Annales UMCS Informatica AI XI, 2 (2011)
143–152; DOI: 10.2478/v10065-011-0003-x

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

On LDPC codes corresponding to affine parts of generalized polygons

Monika Polak*, Vasyl Ustimenko†

Institute of Mathematics, Maria Curie-Sklodowska University,
pl. M. Curie-Sklodowskiej 1, 20-031 Lublin, Poland

Abstract

In this paper we describe how to use special induced subgraphs of generalized m-gons to obtain the LDPC error correcting codes. We compare the properties of codes related to the affine parts of q-regular generalised 6-gons with the properties of known LDPC codes corresponding to the graphs $D(5, q)$.

1. Introduction

Tools of Coding theory have to be used together with cryptographic algorithms. Even a unique error during the transmission of ciphertext can make the decryption impossible. It is interesting that the same families of special graphs can be effectively used both in Coding Theory and Cryptography.

Information is always transmitted through the communication channel, which can be air, telephone line, beam of light or cable. It is usually very important for the recipient to receive exactly the same message as was given. Unfortunately, messages are usually exposed to interference, which could cause errors in the transmission. In order to minimize the number of errors in the transmission we can use error correcting codes.

*E-mail address: monika.katarzyna.polak@gmail.com
†E-mail address: ustymenko_vasyl@yahoo.com

Whole information in the computer is represented as zero-ones sequences. Coding of information using the linear error correcting codes means adding to the sequences of k elements some extra bits in a certain way. These bits do not carry information and only have checking properties. We denote $[n, k]$ the code, which has a length of code words n and k information bits. In that code we have $r = n - k$ parity checks. The ratio r/n is called *code rate* and is denoted by R_C. It is interesting to look for codes with the best correction properties at the lowest code rate for economic reasons. In 1948 Claude Schannon in his works defined the concept of capacity and 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. Let T be the time of transmiting a single bit. Then *the rate of information transmission* is $R_t = \frac{k}{nT}$. Unfortunately, he did not show a way of constructing such codes. The most known classes of error correcting codes are Turbocodes and Low Density Parity Check Codes (LDPC codes). In this article we are only interested in the LDPC codes.

The LDPC codes were introduced in 1963 by Robert G. Gallanger. These codes have a high possibility of selection of parameters n and r, making it possible to create codes with a large block size and excellent correction properties. Their advantage is the existence of efficient decoding algorithms of linear complexity of the block length n.

There are few methods that allows to obtain the LDPC codes, but it is possible to construct very good codes from the families of graph that already exist or construct graphs with specific properties useful for this purpose. The ability to use graphs to construct error correcting codes was first discussed by Tanner [8]. This is the area where we can work because only specified graphs are suitable for creating a good code. Usually for this purpose, *simple graphs* are used, which means graphs undirected and containing no graph loops or multiple edges. The graph must be bipartite, sparse, without small cycles and biregular or regular with the possibility to obtain biregularity. The length of the shortest cycle in the graph is called *girth*. It is interesting to look for graphs with high girth because the codes based on them have better error correcting properties.

Every linear error correcting code can be represented in three ways: by the generator matrix G, parity checks matrix H or Tanner graph $\Gamma(V, E)$. *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 parity checks and the column to the codeword bits. If bit number j in the codeword is checked by the parity check number i then in the position (i, j) in matrix H is one if not there is zero. Each bit is

checked by a unique set of control equations. In the regular LDPC code every row has the same constant weight r and every column has the same constant weight s. Switching column does not change code properties and provides an equivalent code. We assume that every codeword is from the set

$$\mathcal{C} = \{y \in \mathbf{F}_2^n |\ Hy^T = 0\}.$$

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 dependent. It is known that if $G = [I_k|A]$ is a generator matrix in the standard form for the $[n, k]$ code $\mathcal{C}$ then $H = [-A^T|I_{n-k}]$ is a parity check matrix for $\mathcal{C}$.

Bipartite graph we call the graph $\Gamma(V, E)$ in which a set of nodes V can be divided into two subsets $V = V_1 \cup V_2$ in such a way that no two vertices from each set V_i, $i = 1, 2$ are connected by edge.

