Math. Maced.
Vol. 6 (2008)
73-79

ADDITIVE CIRCULANT GRAPH CODES OVER GF(4)

ZLATKO VARBANOV[1]

Dedicated to Academician Blagoj Popov on the Occasion of His 85^{th} Birthday

Abstract. In this paper we consider an algorithm for constructing additive circulant graph codes over GF(4). Also, we present some new results obtained by this algorithm.

1. Introduction

It is well-known [4] that additive self-orthogonal codes over $GF(q^2)$ can be used to represent a class of quantum error-correcting codes. Several papers (for instance, [4, 6, 7]) were devoted to classifying or constructing additive self-dual codes over $GF(4)$. Additive self-dual codes over $GF(9), GF(16)$ and $GF(25)$ were classified in [5]. Moreover, it was shown in [12] that certain vectors in some additive self-dual codes over $GF(4)$ hold generalized t-designs as well as classical t-designs with possibly repeated blocks. These facts motivate the construction of additive self-dual codes over $GF(4)$.

The problem of classification of additive self-dual codes is to construct all nonequivalent codes of given length and minimum distance. This problem has two ingredients: to generate the objects of interest so that at least one object is generated from every class of equivalence, and to remove equivalent objects from consideration. All additive self-dual codes over $GF(4)$ of length n have previously been classified (up to equivalence) by Calderbank et al. [4] for $n \leq 5$, by Höhn [10] for $n \leq 7$, and by Glynn et al. [8] for $n \leq 9$. Gaborit et al. [7] classified all extremal codes of length 8, 9, 11, and 12. Gulliver and Kim [9] classified many circulant and 4-circulant codes of length $n \leq 27$. Using graph representation, Danielsen and Parker [6] gave a full classification of the codes of length $n \leq 12$. Varbanov [16] classified all extremal (optimal) codes of length 13 and 14, and constructed many extremal codes of length $15 \leq n \leq 21$. Huffman [11] constructed many new additive self-dual codes of length $n \leq 30$ with an automorphism of odd prime order.

The purpose of this paper is to describe an algorithm for constructing additive self-dual codes with some special properties. Also, we present new results obtained by this algorithm. The paper is structured in the following way. Section 2 consists of some basic definitions and preliminary results. In Section 3 we give a description of an algorithm for constructing additive circulant graph codes and an analysis of its complexity. Section 4 contains new results for additive self-dual codes obtained by the algorithm.

2000 *Mathematics Subject Classification.* 94B05, 68W01, 68Q25.
Key words and phrases. graph code, circulant matrix, algorithm.
[1]Supported by RD491-09/2008 Project, Veliko Tarnovo University

2. Preliminaries

Let $GF(4) = \{0, 1, \omega, \bar{\omega}\}$ with convention that $\bar{\omega} = \omega^2 = 1 + \omega$. We recall some definitions on additive codes over $GF(4)$ from [4, 7].

An *additive code* C *over* $GF(4)$ *of length* n is an additive subgroup of $GF(4)^n$. As C is a free $GF(2)$-module, it has size 2^k for some $0 \leq k \leq 2n$. We call C an $(n, 2^k)$ code. It has a basis, as a $GF(2)$-module, consisting of k basis vectors; a generator matrix of C is a $k \times n$ matrix with entries in $GF(4)$ whose rows are a basis of C.

There is a natural inner product arising from the trace map. The trace map $Tr : GF(4) \to GF(2)$ is given by $Tr(x) = x + x^2$. In particular $Tr(0) = Tr(1) = 0$ and $Tr(\omega) = Tr(\bar{\omega}) = 1$. The *conjugate* of $x \in GF(4)$, denoted $\bar{x}$, is the following image: $\bar{0} = 0, \bar{1} = 1$, and $\bar{\bar{\omega}} = \omega$.

