ALGEBRAIC AND COMBINATORIAL CODING THEORY

PROCEEDINGS

Eleventh International Workshop
June 16 - 22, 2008
Pamporovo, Bulgaria

ALGEBRAIC AND COMBINATORIAL CODING THEORY

PROCEEDINGS

Eleventh International Workshop
June 16 - 22, 2008
Pamporovo, Bulgaria

PREFACE

The Eleventh International Workshop on Algebraic and Combinatorial Coding Theory (ACCT2008) is organized by the **Institute of Mathematics and Informatics** of the Bulgarian Academy of Sciences and the **Institute for Information Transmission Problems** of the Russian Academy of Sciences.

The previous workshops were held in Varna, Bulgaria (1988), St. Petersburg, Russia (1990), Voneshta Voda, Bulgaria (1992), Novgorod, Russia (1994), Sozopol, Bulgaria (1996), Pskov, Russia (1998), Bansko, Bulgaria (2000), Tsarskoe Selo, Russia (2002), Kranevo, Bulgaria (2004) and Zvenigorod, Russia (2006).

ACCT2008 is held in Bulgaria, in the beautiful mountain resort of Pamporovo.

Organizing Committee:

L. Bassalygo (Moscow)
Co-Chairman
S. Dodunekov (Sofia)
Co-Chairman
S. Kapralov (Gabrovo)
V. Lebedev (Moscow)
B. Kudryashov (St. Petersburg)
I. Landjev (Sofia)
V. Zyablov (Moscow)
S. Topalova (V. Tarnovo)

Programme Committee:

N. Manev (Sofia)
Co-Chairman
G. Kabatiansky (Moscow)
Co-Chairman
S. Bouyuklieva (V. Tarnovo)
P. Boyvalenkov (Sofia)
E. Kolev (Sofia)
V. Levenshtein (Moscow)
F. Solov'eva (Novosibirsk)
V. Zinoviev (Moscow)

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences

Printed in Bulgaria

CONTENTS

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 7-12*

On the least covering radius of the binary linear codes of dimension 6

Tsonka Baicheva, Iliya Bouyukliev tsonka,iliyab@math.bas.bg
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
P.O. Box 323, 5000 Veliko Tarnovo, BULGARIA

Abstract. In this work a heuristic algorithm for obtaining lower bounds on the covering radius of a linear code is developed. Using this algorithm the least covering radii of the binary linear codes of dimension 6 are determined. Upper bounds for the least covering radii of binary linear codes of dimensions 8 and 9 are derived.

1 Introduction

In this work we address two problems: the mathematical question of determining $t_2[n,k]$, the smallest covering radius of any binary linear $[n,k]$ code, and the more practical problem of constructing codes having a specified length and dimension and the least covering radius. More precisely we determine all values of the function $t_2[n,6]$ and give constructions for such codes. An important part of the determination of the values of $t_2[n,6]$ is the suggested heuristic algorithm for computation of lower bound of the covering radius of a linear code. We also derive upper bounds for $t_2[n,8]$ and $t_2[n,9]$.

2 Some preliminary results

Let F_q^n be the n-dimensional vector space over the finite field with q elements. A *linear code* C is a k-dimensional subspace of F_q^n. The ball of radius t around a word $y \in F_q^n$ is defined by

$$\{x | x \in F_q^n, d(x,y) \leq t\}.$$

Then the covering radius $R(C)$ of a code C is defined as the least possible integer number such that the balls of radius $R(C)$ around the codewords cover the whole F_q^n, i.e.

$$R(C) = \max_{x \in F_q^n} \min_{c \in C} d(x,c).$$

A coset of the code C defined by the vector $x \in F_q^n$ is the set $x+C = \{x+c \mid c \in C\}$. A coset leader of $x+C$ is a vector in $x+C$ of smallest weight. When the code is linear its covering radius is equal to the weight of the heaviest coset leader.

The function $t_q[n,k]$ is defined as the least value of $R(C)$ when C runs over the class of all linear $[n,k]$ codes over F_q for a given q.

Definition. [1] Let C be a binary code of length n and covering radius R. For $i = 1,\ldots,n$ let $C_0^{(i)}$ (respectively $C_1^{(i)}$) denote the set of codewords in which the i-th coordinate is 0 (respectively 1). The integer

$$N^{(i)} = \max_{x\in F_2^n}\{d(x,C_0^{(i)}) + d(x,C_1^{(i)})\}$$

is called the *norm* of C with respect to the i-th coordinate and

$$N_{min} = \min_i N^{(i)}$$

is called the minimum norm of C. (We use the convention that $d(x,0) = \infty$.) The code C has norm N if $N_{min} \leq N$ and the coordinates i for which $N^{(i)} < N$ are called *acceptable* with respect to N.

The code C is *normal* if it has norm $2R+1$. If $N^{(i)} \leq 2R+1$, then we say that the coordinate i is acceptable with respect to $2R+1$, or that C is normal with respect to the i-th coordinate.

In the following theorem results about the normality of binary linear codes are summarized.

Theorem 1. [1] *If C is an $[n,k,d]$ code with $n \leq 15$, $k \leq 5$ or $n-k \leq 9$, then C is normal.*

One of the main reasons for studying normal codes is the *amalgamated direct sum* (ADS) construction introduced by Graham and Sloane [2].

Theorem 2 [1] *Assume that A is a normal binary $[n_A,k_A]R_A$ code with the last coordinate acceptable, and B is a normal binary $[n_B,k_B]R_B$ code with the first coordinate acceptable. Then their amalgamated direct sum (ADS)*
$A\dot{\oplus}B = \{(a,0,b)|(a,0) \in A, (0,b) \in B\} \cup \{(a,1,b)|(a,1) \in A, (1,b) \in B\}$
is an $[n_A+n_B-1, k_A+k_B-1]R$ code with $R \leq R_A+R_B$. More generally, if the norm of A with respect to the last coordinate is N_A and the norm of B with respect to the first coordinate is N_B, then the code $A\dot{\oplus}B$ has norm N_A+N_B-1 and hence covering radius at most $\frac{1}{2}(N_A+N_B-1)$. In particular, if the covering radius of $A\dot{\oplus}B$ equals R_A+R_B, then $A\dot{\oplus}B$ is normal and the overlapping coordinate is acceptable.

3 Least covering radius of the binary linear codes of dimension 6

The results about the least covering radius of binary linear codes are summarized in Table 7.1 from [1] where the exact values or bounds for $t_2[n,k]$ for codes of lengths up to 64 are given. We use this table as a source of our investigation and give the results for codes of dimension 6 in the following table.

Bounds on $t_2[n, 6]$ for $n \leq 64$

n	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
t_2[n, 6]	1	1	1	2	2	3	3	3	4	4
n	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
t_2[n, 6]	5	5	5	6	6	6-7	7	7-8	7-8	8-9
n	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**
t_2[n, 6]	8-9	9-10	9-10	9-11	10-11	10-12	11-12	11-13	11-13	12-14
n	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**
t_2[n, 6]	12-14	13-15	13-15	14-16	14-16	14-17	15-17	15-18	16-18	16-19
n	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**
t_2[n, 6]	17-19	17-20	17-20	18-21	18-21	19-22	19-22	20-23	20-23	20-24
n	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**		
t_2[n, 6]	21-24	21-25	22-25	22-26	23-26	23-27	23-27	24-28		

The values of $t_2[n, k]$ for codes of dimensions up to 5 are determined in [2] and also an upper bounds for $t_2[n, k]$ for codes of dimensions 6 and 7 are derived. Namely, in [2, Theorem 23] it is proved that

$$t_2[n, 6] \leq \left\lfloor \frac{n-8}{2} \right\rfloor \text{ for } n \geq 18, \text{ and } t_2[n, 7] \leq \left\lfloor \frac{n-9}{2} \right\rfloor \text{ for } n \geq 19.$$

In this work we show that the bound for codes of dimension 6 is sharp. The approach we use is similar to the approach from [2] and it is based on the determination of the covering radii of the projective codes of dimension 6. We will note that the covering radii of the binary projective codes of dimensions up to 5 are determined in [3] and [4].

Theorem 3. $t_2[n, 6] = \left\lfloor \frac{n-8}{2} \right\rfloor$, *for* $n \geq 18$.

Proof. For codes of lengths 18-21 values of $t_2[n, 6]$ are known and they fulfill the condition of the Theorem. For the rest of the codes of lengths up to 64 the upper bounds from the Table coincide with the value given in the theorem. What remains is to prove that these upper bounds are sharp. Let us consider the first open case [22, 6] codes. If a [22, 6] code C contains a repeated coordinate, then $R(C) \geq t_2[20, 6] + 1 = 7$. Thus, if there exits a [22, 6] code of covering radius 6 it must be a projective one. Classification of all binary projective codes of dimension up to 6 is done in [5]. We use the results of this classification where 2852541 nonequivalent binary [22, 6] codes are found, to show that there is no code of covering radius 6 among them. Therefore $t_2[22, 6] = 7$. Let now C be a [24, 6] code. The same reasoning shows that

$t_2[24,6] = 8$ and as $t_2[25,6] \geq t_2[24,6]$ we get $t_2[25,6] = 8$. We repeat this until $t_2[54,6] = t_2[55,6] = 23$.

Let C be a $[56,6]$ code. If it contains a repeated coordinate, then $R(C) \geq t_2[54,6] + 1 = 23 + 1 = 24$. Otherwise, C is a shortened version of the $[63,6]$ Simplex code whit covering radius 31 and thus $R(C) \geq 31 - 7 = 24$. Therefore $t_2[56,6] = 24$ and $t_2[57,6] = 24$. Similarly $t_2[58,6] = t_2[59,6] = 25$, $t_2[60,6] = t_2[61,6] = 26$ and $t_2[62,6] = t_2[63,6] = 27$.

For $n \geq 64$, every $[n,6]$ code must contain a repeated coordinate and $t_2[n,6] \geq t_2[n-2,6]+1$, which implies $t_2[n,6] \geq \lfloor (n-8)/2 \rfloor$ for all n. Therefore the upper bound is sharp, which completes the proof. ◇

The other aim of our investigation is to construct codes having covering radii equal to the least one. Here we will show how to do this. It is proved in [6] that the constructed in [2] $[14,6,5]3$ code is unique. A $[16,6]4$ or $[18,6]5$ code can be obtained by the $[14,6]3$ code by adding repeated coordinates. Again in [2] the generator matrix of $[19,6,7]5$ code is presented. As $[19,6]5$ codes must be projective ($t_2[19,5] \geq t_2[17,6] + 1 = 6$), we use the classification from [5] to determine the covering radii of all 366089 projective $[19,6]$ codes. It turned out that there is only one code with covering radius 5 and therefore the $[19,6,7]5$ code is unique. Then every $[n,6]$ code for $n > 19$ having the least covering radius can be obtained from the $[18,6]5$ or $[19,6]5$ codes by adding the necessary number of repeated coordinates. By adding repeated coordinates to the $[9,6]1$ code we can obtain $[11,6]2$, $[13,6]3$, $[15,6]4$ and $[17,6]5$ codes, and to the $[8,6]1$ code we can get $[10,6]2$ and $[12,6]3$ codes. We classify all $[8,6]$ and $[9,6]$ codes and among the 25 $[8,6]$ and 99 $[9,6]$ nonequivalent codes there are correspondingly 16 and 4 of covering radius 1.

4 A heuristic algorithm for lower bound of the covering radius of a linear code

In the proof of Theorem 1 we use a computer to show the nonexistence of codes of lengths $22 \leq n \leq 54$, n odd, and given covering radius. There are 236779414 such codes and if we try to determine their covering radii using one of the known for us algorithms it would take years. Here we present a heuristic algorithm which alows us to show the nonexistence of an $[n,k]R$ code C in a reasonable time.

The idea of the algorithm is as fast as possible to find a coset leader of the investigated code of weight greater than R, which means that the covering radius of the code is at least $R+1$. It starts with a randomly chosen vector c from a coset $K_c = \{c + C\}$. We use the evaluation function f to find the current best solution, where the aim is to minimize the number of vectors of minimum weight in the coset. The function $f = wt(K_c)2^k - A(K_c)$ depends on the weight of the coset $wt(K_c)$ and the number $A(K_c)$ of vectors of minimum eight in K_c. Then we search in the set of neighbors $N(c)$ consisting of vectors which differ from c in one coordinate. If in this procedure we obtain a coset of weight greater than R, we are done. Otherwise, we add some noise to c and again try to find a coset of weight greater than R.

Algorithm. LowerBoundCoveringRadius(R_{min})

```
c, c': vector;
br_0, br: integer;
{
  $br_0 := 0$;
  while $br_0 < const_0$
    {
      $br_0 := br_0 + 1$;
      $br := 0$;
      Select a feasible solution $c$;
      while $br < const$
        {
          $br := br + 1$;
          while exists $c' \in N(c)$ such that $f(c') > f(c)$ do  $c := c'$;
          if $wt(K_c) > R_{min}$  break;
          Add some noise to $c$;
        }
    }
}
```

5 Upper bounds for the covering radii of linear codes of dimensions 8 and 9

Theorem 4. $t_2[n,8] \le \left\lfloor \dfrac{n-10}{2} \right\rfloor$ *for* $n \ge 16$ *and* $t_2[n,9] \le \lfloor \frac{n-12}{2} \rfloor$ *for* $n \ge 25$.

