J. Algebra Comb. Discrete Appl.
7(3) • 237–245

Received: 23 May 2019
Accepted: 25 March 2020

Journal of Algebra Combinatorics Discrete Structures and Applications

Generating generalized necklaces and new quasi-cyclic codes*

Research Article

Rumen Daskalov, Elena Metodieva

Abstract: In many cases there is a need of exhaustive lists of combinatorial objects of a given type. We consider generation of all inequivalent polynomials from which defining polynomials for constructing quasi-cyclic (QC) codes are to be chosen. Using these defining polynomials we construct 34 new good QC codes over GF(11) and 36 such codes over GF(13).

2010 MSC: 94B05, 94B65

Keywords: Finite field, Quasi-cyclic linear codes, Necklaces

1. Introduction

Let GF(q) denote the Galois field of q elements and let V(n,q) denote the vector space of all ordered n-tuples over GF(q). The Hamming weight of a vector x, denoted by $wt(x)$, is the number of nonzero entries in x. A linear code C of length n and dimension k over GF(q) is a k-dimensional subspace of V(n,q). Such a code is called $[n,k,d]_q$ code if its minimum Hamming distance is d. For linear codes, the minimum distance is equal to the smallest of the weights of the nonzero codewords. A $k \times n$ matrix G having as rows the vectors of a basis of a linear code C is called a generator matrix for C. Let A_i denote the number of codewords of C with weight i. The weight distribution of C is the list of numbers A_i. The weight distribution $A_0 = 1$, $A_d = \alpha$, ..., $A_n = \gamma$ is expressed as $0^1 d^\alpha \dots n^\gamma$ also.

In order to obtain a q-ary linear code which is capable of correcting most errors for given values of n, k, and q, it is sufficient to obtain an $[n,k,d]_q$ code C with maximum minimum distance d among all such codes or for given values of k, d, and q, to obtain an $[n,k,d]_q$ code C whose length n is the smallest one. The respective codes in these two cases are called optimal.

* *This work was partially supported by the Bulgarian Ministry of Education and Science under Contract in TU–Gabrovo.*

Rumen Daskalov (Corresponding Author), Elena Metodieva; Department of Mathematics and Informatics, Technical University of Gabrovo, Bulgaria (email: daskalov@tugab.bg, metodieva@tugab.bg).

The problem of determining optimal code parameters, known as the Main Linear Coding Theory Problem, has two aspects. One is the construction of codes which optimize minimum distance and the other is proving non-existence of codes of certain parameters ([14], [21]). In the former one often uses computers, but this approach becomes ineffective when the dimension of the codes is large, because, as we know, computing the minimum distance of linear codes is an NP-hard problem [30]. Thus, it becomes expedient to use classes of codes that have a rich mathematical structure. In recent years it has been shown that QC and QT codes form such nice classes. Being generalizations of cyclic and consta-cyclic codes, they contain many good, record-breaking codes [1–6, 10, 11, 15–20, 24, 27].

Markus Grassl and Eric Chen maintain online tables of linear codes. The Grassl's tables [28] contain lower and upper bounds on minimum distances for linear codes over small finite fields ($q \leq 9$). Many of the best-known codes in these tables are QC and QT codes. The Chen's table [9] contains only good and best-known QC and QT codes ($q \leq 13$). These two databases are updated when new codes are discovered.

In recent years there has been an increased interest in codes over GF(11) and GF(13). In [25] and [26] Gulliver constructed QT codes over GF(11) for $k \leq 7$ and QC codes over GF(13) for $k \leq 6$. Venkaiah and Gulliver [31] constructed quasi-cyclic $[pk, k, d]_{13}$ codes of dimensions $k \leq 6$ and $n \leq 150$. In [12] and [13] E. Chen and N. Aydin constructed 45 new OC and QT codes over GF(11), 38 such codes over GF(13) and presented databases for small dimensions and $n \leq 150$. New QT codes over GF(11) and GF(13) are also presented in [7].

Three-dimensional projective codes are closely related to $(n, r)-$arcs in projective finite planes, and a lot of research has been done over finite fields of size up to 19 [8]. Recently an optimal $(78, 8)$-arc in PG(2,11) was constructed in [25] as a $[78, 3, 70]_{11}$ QC code.

