On extremal self-dual $\mathbb{Z}_4$-codes

Matteo Mravić
matteo.mravic@math.uniri.hr

Department of Mathematics, University of Rijeka, Croatia

16.7.2021.

A joint work with Sanja Rukavina

This work has been fully supported by Croatian Science Foundation under the project 6732

The basics

- A binary linear $[n, k]$ code is a $k-$dimensional subspace of $\mathbb{F}_2^n$,
- The Hamming *weight* of a vector $x \in \mathbb{F}_2^n$ is the number of nonzero coordinates in x,
- Binary linear codes for which all codewords have weight divisible by four are called *doubly-even*,
- If the minimum weight d of an $[n, k]$ binary code is known, then we refer to the code as an $[n, k, d]$ binary code,
- The dual code of a binary linear code C of length n is

$$C^{\perp} = \{x \in \mathbb{F}_2^n \mid \langle x, y \rangle = 0 \text{ for all } y \in C\},$$

- C is *self-orthogonal* if $C \subseteq C^{\perp}$, and *self-dual* if $C = C^{\perp}$,

The basics

- A $\mathbb{Z}_4$-*code* C of length n is a $\mathbb{Z}_4$ submodule of $\mathbb{Z}_4^n$.
- Every $\mathbb{Z}_4$ code C contains a set of $k_1 + k_2$ codewords $\{c_1, \ldots, c_{k_1}, c_{k_1+1}, \cdots, c_{k_1+k_2}\}$ such that every codeword in C is uniquely expressible in the form

$$\sum_{i=1}^{k_1} a_i c_i + \sum_{i=k_1+1}^{k_1+k_2} a_i c_i,$$

  where $a_i \in \mathbb{Z}_4$ for $1 \leq i \leq k_1$ and $a_i \in \mathbb{Z}_2$ for $k_1 + 1 \leq i \leq k_1 + k_2$. We say that C is of *type* $4^{k_1} 2^{k_2}$.
- The matrix whose rows are c_i, $1 \leq i \leq k_1 + k_2$, is called a *generator matrix* for C.

The basics

A generator matrix G of a $\mathbb{Z}_4$ code C is in *standard form* if

$$G = \begin{bmatrix} I_{k_1} & A & B_1 + 2B_2 \\ O & 2I_{k_2} & 2D \end{bmatrix},$$

where A, B_1, B_2 and D are matrices with entries from $\mathbb{F}_2$ and O is the $k_2 \times k_1$ zero matrix.
For a $\mathbb{Z}_4$-code C of length n and $x = (x_1, x_2, \dots, x_n)$ we define the Euclidean weight as $wt_E(x) = n_1(x) + 4n_2(x) + n_3(x)$ where $n_i(x) = |\{x_j | x_j = i, j \in \{1, 2, \dots, n\}\}|$, $i = 1, 2, 3$.

The basics

Let C be a $\mathbb{Z}_4$ code of length n. The *dual code* $C^\perp$ of C is defined as

$$C^\perp = \{x \in \mathbb{Z}_4^n \mid \langle x, y \rangle = 0 \text{ for all } y \in C\},$$

where $\langle x, y \rangle = x_1 y_1 + \cdots + x_n y_n \pmod 4$ for $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$. The code C is *self-dual* if $C = C^\perp$.
For every $\mathbb{Z}_4$ code C there are following binary codes associated with C:

- Residue code: $Res(C) = \{c \pmod 2 \mid c \in C\}$,
- Torsion code: $Tor(C) = \{c \in \mathbb{F}_2^n | 2c \in C\}$.