Proof. Let us consider the direct sum of two $[9,4]2$ normal codes. According to Theorem 9 from [2] we obtain $[18,8]4$ normal code and the existence of $[18+2i,8]4+i$ codes for $i > 0$ follows from [2, Theorem 20]. The same way the direct sum of $[8,4]2$ and $[9,4]2$ normal codes gives $[17,8]4$ normal code and there exist $[17+2i,8]4+i$ codes for $i > 0$. From [1, Table 7.1] we have $t_2[16,8] = 3$ which completes the proof for the first upper bound.

Let us now consider the amalgamated direct sum of $[7,4]1$ and $[14,6]3$ normal codes. The result is a $[20,9]4$ code which according to Theorem 2 is normal. A $[25,9]6$ normal code can be obtained by an ADS of $[7,4]1$ code and the constructed in [2] and proved to be unique in this work $[19,6]5$ code. As in the previous case, we can conclude that the $[20+2i,9]4+i$ and $[25+2i,9]6+i$ codes for $i>0$ exist and the upper bound for $t_2[n,9]$ follows. □

References

[1] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering Codes*, North-Holland, Elsevier Science B.V., 1997.

[2] R. L. Graham, N. J. A. Sloane, On the covering radius of codes, *IEEE Trans. Inform. Theory* 31, 1985, 385-401.

[3] N. J. A. Sloane, A new approach to the covering radius of codes, *J. Combin. Theory* Ser. A 42, 1986, 61-86.

[4] K. E. Kilby, N. J. A. Sloane, On the covering radius problem for codes: I Bounds on normalized covering radius, II Codes of low dimension; normal and abnormal codes, *SIAM J. Algebr. Discr. Methods* 8, 1987, 604-627.

[5] I. Bouyukliev, On the binary projective codes with dimension 6, *Discr. Appl. Math.* 154, 2006, 1693-1708.

[6] T. Baicheva, V. Vavrek, On the least covering radius of binary linear codes with small lengths, *IEEE Trans. Inform. Theory* 49, 2003, 738-740.

Fast computing of the positive polarity Reed-Muller transform over $GF(2)$ and $GF(3)$

VALENTIN BAKOEV v_bakoev@yahoo.com
University of Veliko Turnovo "St. Cyril and St. Methodius", BULGARIA

KRASSIMIR MANEV[1] manev@fmi.uni-sofia.bg
Faculty of Mathematics and Informatics, Sofia University, BULGARIA and
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
8 G. Bonchev str., 1113 Sofia, BULGARIA

Abstract. The problem of efficient computing of binary and ternary positive (or zero) polarity Reed-Muller (PPRM) transform is important for many areas. The matrices, determining these transforms, are defined recursively or by Kronecker product. Using this fact, we apply the dynamic-programming strategy to develop three algorithms. The first of them is a new version of a previous our algorithm for performing the binary PPRM transform. The second one is a bit-wise implementation of the first algorithm. The third one performs the ternary PPRM transform. The last two algorithms have better time complexities in comparison with other algorithms, known to us.

1 Introduction

A well-known theorem in the theory of Boolean functions states that any Boolean function $f(x_{n-1}, x_{n-2}, \ldots, x_0)$ can be represented in an unique way by its *Zhegalkin polynomial*:

$$\begin{aligned} f(x_{n-1}, x_{n-2}, \ldots, x_0) &= a_0 \oplus a_1x_0 \oplus a_2x_1 \oplus a_3x_1x_0 \oplus \ldots \\ &\oplus \quad a_ix_{j_1}x_{j_2}\ldots x_{j_k} \oplus \cdots \oplus a_{2^n-1}x_{n-1}x_{n-2}\ldots x_0, \end{aligned} \tag{1}$$

where the coefficients $a_i \in \{0,1\}, 0 \leq i \leq 2^n-1$, $i = 2^{j_1} + 2^{j_2} + \cdots + 2^{j_k}, j_1 > j_2 > \cdots > j_k$, and all variables are *positive* (uncomplemented). This canonical form is also known as *Positive Polarity Reed-Muller* (PPRM) expansion. If each variable x_i, $0 \leq i \leq n-1$, in (1) appears either uncomplemented, or complemented throughout, we obtain a *Fixed Polarity Reed-Muller* (FPRM) expansion. Let $p_i \in \{0,1\}$ denotes the polarity of x_i, $0 \leq i \leq n-1$, i.e. when $p_i = 0$ the polarity is positive (x_i is uncomplemented), and when $p_i = 1$ the

[1] This work was partially supported by the SF of Sofia University under Contract 171/05.2008.

polarity is negative (x_i is complemented). The function $f(x_{n-1}, x_{n-2}, \ldots, x_0)$ has a FPRM expansion of polarity $p, 0 \leq p \leq 2^n - 1$, when the integer p has a n-digit binary representation $p_{n-1}, p_{n-2} \ldots p_0$ and p_i is the polarity of x_i, for $i = n-1, n-2, \ldots, 0$. Thus f has 2^n FPRM possible expansions, each of them is a canonical form.

The FPRM binary transform is an important and known XOR-based expansion, having many applications in digital logic design, testability, fault detection, image compression, Boolean function decomposition, error correcting codes, classification of logic functions, and development of models for decision diagrams [2, 4, 5]. Because of the increasing interest in multiple-valued logic (MVL), the binary FPRM expansion has been extended to represent multiple-valued functions as well. Their FPRM expansions have also many applications in the just mentioned areas.

Every ternary function $f(x)$ of n-variable can also be represented by its canonical FPRM polynomial expansions as follows:

$$f_p(x_{n-1}, x_{n-2}, \ldots, x_0) = \sum_{i=0}^{3^n-1} a_i.\hat{x}_{n-1}^{k_{n-1}} \hat{x}_{n-2}^{k_{n-2}} \ldots \hat{x}_0^{k_0}, \qquad (2)$$

where:

- all additions and multiplications are in $GF(3)$;
- i is the decimal equivalent of the n-digit ternary number $k_{n-1}k_{n-2}\ldots k_0$;
- $\hat{x}_j = x_j + p_j \in \{x_j, x_j + 1, x_j + 2\}$ is the literal of the j-th variable, in dependence of the polarity p_j. The required polarity is given (fixed) by the integer p, $0 \leq p \leq 3^n - 1$, which n-digit ternary representation is $p_{n-1}, p_{n-2} \cdots p_0$;
- the coefficient $a_i \in \{0, 1, 2\}$, $a_i = a_i(p)$ because it depends on the given polarity p;
- $\hat{x}_j^0 = 1$, $\hat{x}_j^1 = \hat{x}_j$ and $\hat{x}_j^2 = \hat{x}_j.\hat{x}_j$.

Optimization of FPRM transforms is an important problem in the area of logic design and spectral transforms. It concerns development of methods for determining the best FPRM representation of a given function among all possible FPRM expansions of it. The best is this one, which has minimal number of product terms or minimal number of literals. There are many approaches to perform such optimization.

Here we consider the problem: *"A Boolean (or ternary) function is given by its vector of functional values. Compute the vector of coefficients of its PPRM expansion"*. We represent three algorithms for fast solving of this problem. They can be used for computing of the rest FPRM expansions of a given function, as do this the algorithms and method in [4, 8]. The main idea of the proposed algorithms can be extended and applied for obtaining of other FPRM expansion, for computing PPRM expansions of MVL functions over other finite fields, and also for fast computing of matrix-vector multiplication when the matrix is defined recursively (by Kronecker product).

2 Binary PPRM transform

Many scientists investigate the computing of binary FPRM transform – by applying of coefficient maps (Karnaugh maps folding, when the number of variables $n \leq 6$), coefficient matrix and tabular techniques [1, 6, 8, 10, 11]. All they consider algorithms for computing the PPRM transform in particular, and most of them apply a coefficient matrix approach. Let f be a n-variable boolean function, given by its vector of values $b = (b_0, b_1, \ldots, b_{2^n-1})$. The *forward* and *inverse* PPRM transform between the coefficient vector $a = (a_0, a_1, \ldots, a_{2^n-1})$ of Eq. (1) and the vector b is defined by the $2^n \times 2^n$ matrix M_n as follows [6, 9, 10]:

$$a^T = M_n.b^T, \quad \text{and} \quad b^T = M_n^{-1}.a^T \quad \text{over } GF(2). \tag{3}$$

The matrix M_n is defined recursively, as well as by Kronecker product:

$$M_1 = \begin{pmatrix} 1\,0 \\ 1\,1 \end{pmatrix}, \quad M_n = \begin{pmatrix} M_{n-1}\; O_{n-1} \\ M_{n-1}\; M_{n-1} \end{pmatrix}, \text{ or } \quad M_n = M_1 \otimes M_{n-1} = \bigotimes_{i=1}^{n} M_1, \tag{4}$$

where M_{n-1} is the corresponding transform matrix of dimension $2^{n-1} \times 2^{n-1}$, and O_{n-1} is a $2^{n-1} \times 2^{n-1}$ zero matrix. Furthermore $M_n = M_n^{-1}$ over $GF(2)$, and hence the forward and the inverse transform are performed in an uniform way. So we shall consider only the forward one. In all papers known to us, there are not complete description of the algorithm for computing of such transform, defined by equalities (3) and (4). These equalities are derived in [6] (Theorem 2) and computing of the transform is illustrated by an example, almost the same is done in [9]. In [1] some equalities, which concern computing of the coefficients of the vector a and relations between them, are derived. Computing of the PPRM transform in [10] is illustrated by its "butterfly" (or "signal flow") diagram only.

Ten years ago we have proposed an algorithm for fast computing the PPRM transform (called by as "Zhegalkin transform") [7]. We developed this algorithm independently of other authors, because their papers in this area were unknown (unaccessible) to us at this time. Here we propose another version of this algorithm, created by the dynamic-programming approach. We also comment its bit-wise implementation, which improves significantly the previous time and space complexity. The same approach will be applied for fast computing of the PPRM transform over $GF(3)$.

Let v be a vector, $v \in \{0,1\}^{2^n}$. We could consider each position of the vector v *labeled* with the corresponding vector of $\{0,1\}^n$, so that the labels are ordered lexicographically. Let $\alpha \in \{0,1\}^k$, $1 \leq k < n$. We will denote by $v_{[\alpha]}$ the sub-vector of these positions in v, first k-coordinates of labels of which are fixed to α. We can rewrite Eq. (3) as follows:

$$a^T = M_n.b^T = \begin{pmatrix} M_{n-1} \; O_{n-1} \\ M_{n-1} \; M_{n-1} \end{pmatrix} \begin{pmatrix} b^T_{[0]} \\ b^T_{[1]} \end{pmatrix} \tag{5}$$
$$= \begin{pmatrix} M_{n-1}.b^T_{[0]} \\ M_{n-1}.b^T_{[0]} \oplus M_{n-1}.b^T_{[1]} \end{pmatrix} = \begin{pmatrix} a^T_{[0]} \\ a^T_{[1]} \end{pmatrix}.$$

Therefore:

$$\begin{aligned} a^T_{[0]} &= M_{n-1}.b^T_{[0]}, \\ a^T_{[1]} &= M_{n-1}.b^T_{[0]} \oplus M_{n-1}.b^T_{[1]} = a^T_{[0]} \oplus M_{n-1}.b^T_{[1]}. \end{aligned} \tag{6}$$

The last two equalities *define recursively* the solution of the problem. They demonstrate how it can be constructed by the solutions of its subproblems. So the problem exhibits the *optimal substructure property* – the first key ingredient for applying the dynamic-programming strategy. The second one – *overlapping subproblems* – is also shown in (6). If we are computing a recursively, we have to compute first $a_{[0]}$ (recursively). Then we have to compute $a_{[1]}$ (recursively) and this will imply computing of $a_{[0]}$ again.

