Annales UMCS Informatica AI XII, 3 (2012) 107–119
DOI: 10.2478/v10065-012-0034-y

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

LDPC Codes Based on Algebraic Graphs

Monika Polak[1]*, Vasyl Ustimenko[1,2]†

[1] *Institute of Mathematics, Maria Curie-Sklodowska University*
pl. M. Curie-Sklodowskiej 5, 20-031 Lublin, Poland
[2] *Institute of Telecommunications and Global Information Space,*
National Academy of Science of Ukraine
Chokolovsky Boulevard 13, Kiev, Ukraine.

Abstract – In this paper we investigate correcting properties of LDPC codes obtained from families of algebraic graphs. The graphs considered in this article come from the infinite incidence structure. We describe how to construct these codes, choose the parameters and present several simulations, done by using the MAP decoder. We describe how error correcting properties are dependent on the graph structure. We compare our results with the currently used codes, obtained by Guinand and Lodge [**1**] from the family of graphs $D(k, q)$, which were constructed by Ustimenko and Lazebnik [**2**].

1 Introduction

The first work in the field of Coding Theory was published in 1948 by C. E. Shannon [**3**], who initiated the development of this theory. Today Coding Theory is very important because we are living in the information society. The information provided by non-ideal communication channels is always the subject to error, therefore, there is a need to use error correcting codes. Tools of Coding Theory have to be used together with cryptographic algorithms because even a unique error during the transmission of ciphertext can make the decryption impossible. One powerful class of error correcting codes are the LDPC codes, which were discovered by Robert Gallanger in his work *Low-Density Parity-Check Codes* [**4**]. They were forgotten for twenty years to get back in the nineties, for example see [**5, 6, 7**]. They have been adopted in many recent communication standards such as DVB-S2, 10GBase-T, 802.16e and 802.11n.

*monika.katarzyna.polak@gmail.com
†ustymenko_vasyl@yahoo.com

Construction of LDPC codes can be random or based on Finite Geometries, cyclic structures or quasi-cyclic structure. Those based on geometric structures have linear decoding complexity, while for random codes decoding complexity is exponential. In this paper we observe recent results on applications of Computer Algebra and Theory of Algebraic Graph and construction of new LDPC. The ability to use graphs in construction of LDPC was first discussed by Tanner [**8**]. Construction of Tanner type codes based on the expander graphs was considered for example by Siper and Spielman [**7**], Guinanad and Lodge [**9, 1**].

The LDPC codes can be regular or irregular. The best results were obtained for the irregular LDPC codes but it is known that their construction often leads to error-floor, which very rarely takes place in the case of regular codes and coding complexity in the case of irregular codes is greater. In the regular LDPC code every row has the same constant weight d_c and every column has the same constant weight d_v.

The error correcting code is $A \subset \mathbb{F}_2^N$, where $\mathbb{F}_2 = \{0, 1\}$ and the codewords are in the classical Hamming metric:

$$d(x, y) = |\{i : x_i \neq y_i\}|.$$

We assume that every codeword is from the set:

$$C = \{y \in \mathbb{F}_2^N | Hy^T = 0\},$$

where H is a parity check matrix for the code C.

In order to improve the transmission quality for each text block of the length N there are added k bits, which do not carry information and only have checking properties. As a result of the action, we get the code words $y \in C$ of the length N. Such a code has $r = N - k$ parity checks and is denoted by $[N, k]$. The ratio r/N is called code rate and is denoted by R_C.

The linear error correcting code can be represented in three ways: by the generator matrix G, parity check matrix H or Tanner graph $\Gamma(V, E)$. The *parity checks matrix* for $[N, k]$ code is $r \times N$ matrix whose words are zeros or ones. Rows of this matrix correspond to the equations of parity check and the columns to the codeword bits. If the bit number j in the codeword is checked by the parity check number i, then in the position (i, j) in the matrix H is one, if not there is zero. Each bit is checked by a unique set of control equations. The switching column does not change code properties and provides an equivalent code. The standard example of Low-Density Parity-Check Codes given by Gallanger [**4**] is code $[16, 12]$ with the matrix H of the form:

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0

The code which has a representation as a sparse matrix or a sparse Tanner graph is called the Low-Density Parity-Check Code. A matrix is sparse if the number of ones in it is small compared to the number of zeros.