In this paper we present 34 new QC codes over GF(11) and 36 new QC codes over GF(13).

2. QC codes

A code C is said to be p-QC if a cyclic shift of any codeword by p positions results in another codeword. Suppose that C is a p-QC $[pm, k]$ code $(m \geq k)$. It is convenient to take the co-ordinate places of C in the following order

$$\begin{array}{l} 1, p+1, 2p+1, \ldots, (m-1)p+1, \\ 2, p+2, \ldots, (m-1)p+2, \\ p, 2p, \ldots, mp. \end{array}$$

Then C will be generated by a matrix of the form

$$[B_1, B_2, \ldots, B_p]$$

where each B_i is a circulant matrix, i.e. a matrix of the form

$$B = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{m-1} \\ b_{m-1} & b_0 & b_1 & \cdots & b_{m-2} \\ b_{m-2} & b_{m-1} & b_0 & \cdots & b_{m-3} \\ \vdots & \vdots & \vdots & & \vdots \\ b_1 & b_2 & b_3 & \cdots & b_0 \end{bmatrix}$$

If the row vector $(b_0 b_1 \cdots b_{m-1})$ is identified with the polynomial $d(x) = b_0 + b_1 x + ... + b_{m-1} x^{m-1}$, then we may write

$$B = \begin{bmatrix} d(x) \\ xd(x) \\ x^2d(x) \\ \vdots \\ x^{m-1}d(x) \end{bmatrix}$$

where each polynomial is reduced modulo $x^m - 1$.

Denote the polynomials associated with the matrices $B_1, B_2, B_3, \ldots, B_p$ by $d_1(x), d_2(x), d_3(x), \ldots, d_p(x)$. These polynomials are called **defining polynomials** of C.

Taking the polynomials $ax^l d_i(x)$ instead of $d_i(x)$ we make a cyclic shift of the columns of B_i and multiply them by a nonzero element of the field. This leads to a generator matrix of an equivalent code. So, the defining polynomials of a QC-code can be chosen from a fixed set of representatives of the equivalence classes of polynomials of degree less than m under the following relation:

$$c_i(x) \approx c_j(x) \iff c_i(x) \equiv ax^l c_j(x) \mod (x^m - 1) \tag{1}$$

It stands to reason that we need an efficient algorithm to produce such a set of polynomials.

3. Necklaces

We identify the polynomials with the strings of their coefficients. In terms of strings the relation (1) is a composition of two actions on strings, namely, rotating the string and multiplying its entries by nonzero elements of the field (scaling the string). Efficient algorithms are known for generating the equivalence classes of the first of the actions. By efficient we mean that the amount of computation used in generating the objects is proportional to the number of objects generated.

Let Σ_q be the alphabet $\{0, 1, 2, \ldots, q-1\}$ and Σ_q^m be the set of q-ary strings of length m. Denote by α_i, $1 \leq i \leq m$, the entries of the string $\alpha \in \Sigma_q^m$, and a^t is the string of length t whose entries are all equal to a. The symbol $\preceq$ is used for lexicographic order in Σ_q^m. Call two strings equivalent if one is a cyclic shift of the other. An equivalence class of strings under this relation is called a **necklace** of m beads in q colors. We identify each necklace with the lexicographically smallest representative in the equivalence class. Thus we call a string $\alpha = \alpha_1\alpha_2\ldots\alpha_m$ a necklace if, for $1 \leq i \leq m$, $\alpha_1 \ldots \alpha_m \preceq \alpha_i \ldots \alpha_m \alpha_1 \ldots \alpha_{i-1}$.

For given m and q the number of necklaces is well known to be

$$N_q(m) = \frac{1}{m} \sum_{d|m} \phi(d) q^{\frac{m}{d}}$$

where ϕ is the Euler totient function.

A simple and elegant algorithm was proposed by Fredricksen, Kessler and Maiorana [22], [23] to generate all necklaces in Σ_q^m. We will refer to this as the FKM algorithm.