If C has a generator matrix G in standard form then, $Res(C)$ and $Tor(C)$ have generator matrices

$$G_{Res} = \begin{bmatrix} I_{k_1} & A & B_1 \end{bmatrix},$$
$$G_{Tor} = \begin{bmatrix} I_{k_1} & A & B_1 \\ O & I_{k_2} & D \end{bmatrix}.$$

Construction theorem

Theorem[1]

Let C be a $\mathbb{Z}_4$-code with generating matrix in standard form

$$G = \begin{bmatrix} I_{k_1} & A & B_1 + 2B_2 \\ O & 2I_{k_2} & 2D \end{bmatrix}.$$

Code C is self-dual if and only if $Res(C)$ is doubly even, $Res(C) = Tor(C)^{\perp}$ and B_2 is such that rows of G are orthogonal.

[1] Pless, V., Leon, J., Fields, J. (1997). All Z4 Codes of Type II and Length 16 Are Known. J. Comb. Theory, Ser. A, 78, 32-50.

Types and extremality of self-dual $\mathbb{Z}_4$ codes

Definition

Let C be a self-dual $\mathbb{Z}_4$ code. We say that C is Type II if all Euclidean weights of words in C are multiples of 8. Otherwise we say that C is Type I $\mathbb{Z}_4$ code.

Theorem[2]

Let C be a self-dual $\mathbb{Z}_4$ code of length n. The following hold:

(i) If C is Type II, then the minimum Euclidean weight of C is at most $8\left\lfloor \frac{n}{24} \right\rfloor + 8$.

(ii) If C is Type I, then the minimum Euclidean weight of C is at most $8\left\lfloor \frac{n}{24} \right\rfloor + 8$ except when $n \equiv 23 \pmod{24}$, in which case the bound is $8\left\lfloor \frac{n}{24} \right\rfloor + 12$. If equality holds in this latter bound, then C is obtained by shortening a Type II code of length $n+1$.

Codes meeting these bounds are called Euclidean-extremal.

[2] W. C. Huffman and V. Pless, Fundamentals of Error-Correcting Codes. Cambridge: Cambridge University Press, 2003.

Brute-force algorithm

It is known that standard form of a generator matrix of a $\mathbb{Z}_4$-code is equivalent to the matrix of the form:

$$G = \begin{bmatrix} F & I_k + 2B \\ 2H & O \end{bmatrix},$$

where F, B, H are matrices over $\mathbb{F}_2$, I_k is $k \times k$ identity matrix and O is zero matrix. In this form the $Res(C)$ and $Tor(C)$ have following generator matrices:

$$G_{Res} = \begin{bmatrix} F & I_k \end{bmatrix},$$
$$G_{Tor} = \begin{bmatrix} F & I_k \\ H & O \end{bmatrix}.$$

Brute-force algorithm

By the construction theorem, in order to obtain a self-dual $\mathbb{Z}_4$-code, one must choose entries in $B = [b_{ij}]$ s.t. rows of G are orthogonal. This gives the following condition:

$$b_{ij} = \begin{cases} b_{ji},\ f_i f_j \equiv 0 \pmod 4, \\ b_{ji} + 1,\ f_i f_j \equiv 2 \pmod 4. \end{cases}$$

So, elements in the lower triangle of B are uniquely determined by the upper triangle elements of B and the inner product of rows in the matrix F. The brute force algorithm consists of checking the extremality of all possible $2^{\frac{k(k-1)}{2}}$ codes obtained from different choices of B.
Problem: Size of the search space, calculating the minimum Euclidean weight of Type I codes is time consuming even for small lengths.

Modification lemma

Lemma

Let C be a $\mathbb{Z}_4$-code of length n with generator matrix in form:

$$G_C = \begin{bmatrix} F & I_k + 2B \\ 2H & O \end{bmatrix}.$$

Let $B' \in M_k(\mathbb{F}_2)$ be the matrix obtained from B by changing a position (i,j), $1 < i < j < k$, from 0 to 1, in such way that the code C' with the generator matrix:

$$G_{C'} = \begin{bmatrix} F & I_k + 2B' \\ 2H & O \end{bmatrix},$$

is self-dual. Let $v \in C$ be of the form:

