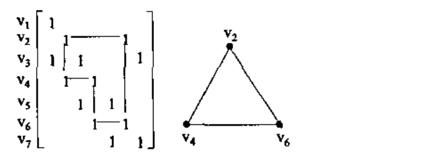

Fig. 1. A 6-cycle in an H matrix and its structure graph.

Let H be the parity check matrix of an LDPC code with v parity check equations, i.e., H is $v \times n$. We represent these parity check equations by a set X of v points, which is called the *point set* of the H matrix. For LDPC codes with column weight $j = 2$, each column of the H matrix is represented by an edge between two points in the set X that correspond to the two nonzero elements in this column. We call the resulting graph the *structure graph* for the LDPC code associated with the H matrix. Figure 1 shows a 6-cycle in the H matrix of an LDPC code with $j = 2$ and its structure graph.

The structure graph helps to identify easily cycles in the H matrices. Two distinct edges between two nodes in a structure graph stand for a 4-cycle. A 6-cycle is just a triangle comprising three points and three edges between any two of them. In general, a K-cycle is a loop composed of $K/2$ points and $K/2$ tail-biting edges.

In this paper we construct two classes of LDPC codes with large girth. In section 2, we introduce the design of the codes with girth $g = 16$ and code rate $r = 1/2$. Then we construct the codes with girth $g = 20$ and code rate $r = 1/3$ in section 3. We provide simulation results in section 4. Finally, section 5 concludes the paper.

2. LDPC CODES WITH GIRTH $g = 16$

In this section, we consider the design of LDPC codes with girth $g = 16$. This means, we design codes whose Tanner graph has no cycles of length smaller than 16. Assume the number of parity check equations is $v = 8p$, where, in

0-7803-7858-X/03/$17.00 ©2003 IEEE

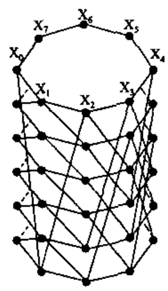

Fig. 2. The cylinder structure.

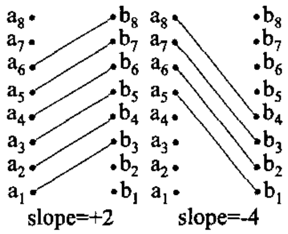

Fig. 3. Edges with specific slopes.

the case under study and as we will discuss below, $p \geq 23$ to guarantee the existence of successful constructions. The point set X has then v points. We divide these points into 8 subsets of equal size, namely $X_0, X_1, \cdots, X_7$, and the points in each of the subsets are aligned in a vertical line. These subsets comprise a loop; each point in subset X_i can only connect to points in the previous or next subset, i.e., $X_{mod(i-1,8)}$ or $X_{mod(i+1,8)}$, and cannot connect to points in X_i. The overall structure looks like a cylinder, and we call it a *cylinder structure*. Figure 2 gives an example of this cylinder structure.

Before describing the construction, we introduce needed definitions.

Definition 1 (section) *All the edges between two neighboring subsets X_i and $X_{mod(i+1,8)}$ in a cylinder structure compose a* section S_i.

Since there are only 8 subsets $X_0, X_1, \cdots, X_7$, there can be only 8 sections $S_0, S_1, \cdots, S_7$.

Definition 2 (slope) *The* slope *s of the edge connecting $a_i \in X_k$ and $b_j \in X_{mod(k+1,8)}$ is defined as $s = j - i$.*

Slopes take values in the range $-(p-1) \leq s \leq (p-1)$, and the number of edges with slope s is $(p - |s|)$. Figure 3 gives examples of edges with slope $+2$ and -4 when $p = 8$.

Definition 3 (admissible slope pair (ASP)) *Assume $v = 8p$, i.e., the size of each subset is p. A slope pair (s_i, s_j) is an* admissible slope pair *(ASP), iff*

$$s_i = -\text{sgn}(s_j) \cdot (p - |s_j|).$$

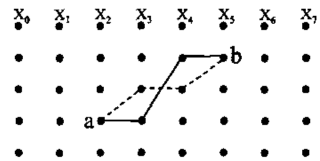

Fig. 4. The planar form of a cylinder structure.