Tanner graph we call the bipartite graph in which one subset V_1 corresponds to the codeword bits and second V_2 to the parity checks. 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.

The code which has a representation as a sparse matrix or a sparse Tanner graph we call the LDPC code. The matrix is called a *sparse* if their ratio of ones to the number of zeros in each row and column is small compared to the length of the rows and columns. The very primary example of LDPC code is $[7, 4]$ the Hamming code with $R_C = \frac{3}{7}$. Sparse graph 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.

In this paper we show that the graph corresponding to the affine part of the generalized m-gons can be used to obtain a very good class of LDPC codes. We discuss how to use them and compare our results with those used by the NSA codes obtained by Guinand and Lodge [2] who used as a base for codes the graphs with suitable properties $D(k, q)$ constructed by Ustimenko and Lezebnik (see [4,5,6]).

2. Description of $AH(q,q)$

Missing definitions of theory of simple graphs can be found by the reader in [13].

The distance between the vertices v_1 and v_2 of the graph is the length of minimal pass from v_1 and v_2. The graph is connected if for an arbitrary pair of vertices v_1, v_2 there is a pass from v_1 to v_2. The diameter of a connected simple graph is the maximum of the distances between vertices in the graph.

We refer to the bipartite graph $\Gamma(V_1 \cup V_2, E)$ as a biregular one if the number of neighbours for representatives of each partition set are constants $r+1$ and $s+1$ (bidegrees). We call the graph regular in the case $r = s$.

Generalized m-gons are connected, biregular, bipartite graphs with girth $2m$, diameter m and bidegress $(r+1, s+1)$. Traditionally, in the case of generalised m-gon $\Gamma(V_1 \cup V_2, E)$ one partition set $V_1 = P$ is called the set of points and the other $V_2 = L$ is called the set of lines. Vertices corresponding to point can be connected by edges only with a vertex from L and the vertex corresponding to the line can be connected only with the vertex from the set P.

When two vertices point (p) and line $[l]$ are connected by the edge we call this incidence pair (p, l) *flag*. We define the distance from flag (p, l) to the vertex $v \in V$ as the sum of distances from p to v and l to v.

Affine generalized m-gon can be obtained in the following way. Let us choose a flag (p, l) in the generazilzed m-gon and remove all points and lines except these which are in the maximal distance from the flag. By this method we obtain a biregular graph with the bidegrees r and s. It is clear that affine generalized m-gons have a girth $\geq 2m$. If the generalised m-gon is edge transitive then the structure/construction of affine generalised m-gon does not depend on the choice of the flag. In the case $m = 6$ there is only one known family of regular generalised m-gons with the bidegrees $r + 1 = s + 1$, where $r = q = p^m$, p is prime, $m \geq 1$. Each representative of this family is an edge transitive graph.

When $m = 6$ we denote generalized m-gon as $GH(q,q)$ and affine generalized m-gon as $AH(q,q)$, where q is a prime power. For more details about this structure we refer to [1]. Note that $q+1$-regular graph $GH(q,q)$ has $1 + q + q^2 + q^3 + q^4 + q^5$ points and the same number of lines. The order of q-regular $AH(q,q)$ is $2q^5$. The following interpretation of $AH(q,q)$ can be used for $p \geq 5$. Let $\mathbb{F}_q$ be the finite field containing q elements. Each point can be identified with $(p) = (x, y, z, u, w) \in$ and each line with $[l] = [a, b, c, d, f]$. Brackets and parentheses allow us to distinguish points and lines. We say point (p) is incident to line $[l]$, and we write $(p)I[l]$ if the following relations on their coordinates

hold:

$$\begin{cases} y - b = xa \\ 2c - z = 2xb \\ u - 3d = -3xc \\ 2w - 3f = 3zb - 3yc + ua \end{cases} \tag{1}$$

