Hybrid Construction of Long LDPC Codes with Very Low Density

Lijun Zhang, Yanjing Zhang
School of Electr. and Inform. Eng.
Beijing Jiaotong Univ.
Beijing, 100044, China
Email: ljzhang@bjtu.edu.cn

L. L. Cheng
Dept. of Electronic Engineering
City Univ. of Hong Kong
Hong Kong, China
Email: itacheng@cityu.edu.hk

***Abstract*—By combining an algebraic method and a random method, a hybrid method is proposed to construct LDPC codes, which can easily ensure the girth is at least six. The complexity of construction for the hybrid code is only a fraction of that for PEG code, which facilitates the construction of long LDPC codes with very low density. Simulation results show that the hybrid code from EG-LDPC code and PEG code has the identical error performance and convergence rate to the PEG code with the same length.**

***Index Terms*—LDPC codes; hybrid construction; EG-LDPC codes; PEG algorithm**

I. Introduction

Low-density parity-check (LDPC) codes are a class of linear block codes with implementable decoders, which provide near-capacity performance on a large set of data-transmission and data-storage channels. LDPC codes were invented by Gallager in his 1960s doctoral dissertation [1] and were mostly ignored during the 35 years that followed. One notable exception is the important work of Tanner in 1981 [2], in which Tanner generalized LDPC codes and introduced a graphical representation of LDPC codes, now called a Tanner graph. The study of LDPC codes was resurrected in the mid 1990s with the work of MacKay, Luby, and others [3–6], who noticed, apparently independently of Gallager's work, the advantages of linear block codes with sparse (low-density) parity-check matrices. Recently, LDPC codes have attracted much attention for their capacity approaching performance, linear decoding complexity, flexibility, and relatively low error floor, among others. The construction for LDPC codes is one of hot spots in LDPC research area.

An LDPC code is a linear block code given by the null space of an $m \times n$ parity-check matrix $\mathbf{H}$ that has a low density of 1's. A regular LDPC code is a linear block code whose parity-check matrix $\mathbf{H}$ has column weight γ and row weight ρ, where $\rho = \gamma n/m$ and $\gamma \ll m$. Many construction methods have been proposed to construct LDPC codes with girth at least six. The structured construction of LDPC codes based on finite geometry is typical. Euclidean geometry (EG) LDPC codes have good performance in AWGN channel [7] and can be decoded by some simplified decoding algorithms, such as one-step majority logic (OSMLG) decoding [8]. But these codes usually have large column weight and relatively high density. For the computer-based construction, the progress edge growth (PEG) algorithm is currently accepted as a vogue algorithm for the random construction of LDPC codes [9]. It resorts to the way of adding edges in turn in the Tanner graph for the codes with a given degree distribution. PEG algorithm makes the counts of the short cycles in the Tanner graph of the codes as few as possible, and the girth as large as possible. However, with the growth of the code length, the computer searching becomes an involved job.

In this paper, a hybrid construction of LDPC code by combining an algebraic method and a random method, is proposed. The hybrid algorithm can easily obtain long LDPC codes with very low density and girth at least six, while the complexity is only the fraction of that for PEG algorithm.

II. The Hybrid Construction

A. Algorithm description

The check matrix $\mathbf{H}$ for the hybrid code is obtained as follows.

1) construct an $m \times n$ base matrix $\mathbf{H_1}$ by algebraic method, which has column weight γ_1, row weight ρ_1, and density τ_1;
2) construct a $\gamma_1 \times n_1$ matrix $\mathbf{H_2}$ by random method, which has column weight γ_2, row weight ρ_2, and density τ_2;
3) for each column of $\mathbf{H_1}$, replace the ith 1-component by the ith row of $\mathbf{H_2}$, $1 \leq i \leq \gamma_1$, and replace the each 0-component remained by an all-zero n_1-tuple, then an $m \times nn_1$ matrix $\mathbf{H}$ is obtained, whose null space defines a hybrid code.

Obvously, the code length of the code obtained by the algorithm above is nn_1, and the density of the check matrix $\mathbf{H}$ is $\tau_1\tau_2$, which can be easily proved.