Generator matrix G for $[N, k]$ code is $k \times N$ zero-ones matrix whose rows create code base. G creates a codeword y for the information vector x of the length k: $y = x \cdot G$. Each information vector corresponds to exactly one codeword. Parity checks matrix and generator matrix are both dependent.

Bipartite graph is called the graph $\Gamma(V, E)$ in which a set of nodes V can be represented as the disjoint union $V = V_1 \cup V_2$ in such a way that no two vertices from each set V_i, $i = 1, 2$ are connected by edge. *Tanner graph* is called the bipartite graph in which one subset V_1 corresponds to the codeword bits and the other V_2 to the parity checks. The vertex from the set V_1 is connected to a vertex from the set V_2 if and only if a bit corresponding to the vertex from V_1 is controlled by the parity check corresponding to the vertex from V_2.

There is a standard way to create the LDPC codes depending on the adjacency matrix of bipartite, biregular Tanner graph. The parity check matrix H is a part of the adjacency matrix of graph used to create a code. If $V_1 > V_2$ then the adjacency matrix has the form:

$$\begin{array}{c} \\ V_2 \\ V_1 \end{array} \begin{array}{c} \begin{array}{cc} V_2 & V_1 \end{array} \\ \begin{pmatrix} 0 & H \\ H^T & 0 \end{pmatrix} \end{array}$$

Matrix H is sparse if and only if corresponding graph is sparse, which means that it has a small number of edges in relation to the number of vertices. The simple relationship describing the density of the graph $\Gamma(V, E)$ is

$$g = \frac{2|E|}{|V|(|V| - 1)},$$

where $|E|$ is the number of edges of graph Γ and $|V|$ is the number of vertices.

2 Descryption of graphs

The missing definitions from the Simple Graphs Theory can be found in [**11**]. The families of graphs described below are the families of simple graphs. *Simple graphs* are undirected graphs containing no graph loops or multiple edges. A distance between the vertices v_1 and v_2 of the graph is the length of the minimal path from v_1 to v_2. A graph is connected if for the arbitrary pair of vertices v_1, v_2 there is a path from v_1 to v_2. The *girth* of a connected, simple graph is a length of the shortest cycle in a graph. We refer to the bipartite graph $\Gamma(V_1 \cup V_2, E)$ as (a, b) biregular one if every vertex from V_1 has a degree a and every vertex from V_2 has a degree b. We call a graph regular in the case $a = b$.

Let K be a commutative ring. The graphs from the families $D(n,K)$, $\widetilde{D(n,K)}$ and $A(n,K)$ are bipartite, $|K|$-regular, sparse and without cycles of the length less than 6 for arbitrary n and K. They come from infinite incidence structures. These graphs are good expanders ([**12, 10**]). The following interpretation of $D(n,K)$ can be found by the reader for example in [**12**]. Traditionally one subset of vertices in the graphs $D(n,K)$ is denoted by $V_1 = P$ and called a set of points and another one $V_2 = L$ is called a set of lines. P and L can be considered as two copies of the Cartesian power K^n, where n is a positive integer. Brackets and parentheses allow us to distinguish points and lines.

If $z \in K^n$, then $(z) \in P$ and $[z] \in L$. Let us use the notions for points and lines introduced in [**12**]:

$$\begin{aligned}(p) &= (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p'_{2,2}, p_{2,3}, ..., p_{i,i}, p'_{i,i}, p_{i,i+1}, p_{i+1,1}...),\\ [l] &= [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l'_{2,2}, l_{2,3}, ..., l_{i,i}, l'_{i,i}, l_{i,i+1}, l_{i+1,1}...].\end{aligned} \tag{1}$$

We now define an infinite incidence structure (P, L, I) in the following way. We say that the point (p) is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations between their coordinates hold:

$$\begin{cases} l_{1,1} - p_{1,1} = l_{1,0}p_{1,0} \\ l_{1,2} - p_{1,2} = l_{1,1}p_{1,0} \\ l_{2,1} - p_{2,1} = l_{0,1}p_{1,1} \\ l_{i,i} - p_{i,i} = l_{0,1}p_{i-1,i} \\ l'_{i,i} - p'_{i,i} = l_{i,i-1}p_{1,0} \\ l_{i,i+1} - p_{i,i+1} = l_{i,i-1}p_{1,0} \\ l_{i+1,i} - p_{i+1,i} = l_{0,1}p'_{i,i} \end{cases} \tag{2}$$