We now define the *trace inner product* of two vectors $x = (x_1, x_2, \ldots, x_n)$, $y = (y_1, y_2, \ldots, y_n)$ in $GF(4)^n$ is

$$x \star y = \sum_{i=1}^{n} Tr(x_i \bar{y}_i) \tag{2.1}$$

If C is an additive code, its *dual* code with respect to (2.1) is the code $C^{\perp} = \{x \in GF(4)^n | x \star c = 0 \text{ for all } c \in C\}$. If C is an $(n, 2^k)$ code, then $C^{\perp}$ is an $(n, 2^{2n-k})$ code. As usual, C is *self-orthogonal* (with respect to (2.1)) if $C \subseteq C^{\perp}$, and *self-dual* if $C = C^{\perp}$. In particular, if C is self-dual, then C is an $(n, 2^n)$ code.

As usual, *weight* of a codeword $c \in C$ ($wt(c)$) is the number of nonzero components of c. The minimum weight d of a code C is the smallest weight of any nonzero codewords of C. If C is an additive $(n, 2^k)$ code with minimum weight d then C is called an $(n, 2^k, d)$ code. C is $Type\ II$ code if C is self-dual and all codewords have even weight; $Type\ II$ codes of length n exist only if n is even [7]. If C is self-dual but some codeword has odd weight, the code is $Type\ I$. There is a bound on the minimum weight of an additive self-dual code ([14], Theorem 33). If d_I and d_{II} are the minimum weights of additive self-dual $Type\ I$ and $Type\ II$ codes, respectively, of length $n > 1$, then

$$d_I \leq \begin{cases} 2\lfloor n/6 \rfloor + 1, & n \equiv 0 \ (mod\ 6); \\ 2\lfloor n/6 \rfloor + 3, & n \equiv 5 \ (mod\ 6); \\ 2\lfloor n/6 \rfloor + 2, & \text{otherwise} \end{cases} \tag{2.2}$$

$$d_{II} \leq \ 2\lfloor n/6 \rfloor + 2$$

A code that meets the appropriate bound is called *extremal*. If the code is not extremal but no code of the given type can exist with a larger minimum weight then the code is called *optimal*. $Type\ II$ codes meeting the bound d_{II} have a unique weight enumerator [7]. This property is not true for $Type\ I$ codes.

We say that two additive codes C_1 and C_2 are *equivalent* provided there is a map sending the codewords of C_1 onto the codewords of C_2 where the map consists of a permutation of coordinates, followed by a scaling of coordinates by elements of $GF(4)$, followed by conjugation of some of the coordinates. The automorphism group of C, denoted $Aut(C)$, consists of all maps which permute coordinates, scale coordinates, and conjugate coordinates that send codewords of C to codewords of C.

A *graph code* is an additive self-dual code over $GF(4)$ with generator matrix $G = \Gamma + \omega I$ where I is the identity matrix and Γ is the adjacency matrix of a simple undirected graph, which must be symmetric with 0's along the diagonal.

Example:

$$\Gamma = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \; G = \begin{pmatrix} \omega & 1 & 1 \\ 1 & \omega & 1 \\ 1 & 1 & \omega \end{pmatrix}$$

A graph code is always self-dual, since its generator matrix has full rank over $GF(2)$ and $C\overline{C}^T$ only contains entries from $GF(2)$ whose traces must be zero.

Schlingemann [15] first proved (in terms of *quantum stabilizer states*) that for any self-dual quantum code, there is an equivalent graph code. This means that there is a one-to-one correspondence between the set of simple undirected graphs and the set of additive self-dual codes over $GF(4)$. We have seen that every graph represents an additive self-dual code over $GF(4)$, and that every additive self-dual code over $GF(4)$ can be represented by a graph.

3. Additive circulant graph codes

A matrix B of the form:

$$B = \begin{pmatrix} b_0 & b_1 & \dots & b_{n-2} & b_{n-1} \\ b_{n-1} & b_0 & b_1 & \dots & b_{n-2} \\ \dots & \dots & \dots & \dots & \dots \\ b_2 & \dots & b_{n-1} & b_0 & b_1 \\ b_1 & b_2 & \dots & b_{n-1} & b_0 \end{pmatrix}$$

is called a *circulant matrix*. The vector $(b_0, b_1, \dots, b_{n-1})$ is called *generator vector* for the matrix B. An additive code with circulant generator matrix is called *circulant* code (see [9]).

An *additive circulant graph (ACG) code* is a code corresponding to graph with circulant adjacency matrix. Circulant graphs must be regular, i.e., all vertices must have the same number of neighbours.