To apply the dynamic-programming strategy we will replace the recursion by an iteration and will compute the vector a *"bottom-up"*. The main idea can be drawn from last two equalities – if we make one more step, expressing M_{n-1} by M_{n-2} and replacing $a_{[0]}$ by $(a_{[00]}, a_{[01]})$, $a_{[1]}$ by $(a_{[10]}, a_{[11]})$, $b_{[0]}$ by $(b_{[00]}, b_{[01]})$, $b_{[1]}$ by $(b_{[10]}, b_{[11]})$, and so on. We conclude that the iteration should perform n steps. Starting from the vector b (as an input), at k-th step, $k = 1, 2, \ldots, n$, we consider the current vector b as divided into two kinds of blocks: *source* and *target*, which alternate with each other. All they have a *size*, equal to 2^{k-1}. At each step, every source block is *added* (by a component-wise XOR) to the next block, which is its target block. The result is assigned to the current vector b. So, after these n steps, the vector b is transformed to the vector a. Assuming that the vector b is represented by an array b of 2^n bytes, the pseudo code of this algorithm is:

```
Binary_PPRM (b, n)
blocksize = 1;
for k = 1 to n do
    source = 0;   //start of the source block
    while source < 2^n do
        target = source + blocksize; //start of the target block
        for i = 0 to blocksize - 1 do
            //component-wise XOR over current blocks
            b[target + i] = b[target + i] XOR b[source + i];
```

```
          //start of the next source block
8)        source = source + 2 * blocksize;
9)     blocksize = 2 * blocksize;
10) return b;                             //b is transformed to a
```

The correctness of the algorithm can be proved easily by induction on n. In its k-th step, $1 \leq k \leq n$, there are 2^{n-k} source blocks and so many target blocks, each of size 2^{k-1}. The algorithm adds (XORs) these source blocks to the corresponding target blocks, and so it performs $2^{k-1}.2^{n-k} = 2^{n-1}$ XORs in the k-th step. Therefore, when the input size is 2^n, the algorithm has a time complexity $\Theta(n.2^{n-1})$ and $\Theta(2^n)$ space complexity. They are many times better than the corresponding complexities, which we shall obtain if we generate the matrix M_n and compute directly the matrix-vector multiplication, given by Eq. (3).

Now we discuss a new version of the given algorithm, obtained by applying a bit-wise representation of the vector b and bit-wise operations. Let $d = 2^j$ be the size (in bits) of the computer word. Then $m = \lceil 2^{n-j} \rceil$ computer words are sufficient to represent the vector b. For simplicity, let $n = j$ (i.e. $m = 1$), and we denote by B the representation of b as a binary number. We use an additional integer *temp*, initialized by $temp = B$. In *temp* we set the values in the target blocks to zero – i.e. we mask them by zeros, and the values in the source blocks we remain the same – we mask them by ones. For that purpose, in the k-th step ($k = 1, 2, \ldots n$) we should use a mask: `mask[k]`$=$ $\underbrace{11\ldots1}_{2^{k-1}}\underbrace{00\ldots0}_{2^{k-1}}\cdots\underbrace{11\ldots1}_{2^{k-1}}\underbrace{00\ldots0}_{2^{k-1}}$, where 2^{k-1} is the block size. After that, we "shift right" them by 2^{k-1} positions and so the source blocks are moved to the places of the target blocks, corresponding to them. Finally, we compute a bit-wise XOR between B and *temp* and store the result in B. So, the body of the main cycle in row 2 of the given above pseudo code (i.e. the rows 3, 4,. . . , 9) could be replaced by:

```
3)   temp = B AND mask[k];          //masks the blocks;
4)   temp = temp SHR blocksize;     //shift right
5)   B = B XOR temp;                //XOR between all blocks.
6)   blocksize = blocksize SHL 1; //double the blocksize
```

We have only four bit-wise operations, repeated n times. Therefore the time complexity of this version of the algorithm is $\Theta(n)$. The array `mask` consists of n computer words and they can be pre-computed once in $\Theta(n^2)$ time and not considered as a part of algorithm. When $n > j$ then $m = 2^{n-j} > 1$ words of memory will be necessary. In this case, during the steps $1, 2, \ldots, j$, the instructions 3, 4 and 5 of the algorithm will be executed for each word separately. During the steps $j+1, j+2, \ldots, n$ the masks are no more necessary, because the blocks are composed of whole words. In such way only $m/2$ XOR operations will be necessary on each of these steps. Finally, the time complexity becomes $\Theta(m.n)$ generally, which is the best one, known to us.

To compare these two versions we have generated all Boolean functions of 5 variables and perform the PPRM transform over each of them. When the new version of the algorithm uses a 32-bit computer word and generates the masks only once, it runs 22 times faster.

3 Ternary PPRM transform

The ternary FPRM and some other transforms are investigated intensively by Falkowski, Fu, etc. [2, 3, 4, 5]. These transforms are determined by the corresponding matrices, defined recursively or by Kronecker product. These matrices are used for building *recursive* algorithms, performing these expansions. Computing of the ternary PPRM transform is an important part for some of them or for other fast algorithms [4]. Let $f(x_{n-1}, x_{n-2}, \ldots, x_0)$ be a ternary function, represented by its vector of values $b = (b_0, b_1, \ldots, b_{3^n-1})$. Analogously to the binary case, the ternary *forward* PPRM transform between the coefficient vector $a = (a_0, a_1, \ldots, a_{3^n-1})$ and the vector b is defined by the $3^n \times 3^n$ matrix T_n as follows [2, 3, 4]:

$$a^T = T_n.b^T \text{ over } GF(3). \tag{7}$$

The matrix T_n is defined recursively, or by Kronecker product:

$$T_1 = \begin{pmatrix} 1\,0\,0 \\ 0\,2\,1 \\ 2\,2\,2 \end{pmatrix}, \quad T_n = \begin{pmatrix} T_{n-1} & O_{n-1} & O_{n-1} \\ O_{n-1} & 2.T_{n-1} & T_{n-1} \\ 2.T_{n-1} & 2.T_{n-1} & 2.T_{n-1} \end{pmatrix}, \text{or} \quad T_n = T_1 \otimes T_{n-1} = \bigotimes_{i=1}^{n} T_1, \tag{8}$$

where T_{n-1} is the corresponding transform matrix of dimension $3^{n-1} \times 3^{n-1}$, and O_{n-1} is a $3^{n-1} \times 3^{n-1}$ zero matrix. It is easy to see that $T_n \neq T_n^{-1}$, and so the forward and inverse ternary PPRM transforms do not coincide.

Let v be a vector, $v \in \{0,1,2\}^{3^n}$. We consider each position of v *labeled* with the corresponding vector of $\{0,1,2\}^n$, so that the labels are ordered lexicographically. Let the vector $\alpha \in \{0,1,2\}^k$ and $1 \leq k < n$. We denote by $v_{[\alpha]}$ the sub-vector of these positions in v, first k-coordinates of labels of which are fixed to α. Using Eq. (8), we rewrite Eq. (7) as:

$$\begin{aligned} a^T = T_n.b^T &= \begin{pmatrix} T_{n-1} & O_{n-1} & O_{n-1} \\ O_{n-1} & 2.T_{n-1} & T_{n-1} \\ 2.T_{n-1} & 2.T_{n-1} & 2.T_{n-1} \end{pmatrix} \begin{pmatrix} b^T_{[0]} \\ b^T_{[1]} \\ b^T_{[2]} \end{pmatrix} = \\ &= \begin{pmatrix} T_{n-1}.b^T_{[0]} & & \\ & 2.T_{n-1}.b^T_{[1]} & +\; T_{n-1}.b^T_{[2]} \\ 2.T_{n-1}.b^T_{[0]} & +2.T_{n-1}.b^T_{[1]} & +2.T_{n-1}.b^T_{[2]} \end{pmatrix} = \begin{pmatrix} a^T_{[0]} \\ a^T_{[1]} \\ a^T_{[2]} \end{pmatrix} \text{ over } GF(3). \end{aligned} \tag{9}$$

Therefore:

$$\begin{aligned} a^T_{[0]} &= T_{n-1}.b^T_{[0]} \\ a^T_{[1]} &= 2.T_{n-1}.b^T_{[1]} + T_{n-1}.b^T_{[2]} \\ a^T_{[2]} &= 2.(T_{n-1}.b^T_{[0]} + T_{n-1}.b^T_{[1]} + T_{n-1}.b^T_{[2]}) \end{aligned} \tag{10}$$

The last equalities determine the solution recursively. The reasons to apply the dynamic-programming strategy are the same as in the binary case. The final solution can be obtained by the solutions of its subproblems (i.e. when the matrix-vector multiplications of the type $T_{n-1}.b^T_{[i]}$ are already computed) by 3 additions of vectors and 2 multiplications of vector by a scalar in $GF(3)$. Thinking about them as source and target blocks, we shall replace them by 4 additions between blocks in $GF(3)$, as it is shown in Fig. 1, for $n = 1$. Obviously, some source and target blocks (of size 1, when $n = 1$) change their roles.

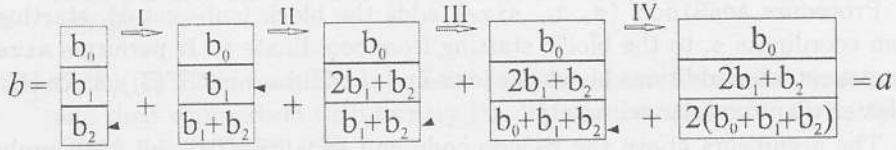

Figure 1: For $n = 1$, vector b is transformed to vector a by 4 additions in $GF(3)$.

The same model of computing will be valid if we expand the equalities (9) and (10) completely for $n = 2$. In the first step we apply the scheme of additions in Fig. 1 for each of sub-vectors $b_{[0]}, b_{[1]}$ and $b_{[2]}$. In the second step we consider the resulting sub-vectors as blocks of size 3, labeled by $0, 1$, and 2, respectively. We compute component-wise additions between the blocks, following the scheme in Fig. 1 and so we obtain the vector a.

We can extend this model of computing for an arbitrary n. Thus we obtain an algorithm, which starts from the given vector b (as an input) and performs n steps. At each step, the current vector b (as a result of a previous step) is divided into blocks of size 3^{k-1}, where k is the number of the step. The blocks are labeled by $0, 1, \ldots, 3^{n-k+1}$. For each triple of consecutive blocks the algorithm performs component-wise additions (in $GF(3)$) between the blocks in the triple, following the scheme in Fig. 1. So, before the last step, the sub-vectors (blocks) $T_{n-1}.b^T_{[i]}$, labeled by $i = 0, 1, 2$, are already computed. In the last step, the algorithm performs the additions between the blocks in the last triple, as they are given in Fig. 1, and so it obtains the vector a. If the vector b is represented by an array b of 3^n bytes, the pseudo code of this algorithm is:

```
Ternary_PPRM (b, n)
blocksize = 1;
for k = 1 to n do
    base = 0;   //start of the 0-blocks in a current triple
    while base < 3^n do
        first = base + blocksize;                //start of 1-block
        second = first + blocksize;              //start of 2-block
        AddBlock(first,second,blocksize );       //adds 1-bl. to 2-bl.
        AddBlock(second,first,blocksize );       //adds 2-bl. to 1-bl.
        AddBlock(base,second,blocksize );        //adds 0-bl. to 2-bl.
        AddBlock(second,second,blocksize );      //adds 2-bl. to itself
        base = base + 3*blocksize;               //start next triple
    blocksize= 3*blocksize;
return b;      //b is transformed to a
```

Procedure `AddBlock (s, t, size)` adds the block (sub-vector), starting from coordinate `s`, to the block, starting from coordinate `t`. It performs `size` component-wise additions by a table look-up (of additions in $GF(3)$), since this is faster than modular arithmetic.

The arguments above the pseudo code and equalities (9) and (10) imply the correctness of the algorithm. Following them, it can be proved strongly by induction on n. The space complexity of the algorithm is $\Theta(3^n)$, the same as the size of input. Its time complexity is derived easily. In the k-th step, $1 \leq k \leq n$, the size of the blocks is 3^{k-1}, and for each triple of blocks the algorithm performs 4.3^{k-1} additions. There are $3^n/(3.3^{k-1}) = 3^{n-k}$ triples, and so the additions in the k-th step are $4.3^{k-1}.3^{n-k} = 4.3^{n-1}$. Therefore the time complexity is $\Theta(n.3^{n-1})$. For comparison, in [4] the authors refer to an algorithm for fast computing of ternary PPRM transform, which performs $n.3^n$ additions and $4n.3^{n-1}$ multiplications.

The matrix T_n^{-1} can be expressed by equalities analogous to these in (8), hence the inverse transform can be performed in way, similar to the performing of forward transform.

4 Conclusions

Here we have used the dynamic-programming strategy to develop three algorithms. They are based on matrices, defined recursively or by Kronecker product, which determine the PPRM transforms over $GF(2)$ and $GF(3)$. The model of building the given algorithms can be extended and applied for fast computing of other FPRM expansions over the considered fields, for other finite fields with prime number of elements, or for fast computing of matrix-vector multiplication when the matrix is defined recursively. Proposed algorithms have better time complexities in comparison with other algorithms, known to us.

References

[1] A. Almaini, P. Thomson, D. Hanson, Tabular techniques for Reed-Muller logic, *Int. J. Electronics* 70, 1991, 23-34.

[2] B. Falkowski, C. Fu, Fastest classes of linearly independent transforms over GF(3) and their properties, *IEE Proc. Comput. Digit. Tech.* 152, 2005, 567-576.

