Construction of Girth 8 LDPC Codes based on Multidimensional Finite Lattices

John Craddock, Mark Flanagan, Stephen J. Redmond, Anthony Fagan

Abstract This paper presents a novel method for constructing Low Density Parity Check (LDPC) codes with a girth of up to 8. These codes are based on the structural properties of Finite Lattices. Results are presented which show that these codes perform well over AWGN channels with iterative decoding.

I. Introduction

Low Density Parity Check (LDPC) codes were first introduced by Robert Gallager of MIT in 1960 [1]. They remained unused until they were rediscovered by MacKay in 1997 . LDPC codes are a class of linear block codes which have been shown to achieve near-capacity performance on the AWGN channel when decoded using a message passing algorithm. Algebraic and combinatorial constructions of LDPC codes have received much attention over the past decade since such constructions generally yield codes with lower implementation complexity than do pseudo-random constructions [2], [3].

LDPC codes are usually decoded using a message passing algorithm. The algorithm acts locally. Each variable and constraint node gathers information from and passes information to its neighbouring nodes. This process continues until it converges to a valid codeword or a maximum number of iterations is reached.

The message passing algorithm is optimum in the case of a Tanner Graph with a tree structure i.e. containing no cycles. One means of improving the performance of the message passing algorithm to decode LDPC codes is to increase the girth of the Tanner graph defined by the code. This increases the number of iterations necessary for information to be passed around a cycle in the code and makes the code more tree-like.

Several geometric methods of producing LDPC codes have been presented, see for example [3] and [2]. In [2] a method for generating an LDPC Code based on a 2 dimensional Finite Lattice is presented. [3] presents a method of generating a girth 6 LDPC code based on Euclidian Geometries and [4] extends modifies this work to produce girth 8 codes.

Section II of this paper describes the construction of two classes of LDPC codes based on a Multidimensional Finite Lattice. The first can be viewed as an extension of the work in [3] from two to m dimensions. A lower bound on the minimum distance of the code is derived as well as equations for the number of proper 8 cycles. This section also introduces a second, more general, class of LDPC codes based on Lattice Geometries. These codes can be viewed as a generalisation of the first codes introduced and also as an extension of the work in [2] and [4] from Euclidian to Lattice Geometries. Section III presents simulation results, and section IV concludes this work.

II. The Lattice Construction

In this section we introduce two methods of constructing LDPC codes of girth 8 based on a multidimensional integer lattice.

2.1 Multidimensional Finite Lattice

Take a subset of the integer lattice of dimension m defined by

$G_m = \{(v_1 \dots v_m) : 0 \leq v_1 \leq d_1 - 1, \dots, 0 \leq v_m \leq d_m - 1)\}$

and assume wlg that $d_1 \leq d_2 \dots d_m$

where $(d_1, \dots, d_m)$ is the depth of the lattice taken in each of the m dimensions, denoted $(D_1, \dots, D_m)$, and $(v_1 \dots v_m)$ is a point in the lattice. The total size of the lattice is given by $n = d_1 \times d_2 \times \cdots \times d_m$ points. All operations in the lattice are conducted mod d_j in each dimension D_j, $1 \leq j \leq m$.

Addition and scalar multiplication are defined: $(a_1, a_2, \dots, a_m) + (b_1, b_2, \dots \ b_m) = (a_1 + b_1, a_2 + b_2, \dots, a_m + b_m)$ and $\alpha(a_1, \dots, a_m) = (\alpha a_1, \dots, \alpha a_m)$.

Lines can be drawn in the lattice. A line of slope $S = (s_1, s_2, \dots, s_m)$ through a point v is the

collection of points v_j such that $v_j = v_1 + \alpha S$, for any integer α.

2.2 Multidimensional Lattice Codes

Define a sub lattice G_{m-1} by setting $v_1 = 0$;

$G_{m-1} = \{(v_1, \ldots, v_m) : v_1 = 0, 0 \leq v_2 \leq d_2 - 1, \ldots, 0 \leq v_m \leq d_m - 1)\}$

Through each of the n/d_1 points in G_{m-1} draw a line section of slope $S = (1, s_2, \ldots, s_m)$ given by $L_v(S) = \{x, v_2 + s_2 x, \ldots v_m + s_m x\}$ where $x = 0, \ldots, d_1 - 1$. This bundle of parallel line sections partitions G_m into n/d_1 sets containing d_1 points each.

