JP Journal of Algebra, Number Theory and Applications

Published Online: September 2016
http://dx.doi.org/10.17654/NT038050473
Volume 38, Number 5, 2016, Pages 473-488 **ISSN: 0972-5555**

A NOTE ON LINEAR CODES FROM JOHNSON GRAPHS

Djoko Suprijanto and Teguh Nugraha

Combinatorial Mathematics Research Group
Faculty of Mathematics and Natural Sciences
Institut Teknologi Bandung
Bandung 40132, Indonesia
e-mail: djoko@math.itb.ac.id

Abstract

To construct linear codes having minimum distance as large as possible, for any given length and dimension is one of the major problems in coding theory. In this paper, we show the adequacy of Dougherty et al.'s method [2] by providing many numerical evidences. We obtain many extremal or nearly-extremal linear codes over several finite fields for modest lengths.

1. Introduction

Let $\mathbb{F}_q$ be a finite field of q elements. A *linear* $[n, k]$ code C over $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$. The value n is called *length* of C and the element of C is called *codeword* of C. A *generator matrix* G of a linear code C is a matrix whose rows are basis of C. The *weight* $wt(c)$ of a codeword $c \in C$ is the number of nonzero components of c. The minimum weight d of all nonzero codewords in C is called *minimum weight* of C. An $[n, k, d]$

Received: March 23, 2016; Accepted: May 14, 2016
2010 Mathematics Subject Classification: 94B05, 94B60.
Keywords and phrases: graphs, linear codes, extremal codes, nearly-extremal codes.
Communicated by K. K. Azad

code is an $[n, k]$ code with minimum weight d. The *weight enumerator* W of C is given by

$$W(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i,$$

where A_i denotes the number of codewords of weight i in C.

The space $\mathbb{F}_q^n$ is equipped by the Euclidean inner product defined by

$$[x, y] = \sum_{i=1}^{n} x_i y_i,$$

for two vectors $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ in $\mathbb{F}_q^n$. The (*Euclidean*) *dual code* $C^{\perp}$ of C is defined as

$$C^{\perp} = \{x \in \mathbb{F}_q^n : [x, c] = 0, \text{ for all } c \in C\}.$$

A code C is called (*Euclidean*) *self-dual* if $C = C^{\perp}$. From now on, what we mean by self-dual is Euclidean self-dual.

Two codes are called *equivalent* if one can be obtained from the other by a permutation of coordinates.

An undirected graph $\Gamma = (V, E)$ is a set of *vertices* V of cardinality n together with a collection E of *edges*, where each edge is an unordered pair of vertices. The vertices v_i and v_j are *adjacent* if $\{v_i, v_j\}$ is an edge. The *degree* of a vertex v is the number of vertices adjacent to v. A graph is called *regular of degree* k if all vertices have the same degree k. The *adjacency* matrix $A = (A_{ij})$ of a graph $\Gamma = (V, E)$ is a symmetric $(0, 1)$-matrix defined as follows: $A_{ij} = 1$ if the ith and jth vertices are adjacent, and $A_{ij} = 0$ otherwise. The *incidence* matrix $M = (M_{ij})$ of a graph $\Gamma = (V, E)$ is defined to be $M_{ij} = 1$ if the ith vertex is incident to the edge e_j, and $M_{ij} = 0$ otherwise.

We provide here some more notations and definitions. *Line graph* of G, denoted by LG, is defined as a graph with vertex set $V(LG) = E(G)$ in which two vertices in LG are adjacent if the related edges in G has a common vertex. It is well known that the adjacency matrix of LG has the form $M^T M - 2I$, where M is an incidence matrix of G. *Tensor product* $G \times H$ of graphs G and H is a graph such that the vertex set $V(G \times H) = V(G) \times V(H)$, and $(u_1, u_2), (v_1, v_2) \in V(G \times H)$ are adjacent if and only if $\{u_1, v_1\} \in E(G)$ and $\{u_2, v_2\} \in E(H)$. We denote by TG the tensor product of graphs G and G.

Among the main problems in coding theory is to construct linear codes with the highest possible minimum distance, for any given length and dimension. It is because the minimum distance is proportional to performance of the codes in detecting and correcting error in communication channels. We may construct linear codes by using graphs.

There are several manners to construct linear codes from graphs. Tonchev [9, 11, 12], for example, stated that we can define two linear codes from the following generator matrices:

(a) $G = (I, A)$,

(b) $G = A$,

where I is the identity matrix of order n. Moreover, he [9, 11, 12] showed the existence of good codes having generator matrices of type (a) above, for large enough value of n, the size of A. Recently, Dougherty et al. [2] introduced methods to construct self-dual codes from graphs. Their construction [2] may be regarded as a kind of generalization of several known methods including generalized doubly circulant method by Gaborit [4]. As results they [2] obtained several self-dual codes with good parameters.