B. RC-constraint and girth analysis

An LDPC code can be described by a Tanner graph [2]. The length of the shortest cycle in a Tanner graph is referred to as its girth. In almost all of the proposed constructions of LDPC codes, the following constraint on the rows and columns of the parity-check matrix $\mathbf{H}$ is imposed. No two rows (or two columns) can have more than one position where they both have 1-components. This constraint on the rows and the

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

columns of $\mathbf{H}$ is called row-column (RC)-constraint. The RC-constraint on $\mathbf{H}$ ensures that the Tanner graph of the LDPC code given by the null space of $\mathbf{H}$ is free of cycles of length 4 and hence has a girth of at least 6 [7, 8].

Proposition Suppose that the girths of the base matrix $\mathbf{H_1}$ and the matrix $\mathbf{H_2}$ are g_1 and g_2, respectively. Thus, the girth of the check matrix $\mathbf{H}$ is

$$g \geq \min\{g_1, g_2\}.$$

Proof: From the view of the columns, each column of the base matrix $\mathbf{H_1}$ is essentially replaced by the small matrix $\mathbf{H_2}$ with some all-zero rows inserted, which cannot import any cycles shorter than g_2, and there do exist cycles of length g_2. From the view of the rows, the 0's in the base matrix $\mathbf{H_1}$ is superseded by all-zero rows, and 1's by rows of $\mathbf{H_2}$, which cannot introduce any cycles shorter than g_1, and there may exist cycles of length g_1 or even larger. If $g_2 \leq g_1$, then $g = g_2$; else if $g_2 > g_1$, then $g \geq g_1$. The two cases can be summarized as $g \geq \min\{g_1, g_2\}$. ■

III. A Typical Combination

From the algorithm description, it can be seen that the column weight γ_1 of the base matrix $\mathbf{H_1}$ should be large enough to accommodate the matrix $\mathbf{H_2}$. As mentioned above, the EG-LDPC codes have relatively large column weight, and can be used to get the base matrix. For matrix $\mathbf{H_2}$, the PEG algorithm is chosen for its graceful girth characteristic. In order to obtain a hybrid code with length as short as possible, let $n_1 = \gamma_1$, and a square matix $\mathbf{H_2}$ will be obtained.

A. EG-LDPC codes

Consider the *k*-dimensional Euclidean geometry EG(*k*,*q*) over the Galois field GF(q). This geometry consists of q^k points and

$$J \triangleq J_{EG}(k,1) = q^{k-1}(q^k-1)/(q-1) \tag{1}$$

lines [7]. A point in EG(*k*,*q*) is simply an *k*-tuple over GF(q). A line in EG(*k*,*q*) is simply a one-dimensional subspace or its coset of the vector space $\mathbf{V}$ of all the *k*-tuple over GF(q). The Galois field GF(q^k) as an extension field of GF(q) is a realization of EG(*k*,*q*). Let α be a primitive element of GF(q^k), then the powers of α

$$\alpha^{-\infty} \triangleq 0, \alpha^0 = 1, \alpha, \alpha^2, \cdots, \alpha^{q-2},$$

represent the q^k points of EG(*k*,*q*), where $\alpha^{-\infty}$ represents the origin point of EG(*k*,*q*).

Let EG*(*k*,*q*) be a *subgeometry* of EG(*k*,*q*) obtained by removing the origin $\alpha^{-\infty}$ and all the lines passing through the origin from EG(*k*,*q*). This subgeometry consists of q^k-1 non-origin points and

$$J_0 \triangleq J_{0,EG}(k,1) = (q^{k-1}-1)(q^k-1)/(q-1) \tag{2}$$

lines not passing through the origin [7]. Let $\mathcal{L}$ be a line in EG*(*k*,*q*) consisting of the points $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_q}$, i.e.,

$$\mathcal{L} = \alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_q}.$$

For $0 \leq t < q^k - 1$,

$$\alpha^t \mathcal{L} = \alpha^{j_1+t}, \alpha^{j_2+t}, \cdots, \alpha^{j_q+t} \tag{3}$$