[3] B. Falkowski, C. Fu, Polynomial expansions over GF(3) based on fastest transformation, *Proc. 33-rd Intern. Symp. Mult.-Val. Logic*, 2003, 40-45.

[4] B. Falkowski, C. Lozano, Column polarity matrix algorithm for ternary fixed polarity Reed-Muller expansions, *J. Circ., Syst., Comp.* 15, 2006, 243-262.

[5] C. Fu, B. Falkowski, Ternary fixed polarity linear Kronecker transforms and their comparison with ternary Reed Muller transform, *J. Circ., Syst., Comp.* 14, 2005, 721-733.

[6] B. Harking, Efficient algorithm for canonical Reed-Muller expansions of Boolean functions, *IEE Proc. Comput. Digit. Tech.* 137, 1990, 366-370.

[7] K. Manev, V. Bakoev, Algorithms for performing the Zhegalkin tranformation, *Proc. XXVII Spring Conf. UBM*, 1998, 229-233.

[8] M. Perkowski, L. Jozwiak, R. Drechsler, A canonical AND/EXOR form that includes both the generalized Reed-Muller forms and Kronecker Reed-Muller forms, *Proc. RM'97*, Oxford Univ., 1997, 219-233.

[9] P. Porwik, Efficient calculation of the Reed-Muller form by means of the Walsh transform, *Int. J. Appl. Math. Comput. Sci.* 12, 2002, 571-579.

[10] S. Rahardja, B. Falkowski, C. Lozano, Fastest linearly independent transforms over GF(2) and their properties, *IEEE Trans. Circuts Syst.* 52, 2005, 1832-1844.

[11] E. Tan, H. Yang, Fast tabular technique for fixed-polarity Reed-Muller logic with inherent parallel processes, *Int. J. Electr.* 85, 1998, 511-520.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 22-25*

New bounds for multiple packings of Euclidean sphere

Vladimir Blinovsky[1] vblinovs@yandex.ru
Institute of Information Transmission Problems,
B. Karetnyi, 19, Moscow, RUSSIA
Simon Litsyn[2] litsyn@eng.tau.ac.il
School of Electrical Engineering, Tel Aviv University,
Ramat Aviv, 69978 ISRAEL

Abstract. Using lower bounds on distance spectrum components of a code on the Euclidean sphere, we improve the known asymptotical upper bounds on the cardinality of multiple packings of the sphere by balls of smaller radii.

Let $\mathbb{R}^n$ be the n-dimensional Euclidean space, and $S^{n-1}(r) \subset \mathbb{R}^n$ be the (closed) Euclidean sphere of radius r with the center in the origin. Let further $\tilde{S}^{n-1}(r, \bar{a})$ be the open ball of radius r centered in $\bar{a} \in \mathbb{R}^n$. Multiple L-packing $\mathcal{K}(L, t)$ by balls of radius t is a finite set (≡code) $\mathcal{K} \subset S^{n-1}(1)$, such that for any subset $\{\bar{x}_1, \ldots, \bar{x}_{L+1}\} \subset \mathcal{K}$ of $L+1$ points (≡codewords) we have

$$\bigcap_{i=1}^{L+1} \tilde{S}^n(t, \bar{x}_i) = \emptyset.$$

In other words, any point on the unit sphere can be at distance not exceeding t from at most L points from $\mathcal{K}$.

Let $R(\mathcal{K}(L,t)) = \frac{\ln|\mathcal{K}(L,t)|}{n}$ be the rate of the multiple packing. The problem is to find bounds on the value

$$R_L(t) = \limsup_{n\to\infty} \max R(\mathcal{K}(L,t)).$$

The value $R_L(t)$ has been studied before, e.g. in connection with list decoding in Gaussian channel, see [1, 2] and references therein. The best known bounds are as follows.

Theorem 1

$$R_L(t) \geq \frac{1}{2} \ln \frac{L}{(L+1)t^2} + \frac{1}{2L} \ln \frac{1}{(L+1)(1-t^2)}. \tag{1}$$

$$R_L(t) \leq \frac{1}{2} \ln \frac{L}{(L+1)t^2}. \tag{2}$$

[1] Supported in part by RFBR grant 06-01-00226
[2] Supported in part by ISF 553-03 grant

Bound (1) was proved in [1], while bound (2) was first proved in [2] and later in [1] using an essentially different approach. In this work we present further improvement of the upper bound (2).

Throughout we identify a point in $\mathbb{R}^n$ with the vector from the origin to the point. Denote

$$\begin{aligned} \alpha_\varphi(\theta) &= 2\arcsin\frac{\sin(\theta/2)}{\sin\varphi}, \quad \beta_\varphi(\theta) = \arccos\frac{\cos\varphi}{\cos(\theta/2)}, \\ j(x,y) &= (1+y)H\left(\frac{y}{y+1}\right) - \ln\left(\frac{1}{2}(x+\sqrt{(1+2y)^2x^2-4y(1+y)}\right) \\ &\quad +(1+2y)\ln\frac{(1+2y)x+\sqrt{(1+2y)^2x^2-4y(1+y)}}{2(1+y)}, \end{aligned}$$

where $H(z) = -z\ln z - (1-z)\ln(1-z)$, $z \in [0,1]$. For a given R denote by ρ_L the unique solution of

$$R = (1+\rho)H\left(\frac{\rho}{1+\rho}\right),$$

and denote by θ_L the unique solution of

$$R = R(\theta_L) = \frac{1+\sin\theta_L}{2\sin\theta_L}H\left(\frac{1-\sin\theta_L}{1+\sin\theta_L}\right).$$

Note that

$$\frac{2\sqrt{\rho_L(\rho_L+1)}}{1+2\rho_L} = \cos\theta_L,$$

and if θ is the minimal angle between a pair of points from $\mathcal{K} \subset S^{n-1}(1)$, the rate of this code satisfies [3]:

$$R \le R(\theta). \tag{3}$$

Denote also

$$b_{\mathcal{K}}(x,\bar{c}) = \left|\left\{\bar{c}_1 : (\bar{c},\bar{c}_1) \in \mathcal{K}\times\mathcal{K} : (\bar{c},\bar{c}_1)/(\|\bar{c}\|\cdot\|\bar{c}_1\|) = x\right\}\right|,$$

where $(\bar{a},\bar{b}) = a_1b_1 + \ldots + a_nb_n$ stands for the scalar product. We will use the following result from [4].

Theorem 2 *For $\mathcal{K} \subset S^{n-1}(r)$, with $\ln|\mathcal{K}| = Rn(1+o(1))$, and ρ and ϕ satisfying*

$$0 \le \rho \le \rho_L(R+\sin\varphi), \quad e^{-R} \le \varphi \le \pi/2,$$

there exists θ. and $\bar{c} \in \mathcal{K}$, such that

$$2\sqrt{\rho(1+\rho)}/(1+2\rho) \le \cos\alpha_\varphi(\theta) \le 1 \quad .$$

and

$$\frac{1}{n}b_{\mathcal{K}}(\cos\theta, \bar{c}) \geq R + 2\ln\sin\varphi + \ln\sin\beta_\varphi(\theta) - j(\cos\alpha_\varphi(\theta), \rho).$$

It is easy to see that $b_{\mathcal{K}}(\cos\theta, \bar{c})$ is the number of codewords in the cross-section of the unit sphere by the hyperplane orthogonal to the vector $\bar{c}$, and intersecting $\bar{c}$ in the point $r \cdot \cos\theta \cdot \bar{c}/\|\bar{c}\|$.

Now we describe a recursive procedure of constructing a simplex of $L+1$ codewords having sufficiently small pairwise distances. We start with a code $\mathcal{K} \subset S^{n-1}(1)$ and consider the cross-section of $S^{n-1}(1)$ by the hyperplane. Then for each φ and ρ, as in Theorem 2, there exists θ, as in Theorem 2, such that for the cross-section of $S^{n-1}(1)$ by the hyperplane which is orthogonal to some vector $\bar{c}$ and being at distance $\cos\theta$ from origin, $b_{\mathcal{K}}(\cos\theta, \bar{c})$ satisfies (4). Note that the cross-section of the sphere by hyperplane is again a sphere of dimension $n-2$, having radius $r \cdot \sin\theta$ and centered in $r \cdot \cos\cdot\theta \cdot \bar{c}/\|\bar{c}\|$. Next we shift the center of this new sphere to the origin, and once again consider the cross-section of this sphere by the hyperplane as in the previous step. Now, the new code $\mathcal{K}_1$ has at least $b_{\mathcal{K}}(\cos\theta, \bar{c})$ codewords and its cross-section contains at least $b_{\mathcal{K}_1}(\cos\theta_1, \bar{c}_1)$ codewords. Notice that on the second step we choose new φ_1 and ρ_1. The procedure can be continued.

Let us provide a formal description of the procedure. On the 0-th step we have a code $\mathcal{K}_0 \subset S^{n-1}(1)$, $\ln|\mathcal{K}_0| \sim [nR]$. On $i-$th step, $i \geq 1$, we obtain a code $\mathcal{K}_i$ of rate

$$\ln|\mathcal{K}_i|/n \sim R_i \geq R_{i-1} + 2\ln\sin\varphi_i - \ln\beta_{\varphi_i}(\theta_i) - j(\cos\alpha_{\varphi_i}(\theta_i), \rho_i).$$

We implement this action $L+1$ times, and on the i-th step, $i \geq 1$, we find (if $R_i > 0$) a new codeword $\bar{c}_i$ such that its distance from $\bar{c}_j$, $0 \leq j < i$, is $d_j = 2r_j \cdot \sin\theta_j/2$. We stop when we fix $L+1$ codewords $\bar{c}_i \in \mathcal{K}_i \subset \mathcal{K}$, $i = 0, 1, \ldots, L$. Note also, that $\mathcal{K}_L \subset \mathcal{K}_{L-1} \subset \cdots \subset \mathcal{K}_0$. What should be done next, is to optimize the set d_j, $j = 0, 1, \ldots, L$, in such a way that the simplex $\bar{c}_i$, $i = 0, 1, \ldots, L$, to be contained in a closed ball of the minimum possible radius t. This means that there exists a point on $S^{n-1}(1)$ which is covered by $L+1$ balls, which yields that an arbitrary code of rate R on the Euclidean sphere is L-packing by the balls of radius strictly less than t. Note that it is necessary to optimize over φ_i and ρ_i in such a way that $R_i > 0$. It is unlikely that this can be done analytically, however this is an affordable task for the computer. One can easily derive the expression for t as a function of pairwise distances $d_{ij} = d_j = \|\bar{c}_i - \bar{c}_j\|, j < i$, see e.g. [5].

Consider the simplest case of $L = 2$. Let us have a code $\mathcal{K}_0 \subset S^{n-1}(1)$ of rate R_0. Set $\varphi_0 = \pi/2$, $\rho_0 = \rho_L - \varepsilon$, for some $\varepsilon \ll \rho_L$. The function $j(x, \rho)$ is increasing with $x \in [0, 1]$,

$$j(1, \rho) = (1+\rho)H\left(\frac{\rho}{\rho+1}\right).$$

Then for the rate R_1 and some $\theta_0 < \theta_L + \delta$,

$$R_1 \geq R - (1+\rho_L)H\left(\frac{\rho_L}{\rho_L+1}\right) + \epsilon_1 = \epsilon_1.$$

Next, for the code $\mathcal{K}_1 \subset S^{n-2}(\sin\theta_0)$ of rate R_1 we choose $\theta_1 = \pi/2$. We have $d_0 = 2\sin(\theta_0/2)$, $d_1 = \sqrt{2}\sin\theta_0$. The points $\bar{c}_0$, $\bar{c}_1$ and $\bar{c}_2$ are the vertices of a triangle with edges d_0, d_0 and d_1. The minimal radius t of the circle passing through these vertices is

$$t = \frac{d_0}{2\sqrt{1-\frac{d_1^2}{4d_0^2}}} = \frac{\sqrt{2}\sin(\theta_0/2)}{\sqrt{1+\sin^2(\theta_0/2)}}.$$

Then

$$\sin\theta_0 = \frac{2\sqrt{2}t\sqrt{1-t^2}}{2-t^2}.$$

Since $\theta_0 \sim \theta$ from (3), we obtain

$$R_2(t) \leq \frac{2-t^2+2\sqrt{2}\sqrt{1-t^2}}{4\sqrt{2}\sqrt{1-t^2}} H\left(\frac{1-\frac{2\sqrt{2}\sqrt{1-t^2}}{2-t^2}}{1+\frac{2\sqrt{2}\sqrt{1-t^2}}{2-t^2}}\right). \quad (4)$$

Comparing it to the specification of (2) for $L=2$,

$$R_2(t) \leq \frac{1}{2}\ln\frac{2}{3t^2},$$

we conclude that (4) is tighter.

References

[1] V. Blinovsky, Multiple packing of the Euclidean sphere, *IEEE Trans. Inform. Theory* 45, 1999, 1334-1337.

[2] N. Blachman, L. Few, Multiple packing of spherical caps, *Mathematika* 10, 1963, 84-88.