The purpose of this note is to provide some more numerical evidence to the powerful of the construction methods of Dougherty et al. [2]. We applied the methods to construct (not necessary self-dual) linear codes. We obtained several extremal or nearly extremal codes over finite fields of modest lengths. As the codes we constructed are used for error correction in data transmission, it is important to know the performance of the resulting codes with respect to the decoding error probability. For that reason we short the codes with respect to their performance in decoding error probability. (See [8], see also [3] for detail account of decoding error probability).

2. Construction Methods

Let $G = (V, E)$ be a simple, undirected graph with $|V| = v$ and $|E| = \varepsilon$. Let A and M be an adjacency and incidence matrix of G, respectively. Define $\overline{A} := J - I - A$.

We describe the following construction. Let $\mathbb{F}_q$ be a finite field. For arbitrary scalar $r, s, t \in \mathbb{F}_q$ define

$$Q_{\mathbb{F}_q}(r, s, t) = (rI + sA + t\overline{A}).$$

We constructed linear codes by using four different methods:

1. $A_{\mathbb{F}_q}$, adjacency matrix of a graph G, as a generator matrix of a code C over $\mathbb{F}_q$.

2. $(A|M)$ as a generator matrix of a code C over $\mathbb{F}_q$.

3. $P_{\mathbb{F}_q}(r, s, t) = (I \,|\, Q_{\mathbb{F}_q}(r, s, t))$ as a generator matrix of a code C over $\mathbb{F}_q$.

4. For $\alpha, \beta, \gamma \in \mathbb{F}_q$,

$$B_{\mathbb{F}_q}(r, s, t) = \left(\begin{array}{c|c|c|c} 1 & 0 \cdots 0 & \alpha & \beta \cdots \beta \\ \hline 0 & & \gamma & \\ \vdots & I & \vdots & Q_{\mathbb{F}_q}(r, s, t) \\ 0 & & \gamma & \end{array}\right)$$

as a generator matrix of a code C over $\mathbb{F}_q$.

The last two methods above are called *pure* and *bordered construction*, respectively.

Then we have the following proposition.

Proposition 2.1. *Let G be a generator matrix of a linear code C of parameter $[n, k, d]$. Then $d \leq \min\{wt(g_i) : 1 \leq i \leq n\}$.*

As a corollary, regarding the above four constructions we have the following.

Corollary 2.1. *Let $G = (V, E)$ with $|V| = v$ and $|E| = \varepsilon$, and let $\delta(G)$ denote the smallest degree of G. Then we have*

1. *Linear code $A_{\mathbb{F}_q}$ has parameters $[v, k, d]$ with $d \leq \delta(G)$.*

2. *Linear code $AM_{\mathbb{F}_q}$ has parameters $[v + \varepsilon, k, d]$ with $d \leq \delta(G)$.*

3. *Linear code $P_{\mathbb{F}_q}(r, s, t)$ has parameters $[2v, v, d]$ with $d \leq v + 1$.*

4. *Linear code $B_{\mathbb{F}_q}(r, s, t)$ has parameters $[2v + 2, v + 1, d]$ with $d \leq v + 2$.*

All observations are focused on the Johnson graphs $J(n, k)$, with $1 \leq k \leq \left\lfloor \frac{n}{2} \right\rfloor$. As usual, the Johnson graph $J(n, k)$ is defined to be the graph with vertex set is equal to the collection of all k-subsets of $\{1, 2, ..., n\}$, and any two vertices are adjacent if they have exactly $k - 1$ elements in common.

3. Results

In this section we provide several concrete examples of extremal or nearly-extremal linear codes we obtained by the above methods. We first explain the notation we use in the tables below. In the first column, $\langle i, q, n, k, r, s, t, \alpha, \beta, \gamma \rangle$ has the following meaning:

- $i = 1$ means the codes are obtained from $A_{\mathbb{F}_q}$ of the graphs $J(n, k)$.