An LDPC code can be created by associating each point in G_m with a variable node and each line section with a constraint node. A H matrix can be constructed by including a variable j in a constraint k iff the point associated with j is contained in the line associated with k. A bundle of parallel line sections of slope S_1 generates a matrix H_1 of column weight 1 and row weight d_1. Taking ρ such bundles can generate a H matrix of column weight ρ given by

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \ldots \\ H_\rho \end{bmatrix} \tag{1}$$

Definition 2.1

j slopes $S_1, S_2, \ldots, S_j$ are linearly independent iff there do not exist integers $\alpha_1, \alpha_2, \ldots, \alpha_j$, such that $\alpha_1 S_1 + \alpha_2 S_2 + \cdots + \alpha_j S_j = (0, 0, \ldots, 0)$, $(\alpha_1, \ldots, \alpha_j) \neq (0, \ldots, 0)$, $-d_j < \alpha_j < d_j$

Choosing the slopes $S_1, \ldots, S_\rho$ subject to certain constraints gives the code some desirable properties.

Constraint 2.1 The slopes chosen span the geometry.

Constraint 2.2 No group of three slopes are linearly dependant.

Constraint 2.3 The element of each slope in each dimension is relatively prime to the depth of the dimension, i.e s_j and d_j have no common factors for all $1 \leq j \leq m$.

Example 2.1

A 4 dimensional finite lattice of size $6 \times 6 \times 6 \times 6$. 4 bundles of parallel lines are drawn with slopes;

$$\begin{pmatrix} S_a \\ S_b \\ S_c \\ S_d \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

Each line contains 6 points. This example gives a regular code with 1296 variable nodes and 864 constraint nodes. As not all rows of H are linearly independent, there are 625 information bits per codeword giving the code a rate of 48.2 percent. The column and row weights are 4 and 6 respectively. It can easily be shown that any three of the bundles of parallel lines chosen are linearly independent so no loop of length 6 can exist, giving the code a girth of 8, see theorem 2.2. This code was used in compiling the results in Fig 3.

Theorem 2.1

LPDC codes of this construction have no cycles of length 4.

Proof

A 4 cycle in an LDPC code exists when 2 variable node share 2 or more constraint nodes in common. This results from the lattice geometry construction when 2 or more lines contain a pair of points in common.

Assume $v_a, v_b \in L_1, L_2$ where $L_{1,2} = v_{1,2} + xS_{1,2}, 0 \leq x \leq d_1 - 1$. Taking the line L_1;

$v_a = v_1 + \alpha S_1$

$v_b = v_1 + \beta S_1$

$v_a - v_b = (\alpha - \beta) S_1 = \gamma S_1$

similarly for L_2; $v_a - v_b = \lambda S_2$

combining gives $\gamma S_1 - \lambda S_2 = \vec{0}$ which violates the linear independence of constraint 2.2.

Theorem 2.2

If the slopes are chosen so that no group of three slopes is linearly dependent i.e. $\alpha S_j + \beta S_k + \gamma S_l \neq \vec{0}$ for all $1 \leq j, k, l \leq \rho$, $\alpha, \beta, \gamma \in \mathbf{Z}$ the resulting LPDC code has no cycles of length 6.

Proof

Similar to the previous proof, three nodes, v_a, v_b and v_c participating in a 6 cycle will give rise to equations

$v_a - v_b = \alpha S_1$

$v_b - v_c = \beta S_2$

$v_c - v_a = \gamma S_3$

Combining these equations gives $\alpha S_1 + \beta S_2 + \gamma S_3 = \vec{0}$ which violates the linear independence condition.

Theorem 2.3

No Pair of variable nodes is joined by a single path of length 4.

Proof

Consider a set of variable nodes v_1 and v_2 connected by a path of length 4. There must exist an intermediate variable node v_3 such that $v_1 + a.S_1 =$

v_3 and $v_3 + b.S_2 = v_2$ for some $a, b \in 0 \ldots d_1 - 1$. A second path of length four exists between these nodes given by $v_1 + b.S_2 = v_4$ and $v_4 + a.S_1 = v_2$. By constraint 2.2, $aS_1 \neq bS_2$. Thus there cannot be a single path of length 4 between two variable nodes.