For given m and q, the FKM algorithm creates a list, $\mathcal{FKM}(q, m)$, consisting of a certain subset of Σ_q^m in lexicographic order. The list begins with the string 0^m and ends with $(q-1)^m$. For a given α

on $\mathcal{FKM}(q, m)$, the successor of α, succ(α), is obtained as follows:

For $\alpha = \alpha_1\alpha_2 \dots \alpha_m \prec (q-1)^m$, $\quad$ $\text{succ}(\alpha) = (\alpha_1 \dots \alpha_{i-1}(\alpha_i + 1))^t \alpha_1 \dots \alpha_j$,
where i is the largest integer $1 \leq i \leq m$ such that $\alpha_i < q-1$ and t, j satisfy $ti + j = m$, $\quad j < i$.

It is shown in [22] that there is no necklace between two elements of $\mathcal{FKM}(q, m)$, so that the list contains all necklaces. Thus, discarding non necklaces of $\mathcal{FKM}(q, m)$ would result in a list of all necklaces in increasing order. In [29] Ruskey, Savage and Wang proved that succ(α) is a necklace if and only if the "i" from the definition of the successor of α is a divisor of m. Including this test, the entire algorithm can be represented by the following PASCAL code:

```
for i:=0 to m do  a[i]:=0;  br:=0;  i:=m;
repeat
   a[i]:=a[i]+1;
   for j:=1 to m-i do a[j+i]:=a[j];
   if m mod i = 0 then
      begin
         br:=br+1;
          for j:=1 to m do write(a[j]);
          writeln;
      end;
     i:=m;
    while a[i]=q-1 do i:=i-1;
until i=0;
```

4. Generalized necklaces

In its turn, scaling the necklaces partitions the set of all necklaces into new equivalence classes. We call them generalized necklaces. To generate their representatives we know no more efficient algorithm than rejecting those necklaces that are not the smallest representatives. A naive approach to testing whether a length m necklace is the smallest representative of an equivalence class according to (1) is to compare the necklace with all of its scaled rotations. However, by taking into consideration some facts we can decrease the number of comparisons that have to be made.

First of all we generate only necklaces with first non-zero element 1. It follows from the FKM algorithm that any necklace has the form

$$\alpha = (\alpha_1\alpha_2 \dots \alpha_i)^t, \quad t \geq 1$$

Any rotation of α has the form

$$\alpha' = \alpha_{s+1} \dots \alpha_i (\alpha_1\alpha_2 \dots \alpha_i)^{t-1} \alpha_1 \dots \alpha_s =$$

$$= (\alpha_{s+1} \dots \alpha_i \alpha_1 \dots \alpha_s)^t$$

for $s = 1, 2, \dots, i-1$.

Thus comparing the strings α and $b\alpha'$, $b \in GF(q) \setminus \{0, 1\}$ it suffices to compare the substring $\beta = \alpha_1\alpha_2 \dots \alpha_i$ with its rotations, multiplied by an element of $GF(q) \setminus \{0, 1\}$.
If p is the index of the first non-zero element of β, than scaled cyclic shifts with starting positions $2, 3, \dots, p$ will obviously follow β in lexicographic order. The cyclic shifts with starting positions

$$i - p + 2, i - p + 3, \dots, i$$

will have $\alpha_i \neq 0$ in position with number less than p and therefore they will also follow $\beta = \alpha_1 \ldots \alpha_i$. So, the only comparisons that have to be made are with cyclic shifts starting from positions

$$l = p+1, p+2, \ldots, i-p+1$$

if $i-p+1 \geq p+1$, i.e. $i \geq 2p$, having

$$\alpha_{l+p-1} > 1,$$

and multiplied only by the inverse of α_{l+p-1}.

The implementation of these considerations yields the following PASCAL code, which produces the desired set of strings (or polynomials):

```
      for i:=0 to n-1 do  a[i]:=0;
      a[n]:=1;  br:=1;   i:=n;   p:=n;
     repeat
        if i=p then begin i:=i-1; p:=p-1; end;
        a[i]:=a[i]+1;
        for j:=1 to n-i do a[j+i]:=a[j];
        if n mod i = 0 then
          begin
               for l:=p+1 to i-p+1 do
                 if a[l+p-1]>1 then
                   begin
                     m:=inv[a[l+p-1]]; k1:=1; k2:=l;
                     while (k1<=i) and (a[k1]=mul[a[k2],m]) do

                       begin
                         k1:=k1+1;
                         k2:=(k2 mod i)+1;
                       end;
                     if k1<= i then
                       if a[k1] > mul[a[k2],m] then goto 10;
                   end;
            begin
              br:=br+1; for j:=1 to n do write(a[j]);
            end;
          end;
10:     i:=n;
        while a[i]=q-1 do i:=i-1;
     until (i=i1) and (i=1);
```

In the table below the number $K_{11}(m)$ and $K_{13}(m)$ of generated objects is given for $m = 6, 7, 8, 9$. The respective number $N_{11}(m)$ and $N_{13}(m)$ of necklaces is also given for comparison. From some seconds up to some minutes are needed for the generation of the objects (CPU, Intel i3, 2.2GHz).