- $i = 2$ means the codes are obtained from $(A|M)$ of the graphs $J(n, k)$.
- $i = 3$ means the codes are obtained from pure construction of the graphs $J(n, k)$.
- $i = 4$ means the codes are obtained from bordered construction of the graphs $J(n, k)$.
- $i = 5$ means the codes are obtained from $A_{\mathbb{F}_q}$ of the graphs $LJ(n, k)$.
- $i = 6$ means the codes are obtained from $(A|M)$ of the graphs $LJ(n, k)$.
- $i = 7$ means the codes are obtained from pure construction of the graphs $LJ(n, k)$.
- $i = 8$ means the codes are obtained from bordered construction of the graphs $LJ(n, k)$.
- $i = 9$ means the codes are obtained from $A_{\mathbb{F}_q}$ of the graphs $TJ(n, k)$.
- $i = 10$ means the codes are obtained from $(A|M)$ of the graphs $TJ(n, k)$.
- $i = 11$ means the codes are obtained from pure construction of the graphs $TJ(n, k)$.
- $i = 12$ means the codes are obtained from bordered construction of the graphs $TJ(n, k)$.

We note that: "ex" and "n-ex" stand for extremal and nearly-extremal. "s-d" stands for self-dual. "T-II" stands for type II, namely the binary self-dual codes which all codewords have weight divisible by 4. "LB" and "UB" are lower bound and upper bound, respectively, obtained from [5] (see also [1]).

3.1. Linear codes over $\mathbb{F}_2$

As mentioned before, the linear codes $A_{\mathbb{F}_q}$ constructed from the Johnson graphs $J(n, k)$ have parameters $\left[\binom{n}{k}, \operatorname{rank}(A), d\right]$ with $d \leq k(n-k)$.

$\langle i, n, k, r, s, t, \alpha, \beta, \gamma\rangle$	Parameters	LB	UB	Type	A_d, A_{d+1}, A_{d+2}
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 1\rangle$	[8, 4, 3]	4	4	n-ex	3, 7, 4
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	4, 6, 4
$\langle 4, 3, 1, 0, 1, 0, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	7, 7, 0
$\langle 3, 4, 1, 0, 1, 0\rangle$	[8, 4, 4]	4	4	s-d, ex	14, 0, 0
$\langle 9, 3, 1\rangle$	[9, 4, 4]	4	4	ex	9, 0, 6
$\langle 1, 2, 5, 2\rangle$	[10, 4, 4]	4	4	ex	5, 0, 10
$\langle 4, 4, 1, 0, 1, 0, 1, 1, 1\rangle$	[10, 5, 3]	4	4	n-ex	4, 6, 8
$\langle 4, 4, 1, 0, 1, 0, 0, 1, 0\rangle$	[10, 5, 3]	4	4	n-ex	4, 14, 8
$\langle 3, 2, 5, 1, 0, 1, 0\rangle$	[10, 5, 4]	4	4	ex	10, 16, 0
$\langle 4, 4, 1, 0, 1, 0, 1, 1, 0\rangle$	[10, 5, 4]	4	4	ex	18, 0, 8
$\langle 5, 2, 4, 2\rangle$	[12, 4, 6]	6	6	ex	12, 0, 3
$\langle 4, 5, 1, 0, 1, 0, 1, 1, 1\rangle$	[12, 6, 3]	4	4	n-ex	5, 10, 0
$\langle 4, 5, 1, 0, 1, 0, 0, 1, 0\rangle$	[12, 6, 3]	4	4	n-ex	5, 10, 16
$\langle 4, 5, 1, 0, 1, 0, 1, 1, 0\rangle$	[12, 6, 4]	4	4	ex	5, 16, 0
$\langle 3, 4, 2, 0, 1, 1\rangle$	[12, 6, 4]	4	4	s-d, ex	15, 0, 32