where $i \geq 2$. The set of vertices (P, L, I) is $P \cup L$ and the set of edges consisting of all pairs $\{(p), [l]\}$ for which $(p)I[l]$. For each positive integer $n > 2$ we obtain a finite incidence structure (P_n, L_n, I_n) as follows. Firstly, P_n and L_n are obtained from P and L, respectively, by projecting each vector onto its n initial coordinates with respect to the natural order. The incidence I_n is then defined by imposing the first $n-1$ incidence equations and ignoring all others. The graph corresponding to the finite incidence structure (P_n, L_n, I_n) is denoted by $D(n,K)$.

The family of graphs $\widetilde{D(n,K)}$ described below comes from the Cartan matrix $\begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$. Let us use the analogous notions for points and lines in the graph $\widetilde{D(n,K)}$:

$$\begin{aligned}(p) &= (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p_{2,3}, ..., p_{i,i}, p_{i,i+1}, p_{i+1,1}...),\\ [l] &= [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,1}, l_{2,2}, l_{2,3}, ..., l_{i,i}, l_{i,i+1}, l_{i+1,1}...].\end{aligned} \tag{3}$$

The infinite incidence structure $\widetilde{(P, L, I)}$ is defined in the following way. We say that the point (p) is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations

between their coordinates hold:

$$\begin{cases} l_{1,1} - p_{1,1} = l_{1,0}p_{1,0} \\ l_{1,2} - p_{1,2} = l_{1,1}p_{1,0} \\ l_{2,1} - p_{2,1} = l_{0,1}p_{1,1} \\ l_{i,i} - p_{i,i} = l_{0,1}p_{i-1,i} - l_{i,i-1}p_{1,0} \\ l_{i,i+1} - p_{i,i+1} = l_{i,i-1}p_{1,0} \\ l_{i+1,i} - p_{i+1,i} = l_{0,1}p_{i,i} \end{cases} \tag{4}$$

where $i \geq 2$. The graph corresponding to the finite incidence structure $(P_n, \widetilde{L_n, I_n})$ obtained by the way described above is denoted by $\widetilde{D(n,K)}$.

For the graph $A(n,K)$ let us use the notion for points and lines:

$$\begin{aligned} (p) &= (p_{0,1}, p_{1,1}, p_{1,2}, p_{2,2}, p_{2,3}, ..., p_{i,i}, p_{i,i+1}, ...), \\ [l] &= [l_{1,0}, l_{1,1}, l_{1,2}, l_{2,2}, l_{2,3}, ..., l_{i,i}, l_{i,i+1}, ...]. \end{aligned} \tag{5}$$

For this graph we define an incidence structure $(P, L, I)_A$ as follows. We say that the point (p) is incident with the line $[l]$, and we write $(p)I[l]$, if the following relations hold:

$$\begin{cases} l_{i,i} - p_{i,i} = l_{1,0}p_{i-1,i} \\ l_{i,i+1} - p_{i,i+1} = l_{i,i}p_{0,1} \end{cases} \tag{6}$$

Denote this infinite incidence structure $(P, L, I)_A$ as $A(K)$ and it can be identified with the bipartite incidence graph of $(P, L, I)_A$. $A(K)$ is an infinite tree. For each positive integer $n > 2$ we obtain an finite incidence structure $(P_n, L_n, I_n)_A$ as above. The incidence graph corresponding to the structure $(P_n, L_n, I_n)_A$ is denoted by $A(n,K)$.

In the case $K = \mathbb{F}_q$, where q is the prime power we denote $D(n, \mathbb{F}_q)$, $\widetilde{D(n, \mathbb{F}_q)}$ and $A(n, \mathbb{F}_q)$ simply as $D(n,q)$, $\widetilde{D(n,q)}$, $A(n,q)$ accordingly.