Table 1: $N_q(m)$ and $K_q(m)$

m	$N_{11}(m)$	$K_{11}(m)$	m	$N_{13}(m)$	$K_{13}(m)$
7	2783891	278389	6	804895	67116
8	26796726	2679859	7	8964085	747007
9	261994491	26199449	8	101969959	8497806

The next explicit formula for the number of generalized necklaces (defining polynomials for QC codes) was given recently in [31] (for QT codes see [32])

$$K_q(m) = \frac{1}{(q-1)m} \sum_{d|m} \phi(d) \left(q^{\frac{m}{d}} - 1\right) \gcd(d, q-1)$$

5. New QC codes over GF(11) and GF(13)

In this section we present 34 new QC codes over GF(11) in dimensions $k = 8, 9$ and 36 new QC codes over GF(13) in dimensions $k = 7, 8$. The new codes are obtained by non-exhaustive local computer search. By reason of space the defining polynomials and weight distributions only of some of the codes are given. For the rest of the codes, the respective information is available on request from the authors. All constructed codes will be send to E. Chen to be included in the respective table[9]. The elements of the fields are denoted by 0, 1, 2, ... , 9, $10 = a$, $11 = b$, $12 = c$.

Codes over GF(11)

There exist quasi-cyclic codes with parameters: $[16,8,8]_{11}$, $[24,8,14]_{11}$, $[32,8,20]_{11}$, $[40,8,26]_{11}$, $[48,8,33]_{11}$, $[56,8,39]_{11}$, $[64,8,46]_{11}$, $[72,8,53]_{11}$, $[80,8,59]_{11}$, $[88,8,66]_{11}$, $[96,8,73]_{11}$, $[104,8,80]_{11}$, $[112,8,87]_{11}$, $[120,8,94]_{11}$, $[128,8,100]_{11}$, $[136,8,107]_{11}$, $[144,8,114]_{11}$, $[152,8,121]_{11}$.

A $[16,8,8]_{11}$ **optimal code**: aaaaaaa9, aaa9a362;
0^{1} 8^{7400} 9^{55200} 10^{367360} $11^{2031680}$ $12^{8526000}$ $13^{26073600}$ $14^{56016800}$ $15^{74628160}$ $16^{46652680}$

A $[24,8,14]_{11}$ **code**: aaa986a2, aaaa9a68, aaaa7022;
0^{1} 14^{4880} 15^{29840} 16^{159390} 17^{753120} $18^{2924120}$ $19^{9254640}$ $20^{23128500}$ $21^{44045200}$ $22^{60063480}$ $23^{52233280}$ $24^{21762430}$

A $[32,8,20]_{11}$ **code**: aaa98558, aaaa6960, aaaa9784, a9073007;
0^{1} 20^{2540} 21^{13360} 22^{66000} 23^{280960} $24^{1077200}$ $25^{3415120}$ $26^{9184520}$ $27^{20465280}$ $28^{36499940}$ $29^{50328800}$ $30^{50421000}$ $31^{32438240}$ $32^{10165920}$

A $[40,8,26]_{11}$ **code**: aaa98503, aaaa7006, aaaa9a28, a9074154, aaaa9232;
0^{1} 26^{1360} 27^{4880} 28^{24300} 29^{113760} 30^{406400} $31^{1287440}$ $32^{3650050}$ $33^{8810720}$ $34^{18211080}$ $35^{31126480}$ $36^{43291000}$ $37^{46825520}$ $38^{36935880}$ $39^{18923760}$ $40^{4746250}$