In section S_i, all edges with the slopes in an ASP will increase the degree of each point in X_i and $X_{mod(i+1,8)}$ by 1. Slope 0 itself can be regarded as the special ASP (0,0).

Definition 4 (mirror slope) *In an ASP, the two slopes are* mirror slopes *of each other.*

We introduce two types of cycles in a cylinder structure. Type I cycles are defined as those that pass all the 8 sections $S_0, S_1, \cdots, S_7$ in the cylinder structure. For girth $g = 16$, these cycles are of no concern, since their minimal length is 16, and we only need to avoid cycles with length $l \leq 14$. Type II cycles are those that only pass through some of the 8 sections, but these sections must be consecutive. This type of cycles can have only length $4m$, where m is an arbitrary positive integer, since they are composed of an even number of edges. Therefore, to design an LDPC code with girth $g = 16$, we only need to avoid Type II cycles with length 4, 8 and 12.

Our method is as follows. For each section of the cylinder, we try to find as many ASPs as possible without introducing cycles with length 4, 8, and 12. All sections have the same number of ASPs. In fact, each section has only two ASPs, and we only introduce edges with slopes belonging to these ASPs. In this case, the degree of each point is 4, and the code rate is 1/2.

We develop a searching algorithm to find the two ASPs of each section. At first, we convert the cylinder structure to a planar form, as shown in Figure 4. Note that any points in subset X_i can only connect to the points in $X_{mod(i-1,8)}$ or $X_{mod(i+1,8)}$. Any Type II cycle can be regarded as two different paths joining a pair of points. Figure 4 gives an example of a cycle of length 12 composed by the two indicated paths between points a and b.

To design an LDPC code with girth $g = 16$, we find two ASPs for each of the sections $S_0, S_1, \cdots, S_7$. One of these two ASPs can be chosen to be the special ASP (0,0), whose two slopes are simply slope 0. This corresponds to connecting all the points in each row of Figure 4 by horizontal edges—note that the first and last point of each row are also connected. This ASP can only introduce cycles of length 16. We consider now how to introduce the second ASP for each section. We do so, one section at a time. We can start with any of the sections, say section S_0. Except

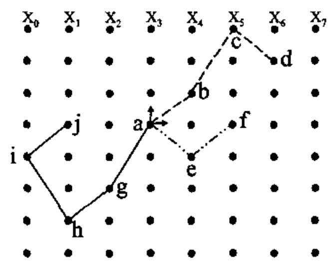

Fig. 5. The paths and the coordinate system.

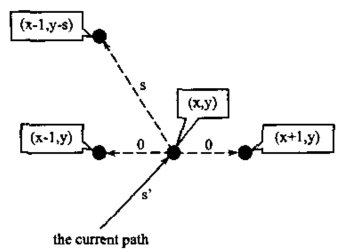

Fig. 6. The derivation of new paths.

for ASP (0,0), the second ASP for the first section S_0 can be chosen arbitrarily. Because we assume $p \geq 23$, and the other sections contain up to now only ASP (0,0), this second ASP will only introduce cycles of length $l \geq 16$.

We consider now introducing the second ASP for the remaining sections $S_1, \cdots, S_7$. Assume we have already done so for several sections. We consider now section S_i.

Pick an arbitrary point a in X_i, call it the reference point, and make it the origin of a two-dimensional coordinate system we now introduce, see figure 5. From the reference point a, we can draw many paths, for example, paths $abcd$, or aef, or $aghij$. If two paths meet at a point, they form a cycle whose length is twice the number of edges in the cycle.

Any path starting from a can be represented by the sequence of coordinates of the points in the path. For example, the coordinates of points b, c, and d are $(1,1)$, $(1,-1)$, and $(-1,-2)$, respectively.

Assume a path reaches point (x,y) at the current step, and 4 edges converge to the point (x,y); each edge is labelled by its slope, as shown in Figure 6. Except for the edge that the current path comes from, i.e., the solid line in Figure 6 with slope s, each of the additional edges (dashed lines) is a candidate to be added to the current path to form a new path at the next step, i.e., the current path can be expanded to three new paths ending at points $(x+1,y)$, $(x-1,y)$ and $(x-1,y-s)$, respectively.