where all coordinates are elements of $\mathbb{F}_q$. $AH(q,q)$ is regular but has a structure that allows us to remove points and lines in such a way that we can obtain an arbitrary bidegree. We can do it exactly the same as it was done with $D(k,q)$ in [6]. Let L be a set of all lines and P a set of all points. To obtain the desired bidegree (r,s) we must put restriction on coordinates. Let $R \subset \mathbb{F}_q$ and $S \subset \mathbb{F}_q$ be an r-element and s-element subsets respectively and let V_P and V_L be sets of points and lines in a new bipartite graph. They are the following sets:

$$V_P = \{(p) \in P | x \in R\}$$
$$V_L = \{[l] \in L | a \in S\}.$$

If the set of points is bigger than the set of lines, then points correspond to codeword bits and lines correspond to parity checks. Otherwise, lines correspond to codeword bits and points correspond to parity checks.

3. Code construction

To create the LDPC code of dimension d containing (n,k) the Hamming code as component codes we must use $AH(q,q)$, where q is the first prime which is greater than n. Then we reduce the bidegree to (d,n). Bidegree reduction can only increase the girth. After reduction the bidegree graph can be disconnected. When we put restriction on the coordinates x of point, the graph will be divided into several components. But when we put restriction on the first coordinate a of lines the graph remains connected. This is due to a lack of symmetry $AH(q,q)$. Next we take one component containing a chosen vertex (point or line) and find all other vertices for which there is a path to the chosen vertex. We use this component to create a parity checks matrix. If $|V_P| > |V_L|$ then points correspond to code words, bits and lines to parity checks, if not then lines correspond to code words bits and points to parity checks. We decide to put one or zero in parity check matrix by checking if relations (1) on their coordinates hold. Every bit from the codeword is checked by d parity checks. In the case of graphs $D(k,q)$, the resulting graphs are always disconnected. It is interesting that the properties of codes obtained from $D(5,q)$ and $AH(q,q)$ through the restriction on points coordinates are similar as can be seen in Figs 1 and 2. Table 1 contains the data about the resulting graphs.

The Graphs $D(5,7)$ and $AH(7,7)$ with $x \in R$ after the reduction bidegree to $(2,7)$ split into 49 components and $D(4,7)$ splits into 7 ones. They all give equivalent codes and it doesn't make any difference which component we choose. The graphs $AH(17,17)$ with restriction on the points coordinate $x \in R$, $|R| = 2$ and $D(5,17)$ after reduction bidegree to $(2,15)$ split into 289 components. The larger field we use the better code rate we obtain, for example taking $\mathbb{F}_7$ for the codes based on these graphs, the code rate is $R_C \approx 0.286$ but taking $\mathbb{F}_{17}$ we have $R_C = 0.1(3)$.

4. Results

Transmission quality depends mainly on code, decoding algorithm and the level of noise in a communication channel. Code error correcting properties 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 bit errors to the total number of transferred bits. Simulation usually carried out for Gaussian Channel where noise is modelled by Gaussian White Noise so our simulations

Table 1. Graphs property after receiving bidegrees $(2,7)$ and $(2,15)$ respectively.

Initial graph	Girth	Restriction on coordinates	Number of lines in fixed component	Number of points in fixed component	Code rate
$AH(17,17)$	12	$a \in S$, $\|S\| = 2$ $x \in R$, $\|R\| = 15$	167042	1252815	0.1(3)
$AH(17,17)$	12	$x \in R$, $\|R\| = 2$ $a \in S$, $\|S\| = 15$	4335	578	0.1(3)
$D(5,17)$	10	$x \in R$, $\|R\| = 2$ $a \in S$, $\|S\| = 15$	4335	578	0.1(3)
$D(5,17)$	10	$a \in S$, $\|S\| = 2$ $x \in R$, $\|R\| = 15$	578	4335	0.1(3)
$AH(7,7)$	12	$a \in S$, $\|S\| = 2$	4802	16807	≈ 0.286
$AH(7,7)$	12	$x \in R$, $\|R\| = 2$	343	98	≈ 0.286
$D(5,7)$	10	$x \in R$, $\|R\| = 2$	343	98	≈ 0.286
$D(5,7)$	10	$a \in S$, $\|S\| = 2$	98	343	≈ 0.286
$D(4,7)$	8	$x \in R$, $\|R\| = 2$	98	343	≈ 0.286
$D(4,7)$	8	$a \in S$, $\|S\| = 2$	343	98	≈ 0.286

were done using the BPSK modulation over the AWGN channel and simple MAP decoder implementation.