is also a line in EG*(*k*,*q*), where the powers of α in $\alpha^t\mathcal{L}$ are taken modulo q^k-1. The q^k-1 lines $\mathcal{L}, \alpha\mathcal{L}, \alpha^2\mathcal{L}, \cdots, \alpha^{q^k-2}\mathcal{L}$ are all different. Since $\alpha^{q^k-1} = 1, \alpha^{q^k-1}\mathcal{L} = \mathcal{L}$. These q^k-1 lines form a *cyclic class*. For any line $\mathcal{L}$ in EG*(*k*,*q*) not passing through the origin, we define the following (q^k-1)-tuple over GF(2),

$$v_{\mathcal{L}} = (v_0, v_1, \cdots, v_{q^k-2}),$$

whose components correspond to the q^k-1 non-origin points, $\alpha^0, \alpha, \alpha^2, \cdots, \alpha^{q^k-2}$, in EG*(*k*,*q*), where $v_i = 1$ if α^i is a point on $\mathcal{L}$, otherwise $v_i = 0$. We form a $(q^k-1) \times (q^k-1)$ matrix $\mathbf{H}_1$ over GF(2) with the incidence vectors $\mathbf{v}_{\mathcal{L}}, \mathbf{v}_{\alpha\mathcal{L}}, \cdots, \mathbf{v}_{\alpha^{q^k-2}\mathcal{L}}$ of the lines $\mathcal{L}, \alpha\mathcal{L}, \alpha^2\mathcal{L}, \cdots, \alpha^{q^k-2}\mathcal{L}$ in a cyclic class as rows arranged in cyclic order. Then $\mathbf{H}_1$ is a $(q^k-1) \times (q^k-1)$ circulant over GF(2) with both column weight and row weight equal to q. $\mathbf{H}_1$ satisfies the RC-constraint and the girth $g_1 = 6$ [7].

B. The PEG algorithm

The PEG algorithm is initialized by the number of variable nodes, *n*, the number of check nodes, *m*, and a variable node-degree sequence D_v, which is the list of degrees for each of the *n* variable nodes. Given these parameters, the algorithm adds edges one by one, and each edge is included in a manner that maximizes the local girth. Thus, the PEG algorithm is a greedy algorithm for creating a Tanner graph with a large girth.

The low-degree variable nodes are the most susceptible to error, because they receive the least amount of neighborly help. Edge placement begins with the lowest-degree variable nodes and progresses to variable nodes of increasing (or non-decreasing) degree. The algorithm does not move to the next variable node until all of the edges of the current variable node have been attached. The first edge attached to a variable node is connected to a lowest-degree check node under the current state of the graph. Subsequent attachments of edges to the variable node are done in such a way that the local girth for that variable node is maximum. Thus, if the current state of the graph is such that one or more check nodes cannot be reached from the current variable node by traversing the edges connected so far, then the edge should be connected to an unreachable check node so that no cycle is created. Otherwise, if all check nodes are reachable from the current variable node along some number of edges, the new edge should be connected to a check node of lowest degree that results in the largest girth seen by the current variable node. This lowest-degree check node strategy will yield a fairly uniform check node degree distribution.

Some hybrid LDPC codes got from EG-LDPC codes and PEG codes with various lengths and rates are listed in Table I. We are impressed by the third code, (65472, 64449) hybrid code. This code is 65472 bits long, and the construction complexity is just the same as that of the PEG code with length 64 if the construction complexity of the EG-LDPC

TABLE I
CODES CONSTRUCTED BY THE HYBRID METHOD.

$\mathbf{H_1}$ (EG-LDPC)	$\mathbf{H_2}$ (PEG)	Hybrid codes	Column weight	Rate	Density
63×63	8×8	(504, 411)	2	0.875	3.17%
255×255	16×16	(4080, 3825)	3	0.94	1.17%
1023×1023	64×64	(65472, 64449)	3	0.98	0.29%

code is neglected. The direct construction of the code with PEG algorithm is unbelievable. The density of the code is up to the amazing extent of one in a thousand.

IV. SIMULATION RESULTS

The second code in Table I is adopted to show the validation of the hybrid algorithm. Based on $\mathrm{EG}(2, 16)$, a 255×255 base matrix $\mathbf{H}_1$ is built. Both the column weight and the row weight are 16. Then we construct a 16×16 matrix $\mathbf{H}_2$ with column weight 3 and column weight 3 by PEG algorithm. For each column of $\mathbf{H}_1$, replacing the ith 1 by the ith row of $\mathbf{H}_2$, a 255×4080 matrix $\mathbf{H}$ with column weight 3 and row weight 48 is obtained, whose null space defines a (4080, 3825) hybrid code. The density of the code is 1.17%. The standard sum-product algorithm (SPA) with maximum iteration number of 50 is used for decoding.