$D(n,K)$, $\widetilde{D(n,K)}$ and $A(n,K)$ have different structures for $n \geq 6$. For all n they are $|K|$-regular but have a structure that allows us to remove points and lines in such a way that we can obtain an arbitrary bidegree (a,b) for $1 \leq a, b \leq |K|$. We can make it as shown in **[13]**. When L is a set of all lines and P is a set of all points to obtain the desired bidegree (a,b) we must put restriction on coordinates. Let $A \subset \mathbb{F}_q$ and $B \subset \mathbb{F}_q$ be an a-element and b-element subsets respectively and let V_P and V_L be sets of points and lines in the new bipartite graph. They are the following sets:

$$\begin{aligned} V_P &= \{(p) \in P | c_p \in A\} \\ V_L &= \{[l] \in L | c_l \in B\}, \end{aligned}$$

where c_p is fixed coordinate of points and c_l is fixed coordinate of lines. Usually restriction are imposed on first coordinates.

3 Code construction and results

The described families consist of simple, bipartite graphs of high girth (girth bigger or equal 6 for any parameters so there are no short cycles) . It is good for codes because

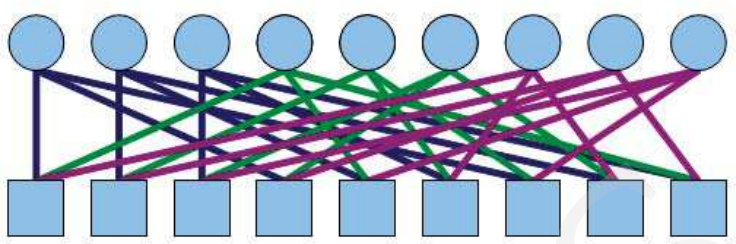

Fig. 1. Graph $D(2,3) = \widetilde{D(2,3)} = A(2,3)$

short cycles decrease the speed of decoding algorithms. The girth in the graphs from the described families increases with the increasing n. In fact, the adverse influence of short cycles decreases with the increasing length of the codewords N, but they also contribute to the formation of the error-floor. To create the LDPC code with the codeword of length N we use $\widetilde{D(n,K)}$ ($D(n,K)$, $A(n,K)$), where $n^{|K|} > N$. Each of these graphs is already bipartite. To obtain biregularity we can use the above described method. Let us denote the number of parity checks by d and it should be at least 2. We reduce the bidegree to $(d,|K|)$ by the method shown above. We can also reduce the bidegree to (d,e), where $d,e \leq |K|$ by taking smaller subsets A, B of $\mathbb{F}_q$. We can impose restrictions on the value of chosen coordinate of lines or points. Bidegree reduction can only increase the girth so there is no risk that short cycles will appear. After the bidegree reduction the graph can become disconnected and divided into several components. Choose one vertex and take the component containing this selected vertex (point or line)and find all other vertices for which there is a path to the chosen one. We use this component to create a parity check matrix. If $|V_P| > |V_L|$ then the points correspond to the code words, the bits and lines to the parity checks, if not then the lines correspond to the code words, the bits and points to the parity checks. We decide to put one or zero in a parity check matrix by checking if relations (1), (2), or (3) (depending on what family of graphs we create code) on their coordinates hold.

Transmission quality depends mainly on code, decoding algorithm and the level of noise in a communication channel. Properties of error-correcting codes are tested by determining the relationship between the noise level and the bit error rate. The bit error rate (BER) is the ratio of the number of the error bits to the total number of the transferred bits. Simulation is usually carried out for the Gaussian Channel where disruptions are modelled by the White Gaussian Noise. Our simulations were done using the BPSK modulation over the AWGN channel and simple MAP decoder implementation with 10 iterations. Let y be the received codeword. The MAP decoder works according to the rule which returns an output value $\hat{x}$ of a code word x for which the *a posterior* probability $P = (x|y,H)$ is maximized.

Traditionally d_v means the number of ones per column in the matrix H and d_c the number of ones per row. In general, it is assumed that $d_v < d_c$ in the case the

number of bits of information is greater than the number of control bits for economic reasons. In 1948 Claude Shannon **[3]** in his works proved that there exists a code allowing the transfer of information from any small error probability if the rate of information transmission is below the capacity. The parameter related to the error correction properties of the code is the minimum distance d_{min} between the codewords measured in the Hamming metric. Correction properties are better for the codes that have a higher minimum distance or a very small number of codewords which are in the minimum distance from each other. A study of asymptotic performance **[14]** shows the future for the degree distribution: $d_v \geq 2$ is enough. Every bit in the codeword should be checked by a unique set of control equations. If $d_v = 2$ then every bit is checked by 2 equations and the condition: $\binom{r}{2} > N$ is necessary. It is easy to see that for every code in this article $r^2 > 2N$. If $d_v \geq 3$ then d_{min} for the code increases linearly with the increasing N, so the error probability decreases exponentially with increasing length of the block. Figs 2, 3 and 4 show that the presented codes give good results for $d_v = 2$. The codes considered in this article have $d_v = \min(a, b)$.