[3] G. Kabatyansky, V. Levenshtein, Bounds for packing on the sphere and in the space, *Probl. Inform. Transm.* 14, 3-25, 1978.

[4] Y. Ben-Haim, S. Litsyn, Improved upper bounds on the reliability function of Gaussian channel, *IEEE Trans. Inform. Theory* 54, 5-12, 2008.

[5] I. Bocharova, R. Johannesson, B. Kudryashov, M. Loncar, An improved bound on the list error probability and list distance properties, *IEEE Trans. Inform. Theory* 54, 2008, 13-32.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 26-28*

Enumeration of some optimal ternary constant-weight codes

Galina Bogdanova, Todor Todorov[1] galina, todor@math.bas.bg
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
P.O. Box 323, 5000 Veliko Tarnovo, BULGARIA

Abstract. We consider the problem of classification of optimal ternary constant-weight codes. We use combinatorial and computer methods to find inequivalent codes for some cases for $3 \leq d \leq n \leq 9$.

1 Introduction

A ternary (n, M, d) code consists of M vectors (called codewords) of length n over the alphabet $\{0,1,2\}$, such that any two codewords differ in at least d positions.

A code is called *constant weight* if all the codewords have the same Hamming weight. Constant weight codes have been studied by many authors [10],[11],[7],[2],[1].

We will use the following notation for the parameters of a ternary constant-weight (TCW)code: (n, M, d, w). Let $A_3(n, d, w)$ denote the largest possible value M, for which there exists an (n, M, d, w) code. TCW codes of size $M = A_3(n, d, w)$ are called *optimal*.

Initially, bounds and exact values of the function $A_3(n, d, w)$ were presented in [7] and the recent results may be found in [8]. In this paper we explore the problem of enumerating (up to equivalence) optimal TCW codes with $3 \leq d \leq n \leq 9$.

Combinatorial and computer methods can be used to classify optimal codes. Enumeration of TCW codes by computer methods is presented in Section 2. The results which have been obtained are presented in Section 3.

2 Enumeration of TCW codes by computer methods

Definition 1 *Two ternary constant-weight codes are equivalent if one of them can be obtained from the other by transformations of the following types:*

- *permutation of the coordinates of the code;*
- *permutation of the alphabet symbols appearing in a fixed position.*

[1] This work was partly supported by Bulgarian National Science Fund (the number of project IO-03-02/2006)

Bounds and exact values for the size of the codes could be found in [8, 5].

A known upper bound $A_3(n,d,w) \leq M$ may be improved after an exhaustive computer search for a code with these parameters and size M. This search can in fact be conveniently described as a search for a clique of size M in the following graph. Consider the graph where the vertex set corresponds to the words of length n and Hamming weight w and two vertices are joined by an edge if the Hamming distance between the corresponding words is greater than or equal to d. With a maximum clique algorithm, we would find the exact value of $A_3(n,d,w)$ but this direct approach is computationally feasible only for very small parameters. We may then perhaps relax the goal and just try to lower the upper bound. In any case, to speed up the search, it is essential to handle the large automorphism group of the constructed graph. This may be done in the following way by utilizing the Johnson-type bounds and removing equivalent copies of partial codes. We know that an (n,M,d,w) code can be shortened to get $(n-1,M',d,w)$ and $(n-1,M'',d,w-1)$ subcodes, where

$$M' \geq \frac{n-w}{n}.M, M'' \geq \frac{w}{n(q-1)}.M$$

Therefore, we may construct a code C by classifying all such subcodes (for one of these two alternatives), and then use the clique-finding approach to find the rest of the words in C.

The method we use is described in [7], [8], [3] and [9].

The two basic steps are:

- Finding all inequivalent possibilities for subcode C';
- Extending any of them to the size of C.

For the application of this method it is crucially important to have an effective algorithm for determining code equivalence.

We implement the steps 1 and 2 using our own, specifically developed, computer algorithms and programs. These algorithms are implemented in the computer package QPlus [4]. Some of the results are also verified using Q-Extension software [6].

3 Results

Let $\#(n,M,d,w)$ denote the number of inequivalent TCW codes with the specified parameters. The computer results are described by the following Theorem:

Theorem 1 *#(3,3,3,2)=1, #(4,4,3,2)=1, #(4,2,4,2)=1, #(4,8,3,3)=1, #(4,2,4,3)=1, #(5,5,3,2)=1, #(5,2,4,2)=1, #(5,12,3,3)=1, #(5,5,4,3)=1, #(5,2,5,3)=1, #(5,10,3,4)=64, #(5,5,4,4)=1, #(5,2,5,4)=1, #(6,6,3,2)=2, #(6,3,4,2)=1, #(6,18,3,3)=54, #(6,8,4,3)=3, #(6,4,5,3)=1, #(6,2,6,3)=1, #(6,15,4,4)=1, #(6,4,5,4)=1, #(6,3,6,4)=1,* $\#(6,24,3,5) \geq 20$, *#(6,12,4,5)=1,*

#(6,3,5,5)=1, #(6,2,6,5)=1, #(7,7,3,2)=2, #(7,3,4,2)=1, #(7,14,4,3)=1, #(7,4,5,3)=2, #(7,2,6,3)=1, #(7,7,5,4)=45, #(7,3,6,4)=3, #(7,2,7,4)=1, #(7,3,6,5)=4, #(7,2,7,5)=1, #(7,9,5,5)=2, $\#(7,14,4,6) \geq 74$, *#(7,7,5,6)=1, #(7,2,6,6)=1, #(7,2,7,6)=1, #(8,8,3,2)=3, #(8,4,4,2)=1, #(8,5,5,3)=1, #(8,2,6,3)=1, #(8,5,6,4)=2, #(8,2,7,4)=2, #(8,2,8,4)=1, #(8,8,6,5)=5, #(8,3,7,5)=3, #(8,2,8,5)=1, #(8,8,6,6)=22, #(8,3,7,6)=2, #(8,2,8,6)=1, #(8,16,5,7)=1, #(8,4,6,7)=2, #(8,2,7,7)=2, #(8,2,8,7)=1, #(9,9,3,2)=4, #(9,4,4,2)=1, #(9,6,5,3)=2, #(9,3,6,3)=1, #(9,3,7,4)=1, #(9,2,8,4)=1, #(9,5,7,5)=1,#(9,3,8,5)=1, #(9,2,9,5)=1, #(9,6,7,6)=12, #(9,3,8,6)=4, #(9,3,9,6)=1, #(9,5,7,7)=11, #(9,3,8,7)=1, #(9,2,9,7)=1, #(9,3,7,8)=1, #(9,2,8,8)=2, #(9,2,9,8)=1.*

References

[1] E. Agrell, A. Vardy, K. Zeger, Upper bounds for constant-weight codes, *IEEE Trans. Inform. Theory* 46, 2000, 2373-2395.

[2] G. Bogdanova, New bounds for the maximum size of ternary constant weight codes, *Serdica Math. J.* 26, 2000, 5-12.

[3] G. Bogdanova, S. Kapralov, Enumeration of optimal ternary constant-composition codes, *Probl. Inform. Transm.* 39, 2003, 346-351.

[4] G. Bogdanova, T. Todorov, V. Todorov, Web-based application for coding theory studying, *Proc. Intern. Congress MASSEE2003*, Borovets, Bulgaria, 2003, 94-99.

[5] G. Bogdanova, T. Todorov, V. Zinoviev, On construction of q-ary equidistant codes, *Probl. Inform. Transm.* 43, 2007, 13-36.

[6] I. Bouyukliev, About the code equivalence, advances in coding theory and cryptology, *Series Coding Theory Cryptol.* 3, Hackensack: World Scientific, 2007 (to appear).

[7] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, W. D. Smith, A new table of constant weight codes, *IEEE Trans. Inform. Theory* 36, 1990, 1334-1380.

[8] P. R. J. Östergard, M. Svanström, Ternary constant weight codes, *Electr. J. Combin.* 9, 2002, R41, 23pp.

[9] M. Svanström, P. R. J. Östergård, G. Bogdanova, Bounds and constructions for ternary constant-composition codes, *IEEE Trans. Inform. Theory* 48, 2002, 101-111.

[10] V. A. Zinoviev, Cascade equal-weight codes and maximal packings, *Probl. Contr. Inform. Theory* 12, 1983, 3-10.

[11] V. A. Zinoviev, On generalization of Johnson upper bound for constant weight codes, *Probl. Inform. Transm.* 20, 1984, 105-108.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 29-34*

Minimal/nonminimal codewords in the second order binary Reed-Muller codes: revisited

Yuri Borissov youri@math.bas.bg
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
8 G. Bonchev str., 1113 Sofia, BULGARIA

Abstract. The result on the weight distribution of minimal codewords in the second order binary Reed-Muller code $RM(2,m)$, was announced for the first time by Ashikhmin and Barg at ACCT'94. They gave only a sketch of the proof and later on a short and nice complete proof of geometric nature was exhibited in their paper: A. Ashikhmin and A. Barg, "Minimal Vectors in Linear Codes", IEEE Trans. on Information Theory, vol. 44, September 1998, pp. 2010-2017. The paper presents a different comprehensive proof of this result based on Dickson's Theorem.

1 Introduction

For the first time the sets of minimal codewords in linear codes were considered in connection with a decoding algorithm [8]. A more detailed description of the role of minimal codewords in the so-called "gradient-like" decoding algorithms can be found in [2] and [3, Ch 7]. Recently, the interest in minimal codewords with respect to decoding algorithms was resumed by [12]. Additional interest to them was sparked by the work of J. Massey [10], where it was shown that minimal codewords describe so-called minimal access structure in secret-sharing schemes based on linear codes (see e.g. [11] for definitions).

It seems to be quite difficult to describe the set of minimal codewords for an arbitrary linear code even in the binary case. The problem has been completely solved only for q-ary Hamming codes and for the second order binary Reed-Muller codes [1]. An attempt to characterize minimal codewords for two-error-correcting binary BCH codes ended with only a partial result [4],[5]. Another partial result was established in [6] for the number of non-minimal codewords of weight $2d_{min}$ in the r^{th} order binary Reed-Muller code $RM(r,m)$. The weight distributions of minimal codewords in some third-order binary Reed-Muller codes are determined by computer assistance in [7] and [13].

In this note, we return to the problem of describing the set of minimal/non-minimal codewords in the second order binary Reed-Muller code. A short and nice proof for this case suggested by Juriaan Simonis was exhibited in [2]. That proof is of geometric nature while here we present another comprehensive proof founded on Dickson's Theorem.

2 Background

We assume the reader is familiar with basic definitions, notations and facts about linear codes [9]. We shall need the following definitions.

Definition 2.1 *A support of an n-vector $\mathbf{c}$ over the finite field $\mathbf{F}_q$ is defined as the subset of its nonzero coordinates. A support of a Boolean function is the support of its truth table.*

Definition 2.2 *A nonzero codeword $\mathbf{c}$ of a binary linear code $\mathbf{C}$ is called minimal in $\mathbf{C}$ if its support does not cover the support of another nonzero codeword. Otherwise, $\mathbf{c}$ is called non-minimal.*

Proposition 2.3 *([1], [4])*

1) If $\mathbf{c}$ is minimal codeword in a linear $[n,k]$-code then its weight satisfies $wt(\mathbf{c}) \leq n-k+1$.

2) Any non-minimal codeword $\mathbf{c}$ in a binary linear code can be represented as a sum of two codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ having disjoint supports contained in the support of $\mathbf{c}$.

3) The automorphisms of a linear code preserve the property of the codewords to be minimal or not.

4) All codewords of a binary linear code with weight $< 2d_{min}$ are minimal.

For basic definitions and facts about second order binary Reed-Muller code (including Dickson's Theorem) we refer to [9, Ch. 15.2].

Let A_w be the number of codewords of weight w in $RM(2,m)$. Then $A_w = 0$ unless $w = 2^{m-1}$ or $w = 2^{m-1} \pm 2^{m-h-1}$ for some h, $0 \leq h \leq \lfloor m/2 \rfloor$.

Here, we shall remind also the theorem for weight distributions of the cosets of $RM(1,m)$ in $RM(2,m)$.

Theorem 2.4 *If the symplectic matrix determining coset $\mathcal{B}$ of $RM(1,m)$ in $RM(2.m)$ has rank $2h$ then the weight distribution of $\mathcal{B}$ is as follows:*

Weight	Number of Vectors
$2^{m-1} - 2^{m-h-1}$	2^{2h}
2^{m-1}	$2^{m+1} - 2^{2h+1}$
$2^{m-1} + 2^{m-h-1}$	2^{2h}

From Theorem 2.4 it follows immediately the corollary.

Corollary 2.5 *The number of codewords of weight 2^{m-1} in the cosets having rank $2h$ is equal to $A_{2^{m-1}-2^{m-h-1}}(2^{m-2h+1}-2)$.*

3 The proof

We shall make use of the following lemma.