When we use the search algorithm to look for the second ASP for S_i, we first list all possible ASPs and test each of them, till we find one that is acceptable. For each candidate ASP, we list all the possible paths starting from the reference point in subset X_i. For a given ASP, if there are no two paths that meet each other in the first m steps, i.e., no cycles with length $l \leq 4m$ are introduced, this ASP is an appropriate choice. Otherwise, we discard it and check another ASP, until an ASP is found for S_i. We then move on to the next section S_{i+1}, till we reach S_7.

To minimize the search space for the algorithm, we provide as many constraints as possible to the set of possible ASPs in each section. We use A_i to represent the set of selected ASPs for section S_i, and we call it the *ASP set*. Each ASP set contains (0,0) and another ASP. It can be shown that the constraints on ASPs in A_i are as follows.

(1) If $(s,s') \in A_i$, then $(s,s'), (-s,(-s)') \notin \widetilde{A}$, where $\widetilde{A} = A_{mod(i-2,8)} \cup A_{mod(i-1,8)} \cup A_{mod(i+1,8)} \cup A_{mod(i+2,8)}$.

(2) Assume $v = 8p$. To check if the ASP (s,s') is a possible choice to include in A_i, we need to calculate the least common multiple of $|s|$ and $|s'|$, i.e., $\text{lcm}(|s|,|s'|)$. Then the minimal length L of the cycles resulting from ASP (s,s') and (0,0) in A_i is

$$L = 4\left[\text{lcm}(|s|,|s'|)/|s| + \text{lcm}(|s|,|s'|)/|s'|\right].$$

The ASP (s,s') is a possible choice only when $L > 14$.

There is also a constraint on the paths. For a given path, if any two points in this path are the same, then this path should be discarded.

We consider the initializations of the search algorithm. Initially, each of the ASP sets $A_0, \cdots, A_7$ includes only ASP (0,0). Based on the constraints for ASP selection discussed previously, we establish the set B_0 of all possible ASP choices for A_0, and call it the *candidate ASP set* for A_0. The searching algorithm is described in Construction 1.

Before moving on to the construction, care must be taken regarding the value of p that must be large enough to guarantee that there exists an appropriate choice of the second ASP for each section. We use the search algorithm to look for the minimal valid value of p. For girth $g = 16$, we find that p must be no lower than 23. Since the number of parity check equation $v = 8p$ and the block length $n = 2v = 16p$, we have $v \geq 184$ and $n \geq 368$. We can select a p value according to the desired block length.

Construction 1 (LDPC codes with girth 16)

1. Select an arbitrary ASP from B_0, and add it into A_0. Set $i = 1$.

2. Based on the constraints for ASP selection, establish the candidate ASP set B_i for A_i.

3. Select an arbitrary ASP (s,s') from B_i, and add it temporarily to A_i. If B_i is an empty set, the construc-

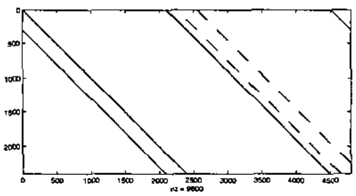

Fig. 7. H matrix for LDPC codes with $g = 16$, $n = 4800$ and $r = 1/2$.

tion is terminated, and an error message is generated. Otherwise, go to 4.

4. Starting from the reference point of subset X_i, derive all the possible paths in the first 3 steps. At each step, derive new paths using the slopes in the relevant ASP sets, and discard invalid paths based on the constraint on paths. If there are no two paths that meet each other in the first 3 steps, (s, s') is an appropriate ASP for A_i, and go to 5. Otherwise, (s, s') introduces cycles of length $l < 16$, so we need to delete (s, s') from A_i and B_i, and go to 3.

5. Let $i = i + 1$. If $i > 7$, go to 6. Otherwise, to 2.

6. End. The sets $A_0, \cdots, A_7$ are the desired ASP sets. The corresponding LDPC code has girth 16.

Achievable code rate: Since there are only two ASPs in each ASP set, the degree of each point in the structure graph is 4, i.e., the row weight k of the corresponding H matrix is 4. Since the column weight is $j = 2$, the code rate is $r = (k - j)/k = 1/2$.