$\langle 4, 4, 2, 0, 0, 1, 0, 1, 1\rangle$	[14, 7, 3]	4	4	n-ex	6, 15, 0
$\langle 3, 7, 1, 0, 1, 0\rangle$	[14, 7, 4]	4	4	ex	21, 0, 0
$\langle 4, 4, 2, 0, 1, 1, 1, 1, 0\rangle$	[14, 7, 4]	4	4	ex	21, 0, 32
$\langle 1, 6, 2\rangle$	[15, 4, 8]	8	8	ex	15, 0, 0
$\langle 3, 8, 1, 0, 1, 0\rangle$	[16, 8, 4]	5	5	s-d, n-ex, T-II	28, 0, 0
$\langle 4, 7, 1, 0, 1, 0, 0, 1, 1\rangle$	[16, 8, 4]	5	5	s-d, n-ex, T-II	28, 0, 0
$\langle 4, 7, 1, 0, 1, 0, 1, 1, 0\rangle$	[16, 8, 4]	5	5	n-ex	28, 0, 0
$\langle 9, 5, 1\rangle$	[25, 16, 4]	4	4	ex	100, 0, 600
$\langle 8, 4, 2, 0, 0, 1, 1, 1, 1\rangle$	[26, 13, 6]	7	7	n-ex	54, 64, 135
$\langle 8, 4, 2, 1, 1, 0, 1, 1, 0\rangle$	[26, 13, 6]	7	7	n-ex	70, 0, 471
$\langle 8, 4, 2, 0, 0, 1, 1, 1, 0\rangle$	[26, 13, 6]	7	7	n-ex	102, 0, 247
$\langle 1, 8, 2\rangle$	[28, 6, 12]	12	12	ex	28, 0, 0
$\langle 4, 6, 2, 0, 0, 1, 1, 1, 1\rangle$	[32, 16, 7]	8	8	n-ex	35, 465, 120
$\langle 4, 6, 2, 0, 0, 1, 1, 1, 0\rangle$	[32, 16, 7]	8	8	n-ex	120, 380, 120
$\langle 4, 6, 2, 0, 0, 1, 0, 1, 0\rangle$	[32, 16, 7]	8	8	n-ex	155, 465, 0
$\langle 4, 6, 2, 0, 0, 1, 0, 1, 1\rangle$	[32, 16, 8]	8	8	s-d, ex	620, 0, 0
$\langle 12, 4, 1, 0, 1, 0, 0, 1, 1\rangle$	[34, 17, 7]	8	8	n-ex	32, 300, 320
$\langle 12, 4, 1, 1, 0, 1, 0, 1, 0\rangle$	[34, 17, 7]	8	8	n-ex	32, 620, 320
$\langle 12, 4, 1, 1, 0, 1, 0, 1, 1\rangle$	[34, 17, 8]	8	8	ex	332, 640, 0
$\langle 3, 6, 3, 1, 0, 1\rangle$	[40, 20, 8]	9	10	s-d, n-ex, T-II	285, 0, 0
$\langle 3, 6, 3, 0, 1, 0\rangle$	[40, 20, 8]	9	10	s-d, n-ex	285, 0, 1024

3.2. Linear codes over $\mathbb{F}_3$

$\langle i, n, k, r, s, t, \alpha, \beta, \gamma\rangle$	Parameters	LB	UB	Type	A_d, A_{d+1}, A_{d+2}
$\langle 4, 3, 1, 1, 0, 0, 0, 1, 1\rangle$	[8, 4, 3]	4	4	n-ex	6, 10, 30
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 1\rangle$	[8, 4, 3]	4	4	n-ex	6, 12, 22
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	8, 18, 16
$\langle 4, 3, 1, 0, 1, 0, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	12, 22, 30
$\langle 3, 4, 1, 0, 1, 0\rangle$	[8, 4, 4]	4	4	n-ex	22, 24, 20
$\langle 3, 4, 1, 1, 2, 0\rangle$	[8, 4, 4]	4	4	n-ex	24, 16, 32
$\langle 2, 4, 1\rangle$	[10, 4, 6]	6	6	ex	60, 0, 0
$\langle 3, 5, 1, 0, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	20, 20, 80
$\langle 3, 5, 1, 1, 2, 0\rangle$	[10, 5, 4]	5	5	n-ex	20, 22, 70
$\langle 4, 4, 1, 0, 1, 0, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	30, 24, 54
$\langle 4, 4, 1, 1, 2, 0, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	32, 16, 64
$\langle 9, 4, 1\rangle$	[16, 9, 4]	5	5	n-ex	72, 0, 288
$\langle 11, 3, 1, 0, 0, 1\rangle$	[18, 9, 5]	6	6	n-ex	54, 72, 36
$\langle 11, 3, 1, 1, 0, 1\rangle$	[18, 9, 6]	6	6	ex	108, 144, 486
$\langle 12, 3, 1, 1, 0, 1, 0, 1, 1\rangle$	[20, 10, 6]	7	7	n-ex	72, 168, 396
$\langle 12, 3, 1, 1, 0, 1, 1, 1, 0\rangle$	[20, 10, 6]	7	7	n-ex	108, 252, 510
$\langle 12, 3, 1, 0, 0, 1, 0, 1, 1\rangle$	[20, 10, 6]	7	7	n-ex	126, 78, 324
$\langle 12, 3, 1, 0, 0, 1, 1, 1, 1\rangle$	[20, 10, 6]	7	7	n-ex	126, 90, 312
$\langle 3, 5, 2, 0, 1, 0\rangle$	[20, 10, 6]	7	7	n-ex	150, 40, 300
$\langle 3, 5, 2, 1, 1, 2\rangle$	[20, 10, 6]	7	7	n-ex	180, 0, 240
$\langle 12, 3, 1, 1, 0, 1, 0, 1, 0\rangle$	[20, 10, 6]	7	7	n-ex	216, 168, 702