A $[48,8,33]_{11}$ **code**: a9074154, aaa986a2, aaaa9232, aaaa7022, aaaa9a68, a92a7a64;
0^{1} 33^{2400} 34^{10200} 35^{44800} 36^{152880} 37^{501200} $38^{1448040}$ $39^{3704240}$ $40^{8343150}$ $41^{16214080}$ $42^{27158960}$ $43^{37882960}$ $44^{42926260}$ $45^{38227600}$ $46^{24916320}$ $47^{10623840}$ $48^{2201950}$

A $[56,8,39]_{11}$ **code**: a9074154, aaa98449, aaaa7011, aaaa9a28, aaaa9162, aa461627, aaa80096;
0^{1} 39^{1120} 40^{3720} 41^{16160} 42^{63720} 43^{192000} 44^{575440} $45^{1534240}$ $46^{3666320}$ $47^{7828560}$ $48^{14643050}$ $49^{23845680}$ $50^{33507360}$ $51^{39412640}$ $52^{37813820}$ $53^{28560000}$ $54^{15893160}$ $55^{5777200}$ $56^{1024690}$

A $[64,8,46]_{11}$ **code**: aaa98449, aa606690, aaaa97a4, a9072196, aaaa9149, aa461627, aaa80091, aa2a9a89;
0^{1} 46^{2000} 47^{6640} 48^{23420} 49^{74960} 50^{234280} 51^{628720} $52^{1577940}$ $53^{3584880}$ $54^{7284360}$ $55^{13222240}$ $56^{21322560}$ $57^{29816640}$ $58^{36078240}$ $59^{36648480}$ $60^{30601420}$ $61^{19981760}$ $62^{9712480}$ $63^{3080160}$ 64^{477700}

A $[72,8,53]_{11}$ **code**: aaa986a2, aaaa7022, aaaa9a68, a9074154, aaaa9232, aa461651, aaa800a5, aa301291, aa6a5912;
0^{1} 53^{2480} 54^{9760} 55^{29760} 56^{93600} 57^{258960} 58^{665280} $59^{1602880}$ $60^{3442340}$ $61^{6787200}$ $62^{12060720}$ $63^{19044240}$ $64^{26783290}$ $65^{33069280}$ $66^{35145400}$ $67^{31353600}$ $68^{23078000}$ $69^{13339120}$ $70^{5749880}$ $71^{1620720}$ 72^{222370}

A $[80,8,59]_{11}$ **code**: a8606780, aaaa6a63, aa8896a9, a9074154, aaa25520, aa461651, aaa7a862, aa2a9a89, aa6a5384, aa3a2803;
0^{1} 59^{1200} 60^{3520} 61^{12080} 62^{40120} 63^{102640} 64^{283920} 65^{691600} $66^{1585440}$ $67^{3298320}$ $68^{6291980}$ $69^{10963040}$ $70^{17246760}$ $71^{24241840}$

$72^{30352890}$ $73^{33273280}$ $74^{31397080}$ $75^{25181760}$ $76^{16544120}$ $77^{8602400}$ $78^{3299480}$ 79^{839040} 80^{106370}

A $[88, 8, 66]_{11}$ **code**: a8606759, aaaa6a60, aa8896a6, a9074150, aaa255a5, aa461649, aaa7a857, aa2a9a86, aa6a5379, aa3a2803, aaa97887;
0^1 66^{1440} 67^{4960} 68^{16000} 69^{42400} 70^{118120} 71^{296160} 72^{698300} $73^{1567200}$ $74^{3124800}$ $75^{5893520}$ $76^{10031240}$ $77^{15640320}$ $78^{21953920}$ $79^{27887520}$ $80^{31397440}$ $81^{31029840}$ $82^{26457160}$ $83^{19099600}$ $84^{11413680}$ $85^{5327520}$ $86^{1884000}$ 87^{424960} 88^{48780}

There exist quasi-cyclic codes with parameters: $[9p, 9, 7p-6]_{11}$ for $p = 2, 3, ..., 7$, $[9p, 9, 15p/2-9]_{11}$ for $p = 8, 10, 12, 14, 16$ and $[9p, 9, 15(p-1)/2-2]_{11}$ for $p = 9, 11, 13, 15, 17$.