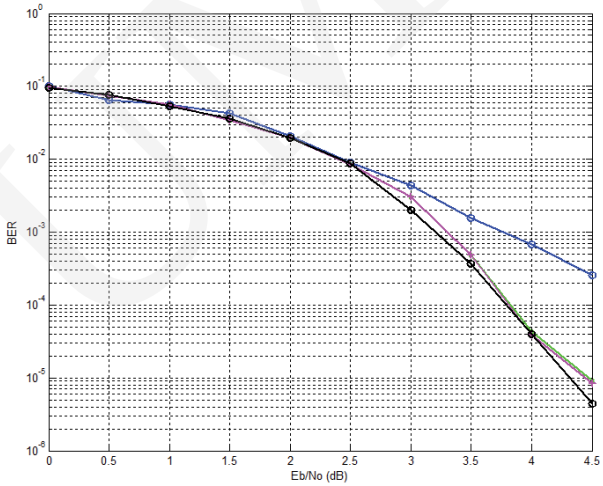

Fig. 2. Bit error rate for $[50, 125]$ code (blue) based on $\widetilde{D(6,5)}$, $[1250, 3125]$ code (red) based on $\widetilde{D(7,5)}$, $[1250, 3125]$ code (purple) based on $\widetilde{D(8,5)}$ and $[1250, 3125]$ code (black) based on $\widetilde{D(10,5)}$

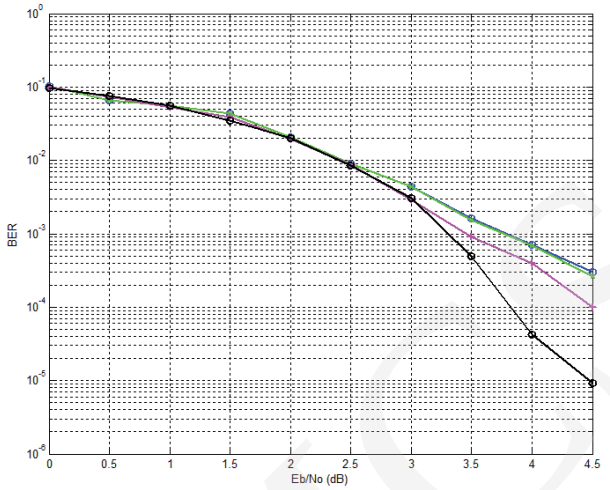

Fig. 3. Bit error rate for $[50, 125]$ code (green) based on $D(6, 5)$, $[50, 125]$ code (blue) based on $D(7, 5)$, $[250, 625]$ code (purple) based on $D(8, 5)$ and $[1250, 3125]$ code (black) based on $D(9, 5)$

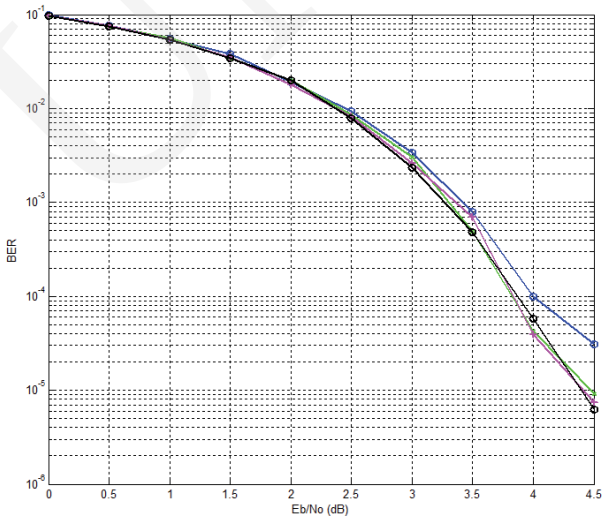

Fig. 4. Bit error rate for $[250, 625]$ code (blue) based on $A(6, 5)$, $[1250, 3125]$ code (green) based on $A(7, 5)$, $[1250, 3125]$ code (purple) based on $A(8, 5)$ and $[1250, 3125]$ code (black) based on $A(9, 5)$

Tables 1, 2, 3 and 4 show the properties of the sample codes.