Given the number of parity check equations v, we can generate many different LDPC codes, all with code rate 1/2 and girth 16, by choosing different initial ASPs and making different choices in Step 1 and 3 of Construction 1. When $v = 184$, i.e., $p = 23$, one solution for the 8 ASP sets is $\{(0,0),(-11,12)\}, \{(0,0),(-10,13)\}, \{(0,0),(-9,14)\}, \{(0,0),(-11,12)\}, \{(0,0),(-10,13)\}, \{(0,0),(-9,14)\}, \{(0,0),(-8,15)\}, \{(0,0),(-5,18)\}$, respectively. The corresponding H matrix is well structured and completely determined by p and ASP sets $A_0, \cdots, A_7$. Figure 7 gives one of the many possible H matrices when $v = 2400$, $n = 4800$, and $r = 1/2$.

3. LDPC CODES WITH GIRTH $g = 20$

With small modifications on Construction 1, we can obtain LDPC codes with girth 20 and code rate 1/3.

Assume $v = 10p$, where $p \geq 13$ to guarantee the existence of successful constructions, and divide these v points into 10 subsets of equal size. Although the girth is now $g = 20$, larger than the girth $g = 16$ in the previous section, the value of minimum p is smaller because the code rate is lower. We can establish a cylinder structure with 10 sections. Our task is to find 10 ASP sets, i.e., $A_0, \cdots, A_9$. The ASP sets with even indices include two ASPs, and the ASP sets with odd indices include only one ASP. The first ASP in each ASP set can be chosen to be the special ASP (0,0). Therefore, we only need to find the second ASP for those ASP sets with even index. The constraints on ASP selection and paths are the same as described in the previous section. The search algorithm is described in Construction 2.

Construction 2 (LDPC codes with girth 20)

1. Establish a candidate ASP set B_0 for A_0. Select an arbitrary ASP from B_0, and add it to A_0. Set $i = 2$.

2. Based on the constraints on ASP selection, establish the candidate ASP set B_i for A_i.

3. Select an arbitrary ASP (s, s') from B_i, and add it temporarily to A_i. If B_i is an empty set, the construction is terminated, and an error massage is generated. Otherwise, go to 4.

4. Starting from the reference point of subset X_i, derive all possible paths in the first 4 steps. At each step, derive new paths using the slopes in the relevant ASP sets, and discard invalid paths based on the constraint on paths. If there are no two paths that meet each other in the first 4 steps, (s, s') is an appropriate ASP for A_i, and go to 5. Otherwise, (s, s') introduces cycles of length $l < 20$, so we need to delete (s, s') from A_i and B_i, and go to 3.

5. Let $i = i + 2$. If $i > 9$, go to 6. Otherwise, to 2.

6. End. The sets $A_0, \cdots, A_9$ are the desired ASP sets. The corresponding LDPC code has girth 20.

Achievable code rate: Since there are two ASPs in each ASP set with even index and only one ASP in each ASP set with odd index, the degree of each point in the structure graph is 3, i.e., the row weight k of the corresponding H matrix is 3. Since the column weight is $j = 2$, the code rate is $r = (k - j)/k = 1/3$.

To make sure that we can find two ASPs for each ASP set with even index, v must be large enough. We find that the minimal value of v is 130, i.e., $p = 13$. In this case, the 10 ASP sets are $\{(0,0),(-6,7)\}, \{(0,0)\}, \{(0,0),(-5,8)\}, \{(0,0)\}, \{(0,0),(-6,7)\}, \{(0,0)\}, \{(0,0),(-5,8)\}, \{(0,0)\}, \{(0,0),(-2,11)\}, \{(0,0)\}$, respectively. The corresponding H matrix is well structured and completely determined by p and the ASP sets $A_0, \cdots, A_9$. Figure 8 gives one of the many possible H matrices when $v = 3970$, $n = 5955$, and $r = 1/3$.