$$v = c_i g_i + c_j g_j + \sum_{\substack{m=1 \\ m \neq i,j}}^{k} c_m g_m + \sum_{m=k+1}^{n-2k} c_m g_m,$$

where g_s, $s \in \{1,2,\dots,n-2k\}$, is the s-th row of the matrix G_C. Let $I = \left\{t \in \{1,2,\dots,k\} - \{i,j\} \,\middle|\, (c_t g_t)_i = 2\right\}$, and $J = \left\{t \in \{1,2,\dots,k\} - \{i,j\} \,\middle|\, (c_t g_t)_j = 2\right\}$, where $(c_t g_t)_i$ and $(c_t g_t)_j$ stand for the i-th and j-th coordinate of the codeword $c_t g_t$ respectively. Let $v' \in C'$ be:

$$v' = c_i g_i' + c_j g_j' + \sum_{\substack{m=1 \\ m \neq i,j}}^{k} c_m g_m + \sum_{m=k+1}^{n-2k} c_m g_m.$$

Then $d_E(v') = d_E(v) + r$, where $r = 0$ for all c_i and c_j except those given in the following Table.

$B(i,j) = B(j,i)$				
c_i	c_j	$\lvert I \rvert \pmod 2$	$\lvert J \rvert \pmod 2$	r
0	1,3	0	x	4
		1	x	-4
1,3	0	x	0	4
		x	1	-4
2	1,3	0	x	-4
		1	x	4
1,3	2	x	0	-4
		x	1	4

$B(i,j) \neq B(j,i)$				
c_i	c_j	$\lvert I \rvert \pmod 2$	$\lvert J \rvert \pmod 2$	r
0	1,3	0	x	-4
		1	x	4
1,3	0	x	0	-4
		x	1	4
2	1,3	0	x	4
		1	x	-4
1,3	2	x	0	4
		x	1	-4

Table: Changes of weights in the modification Lemma

Small example

$$G=\begin{bmatrix} 1&1&1&0&1&1&1&0&1&0&1&0&0&0&0&0\\ 1&0&0&1&1&1&1&1&0&1&0&1&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&1&2&2&1&0&0&0\\ 0&1&1&1&1&0&0&1&1&1&0&0&2&1&0&0\\ 0&1&1&1&0&0&1&1&1&1&0&0&2&2&1&0\\ 0&0&0&0&0&1&0&0&1&1&2&2&2&2&2&1\\ 0&2&2&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&2&0&0&0&2&0&0&0&0&0&0&0&0\\ 2&2&0&0&0&2&0&0&2&0&0&0&0&0&0&0\\ 2&0&0&2&0&2&0&0&0&2&0&0&0&0&0&0 \end{bmatrix},$$

$$v=(3221333121210000)=2g_1+g_2,\ wt_E(v)=24$$

Small example

$$G'=\begin{bmatrix} 1&1&1&0&1&1&1&0&1&0&1&2&0&0&0&0\\ 1&0&0&1&1&1&1&1&0&1&2&1&0&0&0&0\\ 1&0&0&0&0&0&0&0&1&1&2&2&1&0&0&0\\ 0&1&1&1&1&0&0&1&1&1&0&0&2&1&0&0\\ 0&1&1&1&0&0&1&1&1&1&0&0&2&2&1&0\\ 0&0&0&0&0&1&0&0&1&1&2&2&2&2&2&1\\ 0&2&2&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&2&0&0&0&2&0&0&0&0&0&0&0&0\\ 2&2&0&0&0&2&0&0&2&0&0&0&0&0&0&0\\ 2&0&0&2&0&2&0&0&0&2&0&0&0&0&0&0 \end{bmatrix},$$

$$v=(3221333121210000)=2g_1+g_2,\ wt_E(v)=24$$
$$v'=(3221333121010000)=2g_1'+g_2',\ wt_E(v)=20.$$

Modified search algorithm

We say that two matrices B and B' are neighbors if their upper diagonal elements differ in exactly one element. The method of generating a self-dual $\mathbb{Z}_4$-code is unchanged and consists of choosing lower diagonal elements of matrix B as previously explained.