Comment 2.1

As some variable nodes in a non-disjoint graph must be joined by a path of length four and such paths cannot occur singly this result proves that cycles of length 8 exist. Combined with theorems 2.1 and 2.2, theorem 2.3 proves that the girth of Multidimensional Lattice Geometry codes is 8.

Theorem 2.4

In the case where no four slopes are linearly dependant each pair of variable nodes is joined by either zero or exactly two paths of length 4.

Comment 2.2

This case includes the group of codes where $\rho = m$ which is the group of codes with the highest rate achievable by this construction.

Proof

Combining equations from theorem 2.3 gives $a.S_1 + b.S_2 = v_1 - v_2$. Assume there is a third path of length four between two variable nodes, as above this path would give rise to an equation $c.S_3 + d.S_4 = v_1 - v_2$ for some $c, d \in 0 \ldots d_1 - 1$. Adding these gives:

$$\begin{pmatrix} a & b & c & d \end{pmatrix} . \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3)$$

for some

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \neq \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix} \quad (4)$$

This violates the condition that any four slopes are linearly independent. Therefore there cannot be any third path of length four between v_1 and v_2.

Theorem 2.5

In the case where no four lines in the geometry are linearly dependant the number of proper cycles of length 8 in the code is given by $(\rho(\rho-1)(d_1-1)^2 n)/8$

Proper cycles of length 8 are defined as distinct sets of four variable nodes which participate in a length 8 cycle.

Proof

Consider the tree expansion centered on a variable node v, Fig 1.

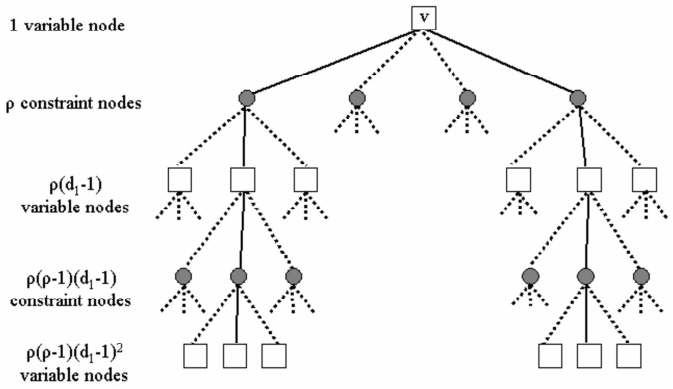

Fig. 1. Tree expansion of a Multidimensional Lattice Code about node v

The first line of this expansion consists of the ρ constraint nodes directly connected to v. The second line of $\rho(d_1 - 1)$ variable nodes, all of which are unique as there are no cycles of length 4 in the code. The third line consists of $\rho(d_1-1)(\rho-1)$ constraint unique nodes. They are all unique as any repetition would give rise to a cycle of length 6 in the code. The fourth line consists of $\rho(\rho-1)(d_1-1)^2$ variable nodes. By theorem 2.4, these nodes must consist of $(\rho(\rho-1)(d_1-1)^2)/2$ unique nodes, each occurring twice. A path of length 8 passing through v can be traced between each pair, representing a length 8 cycle. The variable node v therefore participates in $(\rho(\rho-1)(d_1-1)^2)/2$ proper cycles of length 8. Multiplying by, n, the number of nodes and dividing by four, as each cycle contains four variable nodes, gives the total number of proper eight cycles as $(\rho(\rho-1)(d_1-1)^2 n)/8$

Theorem 2.6

The minimum distance of finite geometry codes is lower bounded by $d_{min} \geq 2^m$

Proof:

It is equivalent to prove that no non-zero codeword exists with Hamming weight less than 2^m. We shall prove this by induction on m, the dimension of the geometry.

First, take the case where m=1. This lattice consists of d_1 points which lie in a straight line. Any non-zero codeword must contain at least one non-zero element. As this non-zero element must participate in at least one parity check equation for the code to be non-disjoint, there must be at lease one further non-zero element for that equation to be satisfied.