A $[18, 9, 8]_{11}$ **code**: aaaa98708, aaaaa7023;
0^1 8^{1440} 9^{22320} 10^{187740} $11^{1340640}$ $12^{7871640}$ $13^{36363600}$ $14^{129741480}$ $15^{346074720}$ $16^{648874890}$ $17^{763370460}$ $18^{424098760}$

A $[45, 9, 29]_{11}$ **code**: aaaaa7023, aaaaa9949, aaaa98708, aa98113a9, aaa9a9a68;
0^1 29^{2880} 30^{11160} 31^{53550} 32^{236160} 33^{930270} $34^{3294720}$ $35^{10311930}$ $36^{28639260}$ $37^{69744510}$ $38^{146759040}$ $39^{263601600}$ $40^{395301510}$ $41^{481704840}$ $42^{459168780}$ $43^{320269140}$ $44^{145588050}$ $45^{32330290}$

A $[108, 9, 81]_{11}$ **code**: aaaa98708, aaaaa7023, aaaaa9949, aa98113a9, aaa9a9a59, aaa520467, aaa249519, aa9a9a937, aaa721a86, aa9a9a823, aaa9a8973, aa9a89a82
0^1 81^{1800} 82^{5400} 83^{17280} 84^{50580} 85^{142650} 86^{394740} 87^{976260} $88^{2353500}$ $89^{5258880}$ $90^{11138730}$ $91^{21897720}$ $92^{40764510}$ $93^{69821070}$ $94^{111537090}$ $95^{164458260}$ $96^{222387870}$ $97^{275444100}$ $98^{309390300}$ $99^{312102790}$ $100^{280789380}$ $101^{222478830}$ $102^{152968320}$ $103^{88963020}$ $104^{42785460}$ $105^{16267290}$ $106^{4608720}$ 107^{864810} 108^{78330}

Codes over GF(13)

There exist quasi-cyclic codes with parameters: $[14, 7, 7]_{13}$, $[21, 7, 13]_{13}$, $[28, 7, 18]_{13}$, $[7p, 7, 6p-6]_{13}$ for $p = 5, 6, ..., 15$ and $[7p, 7, 6p-5]_{13}$ for $p = 16, 17, 18, 19$.

A $[14, 7, 7]_{13}$ **code**: cccccb, cccbca7;
0^1 7^{2100} 8^{21336} 9^{164220} 10^{983556} $11^{4319196}$ $12^{12922308}$ $13^{23875068}$ $14^{20460732}$

A $[21, 7, 13]_{13}$ **code**: cc484c1, ca650c9, ccb0b70;
0^1 13^{6636} 14^{39396} 15^{210336} 16^{957684} $17^{3356220}$ $18^{9025884}$ $19^{17011344}$ $20^{20458452}$ $21^{11682564}$

A $[28, 7, 18]_{13}$ **code**: cc48499, ca650c7, cc78aca, cccca17;
0^1 18^{1848} 19^{7896} 20^{49560} 21^{219168} 22^{847980} $23^{2627268}$ $24^{6584004}$ $25^{12663000}$ $26^{17505348}$ $27^{15570492}$ $28^{6671952}$

A $[35, 7, 24]_{13}$ **code**: cc4849a, ca650c9, cc78acb, cccca15, cb04650;
0^1 24^{2772} 25^{11172} 26^{52500} 27^{209160} 28^{713412} $29^{2061696}$ $30^{4989516}$ $31^{9623040}$ $32^{14413560}$ $33^{15735888}$ $34^{11134032}$ $35^{3801768}$

A $[42, 7, 30]_{13}$ **code**: cc48499, cc78b00, cccca18, cb04647, ccbb381, ca650c9;
0^1 30^{2352} 31^{12768} 32^{51912} 33^{194376} 34^{599592} $35^{1619184}$ $36^{3827880}$ $37^{7441812}$ $38^{11757564}$ $39^{14433048}$ $40^{13019076}$ $41^{7613844}$ $42^{2175108}$

A $[112, 7, 91]_{13}$ **code**: cccca24, ca650c9, cc484c1, ccc76a8, ccbc4b2, ccbb385, cccbcb4, cc78b01, ccb9656, ccbc163, ccbc962, ccbac74, ccbab1a, ccbab23, ccccbc1, cb38697;
0^1 91^{4200} 92^{14448} 93^{35784} 94^{91056} 95^{196224} 96^{407736} 97^{825636} $98^{1507632}$ $99^{2539824}$ $100^{3945984}$ $101^{5683272}$ $102^{7303128}$ $103^{8580348}$ $104^{8870568}$ $105^{8095248}$ $106^{6441624}$ $107^{4324320}$ $108^{2388792}$ $109^{1068228}$ 110^{336672} 111^{79716} 112^{8076}