- Start with the matrix B s.t. all upper diagonal elements are equal to 0,
- In each iteration of the algorithm do the following:
  - Generate a $\mathbb{Z}_4$-code with the chosen matrix B, and set $D = \left|\left\{v \in C | 0 < wt_E(v) < 8\left\lfloor\frac{n}{24}\right\rfloor + 8\right\}\right|$,
  - If $D = 0$ then C is extremal,
  - Calculate sets:

$$\begin{aligned}
S_4 &= \{v \in C | wt_E(v) = 4\}, \\
S_{E-4} &= \left\{v \in C | wt_E(v) = 8\left\lfloor\frac{n}{24}\right\rfloor + 4\right\}, \\
S_E &= \left\{v \in C | wt_E(v) = 8\left\lfloor\frac{n}{24}\right\rfloor + 8\right\},
\end{aligned}$$

  - For every upper diagonal element of matrix B which is equal to 0 calculate the neighbor B' which have that element equal to 2. If B' is unchecked, using the modification lemma, calculate following numbers:

$$\begin{aligned}
d_4 &= |\{v \in S_4 | wt_E(v') \text{ changes by -4}\}|, \\
d_{E-4} &= |\{v \in S_{E-4} | wt_E(v') \text{ changes by +4}\}|, \\
d_E &= |\{v \in S_E | wt_E(v') \text{ changes by -4}\}|, \\
d &= D - d_4 - d_{E-4} + d_E,
\end{aligned}$$

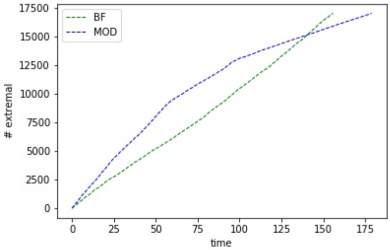
  - All B' that have $d = 0$ are extremal,
  - Mark all neighbors of B as checked,
  - Repeat the process with the first unchecked matrix B.

Comparing the algorithms

We tested the algorithm on the code of length 16, with a $[16, 6, 4]$ residue code generated with:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Comparing the algorithms



- Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz processor, and 16GB RAM memory with frequency 2400MHz, MAGMA,
- Brute force: 155.844s,
- Modified: 200.860s,
- Up until 127.438s of the execution, the modified algorithm was better,
- Worsen over time due to the exploit of unchecked neighbors,
- Due to the vast search space, this can never happen for codes of bigger lengths and with larger dimension of the residue code.

Self-dual $\mathbb{Z}_4$-codes of length 32

Code	[n,k,d]	0	4	8	12	16	20	24	28	32
C_1	$[32,6,16]$	1				62				1
C_2, C_7	$[32,9,8]$	1		28		454		28		1
C_3	$[32,12,4]$	1	28	84	420	3030	420	84	28	1
C_4	$[32,15,4]$	1	56	924	3976	22854	3976	924	56	1
C_5	$[32,9,4]$	1	7		49	398	49		7	1
C_6	$[32,15,4]$	1	42	560	5558	20446	5558	560	42	1
C_8	$[32,10,4]$	1	14	4	98	790	98	4	14	1
C_9, C_{13}	$[32,16,4]$	1	56	1180	11144	40774	11144	1180	56	1
C_{10}	$[32,7,8]$	1		4		118		4		1
C_{11}	$[32,10,8]$	1		32	112	734	112	32		1
C_{12}	$[32,13,4]$	1	28	228	868	5942	868	228	28	1
C_{14}	$[32,10,8]$	1		60		902		60		1
C_{15}	$[32,10,4]$	1	8	28	56	838	56	28	8	1
C_{16}	$[32,16,4]$	1	120	1820	8008	45638	8008	1820	120	1
C_{17}	$[32,7,4]$	1	1		7	110	7		1	1
C_{18}	$[32,10,4]$	1	8	7	140	712	140	7	8	1
C_{19}	$[32,13,4]$	1	36	196	924	5878	924	196	36	1
C_{20}	$[32,10,4]$	1	1	42	63	810	63	42	1	1
C_{21}	$[32,16,4]$	1	50	1120	11438	40318	11438	1120	50	1