Therefore, in the case $m = 1$, the shortest non-

zero codeword must have a Hamming weight of $d_{min} \geq 2$.

Now take an k dimensional lattice, G_k, and a $k-1$ dimensional sub lattice, G_{k-1} of minimum distance $d_{min}^{(G_{k-1})}$. For the code to be non-disjoint there must be at least one bundle of parallel line sections in G_k passing through G_{k-1}. Constraint 2.3 ensures that each line section contains only 1 point in G_{k-1}. For the associated parity-check equations to be satisfied, each non-zero element in a codeword over G_{k-1} must give rise to at least one further non-zero element in a codeword over G_k. Therefore, the codeword with hamming weight $d_{min}^{(G_{k-1})}$ in G_{k-1} gives rise to a codeword of weight $d_{min}^{(G_k)} \geq 2d_{min}^{(G_{k-1})}$.

Therefore, by induction on m, for an m dimensional geometry $d_{min} \geq 2^m$.

Extended Multidimensional Lattice Codes

The Multidimensional Lattice Codes described above have the constraint that the slope in the dimension D_1 takes the value 1. By removing this constraint the number of slopes meeting constraint 2.2 increases giving a wider range of code rates for a given value of m.

Irregular codes can be produced by varying the depth of the lattice in each dimension, however this paper shall deal with regular codes where the depth of the lattice in each dimension is a constant, d. Theorems 2.1 to 2.6 remain valid for regular Extended Multidimensional Lattice Codes as they are not dependant on the slope in dimension D_1.

These codes can be viewed as a transfer of the work of [2] and [4] from Euclidian Geometries to the Finite Lattice.

Example 2.2

Fig 2 shows a 2 dimensional finite lattice of size 3×3. 2 bundles of parallel lines are drawn. Each line contains 3 points. This is example gives a regular code with 9 variable nodes and 6 constraint nodes. The column and row weights are 2 and 3 respectively. It can easily be seen that the bundles of parallel lines chosen are linearly independent so no loop of length 6 can exist, see theorem 2.3. The shortest cycle in this code is of length 8 and a cycle of length 8 is highlighted. As the H matrix of this code has a column weight of 2 it is impractical in real communication systems. It is shown for illustrative purposes.

For clarity, the variable nodes are numbered and the H matrix of this code is reproduced below.

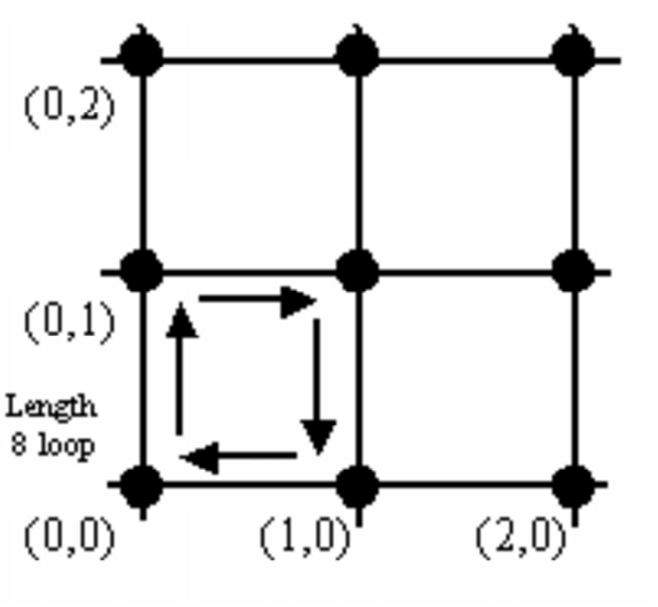

Fig. 2. A finite geometry of size 3×3 showing two linearly independent bundles of parallel lines. A cycle of length 8 is highlighted.

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Example 2.3

A 4 dimensional finite lattice of size $8 \times 8 \times 8 \times 8$. 5 bundles of parallel lines are drawn with slopes;

$$\begin{pmatrix} S_a \\ S_b \\ S_c \\ S_d \\ S_e \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{6}$$

Each line contains 8 points. This example gives a regular code with 4096 variable nodes and 2560 constraint nodes. As not all rows of H are linearly independent, there are 2101 information bits per codeword giving the code a rate of 51.3 percent. The column and row weights are 5 and 8 respectively. It can easily be shown that any three of the bundles of parallel lines chosen are linearly independent so no loop of length 6 can exist, giving the code a girth of 8, see theorem 2.3. This code was used in compiling the results in Fig 4.