There exist quasi-cyclic codes with parameters: $[16, 8, 8]_{13}$, $[24, 8, 14]_{13}$, $[32, 8, 20]_{13}$, $[40, 8, 27]_{13}$, $[48, 8, 33]_{13}$, $[56, 8, 40]_{13}$, $[64, 8, 47]_{13}$, $[72, 8, 53]_{13}$ and $[8p, 8, 7p-10]_{13}$ for $p = 10, 11, ..., 19$.

A $[16, 8, 8]_{13}$ **code**: cccccc77, cc82b0ca;
0^1 8^{7116} 9^{80352} 10^{583632} $11^{4004448}$ $12^{19892040}$ $13^{73427424}$ $14^{188896848}$ $15^{302188704}$ $16^{226650156}$

A $[24, 8, 14]_{13}$ **code**: ccccccc6, ccb50a09, ccc538ca;
0^{1} 14^{3312} 15^{32544} 16^{205248} $17^{1154400}$ $18^{5379504}$ $19^{20390304}$ $20^{61267056}$ $21^{139904640}$ $22^{228977184}$ $23^{238950528}$ $24^{119466000}$

A $[64, 8, 47]_{13}$ **code**: cccccca, ccb50a15, ccc7b8b6, ccb75441, cc629b0b, cc847138, ccc538ca, cc6c9aa0;
0^{1} 47^{3168} 48^{12156} 49^{50016} 50^{189360} 51^{593184} $52^{1788792}$ $53^{4855392}$ $54^{11944992}$ $55^{25823712}$ $56^{50130300}$ $57^{84293952}$ $58^{122167440}$ $59^{148750176}$ $60^{149087808}$ $61^{117161664}$ $62^{68071872}$ $63^{25970304}$ $64^{4836432}$

A $[80, 8, 60]_{13}$ **code**: cccccc56, ccb50968, ccc538c8, ccc7c166, ccb75438, cc629a31, cc8471a8, cc6c9a95, cccb9586, ccb57b73;
0^{1} 60^{1584} 61^{3840} 62^{19200} 63^{62112} 64^{199032} 65^{583776} $66^{1584240}$ $67^{3965664}$ $68^{9137136}$ $69^{19061568}$ $70^{35892528}$ $71^{60720384}$ $72^{91043424}$ $73^{119819040}$ $74^{135862176}$ $75^{130555584}$ $76^{102976704}$ $77^{64252800}$ $78^{29645280}$ $79^{8981952}$ $80^{1362696}$

References

[1] N. Aydin, I. Siap, D. K. Ray–Chaudhuri, The structure of 1–generator quasi–twisted codes and new linear codes, Des. Codes Cryptogr. 24 (2001) 313–326.

[2] N. Aydin, I. Siap, New quasi–cyclic codes over F_5, Appl. Math. Lett. 15 (2002) 833–836.

[3] N. Aydin, J. Murphree, New linear codes from constacyclic codes, J. Franklin Inst. 351(3) (2014) 1691–1699.

[4] N. Aydin, N. Connolly, M. Grassl, Some results on the structure of constacyclic codes and new linear codes over GF(7) from QT codes, Adv. Math. Commun. 11(1) (2017) 245–258.

[5] N. Aydin, N. Connolly, J. Murphree, New binary linear codes from quasi–cyclic codes and an augmentation algorithm, Appl. Algebra Engrg. Comm. Comput. 28(4) (2017) 339–350.

[6] N. Aydin,J. Lambrinos, O. VandenBerg, On equivalence of cyclic codes, generalization of a quasi–twisted search algorithm, and new linear codes, Des. Codes Cryptogr. 87 (2019) 2199–2212.

[7] N. Aydin, D. Foret, New linear codes over GF(3), GF(11) and GF(13), J. Algebra Comb. Discrete Struct. Appl. 6(1) (2019) 13–20.