It is easy to see that such matrix has the following property: $b_i = b_{n-i}, \forall\, i = 1, \dots, n-1$, and $b_0 = \omega$. Then, the entries in the generator matrix of ACG code depend on the coordinates $(b_1, b_2, \dots, b_{\lfloor n/2 \rfloor})$ only. Therefore, we can restrict our search space to the $2^{\lfloor n/2 \rfloor}$ codes over $GF(4)$ of length n corresponding to graphs with circulant adjacency matrices.

Besides a smaller search space, the special form of the generator matrix of a graph code makes it easier to determine the minimum distance, since any codeword obtained as a linear combination of i rows of the generator matrix must have weight at least i. If, for example, we want to check whether a code has minimum distance at least d, we only need to consider combinations of $d-1$ or fewer rows of its generator matrix.

Our search algorithm is the following:

INPUT: positive integers n and d $(1 < d < n)$.

OUTPUT: all possible ACG codes of length n and minimum distance $\geq d$.

Step 1. If n is even, take a binary vector $g^{(0)} = (g_1, g_2, \dots g_{\frac{n}{2}})$ and extend it to a vector $g = (\omega, g_1, g_2, \dots, g_{\frac{n}{2}-1}, g_{\frac{n}{2}}, g_{\frac{n}{2}-1}, \dots, g_2, g_1)$. If n is odd then $g^{(0)} = (g_1, g_2, \dots g_{\frac{n-1}{2}})$, and $g = (\omega, g_1, g_2, \dots, g_{\frac{n-1}{2}}, g_{\frac{n-1}{2}}, \dots, g_2, g_1)$

Step 2. Construct a circulant matrix G (a generator matrix of an ACG code) with generator vector g.

Step 3. Compute all linear combinations of $1, 2, ..., d-1$ rows of G and check their weights. If all weights are $\geq d$ then the minimum distance is at least d.

Step 4. If $g^{(0)}$ is not all-one vector – $g^{(0)} = g^{(0)} + 1$, Step 1.

END.

For Step 3, to compute all linear combinations of at most $d-1$ rows of G, we should use exactly t embedded cycles (for any $t < d$). This is not appropriate for practical purposes – certain number of cycles have to be added to (or removed from) the program for any change of t. To eliminate this problem we use the algorithm *GCQARY_NONREC* ([2], p.14) that is a non-recursive emulation of t embedded cycles. Also, to check a weight of given nonbinary vector we do not need to check every coordinate position of the vector. We can use bit-wise representation of the codewords and faster algorithm [3]. In step 4, the operation $g^{(0)} = g^{(0)} + 1$ actualy means that we take the next binary vector (by lexicographic order). Initially, $g^{(0)} = (0, 0, \ldots, 0, 1)$.

To obtain all nonequivalent codes among the constructed codes of given length we use a transformation into linear binary codes and the program package Q-Extension [1]. The transformation from additive code C over $GF(4)$ into a binary code is done by applying the following map $\beta : 0 \to 000, 1 \to 011, \omega \to 101, \bar{\omega} \to 110$ to the coordinates of C. The map β sends an $(n, 2^k)$ additive code C to $\beta(C)$, a linear binary $[3n, k]$ code. It is known [4] that if two additive codes are equivalent then two binary images are equivalent, and conversely. We transform the codes into their binary images and check them for equivalence using Q-Extension.

4. Results

Gulliver and Kim [9] performed a computer search of circulant self-dual additive codes over $GF(4)$ of length up to 30. Their search was not restricted to graph codes, so our search space is a subset of theirs. On the other hand, in some cases our search include all circulant graph codes of given length (not only extremal or optimal codes).

In this section we construct ACG codes of lengths $13 \leq n \leq 36$ with maximum d that the codes of this type can reach. We give full classification of ACG codes of lengths $13 \leq n \leq 33$, and we construct some codes of lengths $34 \leq n \leq 36$. The complete classification of additive self-dual codes of lengths $2 \leq n \leq 12$ was done in [6]. Also, a classification of extremal/optimal self-dual codes of lengths 13 and 14 was given in [16]. We will compare our results with those obtained in [5, 9].

$n = 13$: There are 8 nonequivalent ACG codes of length 13 – two optimal codes with minimum distance $d = 5$, four codes with $d = 4$, one code with $d = 3$, and one code with $d = 2$.