$\langle 1, 6, 3\rangle$	[20, 13, 4]	5	5	n-ex	90, 0, 1560
$\langle 1, 7, 2\rangle$	[21, 15, 4]	4	4	ex	210, 420, 6244
$\langle 2, 6, 1\rangle$	[21, 6, 10]	11	11	n-ex	42, 42, 140
$\langle 1, 10, 2\rangle$	[45, 36, 4]	5	6	n-ex	1260, 1890, 38220

3.3. Linear codes over $\mathbb{F}_4$

$\langle i, n, k, r, s, t, \alpha, \beta, \gamma\rangle$	Parameters	LB	UB	Type	A_d, A_{d+1}, A_{d+2}
$\langle 6, 3, 1\rangle$	[6, 2, 4]	4	4	ex	9, 0, 6
$\langle 4, 3, 1, 1, w, 1, 1, 1, w\rangle$	[8, 4, 3]	4	4	n-ex	9, 9, 78
$\langle 4, 3, 1, 1, 0, 1, 1, 1, 1\rangle$	[8, 4, 3]	4	4	n-ex	9, 21, 30
$\langle 4, 3, 1, 1, w, 1, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	9, 69, 54
$\langle 4, 3, 1, 0, 1, 1, 1, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	12, 18, 48
$\langle 4, 3, 1, 0, 1, 1, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	21, 21, 126
$\langle 4, 3, 1, 1, w, 1, 1, 1, w^2\rangle$	[8, 4, 4]	4	4	ex	18, 96, 24
$\langle 4, 3, 1, w, w^2, 1, w, w^2, w^2\rangle$	[8, 4, 4]	4	4	s-d, ex	18, 96, 24
$\langle 4, 3, 1, 1, w, 1, 1, 1, 1\rangle$	[8, 4, 4]	4	4	ex	30, 48, 96
$\langle 4, 3, 1, 0, 1, 1, 0, 1, 1\rangle$	[8, 4, 4]	4	4	s-d, ex	42, 0, 168
$\langle 4, 3, 1, 1, w, 1, 1, 1, 0\rangle$	[8, 4, 4]	4	4	ex	54, 24, 72
$\langle 9, 3, 1\rangle$	[9, 4, 4]	5	5	n-ex	27, 0, 54
$\langle 3, 5, 1, 1, w, 1\rangle$	[10, 5, 4]	5	5	n-ex	30, 0, 300
$\langle 4, 4, 1, 1, w, 1, 1, 1, 1\rangle$	[10, 5, 4]	5	5	n-ex	30, 24, 180
$\langle 3, 5, 1, 0, 1, 1\rangle$	[10, 5, 4]	5	5	n-ex	30, 48, 60
$\langle 4, 4, 1, 1, w, 1, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	30, 96, 108

$\langle 4, 4, 1, 0, 1, 1, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	54, 0, 228
$\langle 5, 4, 2\rangle$	[12, 4, 6]	7	7	n-ex	36, 0, 9
$\langle 11, 3, 1, 0, 1, w\rangle$	[18, 9, 7]	8	8	n-ex	432, 594, 5520
$\langle 12, 3, 1, 1, w, 0, 1, 1, 1\rangle$	[20, 10, 7]	8	8	n-ex	72, 1206, 2160
$\langle 12, 3, 1, 1, w, w^2, w, w^2, w^2\rangle$	[20, 10, 7]	8	8	n-ex	108, 1170, 1944
$\langle 12, 3, 1, 1, w, 0, 1, 1, w\rangle$	[20, 10, 7]	8	8	n-ex	144, 702, 4320
$\langle 12, 3, 1, 1, w, w^2, 1, 1, w\rangle$	[20, 10, 7]	8	8	n-ex	180, 666, 4104
$\langle 12, 3, 1, 0, 1, w, 1, 1, 0\rangle$	[20, 10, 7]	8	8	n-ex	432, 846, 5952
$\langle 12, 3, 1, 0, 1, w, 0, 1, 0\rangle$	[20, 10, 7]	8	8	n-ex	684, 1026, 10488
$\langle 12, 3, 1, 0, 1, w^2, 0, w^2, 0\rangle$	[20, 10, 7]	8	8	n-ex	684, 1026, 10488
$\langle 12, 3, 1, 0, 1, w, 0, 1, 1\rangle$	[20, 10, 8]	8	8	n-ex	1710, 0, 20976
$\langle 4, 5, 2, 1, w, 0, 0, 1, 1\rangle$	[22, 11, 7]	8	9	n-ex	90, 780, 2130
$\langle 4, 5, 2, 0, 1, w, 0, 1, 1\rangle$	[22, 11, 7]	8	9	n-ex	150, 720, 1950
$\langle 4, 5, 2, 0, 1, w^2, 0, w^2, w^2\rangle$	[22, 11, 7]	8	9	n-ex	150, 720, 1950
$\langle 4, 5, 2, 1, w, 0, 1, 1, 1\rangle$	[22, 11, 7]	8	9	n-ex	180, 690, 1860
$\langle 4, 5, 2, 1, w^2, 0, w, w^2, w^2\rangle$	[22, 11, 7]	8	9	n-ex	180, 690, 1860
$\langle 7, 4, 2, 0, 1, w\rangle$	[24, 12, 8]	9	9	n-ex	981, 768, 5184