The bit error rate (BER) and frame error rate (FER) performances of the code over the AWGN channel with BPSK modulation are shown in Fig. 1. A MacKay code and a PEG code with the same code lengths, rates, column weights, and row weights are also included for comparison. It can be seen that the curves of the codes are on the top of each other. The hybrid code has the same or ever better error performance in comparison with the MacKay code and PEG code.

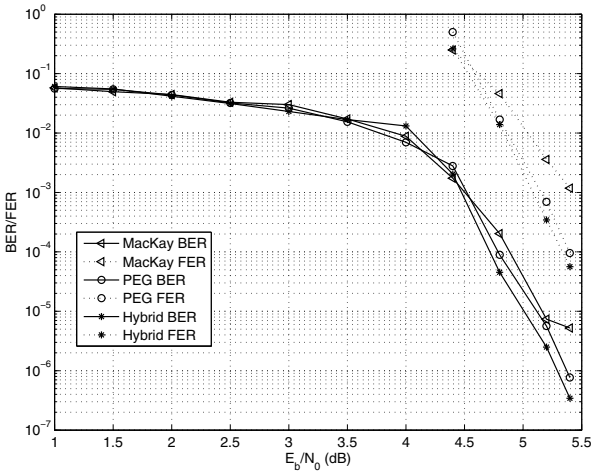

Fig. 1. Performance comparison among (4080, 3825) MacKay code, (4080, 3825) PEG code, and (4080, 3825) hybrid codes with same column weight and row weight over the AWGN channel.

In Fig.2 is shown the average numbers of iterations required for decoding the (4080, 3825) hybrid code and (4080, 3825) PEG code. It can be seen that the hybrid code converges as fast as the PEG code does.

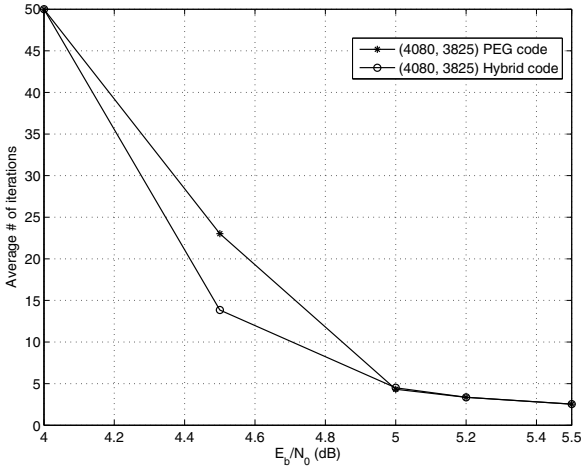

Fig. 2. Average numbers of iterations required for decoding the (4080, 3825) hybrid code and (4080, 3825) PEG code.

V. CONCLUSIONS

A hybrid method by combining an algebraic method and a random method is presented, which facilitates the construction of long LDPC codes with very low density, while keeping graceful performance. The complexity of construction for the hybrid code is only a fraction of that for PEG code. However, the hybrid construction requires large column weight of the base matrix, which restricts the flexibility of the method. Furthermore, the extension of the hybrid method to irregular LDPC codes is still under consideration.

ACKNOWLEDGMENT

This work is supported by the fundamental research funds for the central universities.

REFERENCES

[1] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-18, pp. 21–28, Jan. 1962.

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

[3] D. J. C. MacKay and R. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding. 5th IMA Conference, number 1025 in Lecture Notes in Computer Science*. Springer, 1995, pp. 100–111.

[4] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 399–431, Mar. 1999.

[5] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Trans. Inf. Theory*, vol. 42, no. 11, pp. 1732–1736, Nov. 1996.

[6] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 28, no. 4, Oct. 1998.

[7] Y. Kou, S. Lin, and M. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[8] S. Lin and J. D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.

[9] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular Progressive Edge-Growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.