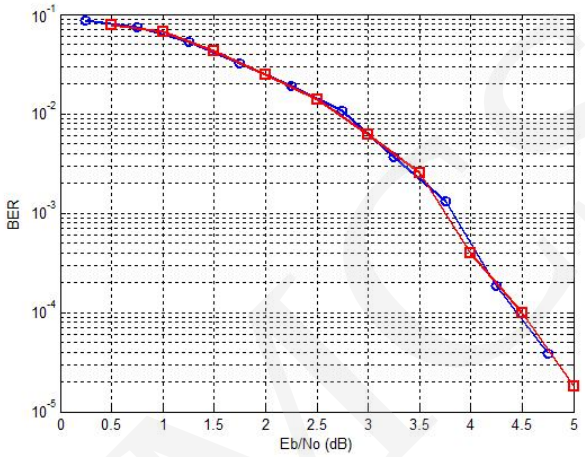

Fig. 1. Bit error rate for $[343, 98]$ code (circle) based on $AH(7, 7)$ and $[343, 98]$ code (square) based on $D(5, 7)$, both with $x \in \{0, 1\}$.

In $D(k, q)$ there is no difference if we put restriction on points or lines. When we take lines from a smaller partition set in a reduced $AH(q, q)$, we obtain a better code but with exactly the same code rate $R_C \approx 0.286$ as if we take points from R, $|R| = 2$. Fig. 3 shows the results.

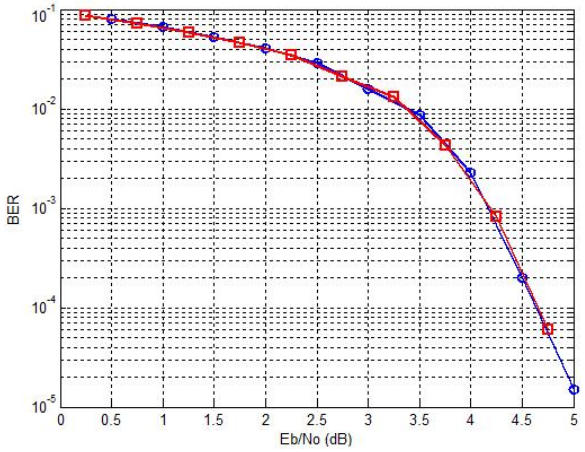

Fig. 2. Bit error rate for $[4335, 578]$ code (circle) based on $AH(17, 17)$ and $[4335, 578]$ code (square) based on $D(5, 17)$, both with $x \in \{0, 1\}$ and $a \in \{0, 14\}$.

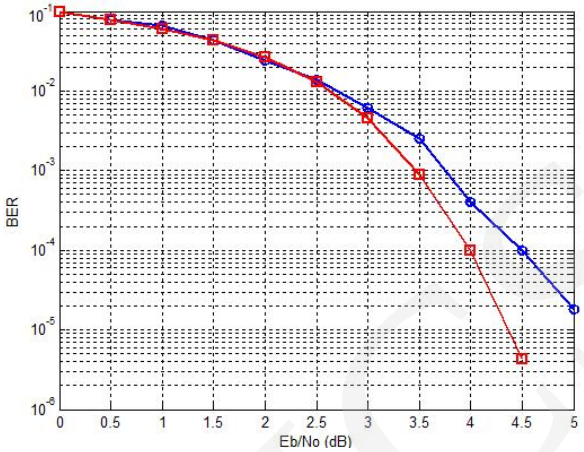

Fig. 3. Bit error rate for $[16807, 4802]$ code (square) with $a \in \{0, 1\}$ and chosen vertex $[l] = [0, 0, 0, 0, 0]$ and $[343, 98]$ code (circle) with $x \in \{0, 1\}$ and chosen vertex $(p) = (0, 0, 0, 0, 0)$, both based on $AH(7, 7)$.