$n = 14$: There are 30 nonequivalent ACG codes – 3 codes with $d = 6$ (all codes are *Type II*), 3 codes with $d = 5$, 14 codes with $d = 4$ (6 *Type I* and 8 *Type II*), 2 codes with $d = 3$, and 8 codes with $d = 2$ (3 *Type I* and 5 *Type II*).

$n = 15$: There are 39 nonequivalent ACG codes of this length – 2 codes with $d = 6$, 10 codes with $d = 5$, 10 codes with $d = 4$, 10 code with $d = 3$, and 7 codes with $d = 2$.

$n = 16$: There are 6 nonequivalent extremal codes with $d = 6$ (one code is *Type I* and 5 codes are *Type II*).

$n = 17$: There is a unique ACG $(17, 2^{17}, 7)$ code. This code is equivalent to the code constructed in [9].

$n = 18$: No extremal *Type I* ($d = 7$) or *Type II* ($d = 8$) codes of this length (see the bounds 2.2). There are 52 ACG $(18, 2^{18}, 6)$ codes (16 codes are *Type I* and 36 codes are *Type II*).

$n = 19$: There are 4 nonequivalent extremal codes with $d = 7$.

$n = 20$: There are 2 codes with $d = 8$. Their group orders are 40 and 6840, and they are equivalent to the codes constructed in [9].

$n = 21$: No extremal code (with $d = 8$) of this length. There are 11 nonequivalent ACG codes with $d = 7$.

$n = 22$: There are 14 nonequivalent extremal codes with $d = 8$ (all of them are *Type II*).

$n = 23 - 27$: No ACG codes with $d = 9$. There are 2 nonequivalent $(23, 2^{23}, 8)$ codes, 51 nonequivalent $(24, 2^{24}, 8)$ codes, 31 nonequivalent $(25, 2^{25}, 8)$ codes, 210 nonequivalent $(26, 2^{26}, 8)$ codes, and 140 nonequivalent $(27, 2^{27}, 8)$ codes.

$n = 28 - 29$: There are a unique $(28, 2^{28}, 10)$ code and a unique $(29, 2^{29}, 11)$ code. The group orders of these code are 56 and 812, respectively. They are equivalent to the codes with the same parameters constructed in [9].

$n = 30$: We construct 4 ACG codes of this length and $d = 12$. These codes have the same weight enumerator as the extended quadratic residue code $XQ29$. It is known [13] that the codes with generator matrices in graph form cannot be GF(4)-linear. As linearity is not invariant under the equivalence of additive codes, these 4 codes may be equivalent to the code $XQ29$. We were unable to determine the equivalence in this case.

$n = 31$: No extremal ACG code (with $d = 12$, see the bounds (2.2)), and no code with $d = 11$. There are exactly 62 nonequivalent codes with $d = 10$.

$n = 32$: No extremal ACG code with $d = 12$, and no code with $d = 11$. There are exactly 108 nonequivalent codes with $d = 10$ (2 codes are *Type I* and 106 codes are *Type II*).

$n = 33$: No extremal ACG code with $d = 12$, and no code with $d = 11$. There are exactly 76 nonequivalent codes with $d = 10$.

$n = 34 - 36$: No ACG code with $d > 10$. For $d = 10$, we construct 144 nonequivalent codes of length 34, 12 nonequivalent codes of length 35, and 4 nonequivalent codes of length 36.

We summarize the obtained results in Table 1. The maximum reached minimum distance is the same as in [5, 9]. But the classification results for some lengths are different than the results in [9]. For instance, 51 nonequivalent circulant codes of length 24 are constructed in [9], and all of them are *Type II*. In our work, we also construct 51 nonequivalent codes of this length but five of them are *Type I* (**these are the first constructed examples**), and 46 codes are *Type II*. This shows that the circulant graph code construction cannot produce the same nonequivalent codes as strong as the more general circulant code construction.

Also, in our work we improve the lower bound on the number of nonequivalent codes of length 26. We construct 210 codes of length $n = 26$ (49 codes are *Type I* and 161 codes are *Type II*). In [9], the number of the codes of this length is 14 (*Type I*) and 49 (*Type II*), respectively. In Table 2 we summarize the results about the codes of even length $14 \leq n \leq 26$.