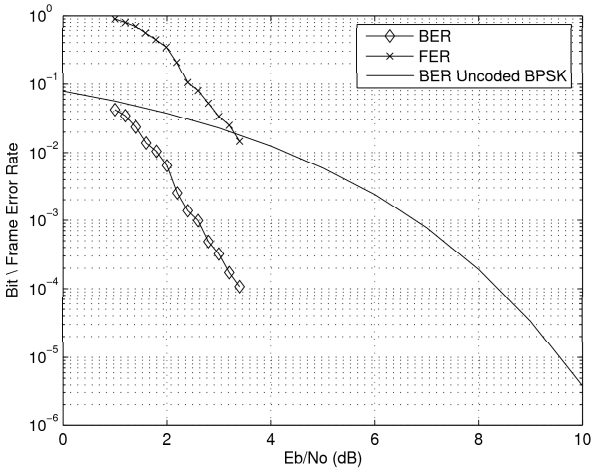

Fig. 3. BER for length 1296 rate 0.48 code.

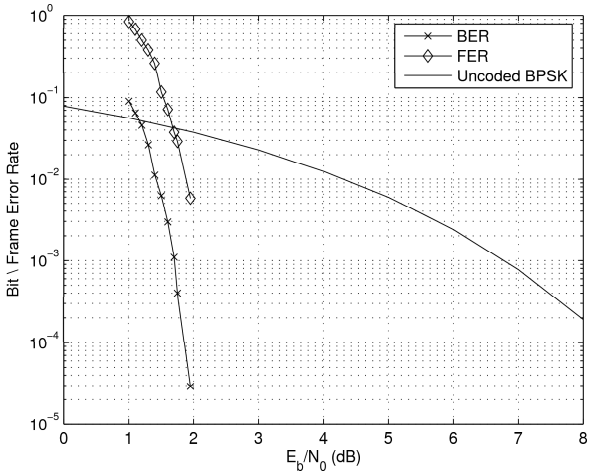

Fig. 4. BER for length 4096 rate 0.51 code.

III. Results

In this section, error correcting performance is demonstrated for the two example girth-8 codes given in examples 2.1 and 2.3. BPSK transmission over the AWGN channel was assumed, and in each case decoding continued until either a valid codeword was detected or a maximum of 20 iterations were completed.

Fig 3 shows the performance of the length 1296 code of example 2.1. The 1296×864 parity check matrix has a rank of 625, giving the code a rate of 48.2%. The second code tested is given by example 2.3. Its 4096×2560 H matrix has a row rank of 2101, giving a rate of 51.3%. Its performance is given in Fig 4.

IV. Conclusion

Two classes of regular LDPC codes based upon finite integer lattices have been presented. These codes were shown to exhibit desirable properties such as a girth of eight in the Tanner Graph, a lower bound on the minimum distance and, given certain constraints, a sparsity of eight cycles.

References

[1] R. Gallager. "Low-density parity-check codes," *IEEE Transactions on Information Theory* **Volume 8, Issue 1**, January, 1962 : 21–28.

[2] H. Tang and J. Xu and Y. Kou and S. Lin and K. Abdel-Ghaffa "On algebraic construction of Gallager and circulant low-density parity-check codes," *IEEE Transactions on Information Theory* **Volume 50, Issue 6**, June, 2004 : 1269–1279.

[3] B. Vasic and O. Milenkovic. "Combinatorial Construction of Low-Density Parity-Chack Codes for Iterative Decoding," *IEEE Transactions on Information Theory* **Volume 50, Issue 6**, June, 2004 : 1156–1167.

[4] M. Flanagan and J. Craddock and C. P. Fewer and S. Redmond. "A Euclidean Geometry Based Algebraic Construction Technique for Girth-8 Gallager LDPC Codes," *IEEE Information Theory Workshop, Chengdu, China* , October 2006.