Table 1. Properties of the graphs $\widetilde{D(n,5)}$ after receiving bidegree $(2,5)$ used for the presented sample codes

Based graph	$\|P\| = \|L\|$	Size of desired H	Block length
$\widetilde{D(2,5)}$	25	10×25	25
$\widetilde{D(3,5)}$	125	10×25	25
$\widetilde{D(4,5)}$	625	50×125	125
$\widetilde{D(5,5)}$	3125	50×125	125
$\widetilde{D(6,5)}$	15625	50×125	125
$\widetilde{D(7,5)}$	78125	1250×3125	3125
$\widetilde{D(8,5)}$	390625	1250×3125	3125
$\widetilde{D(9,5)}$	1953125	1250×3125	3125
$\widetilde{D(10,5)}$	9765625	6250×15625	15625

Table 2. Properties of the graphs $D(n,5)$ after receiving bidegree $(2,5)$ used for the presented sample codes

Based graph	$\|P\| = \|L\|$	Size of desired H	Block length
$D(2,5)$	25	10×25	25
$D(3,5)$	125	10×25	25
$D(4,5)$	625	50×125	125
$D(5,5)$	3125	50×125	125
$D(6,5)$	15625	50×125	125
$D(7,5)$	78125	50×125	125
$D(8,5)$	390625	250×625	625
$D(9,5)$	1953125	1250×3125	3125
$D(10,5)$	9765625	1250×3125	3125

Table 3. Properties of the graphs $A(n,5)$ after receiving bidegree $(2,5)$ used for the presented sample codes

Based graph	$\|P\| = \|L\|$	Size of desired H	Block length
$A(2,5)$	25	10×25	25
$A(3,5)$	125	10×25	25
$A(4,5)$	625	50×125	125
$A(5,5)$	3125	50×125	125
$A(6,5)$	15625	250×625	625
$A(7,5)$	78125	1250×3125	3125
$A(8,5)$	390625	1250×3125	3125
$A(9,5)$	1953125	1250×3125	3125
$A(10,5)$	9765625	6250×15625	15625

Figs 2, 3 and 4 show BER for the codes obtained from the graphs described in Tables 1, 2, 3 accordingly. With the increasing parameter n, the graphs for the same field produce the codes with better correcting properties (Figs 2, 3 and 4) and bigger block length N. The graphs $A(n,q)$ have the most consistent structure, so that they

Table 4. Properties of the graphs $\widetilde{D(5,q)}$, $D(5,q)$ and $A(5,q)$ after receiving bidegree to $(2,5)$ and $(2,q)$used for the presented sample codes

Based graph	$\lvert P\rvert = \lvert L\rvert$	Biregularity	Size of desired H	Block length	R_C
$\widetilde{D(5,5)}$, $D(5,5)$, $A(5,5)$	3125	(2,5)	50×125	125	0.4
$\widetilde{D(5,7)}$, $D(5,7)$, $A(5,7)$	16807	(2,5)	98×245	245	0.4
$\widetilde{D(5,11)}$, $D(5,11)$, $A(5,11)$	161051	(2,5)	242×605	605	0.4
$\widetilde{D(5,7)}$, $D(5,7)$, $A(5,7)$	16807	(2,7)	98×343	343	0.28
$\widetilde{D(5,11)}$, $D(5,11)$, $A(5,11)$	161051	(2,11)	242×1331	1331	0.18

give codes with the best correcting properties. For example, the graph $A(8,5)$ after the reduction of bidegrees to $(2,5)$ splits into 125 components and $D(8,5)$ into 625, $A(10,3)$ after the reduction of bidegrees to $(2,3)$ splits into 81 components and $D(10,3)$ into 243. The codes obtained from the graphs $D(n,q)$ give the worst results in the case of considered families of graphs.