References

[1] I. Bouyukliev, *What is Q-extension?*, Serdica J. Computing **1** (2007), 115–130.

[2] I.Bouyukliev, V.Bakoev, *A method for efficiently computing the number of codewords of fixed weights in linear codes*, Discrete Applied Mathematics **156** (2008), 2986–3004.

TABLE 1. Nonequivalent ACG codes of lengths $13 \leq n \leq 36$ for the maximum reached d

n	d	number	n	d	number	n	d	number
13	5	2	21	7	11	29	11	1
14	6	3	22	8	14	30	12	≥ 1
15	6	2	23	8	2	31	10	62
16	6	6	24	8	51	32	10	108
17	7	1	25	8	31	33	10	76
18	6	52	26	8	210	34	10	≥ 144
19	7	4	27	8	140	35	10	≥ 12
20	8	2	28	10	1	36	10	≥ 4

TABLE 2. Nonequivalent *Type I* and *Type II* ACG codes of even lengths

n	d_I	number	d_{II}	number
14	6	0	6	3
16	6	1	6	5
18	6	16	6	36
20	8	0	8	2
22	8	0	8	14
24	8	**5**	8	46
26	8	**49**	8	**161**

[3] I.Bouyukliev, V.Bakoev, *Efficient computing of some vector operations over $GF(3)$ and $GF(4)$*, Serdica J. Computing **2** (2008), 101–108.

[4] A.Calderbank, E.Rains, P.Shor, N.Sloane, *Quantum error correction via codes over $GF(4)$*, IEEE Trans. Inform. Theory. **44** (1998), 1369–1387.

[5] L.Danielsen, *Graph–Based Classification of Self-Dual Additive Codes over Finite Fields*, submitted Jan.2008

[6] L.Danielsen, M.Parker, *On the classification of all self-dual additive codes over GF(4) of length up to 12*, J.Combin.Theory, Series **A 113 (7)** (2006), 1351–1367, arXiv:math.CO/0504522.

[7] P.Gaborit, W.C.Huffman, J.L.Kim, and V.Pless, *On additive GF(4)-codes*, DIMACS Workshop on Codes and Association Schemes, DIMACS Series in Discrete Math. and Theoret. Computer Science, American Mathematical Society **56** (2001), 135–149.

[8] D.G.Glynn, T.A.Gulliver, J.G.Maks, M.K.Gupta, *The geometry of additive quantum codes*, submitted to Springer-Verlag, 2004

[9] T.A.Gulliver, J.L.Kim, *Circulant based extremal additive self-dual codes over GF(4)*, IEEE Trans. on Inform. Theory **40** (2004), 359–366.

[10] G. Höhn, *Self-dual codes over the Kleinian four group*, Math. Ann. **327 (2)** (2003), 227–255, arXiv:math.CO/0005266.

[11] W.C.Huffman, *Additive self-dual codes over F_4 with an automorphism of odd prime order*, Advances in Mathematics of Communications, **Volume 1, No. 3** (2007), 357–398

[12] J.L. Kim and V. Pless, *Designs in additive codes over GF(4)*, Design, Codes, and Cryptography **30** (2003), 187–199.

[13] M.van den Nest, *Local Equivalence of Stabilizer States and Codes*, Ph.D. thesis, K. U. Leuven, Belgium, May 2005.

[14] E.M. Rains, N.J.A. Sloane, *Self–dual codes*, in Handbook of Coding Theory, ed. V. S. Pless and W. C. Huffman, Amsterdam: Elsevier (1998), pp. 177–294.

[15] D. Schlingemann, *Stabilizer codes can be realized as graph codes*, Quantum Inf. Comput. **2 (4)** (2002), pp. 307–323, arXiv:quant-ph/0111080.
[16] Z.Varbanov, *Some new results for additive self-dual codes over GF(4)*, Serdica J. Computing **1** (2007), 213–227.

Department of Mathematics and Informatics,
University "'St.Cyril and St.Methodius"',
Veliko Tarnovo, Bulgaria
E-mail address: vtgold@yahoo.com, zl.varbanov@uni-vt.bg