3.4. Linear codes over $\mathbb{F}_5$

$\langle i, n, k, r, s, t, \alpha, \beta, \gamma\rangle$	Parameters	LB	UB	Type	A_d, A_{d+1}, A_{d+2}
$\langle 4, 3, 1, 1, 2, 0, 1, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	4, 60, 64
$\langle 4, 3, 1, 1, 0, 0, 0, 1, 1\rangle$	[8, 4, 3]	4	4	n-ex	12, 16, 100

$\langle 4, 3, 1, 1, 0, 0, 1, 1, 4\rangle$	[8, 4, 3]	4	4	n-ex	12, 12, 116
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	12, 36, 80
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 1\rangle$	[8, 4, 3]	4	4	n-ex	12, 24, 68
$\langle 4, 3, 1, 1, 2, 0, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	16, 76, 180
$\langle 4, 3, 1, 1, 3, 0, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	12, 92, 156
$\langle 4, 3, 1, 0, 1, 0, 0, 1, 0\rangle$	[8, 4, 3]	4	4	n-ex	24, 44, 228
$\langle 3, 4, 1, 1, 3, 0\rangle$	[8, 4, 4]	4	4	ex	28, 112, 168
$\langle 4, 3, 1, 1, 2, 0, 0, 1, 1\rangle$	[8, 4, 4]	4	4	ex	32, 96, 192
$\langle 4, 3, 1, 0, 1, 0, 1, 1, 2\rangle$	[8, 4, 4]	4	4	ex	36, 80, 216
$\langle 3, 4, 1, 0, 1, 0\rangle$	[8, 4, 4]	4	4	ex	40, 64, 240
$\langle 3, 4, 1, 1, 4, 0\rangle$	[8, 4, 4]	4	4	s-d, ex	48, 32, 288
$\langle 3, 4, 1, 2, 3, 0\rangle$	[8, 4, 4]	4	4	ex	48, 32, 288
$\langle 4, 3, 1, 1, 3, 0, 1, 1, 0\rangle$	[8, 4, 4]	4	4	ex	72, 56, 172
$\langle 2, 4, 1\rangle$	[10, 4, 6]	6	6	ex	100, 80, 240
$\langle 4, 4, 1, 1, 3, 0, 0, 1, 1\rangle$	[10, 5, 4]	5	5	s-d, n-ex	40, 8, 400
$\langle 4, 4, 1, 1, 3, 0, 0, 1, 2\rangle$	[10, 5, 4]	5	5	n-ex	40, 8, 400
$\langle 4, 4, 1, 0, 1, 0, 0, 1, 1\rangle$	[10, 5, 4]	5	5	n-ex	40, 20, 340
$\langle 4, 4, 1, 1, 3, 0, 1, 1, 2\rangle$	[10, 5, 4]	5	5	n-ex	40, 28, 300
$\langle 4, 4, 1, 0, 1, 0, 1, 1, 4\rangle$	[10, 5, 4]	5	5	n-ex	40, 32, 280
$\langle 4, 4, 1, 0, 1, 0, 1, 1, 3\rangle$	[10, 5, 4]	5	5	n-ex	40, 40, 240
$\langle 4, 4, 1, 1, 3, 0, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	44, 112, 300
$\langle 4, 4, 1, 0, 1, 0, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	56, 64, 360
$\langle 4, 4, 1, 1, 4, 0, 1, 1, 0\rangle$	[10, 5, 4]	5	5	n-ex	64, 32, 400