5. Remarks

In [2] as coordinates the authors used elements from $\mathbb{F}_q$ where q is the first prime greater than n. We take q which is the first prime power greater than n. Fig. 4 shows that for the code based on $D(3, 16)$ we obtain as good results as for $D(3, 17)$. $D(3, 16)$ gives $[256, 32]$ code with a slightly better code rate than code $[255, 34]$ arising from $D(3, 17)$.

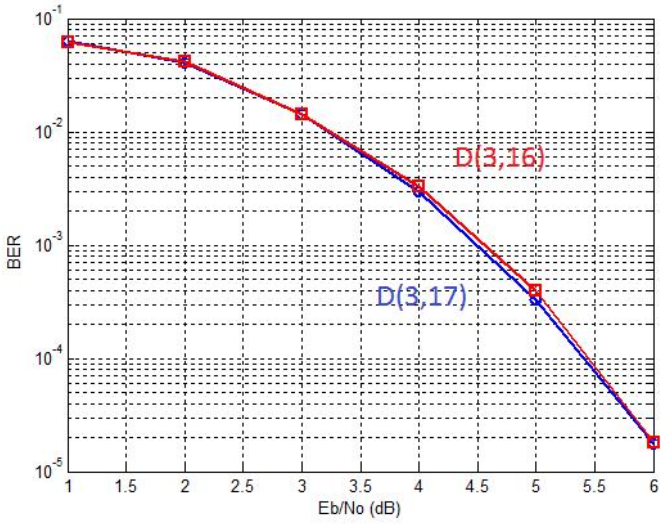

Fig.4 . Bit error rate for $[256, 32]$ code (square) based on $D(3, 16)$ and $[255, 34]$ code (circle) based on $D(3, 17)$.

Obviously in each code we can reduce the bidegree of graphs to $(3, 7)$ or $(3, 15)$ depending on a field and a graph. But then the code rate increases.

It is also possible to construct the LDPC codes based on the graphs $AO(q, q^2)$, $q = 2^{2k+1}$ arising from generalized 8-gons whose girth is 16, so it seems that this kind of codes has better error correcting properties. The graph $AO(q, q^2)$ has the bidegree (q, q^2). It also has structure which allows easily to remove points and lines to obtain arbitrary bidegree exactly in the same way as it was done with $D(k, q)$ and $AH(q, q)$.

References

[1] Ustimenko V., Woldar A., Extremal properties of regular and affine generalized polygons as tactical configurations, European Journal of Combinatorics 24 (2003):99.

[2] 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.

[3] 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.

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

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

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

[7] Lazebnik F., Ustimenko V. A., Woldar A. J., A new series of dense graphs of high girth , Bulletin (New Series) of the AMS, 32(1) (1995): 73.

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

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

[10] Huffman W. C., Pless V., Fundamentals of error correcting codes, first edition, Cambridge University Press, Cambridge, 2003.

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

[12] Shannon C. E., Weaver Warren, The Mathematical Theory of Communication, Univ Of Illinois Pr 1963.

[13] Brower A.,Cohen A., Nuemaier A., Distance regular graphs, Springer, Berlin, 1989.

[14] Bollobas B., Extremal Graph Theory. Academic Press, 1978.

[15] Shokrollahi A., LDPC Codes: An Introduction, Digital Fountain Inc, Fremont (2002), available from:
`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.1008`.

[16] Shaska T., Ustimenko V., On some applications of graph theory to cryptography and turbocoding, Special issue of Albanian Journal of Mathematics:Proceedings of the NATO Advanced Studies Institute "New challenges in digital communications", May 2008, University of Vlora 2(3) (2008): 249.

[17] Ustimenko V. A., On the extremal regular directed graphs without commutative diagrams and their applications in coding theory and cryptography, Albanian. J. of Mathematics, Special Issue Algebra and Computational Algebraic Geometry 1(N4) (2007): 387.

[18] Shaska T., Huffman W. C., Joener D., Ustimenko V. (editors), Advances in Coding Theory and Cryptography, Series on Coding and Cryptology 3 (2007): 181.