4. SIMULATION RESULTS

We studied by simulations the bit-error-rate (BER) decoding performance of the two classes of LDPC codes in AWGN channels, adopting the sum-product decoding algorithm and the rate-adjusted signal to noise ratio used in [7].

Figure 9 compares the BER performance of an LDPC code with girth $g = 16$ and a randomly constructed LDPC

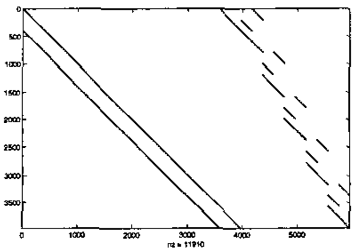

Fig. 8. H matrix for LDPC codes with $g = 20$, $n = 5955$ and $r = 1/3$.

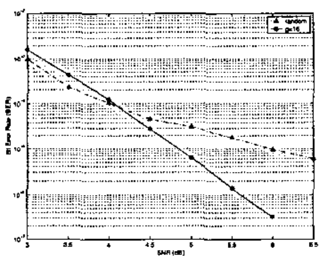

Fig. 9. BER performance comparison between LDPC codes with girth 16 and randomly constructed LDPC codes.

code over AWGN channels. Both of them have column weight $j = 2$, block length $n = 4368$ and code rate $r = 1/2$. The code with girth 16 has worse performance in the low SNR region. However, in the high SNR region, it outperforms the randomly generated code by 0.8 dB at BER=10^{-5}.

We have similar results for LDPC codes with girth 20. Figure 10 compares the BER performance of an LDPC code with girth $g = 20$ and a randomly constructed LDPC code over a AWGN channel. Both of them have column weight $j = 2$, block length $n = 4395$, and code rate $r = 1/3$. In the high SNR region, the code with girth 20 outperforms the randomly generated code by 1.1 dB at BER=10^{-5}.

For an LDPC code with column weight 2 and girth g,

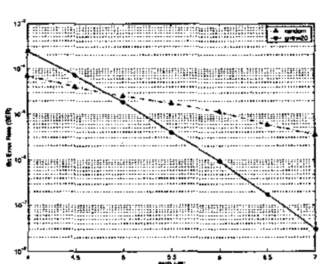

Fig. 10. BER performance comparison between LDPC codes with girth 20 and randomly constructed LDPC codes.

the minimum distance $d_{min} = g/2$. Therefore, the codes with girth 16 and 20 have $d_{min} = 8$ and 10, respectively, much larger than that of a randomly constructed code whose $d_{min} = 2$ due to the presence of 4-cycles. In the high SNR region, d_{min} is the dominant factor for BER performance; consequently, the codes with larger girth give better performance. At this time, we are still investigating the behavior at low SNR. One possible explanation is the cycle distribution. A preliminary analysis shows that the codes with large girth have more cycles of length equal to girth g than randomly constructed codes, which may overwhelm at low SNR the benefit our structured code derives from its large girth.

5. CONCLUSION

We construct two classes of regular LDPC codes with column weight 2 and girth $g = 16$ and $g = 20$, respectively. These codes are well-structured; their parity check matrices can be effectively represented by a set of integers. Our codes outperform randomly constructed codes in the high SNR region in AWGN channels.

6. REFERENCES

[1] R. G. Gallager, *Low-Density Parity Check Codes*, MIT Press, Cambridge, MA, 1963.

[2] D. J. C. Mackay and R. M. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding, 5th IMA Conference, in Lecture Notes in Computer Science*, C. Boyd, Ed., 1995, vol. 1025, pp. 110–111.

[3] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[4] R. J. McEliece, D. J. C. Mackay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Selected Areas Common.*, vol. 16, no. 2, pp. 140–152, Feb 1998.

[5] T. Etzion, A. Trachtenberg, and A. Vardy, "Which codes have cycle-free Tanner graphs?," *IEEE Tans. Inform. Theory*, vol. 45, no. 6, pp. 2173–2181, Sep 1999.

[6] H. Song, J. Liu, and B. V. K. Vijaya Kumar, "Low complexity LDPC codes for magnetic recording," in *IEEE Globecom 2002*, Taipei, Taiwan, R.O.C., Nov 2002.

[7] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, March 1999.