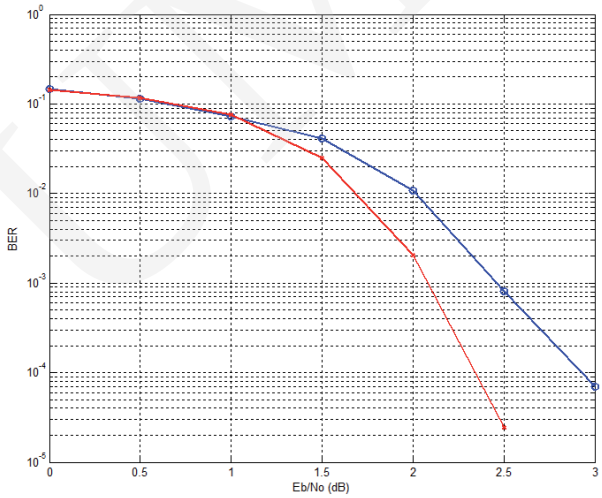

Fig. 5. Bit error rate for $[1875, 3125]$ code (red) based on $\widetilde{D(6,5)}$ and $[375, 625]$ code (blue) based on $\widetilde{D(5,5)}$ after reducing bidegree to $(3,5)$

Use the bidegree (a,b), where $a \geq 3$ makes the desired codes much better (Fig. 5) and of bigger block size. For these two presented codes $d_v = 3$. In the case of the codes which come from the presented families, reducing bidegrees to (a,b) gives the code rate $R_C = \frac{a}{b}$. The codes described in Tables 1, 2, 3 (Figs 2, 3 and 4) have the code rate 0.4. When we use the bigger field $\mathbb{F}_q$ we can obtain a different and often better (more economial) code rate (For example see Table 4) and usually the code correcting

properties do not change much. However, we must be careful because much bigger field and reduced biregularity to $(2, q)$ or $(3, q)$ can give R_C close to zero, but the error correcting properties can be much worse.

The presented codes have a high possibility to choose the code rate R_C. In many well known constructions, the code rate is strictly determined, for example it is equal to 1/2. David MacKay considerd **[6]** very good, randomly generated codes with the code rate 1/2 and 1/4. The codes arising from the graphs with the symmetric adjacency matrix, which was considered in **[15]** have the code rate 1/2 and 1. Fig. 6 shows the codes: $[75, 150]$ (blue), $[500, 1000]$ (green), $[1875, 3750]$ (purple), $[250, 500]$ (black) obtained from the random construction based on the Radford M. Neal's programs available from **[6]**. Radfold M. Neal and David MacKay reinvented the LDPC codes in the mid-1990', **[6]**. Fig. 7 shows the codes based on the presented graphs with the same number of information bits k accordingly to the randomly generated codes. It is easy to see that the codes (blue, green and purple) based ont he graphs (Fig. 7) with the same number of information bits as the random codes (Fig. 6) have better error correcting properties and smaller code rate (Table 5).

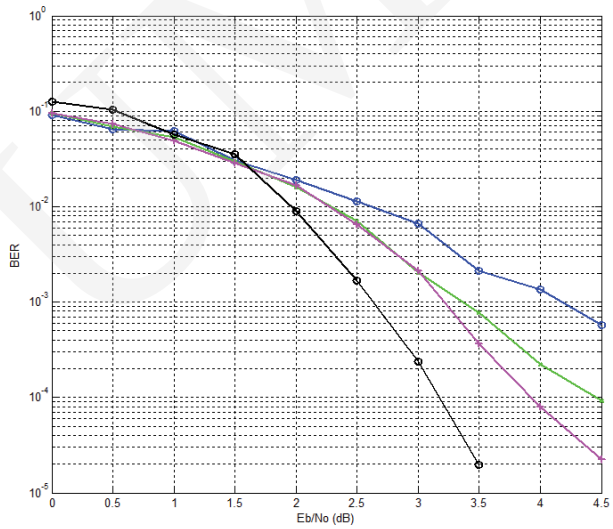

Fig. 6. Bit error rate for $[75, 150]$ code (blue), $[500, 1000]$ code (green), $[1875, 3750]$ code (purple) and $[250, 500]$ code (black), all based on the Radford M. Neal random constructions **[16]** with the code rate $R_C = 1/2$.