Table: Weight distributions of self-orthogonal binary codes $C_1, \dots, C_{21}$

Ban, S., Crnkovic, D., Mravic, M., Rukavina, S., New extremal Type II $\mathbb{Z}_4$-codes of length 32 obtained from Hadamard matrices: Discrete Mathematics, Algorithms and Applications, 2019.

Extremal $\mathbb{Z}_4$-codes obtained by the random search (BF)

The binary code	The number of obtained extremal $\mathbb{Z}_4$ codes	The type	E_{16}	The binary residue code
C_1	118	$4^6 2^{20}$	128216	[32,6,16]
C_2	114	$4^9 2^{14}$	120152	[32,9,8]
C_7	91	$4^9 2^{14}$	120152	[32,9,8]
C_{10}	296	$4^7 2^{18}$	123608	[32,7,8]
C_{14}	304	$4^{10} 2^{12}$	119576	[32,10,8]

Table: Extremal Type II $\mathbb{Z}_4$ codes from $C_1, \ldots, C_{21}$

- Codes of type $4^6 2^{20}$ are known and all equivalent
- Only known code of type $4^7 2^{18}, 4^9 2^{14}, 4^{10} 2^{12}$ have residue code of $d = 4 \Rightarrow$ new codes.

Extremal $\mathbb{Z}_4$-codes obtained by random search (BF)

The binary code	The number of obtained extremal $\mathbb{Z}_4$ codes	At least non equivalent	The type	The binary residue code
C_3	13	10	$4^{12}2^{12}$	[32,12,4]
C_4	6	6	$4^{15}2^{2}$	[32,15,4]
C_8	35	2	$4^{10}2^{12}$	[32,10,4]
C_{12}	5	5	$4^{13}2^{10}$	[32,13,4]
C_{15}	210	2	$4^{10}2^{12}$	[32,10,4]
C_{16}	272	240	$4^{16}2^{0}$	[32,16,4]
C_{18}	44	1	$4^{10}2^{12}$	[32,10,4]
C_{19}	188	177	$4^{13}2^{10}$	[32,13,4]

Table: Extremal Type I $\mathbb{Z}_4$ codes from $C_1, \ldots, C_{21}$

- In Asamov table of Z_4 code there are 2 codes of type $4^{16}2^{0}$,
- Other codes aren't in Asamov table

Extremal $\mathbb{Z}_4$-codes obtained by the modified algorithm

We obtained extremal $\mathbb{Z}_4$-codes with residue codes C_5 and C_{11}:

- C_5, $[32, 9, 4]$, $A_4 = 7$:
    - $4^9 2^{14}$
    - In total 1664 codes are obtained,
    - At least 3 nonequivalent codes, of which one is Type II, and two Type I,
    - Only known $4^9 2^{12}$ Type II code[3] have the residue code with $A_4 = 6$, therefore the obtained Type II code is new,
    - Type I codes are not in the Asamov table.
- C_{11}, $[32, 10, 8]$, $A_4 = 0$:
    - $4^{10} 2^{12}$
    - In total 4800 codes are obtained,
    - At least 3 nonequivalent codes, of which one is Type II, and two Type I,
    - Only known $4^{10} 2^{12}$ Type II code[3] have the residue code with $A_4 = 10$, therefore the obtained Type II code is new,
    - Type I codes are not in the Asamov table.

[3] Harada, M. (2011). On the residue codes of extremal Type II Z4-codes of lengths 32 and 40. Discrete Mathematics, 311(20), 2148–2157.