[8] S. Ball, Table of bounds on three dimensional linear codes or (n, r) Arcs in PG(2, q), available at https://web.mat.upc.edu/simeon.michael.ball/codebounds.html

[9] E. Z. Chen, Database of Quasi–twisted codes, available at http://www.tec.hkr.se/∼chen/research/codes/

[10] E. Z. Chen, A new iterative computer search algorithm for good quasi–twisted codes,Des. Codes Cryptogr. 76(2) (2015) 307–323.

[11] E. Z. Chen, Some new binary linear codes with improved minimum distances, J. Algebra Comb. Discrete Struct. Appl. 5(2) (2018) 65–70.

[12] E. Z. Chen, N. Aydin, New quasi–twisted codes over F_{11} minimum distance bounds and a new database, J. Inf. Optim. Sci. 36(1–2) (2015) 129–157.

[13] E. Z. Chen, N. Aydin, A database of linear codes over F_{13} with minimum distance bounds and new quasi–twisted codes from a heuristic search algorithm, J. Algebra Comb. Discrete Struct. Appl. 2(1) (2015) 1–16.

[14] R. Daskalov, E. Metodieva, The nonexistence of some 5–dimensional quaternary linear codes, IEEE Trans. Inform. Theory 41(2) (1995) 581–583.

[15] R. N. Daskalov, T. A. Gulliver, New good quasi–cyclic ternary and quaternary linear codes, IEEE Trans. Inform. Theory 43 (1997) 1647–1650.

[16] R. Daskalov, T. A. Gulliver, Bounds on Minimum Distance for Linear Codes over GF(5), Appl. Algebra Engrg. Comm. Comput. 9(6) (1999) 547–558.

[17] R. Daskalov, P. Hristov, New one–enerator quasi–cyclic codes over GF(7), Problemi Peredachi Informatsii 38(1) (2002) 59–63. English translation: Probl. Inf. Transm. 38(1) (2002) 50–54.

[18] R. Daskalov, P. Hristov, Some new quasi–twisted ternary linear codes, J. Algebra Comb. Discrete Struct. Appl. 2(3) (2016) 211–216.

[19] R. Daskalov, P. Hristov, Some new ternary linear codes, J. Algebra Comb. Discrete Struct. Appl.

4(3) (2017) 227–234.
[20] R. Daskalov, P. Hristov, E. Metodieva, New minimum distance bounds for linear codes over GF(5), Discrete Math. 275(1–3) (2004) 97–110.
[21] R. Daskalov, E. Metodieva, The nonexistence of ternary [105,6,68] and [230,6,152] codes, Discrete Math. 286(3) (2004) 225–232.
[22] H. Fredricksen, J. Maiorana, Necklaces of beads in k colors and k–ary de Bruijn sequences, Discrete Math. 23 (1978) 207–210.
[23] H. Fredricksen, I. J. Kessler, An algorithm for generating necklaces of beads in two colors, Discrete Math. 61 (1986) 181–188.
[24] P. P. Greenough, R. Hill, Optimal ternary quasi–cyclic codes, Des. Codes Crypt. 2 (1992) 81–91.
[25] T. A. Gulliver, Quasi–twisted codes over F_{11}, Ars Combinatoria 99 (2011) 3–17.
[26] T. A. Gulliver, Quasi–cyclic codes over F_{13}, In Combinatorial Algorithms, Lecture Notes in Computer Science 7056 (2011) 236–246.
[27] T. A. Gulliver, P. R. J. Ostergard, Improved bounds for ternary linear codes of dimension 7, IEEE Trans. Inform. Theory 43 (1997) 1377–1388.
[28] M. Grassl, Bounds on the minimum distances of linear codes, available at http://www.codetables.de
[29] F. Ruskey, C. Savage, T. Wang, Generating necklaces, Journal of Algorithms 13 (1992) 414–430.
[30] A. Vardy, The intractability of computing the minimum distance of a code, IEEE Trans. Inform. Theory 43 (1997) 1757–1766.
[31] V. Venkaiah, T. A.Gulliver, Quasi–cyclic codes over F_{13} and enumeration of definining polynomials, Journal Discrete Algorithms 16 (2012) 249–257.
[32] T. A. Gulliver, V. Venkaiah, Construction of quasi–twisted codes and enumeration of definining polynomials, J. Algebra Comb. Discrete Struct. Appl. 7(1) (2020) 3–20.