$\langle 11, 3, 1, 1, 0, 2\rangle$	[18, 9, 6]	7	8	n-ex	72, 0, 2124
$\langle 11, 3, 1, 1, 3, 4\rangle$	[18, 9, 6]	7	8	n-ex	96, 0, 2232
$\langle 11, 3, 1, 2, 1, 3\rangle$	[18, 9, 6]	7	8	s-d, n-ex	96, 0, 2232
$\langle 11, 3, 1, 1, 1, 2\rangle$	[18, 9, 6]	7	8	n-ex	168, 144, 1440
$\langle 11, 3, 1, 1, 0, 1\rangle$	[18, 9, 6]	7	8	n-ex	180, 144, 1368
$\langle 11, 3, 1, 0, 1, 4\rangle$	[18, 9, 7]	7	8	n-ex	576, 792, 6160
$\langle 7, 5, 1, 1, 3, 4\rangle$	[20, 10, 7]	8	9	n-ex	40, 1500, 2400
$\langle 12, 3, 1, 0, 1, 4, 1, 1, 1\rangle$	[20, 10, 7]	8	9	n-ex	72, 1632, 1908
$\langle 12, 3, 1, 0, 1, 4, 1, 1, 2\rangle$	[20, 10, 7]	8	9	n-ex	96, 1608, 2016
$\langle 12, 3, 1, 1, 0, 2, 1, 1, 2\rangle$	[20, 10, 7]	8	9	n-ex	144, 984, 3816
$\langle 12, 3, 1, 1, 0, 2, 1, 1, 3\rangle$	[20, 10, 7]	8	9	n-ex	168, 960, 3924
$\langle 12, 3, 1, 1, 3, 4, 1, 1, 2\rangle$	[20, 10, 7]	8	9	n-ex	192, 936, 4032
$\langle 12, 3, 1, 0, 1, 4, 1, 1, 0\rangle$	[20, 10, 7]	8	9	n-ex	576, 1128, 6736
$\langle 12, 3, 1, 0, 1, 4, 0, 1, 0\rangle$	[20, 10, 7]	8	9	n-ex	912, 1368, 11704
$\langle 7, 5, 1, 0, 1, 4\rangle$	[20, 10, 8]	8	9	s-d, n-ex	2280, 0, 23408
$\langle 7, 5, 1, 0, 2, 3\rangle$	[20, 10, 8]	8	9	n-ex	2280, 0, 23408
$\langle 7, 4, 2, 1, 4, 0\rangle$	[24, 12, 8]	9	10	n-ex	660, 1200, 6960
$\langle 5, 5, 2\rangle$	[30, 26, 3]	3	3	ex	360, 11280, 234984

3.5. Linear codes over $\mathbb{F}_7$

$\langle i, n, k, r, s, t, \alpha, \beta, \gamma\rangle$	Parameters	LB	UB	Type	A_d, A_{d+1}, A_{d+2}
$\langle 4, 3, 1, 1, 2, 0, 1, 1, 3\rangle$	[8, 4, 4]	5	5	n-ex	36, 192, 552
$\langle 4, 3, 1, 1, 2, 0, 0, 1, 1\rangle$	[8, 4, 4]	5	5	n-ex	42, 168, 588

⟨4, 3, 1, 1, 3, 0, 0, 1, 1⟩	[8, 4, 4]	5	5	n-ex	48, 144, 624
⟨4, 3, 1, 0, 1, 0, 1, 1, 2⟩	[8, 4, 4]	5	5	n-ex	54, 120, 660
⟨4, 3, 1, 0, 1, 0, 0, 1, 1⟩	[8, 4, 4]	5	5	n-ex	60, 96, 696
⟨4, 3, 1, 0, 1, 0, 1, 1, 4⟩	[8, 4, 4]	5	5	n-ex	72, 48, 768
⟨4, 3, 1, 1, 2, 0, 1, 1, 0⟩	[8, 4, 4]	5	5	n-ex	108, 156, 522
⟨2, 4, 1⟩	[10, 4, 6]	6	6	ex	150, 120, 900
⟨3, 5, 1, 1, 4, 0⟩	[10, 5, 4]	5	5	n-ex	60, 0, 780
⟨3, 5, 1, 1, 5, 0⟩	[10, 5, 4]	5	5	n-ex	60, 6, 750
⟨4, 4, 1, 1, 2, 0, 0, 1, 1⟩	[10, 5, 4]	5	5	n-ex	60, 12, 720
⟨4, 4, 1, 0, 1, 0, 1, 1, 2⟩	[10, 5, 4]	5	5	n-ex	60, 24, 660
⟨4, 4, 1, 1, 2, 0, 1, 1, 5⟩	[10, 5, 4]	5	5	n-ex	60, 42, 570
⟨4, 4, 1, 0, 1, 0, 1, 1, 3⟩	[10, 5, 4]	5	5	n-ex	60, 48, 540
⟨3, 5, 1, 0, 1, 0⟩	[10, 5, 4]	5	5	n-ex	60, 30, 630
⟨4, 4, 1, 1, 4, 0, 1, 1, 6⟩	[10, 5, 4]	5	5	n-ex	60, 36, 600
⟨3, 5, 1, 1, 3, 0⟩	[10, 5, 4]	5	5	n-ex	60, 60, 480
⟨4, 4, 1, 1, 4, 0, 1, 1, 0⟩	[10, 5, 4]	5	5	n-ex	60, 192, 828
⟨4, 4, 1, 1, 2, 0, 1, 1, 0⟩	[10, 5, 4]	5	5	n-ex	66, 168, 858
⟨4, 4, 1, 0, 1, 0, 1, 1, 0⟩	[10, 5, 4]	5	5	n-ex	84, 96, 948
⟨4, 4, 1, 1, 6, 0, 1, 1, 0⟩	[10, 5, 4]	5	5	n-ex	96, 48, 1008
⟨11, 3, 1, 0, 1, 6⟩	[18, 9, 7]	8	8	n-ex	432, 972, 13776
⟨11, 3, 1, 1, 3, 4⟩	[18, 9, 7]	8	8	n-ex	432, 972, 13860
⟨3, 5, 2, 1, 4, 5⟩	[20, 10, 8]	9	9	n-ex	1350, 3600, 44592
⟨3, 5, 2, 1, 4, 5⟩	[20, 10, 8]	9	9	n-ex	1350, 3600, 44592