Lemma 3.1 *The rank of symplectic matrix corresponding to the sum of two codewords in $RM(2,m)$ is less than or equal to the sum of the ranks of symplectic matrices associated with these codewords.*

Proof. Let $\mathbf{c_1}$ and $\mathbf{c_2}$ be two arbitrary codewords of $RM(2,m)$. According to [9, Ch. 15.2] the corresponding Boolean functions associated with them are of the form: $S_1(\mathbf{v}) = \mathbf{v}\mathbf{Q_1}\mathbf{v}^T + \mathbf{L_1}\mathbf{v} + \epsilon_1$ and $S_2(\mathbf{v}) = \mathbf{v}\mathbf{Q_2}\mathbf{v}^T + \mathbf{L_2}\mathbf{v} + \epsilon_2$, where $\mathbf{Q_1}$, $\mathbf{Q_2}$ are upper triangular binary matrices, $\mathbf{L_1}$, $\mathbf{L_2}$ are binary m-vectors, ϵ_1, ϵ_2 are binary constants, and $\mathbf{v} = (v_1, \ldots, v_m)$ is the vector of variables. Their corresponding symplectic matrices are:

$$\mathbf{B_1} = \mathbf{Q_1} + \mathbf{Q_1}^T \text{ and } \mathbf{B_2} = \mathbf{Q_2} + \mathbf{Q_2}^T$$

Therefore the symplectic matrix corresponding to the sum:

$$S_1(\mathbf{v}) + S_2(\mathbf{v}) = \mathbf{v}(\mathbf{Q_1} + \mathbf{Q_2})\mathbf{v}^T + (\mathbf{L_1} + \mathbf{L_2})\mathbf{v} + (\epsilon_1 + \epsilon_2)$$

is:

$$\mathbf{B} = (\mathbf{Q_1} + \mathbf{Q_2}) + (\mathbf{Q_1} + \mathbf{Q_2})^T = \mathbf{B_1} + \mathbf{B_2}$$

Taking into account, the well-known inequality for the rank of sum of two matrices, we complete the proof. □

Now, let us recall the result stated by Ashikhmin and Barg in [1].

Proposition 3.2 *Let $\mathbf{C} = RM(2,m)$ be the second order binary Reed-Muller code, and A_w, M_w be the number of its codewords and its minimal codewords of weight w, respectively. Then for $w = 2^{m-1} + 2^{m-1-h}, h = 0,1,2$ and $w = 0$ there are no minimal codewords ($M_w = 0$). Otherwise, $M_w = A_w$, except for the case $w = 2^{m-1}$, where*

$$M_w = \sum_{h=2}^{\lfloor m/2 \rfloor} A_{2^{m-1}-2^{m-h-1}}(2^{m-2h+1} - 2) \qquad (1)$$

Herein, we present a proof of this proposition different from exhibited in [2].

Proof. The smallest two weights in $\mathbf{C}$ are $w_1 = 2^{m-2}$ and $w_2 = 2^{m-1} - 2^{m-3}$ (corresponding to $h = 1,2$). By Proposition 2.3 Part 2), the smallest weights of $\mathbf{C}$ where non-minimal codewords could exist are $2w_1 = 2^{m-1}$ and $w_1 + w_2 = 2^{m-1} + 2^{m-3}$. Now, we shall show that all codewords of weight $w \geq 2^{m-1} + 2^{m-h-1}$, whenever $h = 0, 1$ or 2, are non-minimal in $\mathbf{C}$. Let $\mathbf{c}$ be such a codeword. There are three cases to be considered accordingly to the values of h.

- (1) $wt(\mathbf{c}) = 2^m$ $(h = 0)$. The only codeword of this kind is the all-one vector $\mathbf{1}$ which is obviously non-minimal.
- (2) $wt(\mathbf{c}) = 2^{m-1} + 2^{m-2}$ $(h = 1)$. The corresponding symplectic matrix has rank 2. By Dickson's Theorem [9, Ch. 15.2] it follows the existence of an affine transformation by which the Boolean function associated with the codeword $\mathbf{c}$, is reduced to the form $y_1y_2 + 1$. So, the considered codeword is affinely equivalent to concatenation of identical codewords from $RM(2,2)$ having weight 3. Hence, its property to be minimal or not, is the same as the latter one's property because of Proposition 2.3 Part 3). But the non-minimality of the codewords in $RM(2,2)$ of weight > 1 (like of that considered here) is obvious.
- (3) $wt(\mathbf{c}) = 2^{m-1} + 2^{m-3}$ $(h = 2)$. The corresponding symplectic matrix has rank equal to 4 and the Boolean function associated with such a codeword is affinely equivalent to $y_1y_2 + y_3y_4 + 1$. Similarly to the case (2), the non-minimality follows by that of the corresponding codeword in $RM(2,4)$ but this time according to Proposition 2.3 Part 1), since the weight of the latter equals 10 which is $> 16 - dim(RM(2,4)) + 1 = 6$.

So, it remains to consider the codewords of weight 2^{m-1}. Since the minimum weight of $\mathbf{C}$ is 2^{m-2} by Proposition 2.3 Part 2), we conclude that any non-minimal codeword $\mathbf{c}$ of weight 2^{m-1} must be sum of two codewords of weight 2^{m-2}, say $\mathbf{c}_1$ and $\mathbf{c}_2$. Since the symplectic matrices corresponding to $\mathbf{c}_i$, $i = 1, 2$ have rank 2, by Lemma 3.1 it follows the symplectic matrix $\mathbf{B}$ corresponding to $\mathbf{c}$ has rank ≤ 4 (i.e. the possible rank of $\mathbf{B}$ is $2h$ for some $h = 0, 1$ or 2). Hence, there are again three cases to be considered:

- (1) $h = 0$. According to Dickson's Theorem the corresponding Boolean function is affinely equivalent to $f(\mathbf{y}) = y_1$. The non-minimality of such an "affine" codeword (i.e. $\in RM(1,m)$) follows by the fact that Boolean functions y_1y_2 and $y_1(y_2+1)$ have disjoint supports and their sum is equal to f. By Corollary 2.5 the number of these codewords is $A_0(2^{m+1} - 2)$.
- (2) $h = 1$. The corresponding Boolean function is affinely equivalent to $f(\mathbf{y}) = y_1y_2 + y_3$ and the non-minimality of $\mathbf{c}$ follows by Proposition

2.3 Part 1), since the weight of the corresponding codeword in $RM(2,3)$ equals 4 which is $> 8 - dim(RM(2,3)) + 1 = 2$. For instance, f can be represented as a sum of Boolean functions $y_2y_3 + y_3$ and $y_1y_2 + y_2y_3$ having disjoint supports which are subsets of the support of f. Note that by Corollary 2.5 the number of codewords of this kind is $A_{2^{m-2}}(2^{m-1}-2)$.

- (3) $h = 2$. The Boolean function corresponding to $\mathbf{c}$ is affinely equivalent to $f(\mathbf{y}) = y_1y_2 + y_3y_4 + y_5$. Let Boolean functions corresponding to $\mathbf{c}_1$ and $\mathbf{c}_2$ be f_1 and f_2, respectively. Let us also consider $\mathbf{c}$ as a concatenation of two codewords $\mathbf{c}', \mathbf{c}''$ of $RM(2, m-1)$ over the hyperplanes $y_5 = 0$ and $y_5 = 1$. The subfunction $f(\mathbf{y}|y_5 = 0)$ is equal to $y_1y_2 + y_3y_4$ and thus $wt(\mathbf{c}') = 2^{m-2} - 2^{m-4} < 2^{m-2} = 2 * 2^{m-3} = 2 * dim(RM(2, m-1))$. Hence, $\mathbf{c}'$ is minimal in $RM(2, m-1)$ and therefore wlog we can assume that $f_1(\mathbf{y}|y_5 = 0) \equiv 0$. So, $f_1(\mathbf{y})$ is of the form $y_5L(\mathbf{y})$, where L depends essentially only on y_1, y_2, y_3 and y_4 and its algebraic degree is strictly less than 2. Then, clearly: $f(\mathbf{y}|y_5 = 1) = f_1(\mathbf{y}|y_5 = 1) + f_2(\mathbf{y}|y_5 = 1) = L(\mathbf{y}) + f_2(\mathbf{y}|y_5 = 1)$. Since $f_1(\mathbf{y}|y_5 = 1) \equiv L(\mathbf{y})$ and $wt(\mathbf{c}_1) = 2^{m-2}$, it follows that L is an affine function of weight 2^{m-2}. Furthermore, obviously $wt(\mathbf{c}'') = 2^{m-2} + 2^{m-4}$ and thus the weight of $f_2(\mathbf{y}|y_5 = 1) = 2^{m-4}$. But this is impossible weight for quadratic function in $m-1$ variables. Therefore $\mathbf{c}$ must not be non-minimal codeword i.e. all codewords of this kind are minimal.

Finally, by the above deductions and Corollary 2.5, for the number of minimal codewords of weight 2^{m-1} in $RM(2, m)$, we obtain:

$$\begin{aligned} M_{2^{m-1}} &= A_{2^{m-1}} - \sum_{h=0}^{1} A_{2^{m-1}-2^{m-h-1}}(2^{m-2h+1} - 2) \\ &= \sum_{h=2}^{\lfloor m/2 \rfloor} A_{2^{m-1}-2^{m-1-h}}(2^{m-2h+1} - 2), \end{aligned}$$

which completes the proof. □

References

[1] A. Ashikhmin, A. Barg, Combinatorial aspects of secret sharing with codes, *Proc. Intern. Workshop ACCT*, Novgorod, Russia, September, 1994, 8-11.

[2] A. Ashikhmin, A. Barg, Minimal vectors in linear codes, *IEEE Trans. Inform. Theory* 44, 1998, 2010-2017.

[3] A. Barg, Complexity Issues in Coding Theory, in *Handbook of Coding Theory* (Eds. V. Pless and W. Huffman), Amsterdam, Elsevier Science B.V., 1998.

[4] Y. Borissov, N. L. Manev, On the minimal words of the primitive BCH codes, *Proc. Intern. Workshop ACCT*, Sozopol, Bulgaria, 1996, 59-65.

[5] Yu. Borissov, N. L. Manev, Minimal codewords of the primitive BCH codes, *Probl. Pered. Inform.* 34, 3, 1998, 37-46 (in Russian).

[6] Y. Borissov, N. Manev, S. Nikova, On the non-minimal codewords in binary Reed-Muller codes, *Discr. Appl. Math.* 128, 2003, 65-74.

[7] Y. Borissov, N. Manev, Minimal codewords in linear codes, *Serdica Math. J.* 30, 2004, 303-324.

[8] Tai-Yang Hwang, Decoding linear block codes for minimizing word error rate, *IEEE Trans. Inform. Theory* 25, 1979, 733-737.

[9] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company 1977.

[10] J. Massey, Minimal codewords and secret sharing, *Proc. Sixth Joint Swedish-Russian Workshop Inform. Theory*, Mölle, Sweden, 1993, 246-249.

[11] D. R. Stinson, An explication of secret sharing schemes, *Des. Codes Crypt.* 2, 1992, 357-390.

[12] P. O. Vontobel, R. Smarandache, N. Kiyavash, J. Teutsch, D. Vukobratovic, On the minimal pseudo-codewords of codes from finite geometries, *ISIT 2005, Proc. Intern. Symp. Inform. Theory*, 2005, 980 - 984.

[13] K. Yasunaga, T. Fujiwara, T. Kasami, Local weight distribution of the (256, 93) third-order binary Reed-Muller code, *IEICE Trans. Fundam. Electr., Commun. Computer Sci.* E90-A, 2007, 698-701.

Nonexistence results for spherical 7-designs

SILVIA BOUMOVA silvi@moi.math.bas.bg
PETER BOYVALENKOV peter@moi.math.bas.bg
Institute of Mathematics and Informatics Bulgarian Academy of Sciences,
8 G. Bonchev str., 1113 Sofia, BULGARIA
MAYA STOYANOVA stoyanova@fmi.uni-sofia.bg
Faculty of Mathematics and Informatics, Sofia University,
5 James Bouchier blvd, 1164 Sofia, BULGARIA

Abstract. We obtain new nonexistence results for spherical 7-designs of odd cardinality. Our approach continues similar investigations for smaller strengths. We combine polynomial techniques with some geometric argument to obtain restrictions of the structure of 7-designs with fixed cardinality.

1 Introduction

A spherical τ-design [2] is a spherical code $C \subset \mathbb{S}^{n-1}$ such that for every point $y \in \mathbb{S}^{n-1}$ and for every real polynomial $f(t)$ of degree at most τ, the equality

$$\sum_{x \in C} f(\langle x, y\rangle) = f_0|C|. \tag{1}$$

holds, where f_0 is the first coefficient in the expansion $f(t) = \sum_{i=0}^{k} f_i P_i^{(n)}(t)$ in terms of the Gegenbauer polynomials [1, Chapter 22]. The number τ is called strength of C. When $y \in C$, (1) becomes

$$\sum_{x \in C\setminus\{y\}} f(\langle t_i(x)\rangle) = f_0|C| - f(1), \tag{2}$$

where $t_1(x) \le t_2(x) \le \cdots \le t_{|C|-1}(x)$ are the inner products of $x \in C$ with all other points of C.