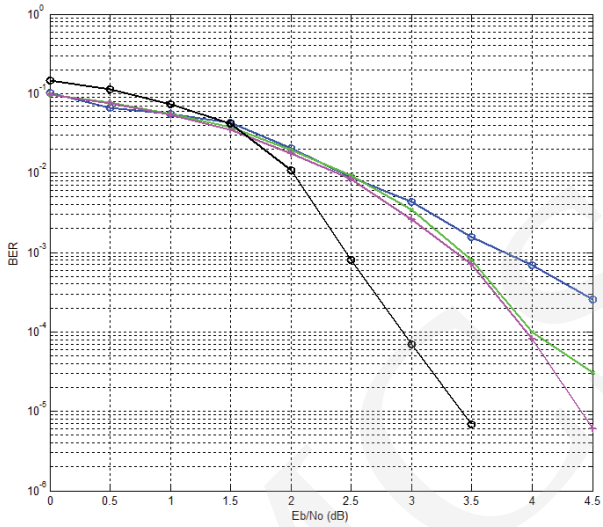

Fig. 7. Bit error rate for $[50, 125]$ code (blue) based on $\widetilde{D(6,5)}$, $[250, 625]$ code (green) based on $A(6,5)$, $[1250, 3125]$ code (purple) based on $A(9,5)$ and $[375, 625]$ code (black) based on $\widetilde{D(5,5)}$

Table 5. Comparison between the presented codes and other effective LDPC

Code	Number of information bits	Block length	R_C	Number of ones per column
random $[75, 150]$	75	150	0.5	2
random $[500, 1000]$	500	1000	0.5	2
random $[1875, 3750]$	1875	3750	0.5	2
random $[250, 500]$	250	500	0.5	3
$[50, 125]$ based on $\widetilde{D(6,5)}$	75	125	0.4	2
$[250, 625]$ based on $A(6,5)$	500	625	0.4	2
$[1250, 3125]$ based on $A(9,5)$	1875	3125	0.4	2
$[375, 625]$ based on $\widetilde{D(5,5)}$	250	625	0.6	3

References

[1] Guinand P., Lodge J., Tanner type codes arising from large girth graphs, Canadian Workshop on Information Theory CWIT '97, Toronto, Ontario, Canada, June 3-6 (1997): 5.

[2] Lazebnik F., Ustimenko V. A., Woldar A. J., A characterization of the components of the graphs $D(k, q)$, Discrete Mathematics 157 (1996): 271.

[3] Shannon C. E., A Mathematical Theory of Communication, Bell System Technical Journal 27 (1948): 379.

[4] Gallager R. G., Low-Density Parity-Check Codes, IRE Trans of Info Thy 8 (1962): 21.

[5] Luby M. G., Mitzenmacher M., Shokrollahi M. A., Spielman D. A., Improved Low-Density Parity-Check Codes Using Irregular Graphs and Belief Propagation, in ISIT 98-IEEE International Symposium of Information Theory, Cambridge, USA (1998): 171.

[6] MacKay D. J. C., Neal R. M., Good Codes Based on Very Sparse Matrices, in Cryptography and Coding 5th IMAConference, BERLIN (1995): 100.

[7] Sipser M., Spielman D. A., Expander codes, IEEE Trans on Info Theory 42 (6) (1996): 1710.

[8] Tanner R. M., A recursive approach to low density codes, IEEE Transactions on Information Theory IT 27 (5) (1984): 533.

[9] Guinand P., Lodge J., Graph theoretic construction of generalized product codes, IEEE International Symposium on Information Theory ISIT'97 Ulm, Germany June 29-July 4 (1997): 111.

[10] Romańczuk U., Ustimenko V., On Extremal Graph Theory, explicit algebraic constructions of extremal graphs and corresponding Turing encryption machines, in Artificial Intelligence, Evolutionary Computing and Metaheuristics, In the footsteps of Alan Turing Series: Studies in Computational Intelligence 427 (2012).

[11] Bollobas B., Extremal Graph Theory, Academic Press (1978).

[12] Lazebnik F., Ustimenko V. A., New examples of graphs without small cycles and of large size, European Journal of Combinatorics 14 (1993): 445.

[13] Lazebnik F., Ustimenko V. A., Explicit construction of graphs with an arbitrary large girth and of large size, Discrete Applied Mathematics 60 (1995): 275.

[14] Brower A., Cohen A., Nuemaier A., Distance regular graphs, Springer, Berlin (1989).

[15] Tonchev V. D., Error-correcting codes from graphs, Discrete Math. 257 (2002): 549.

[16] Zhang Z., Lee P., Anantharam V., Nikolic B., Wainwright M., Dolecek L., Predicting error floors of structured LDPC codes: deterministic bounds and estimates, Journal IEEE Journal on Selected Areas in Communications - Special issue on capaciyy approaching codes archive 27 (6) (2009): 908.