$\langle 3, 5, 2, 1, 2, 4\rangle$	[20, 10, 8]	9	9	n-ex	1470, 4860, 34890
$\langle 3, 5, 2, 1, 5, 6\rangle$	[20, 10, 8]	9	9	n-ex	1530, 2880, 45696
$\langle 3, 5, 2, 1, 2, 0\rangle$	[20, 10, 8]	9	9	n-ex	1650, 2880, 43290
$\langle 3, 5, 2, 0, 1, 6\rangle$	[20, 10, 8]	9	9	n-ex	2340, 0, 49152
$\langle 3, 5, 2, 1, 3, 4\rangle$	[20, 10, 8]	9	9	n-ex	2340, 0, 49860
$\langle 1, 8, 3\rangle$	[56, 49, 4]	5	6	n-ex	12600, 23520, 2629872

4. Concluding Remarks

We obtained many linear codes with good parameters over finite fields of modest lengths. It will be good to apply the methods to construct many more linear codes with good parameters for other lengths. As done in [2], the methods apply to construct good self-dual codes over finite rings as well. It will be beneficial to apply the methods to construct good self-dual codes over finite rings other than the ones considered in [2] as well as the same finite rings for other lengths.

Acknowledgements

The research is supported by Riset Desentralisasi ITB-DIKTI tahun 2015-2016. It has been initiated by partial support from Riset dan Inovasi KK ITB Tahun 2014-2015.

References

[1] A. E. Brouwer, Bounds on the size of linear codes, V. S. Pless and W. C. Huffman, eds., Handbook of Coding Theory, Vol. I, Elsevier, Amsterdam, 1998, pp. 295-461.

[2] S. T. Dougherty, J.-L. Kim and P. Solé, Double circulant codes from two class association schemes, Adv. Math. Commun. 1 (2007), 45-64.

[3] A. Faldum, J. Lafuente, G. Ochoa and W. Willems, Error probabilities for bounded distance decoding, Des. Codes. Cryptogr. 40(2) (2006), 237-252.

[4] P. Gaborit, Quadratic double circulant codes over fields, J. Combin. Theory Ser. A 97 (2002), 85-107.

[5] M. Grassl, Code tables: bounds on the parameters of various types of codes (http://www.codetables.de/)

[6] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-correcting Codes, North-Holland, 1993.

[7] E. M. Rains and N. J. A. Sloane, Self-dual codes, V. S. Pless and W. C. Huffman, eds., The Handbook of Coding Theory, Vol. I, Elsevier, Amsterdam, 1998, pp. 177-294.

[8] D. Suprijanto and Pratikto, A note on new self-dual [28, 14, 10] codes over *GF*(7), Appl. Math. Sci. 8 (2014), 3737-3745.

[9] V. D. Tonchev, Binary codes derived from the Hoffman-Singleton and Higman-Sims graphs, IEEE Inform. Theory 43 (1997), 1021-1025.

[10] V. D. Tonchev, Codes and designs, V. S. Pless and W. C. Huffman, eds., The Handbook of Coding Theory, Vol. II, Elsevier, Amsterdam, 1998, pp. 1229-1267.

[11] V. D. Tonchev, Error-correcting codes from graphs, Discrete Math. 257 (2002), 549-557.

[12] V. D. Tonchev, A Varshamov-Gilbert bound for a class of formally self-dual codes and related quantum codes, IEEE Trans. Inform. Theory 48 (2002), 975-977.