Polynomial techniques use suitable polynomials in (1) and (2) for obtaining bounds on some inner products. Restrictions on the structure of spherical designs via polynomial techniques were described in 1997 by Fazekas-Levenshtein [8] (see also [9]) and proved to work for nonexistence results by Boyvalenkov-Danev-Nikova [6] (see also [3, 4, 5]). In this paper we continue investigations from [5] by obtaining new nonexistence results for 7-designs in dimensions $n \le 20$.

2 Preliminaries

Let $C \subset \mathbf{S}^{n-1}$ be a 7-design. Then

$$|C| \geq 2\binom{n+2}{3} = \frac{n(n+1)(n+2)}{3} \tag{3}$$

by the Delsarte-Goethals-Seidel bound [2].

We use some results and notations from [3, 4, 5, 8, 9]. The parameters α_i are roots of certain Jacobi polynomials and the definition of the weight ρ_0 can be found in [9].

Lemma 1. [3] *Let $C \subset \mathbb{S}^{n-1}$ be a τ-design with odd $\tau = 2e-1$. Then for every point $x \in C$ we have $t_1(x) \leq \alpha_0$ and $t_{|C|-1}(x) \geq \alpha_{e-1}$. In particular, we have $s(C) \geq \alpha_{e-1}$. If $|C|$ is odd then there exist a point $x \in C$ such that $t_2(x) \leq \alpha_0$.*

Lemma 2. [4] *Let $C \subset \mathbb{S}^{n-1}$ be a τ-design with odd $\tau = 2e-1$ and odd cardinality $|C|$. Then there exist three distinct points $x, y, z \in C$ such that $t_1(x) = t_1(y)$ and $t_2(x) = t_1(z)$. Moreover, we have $t_{|C|-1}(z) \geq \max\{\alpha_{e-1}, 2\alpha_0^2 - 1\}$. In particular, we have $s(C) \geq \max\{\alpha_{e-1}, 2\alpha_0^2 - 1\}$.*

Theorem 3. [3] *If $C \subset \mathbb{S}^{n-1}$ is a τ-design with odd $\tau = 2e-1$ and odd $|C|$ then $\rho_0|C| \geq 2$.*

It is proved in [5] that the necessary condition $\rho_0|C| \geq 2$ can be replaced by the stronger $\rho_0|C| \geq 3$ for 3-designs (with a few exceptions) in dimensions $8 \leq n \leq 50$ and for 5-designs in dimensions $5 \leq n \leq 25$. In this paper we prove that $\rho_0|C| \geq 3$ is necessary for 7-designs of odd cardinalities in dimensions $5 \leq n \leq 20$. Moreover, we obtain nonexistence results in several cases where $\rho_0|C| > 3$.

It is convenient to use the following notation: $U_{\tau,i}(x)$ (respectively $L_{\tau,i}(x)$) for any upper (resp. lower) bound on the inner product $t_i(x)$. When a bound does not depend on x we omit x in the notation. For example, the first bound from Lemma 1 is $t_1(x) \leq U_{\tau,1} = \alpha_0$ and the last bound from Lemma 2 is $t_{|C|-1}(z) \geq L_{\tau,|C|-1}(z) = \max\{\alpha_{e-1}, 2\alpha_0^2 - 1\}$.

3 Brief description of the algorithm

Assuming the existence of a 7-design on $\mathbf{S}^{n-1}$ with odd $|C|$ and $2\alpha_0^2 - 1 > \alpha_3$, we consider a special triple of points $x, y, z \in C$ as in Lemma 2. We focus on the inner products in $I(z)$. Sometimes we need to consider the point $u \in C$

such that $\langle u, z\rangle = t_2(z)$. We consecutively obtain bounds $L_{7,1}(z) \leq t_1(z)$, $t_2(z) \leq U_{7,2}(z)$ and $L_{7,3}(z) \leq t_3(z)$ using suitable polynomials.

Sometimes we get contradictions at the beginning – already from the first bounds $t_1(z) \geq L_{7,1}(z) > U_{7,1}(z) = \alpha_0 \geq t_1(z)$ (this happens in cases where $\rho_0|C|$ is close from above to 2) and further by $t_2(z) \leq U_{7,2}(z) < L_{7,1}(z) \leq t_1(z)$. When $U_{7,2}(z) \geq L_{7,1}(z)$, we consider two cases for the location of $t_2(z)$ with respect to α_0.

Case 1. If $t_2(z) \in [\alpha_0, U_{7,2}(z)]$ (this can happen only when $\alpha_0 \leq U_{7,2}(z)$) then we obtain new upper bound $t_1(z) \leq U_{7,1}(z) < \alpha_0$ which can be used for obtaining a contradiction. If necessary (in a few cases) we organize an iteration procedure.

Case 2. If $t_2(z) \in [t_1(z), \alpha_0]$, then we consider the point $u \in C$ such that $t_2(z) = \langle z, u\rangle$. It follows from [5, Section 4] that some special quadruple $\{x, y, z, u\} \subset C$ exists such that $\max\{t_{|C|-2}(z), t_{|C|-2}(x)\} \geq 2\alpha_0^2 - 1$. In both cases we continue with new bounds $L_{7,3}(z)$ and $U_{7,1}(z)$ which can be used for obtaining a contradiction. In some case we need more careful consideration of the location of some inner products and iteration procedures.

All symbolic and numerical calculations were performed by MAPLE with high enough precision. All programs and results (symbolic and numerical) are available upon request.

4 The new nonexistence results

After [4], there are 291 open cases in dimensions $3 \leq n \leq 20$, with odd $|C|$ and $2 \leq \rho_0|C| < 3$. In every such case we have $2\alpha_0^2 - 1 > \alpha_3$, i.e. $t_{|C|-1}(z) \geq L_{3,|C|-1}(z) = 2\alpha_0^2 - 1$ by Lemma 2. Applying our algorithm we obtain nonexistence in all cases with only one exception – the case $n = 4$, $|C| = 43$. There are 18 cases of nonexistence with $\rho_0|C| > 3$ as well.

In the table below we give lower bounds on

$$B_{\text{odd}}(n, 7) = \min\{|C| : C \subset \mathrm{S}^{n-1} \text{ is a 7-design, } |C| \text{ is odd}\}.$$

The bounds from [7] (the second column in the table) come from pure linear programming and are better than the Delsarte-Goethals-Seidel bound (3) in dimensions 5, 6, and 7 only. No examples in small dimensions are given in [6, 3, 4] but we know that the best bounds come by the method from [4] (the third column in the table). In the fourth column we give the results from the method from Section 3 when $\rho_0|C| \in [2, 3)$ and the fifth column gives the results from the method from Section 3 when $\rho_0|C| > 3$. So the best bounds are the last entries in the rows.

Table. Lower bounds on $B_{\text{odd}}(n,7)$, $3 \le n \le 20$.

n	[7]	[4]	This paper $\rho_0\|C\| \in [2,3)$	This paper $\rho_0\|C\| \ge 3$
3		23	23	
4		43	43	
5	73	75	77	
6	117	119	123	
7	173	177	183	
8		253	261	
9		347	359	
10		463	477	
11		601	619	621
12		765	789	
13		957	985	987
14		1175	1213	1215
15		1427	1471	1475
16		1713	1767	1769
17		2031	2097	2101
18		2393	2467	2473
19		2791	2879	2885
20		3233	3333	3341

In [6], the asymptotic lower bound $B_{\text{odd}}(n,7) \gtrsim \frac{(1+\sqrt[3]{2})n^3}{6} \approx 0.35068n^3$ was proved. This was obtained again in [3] despite the results in small dimensions from [3] are better than those from [6]. The best known asymptotic lower bound is $B_{\text{odd}}(n,7) \gtrsim 0.35314n^3$ from [4]. The results from this paper suggest that further improvements are possible by our method. However, we still could not overcome the technical difficulties on this way.

Acknowledgment. This research was partially supported by the Bulgarian NSF under Contract MM-1405/2004 and the SF of Sofia University under Contract 171/05.2008.

References

[1] M. Abramowitz, I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1965.

[2] P. Delsarte, J.-M. Goethals, J. J. Seidel, Spherical codes and designs, *Geom. Dedicata* 6, 1977, 363-388.

[3] P. G. Boyvalenkov, S. P. Bumova, D. P. Danev, Necessary conditions for existence of some designs in polynomial metric spaces, *Europ. J. Combin.* 20, 1999, 213-225.

[4] S. Boumova, P. Boyvalenkov, D. Danev, New nonexistence results for spherical designs, in *Constructive Theory of Functions* (B. Bojanov, Ed.) Darba, Sofia, 225-232 (2003).

[5] S. Boumova, P. Boyvalenkov, H. Kulina, M. Stoyanova, Polynomial techniques for investigation of spherical designs, submitted.

[6] P. G. Boyvalenkov, D. P. Danev, S. I. Nikova, Nonexistence of certain spherical designs of odd strengths and cardinalities, *Discr. Comp. Geom.* 21, 1999, 143-156.

[7] P. Boyvalenkov, S. Nikova, Improvements of the lower bounds on the size of some spherical designs, *Math. Balk.* 12, 1998, 151-160.

[8] G. Fazekas, V. I. Levenshtein, On the upper bounds for code distance and covering radius of designs in polynomial metric spaces, *J. Comb. Theory A*, 70, 1995, 267-288.

[9] V. I. Levenshtein, Universal bounds for codes and designs, Chapter 6 (499-648) in *Handbook of Coding Theory*, Eds. V. Pless and W.C. Huffman, Elsevier Science B.V., 1998.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 40-45*

Notes on automorphisms of extremal codes

STEFKA BOUYUKLIEVA stefka@uni-vt.bg
Veliko Tarnovo University, 5000 Veliko Tarnovo, BULGARIA,

WOLFGANG WILLEMS wolfgang.willems@mathematik.uni-magdeburg.de
Otto-von-Guericke Universität, 39016 Magdeburg, GERMANY

Abstract. We prove that if a putative extremal self-dual $[24m, 12m, 4m + 4]$ code has an automorphism of odd prime order p with c cycles and f fixed points then $c \geq f$. In case $p > 12m$ the results we have obtained so far give some evidence that m must be 1 or 2.

1 Introduction

Let C be an extremal (doubly-even) self-dual $[24m, 12m, 4m + 4]$ binary code. By the results of Zhang [9], we know that $m \leq 153$. However, the existence of such codes is proved only for $m = 1$ and $m = 2$, and in these cases we have the extended $[24, 12, 8]$ Golay code with automorphism group M_{24} and the extended quadratic residue code $[48, 24, 12]$ with automorphism group $\mathrm{PSL}(2, 47)$. In [1] we proved that the automorphism group of a binary self-dual doubly-even $[72, 36, 16]$ code is a solvable group of order $5, 7, 10, 14, 56$, or a divisor of 72.

Here we investigate primes which may occur in the order of the automorphism group $G = \mathrm{Aut}(C)$ and the cycle structure of permutations in G. Let $\sigma \in G$ be a permutation of order p where p is an odd prime. The action of σ on the positions produces, say c cycles of length p and f fixed points and in this case we call σ of type $p - (c, f)$. In Section 2 we prove that $c \geq f$ for any automorphism of C of order p. In Section 3 we investigate the possibility $c = f = 1$.

2 The main result

First we consider the case $p = 3$. Let C be a binary self-dual code of length n with an automorphism σ of order 3 with exactly c independent 3-cycles and $f = n - 3c$ fixed points in its factorization. Let $\sigma = \Omega_1\Omega_2 \ldots \Omega_c$, where $\Omega_1, \Omega_2, \ldots, \Omega_c$, are independent cycles of length 3. Two particular subcodes of C play an important role in the following investigations.

Let $F_\sigma(C) = \{v \in C : v\sigma = v\}$. Clearly, $v \in F_\sigma(C)$ iff $v \in C$ is constant on each cycle. Let $\pi : F_\sigma(C) \to \mathbb{F}_2^{c+f}$ be the projection map where if $v \in F_\sigma(C)$ then $(v\pi)_i = v_j$ for some $j \in \Omega_i, i = 1, 2, \ldots, c+f$.

We consider furthermore the vector space

$$E_\sigma(C) = \{v \in C : wt(v|\Omega_i) \equiv 0 \pmod 2), i = 1, \ldots, c, v_j = 0, j = 3c{+}1, \ldots, n\},$$

where $v|\Omega_i$ denotes the restriction of v on Ω_i. Let P be the set of even-weight polynomials in $\mathbb{F}_2[x]/(x^3+1)$, and let $v|\Omega_i = (v_0, v_1, v_2)$ correspond to the polynomial $v_0 + v_1 x + v_2 x^2$ of P for $i = 1, \ldots, c$. Thus we obtain a natural map $\phi : E_\sigma(C)^* \to P^c$. In our particular case, $P = \{0, e = x + x^2, xe, x^2 e\} \cong \mathbb{F}_4$.

Theorem 1 [2] *A binary code C with an automorphism σ is self-dual if and only if the following two conditions hold.*

(i) *$C_\pi = \pi(F_\sigma(C))$ is a self-dual binary code of length $c+f$;*

(ii) *$C_\phi = \phi(E_\sigma(C))$ is a Hermitian quaternary self-dual code of length c over the field $P \cong \mathbb{F}_4$.*

For the minimum distance of the quaternary Hermitian self-dual codes we have the following bound.

Theorem 2 [5] *If C is an $[n, n/2, d]$ Hermitian self-dual code over $\mathbb{F}_4$, then*

$$d \leq 2\lfloor n/6 \rfloor + 2.$$

Using the above theorems we obtain

Corollary 3 *If C is an extremal binary self-dual $[24m, 12m, 4m+4]$ code and σ is an automorphism of C of type $3-(c,f)$ then $c \geq f$.*

Proof: By Theorem 1, C_ϕ must be a Hermitian quaternary self-dual code of length c over the field $P \cong \mathbb{F}_4$. Since $d(C) = 4m+4$, the minimum distance of C_ϕ cannot be less than $2m+2$. By Theorem 2, $2m+2 \leq 2\lfloor c/6 \rfloor + 2 \leq 2c/6 + 2$, hence $c \geq 6m$. It follows that $f = 24m - 3c \leq 24m - 18m = 6m \leq c$. □

To restrict the possible automorphisms for particular codes we need the following theorem.

Theorem 4 [8] *Let C be a binary self-dual $[n, k, d]$ code and let $\sigma \in$ Aut(C) be of type $p-(c,f)$, where p is an odd prime. If $g(s) = \sum_{i=0}^{s-1} \lceil \frac{d}{2^i} \rceil$ then*

(i) *$pc \geq g(\frac{p-1}{2}c)$ and*

(ii) *$f \geq g(\frac{f-c}{2})$ for $f > c$.*

Now let $n = 24m$ and $d = 4m + 4$. Then $g(1) = 4m + 4$, $g(2) = 6m + 6$, $g(3) = 7m + 7$, and for $s \geq 4$ we have

$$g(s) = \sum_{i=0}^{s-1} \left\lceil \frac{4m+4}{2^i} \right\rceil = 7m + 7 + \sum_{i=3}^{s-1} \left\lceil \frac{4m+4}{2^i} \right\rceil = 7m + 7 + \sum_{i=1}^{s-3} \left\lceil \frac{m+1}{2^i} \right\rceil.$$

If $2^l < m + 1 \leq 2^{l+1}$ for $l \in \mathbb{N}_0$ then

$$g(s) \geq 7m + 7 + \sum_{i=1}^{s-3} \frac{m+1}{2^i} = 7m + 7 + (m+1)\frac{2^{s-3} - 1}{2^{s-3}} = (m+1)\frac{2^s - 1}{2^{s-3}}.$$

For $i > l$ we have $\frac{m+1}{2^i} < 2^{l+1-i} \leq 1$ and therefore $\lceil \frac{m+1}{2^i} \rceil = 1$. Hence for $s - 3 > l$ the following inequality holds

$$g(s) = g(l+3) + s - 3 - l \geq \Big(8 - \frac{1}{2^l}\Big)(m+1) - l - 3 + s = A + s$$

Using the above inequalities and Theorem 4 we prove the main result

Main Theorem 5 *If C is an extremal self-dual $[24m, 12m, 4m+4]$ code and σ is an automorphism of C of type $p - (c, f)$, where p is an odd prime, then $c \geq f$.*

Proof: By Corollary 3, we may assume that $p \geq 5$. Suppose that $f > c$. We know that this is impossible if $m \leq 3$. Let $m \geq 4$, hence $l \geq 2$. By Theorem 4, we have the following inequalities

$$pc \geq g\Big(\frac{p-1}{2}c\Big) \text{ and } f \geq g\Big(\frac{f-c}{2}\Big)$$

(a) We claim $\frac{p-1}{2}c > l + 3$: If $\frac{p-1}{2}c \leq l + 3$ then $c \leq \frac{2(l+3)}{p-1} \leq \frac{20}{p-1}$ since by Zhang, $m \leq 153$, hence $l \leq 7$. This inequality is possible only in the following cases: $p = 5, c \leq 5$; $p = 7$, $c \leq 3$; $p = 11$, $c \leq 2$; $p = 13, 17, 19$, $c = 1$. But for $p \geq 5$ we have $pc \geq g(\frac{p-1}{2}c) \geq g(2) = 6(m+1) \geq 30$ which does not hold in all cases.

(b) We claim $\frac{f-c}{2} > l + 3$: If $\frac{f-c}{2} \leq l + 3$ then $f - c \leq 20$ and so $f \leq 21$. But $f \geq g(\frac{f-c}{2}) \geq g(1) = 4(m+1) \geq 20$, a contradiction.

As $f = 24m - pc$ we obtain, by (a) and (b), that

$$pc \geq A + \frac{p-1}{2}c \text{ and } 24m - pc \geq A + 12m - \frac{(p+1)c}{2},$$

hence $\frac{2A}{p+1} \leq c \leq \frac{24m-2A}{p-1}$ and therefore $A(p-1) \leq (12m-A)(p+1)$ or $(2A-12m)p \leq 12m$. Since

$$\begin{aligned} A-6m &= (8-\tfrac{1}{2^l})(m+1)-l-3-6m \\ &\geq (8-\tfrac{1}{4})(m+1)-l-3-6m = \tfrac{1}{4}(7m+19-4l) > 0 \end{aligned}$$

we get $\quad p \leq \dfrac{6m}{A-6m} \leq \dfrac{24m}{7m+19-4l} \leq \dfrac{24m}{7(m-(4l-19)/7)}, \quad$ hence

$$p \leq \frac{24}{7} + \frac{24}{7}\frac{(4l-19)}{(7m-4l+19)} \leq \frac{24}{7} + \frac{24}{7}\frac{9}{(7m-9)} \leq \frac{24}{7} + \frac{24.9}{7.19} = \frac{96}{19} < 6.$$

Thus $p = 5$ and moreover by (a), we have $5c \geq g(2c) \geq A + 2c$, hence $3c \geq A$. The inequality $f = 24m - 5c > c$ implies $c < 4m$. Furthermore, $f = 24m - 5c \geq g(12m - 3c)$. Since $12m - 3c > l + 3$, by (b), we have $24m - 5c \geq A + 12m - 3c$, hence $(12 - a)m - b \geq 2c \geq \frac{2}{3}(am + b)$, where $a = 8 - \frac{1}{2^l}$, $b = a - l - 3$. $A = am + b$. Hence $(36 - 5a)m \geq 5b$ and therefore $(36 - 40 + 5/2^l)m \geq 5(5 - l) - 5/2^l$ which implies $(4.2^l - 5)m \leq 5.2^l(l - 5) + 5$, a contradiction. This proves that $c \geq f$. □

3 Automorphisms of prime order $p > 12m$

Now suppose that $p > \frac{n}{2} = 12m$. Thus, by Theorem 5, σ is of type $p-(1,1)$. Hence $n = 24m = p + 1$, and in particular $p \equiv -1 \bmod 8$. The later yields that $\frac{p-1}{2}$ is odd. As usual let $s(p)$ denote the smallest number $s \in \mathbb{N}$ such that $p \mid 2^s - 1$.

Lemma 6 *For $p > \frac{n}{2} = 12m$ we have $s(p)$ odd.*

Proof: Since $p \equiv -1 \bmod 8$ the prime 2 is a square mod p. This yields that $2^{\frac{p-1}{2}} \equiv 1 \bmod p$. As $s(p) \mid \frac{p-1}{2}$ and $\frac{p-1}{2}$ is odd the proof is complete. □

Lemma 7 *For the group algebra $\mathbb{F}_2\langle\sigma\rangle$, the trivial module is the only irreducible self-dual module.*

Proof: By Lemma 6, we know that $s(p)$ is odd. The assertion now follows directly by Theorem 2.7 of [7]. □

Using Maple we easily find all primes p of the form $2m - 1$ for $m \leq 153$. It turns out that apart from six primes, we always have $s(p) = \frac{p-1}{2}$.

Theorem 8 *Apart from the six exceptions C is an extended* QR *code.*

Proof: Let $K = \mathbb{F}_2$. The ambient space K^n of C can be written as

$$K^n = K\langle\sigma\rangle \oplus K.$$

Since $s(p) = \frac{p-1}{2}$ the non-trivial irreducible $K\langle\sigma\rangle$-modules are of dimension $\frac{p-1}{2}$. Thus by Maschke, we have the decomposition

$$(*) \qquad K\langle\sigma\rangle = K \oplus V \oplus W$$

with irreducible modules V and W both of dimension $s(p) = \frac{p-1}{2}$. By Lemma 7, we have $V \not\cong V^*$ and $W \not\cong W^*$. Since a group algebra is always selfdual we obtain $W \cong V^*$. Furthermore, the decomposition in $(*)$ is unique since the three modules are non-isomorphic. On the other hand, we know that

$$K\langle\sigma\rangle = K \oplus Q \oplus N$$

where Q is the code associated to the squares mod p and N to the non-squares. Since Q is equivalent to N we may assume that $V = Q$. Finally, if C_0 is the subspace of C with 0 in the last position then $C = \langle C_0, c\rangle$ where c is the all one word. This shows that C is an extended QR code. □

Problem 9 Is an extended QR of length $p+1 = 24m$ extremal only for $m = 1$ and $m = 2$?

By known results [4], this is true for $m \leq 21$. But we have to check up to $m = 153$. Fortunately, we do not need to compute the minimum distance in these remaining cases. Instead we only have to find a codeword of weight smaller than $4m + 4$. Apart from the largest case, i.e. $m = 153$, this is always possible if $s(p) = \frac{p-1}{2}$ splits up into a nontrivial product of primes which holds true in about half of the cases we have to consider. Here the Karlin-MacWilliams algorithm (see [3] or [6], chap. 16, section 6) is applicable and the computations have been done partly by Malevich (Minsk) and independently by O'Brien (Auckland). In the other half of cases in which $s(p) = \frac{p-1}{2}$ is a prime the Karlin-MacWilliams algorithm does not work and further theoretical investigations are needed to answer Problem 9.

Summarizing the above theoretical and computational results there is some evidence to

Conjecture 10 *If a binary extremal code C of length $24m$ has an automorphism of prime order $p > 12m$ then $m = 1$ or $m = 2$.*

References

[1] S. Bouyuklieva, E. A. O'Brien, W. Willems, The automorphism group of a binary self-dual doubly even $[72, 36, 16]$ code is solvable, *IEEE Trans. Inform. Theory* 52, 2006, 4244-4248.

[2] W. C. Huffman, Automorphisms of codes with application to extremal doubly-even codes of lenght 48, *IEEE Trans. Inform. Theory* 28, 1982, 511-521.

[3] M. Karlin, F. J. MacWilliams, On finding low weight vectors in quadratic residue codes for $p = 8m - 1$, *SIAM J. Appl. Math.* 25, 1973, 95-104.

[4] J. S. Leon, A probabilistic algorithm for computing the minimum weights of large error-correcting codes, *IEEE Trans. Inform. Theory* 34, 1988, 1354-1359.

[5] F. J. MacWilliams, A. M. Odlyzko, N. J. A. Sloane, H. N. Ward, Self-dual codes over $GF(4)$, *J. Combin. Theory*, Series A25, 1978, 288-318.

[6] F. J. MacWilliams, N. J. A. Sloane, *The theory of error-correcting codes*, North Holland, Amsterdam 1977.

[7] C. Martínez-Pérez, W. Willems, Self-dual extended cyclic codes, *Appl. Algebra Eng. Comm. Computing* 17, 2006, 1-16.

[8] V. Y. Yorgov, A method for constructing inequivalent self-dual codes with applications to length 56, *IEEE Trans. Inform. Theory* 33, 1987, 77-82.

[9] S. Zhang, On the nonexistence of extremal self-dual codes, *Discr. Math.*, 91, 1999, 277-286.

Three-dimensional array codes correcting $2 \times 2 \times 2$-clusters of errors

Igor Boyarinov i.boyarinov@mtu-net.ru
Institute for System Analysis, Russian Academy of Sciences,
60 years of October ave. 9, 117312, Moscow, RUSSIA

Abstract. Linear binary three–dimensional array codes that can correct three-dimensional clusters (or bursts) of errors are presented. New constructions of three-dimensional $2 \times 2 \times 2$–cluster–error–correcting array codes with excess redundancy $\tilde{r}_{n_1,n_2,n_3}(2,2,2) = 17$ are given.

1 Introduction

There are data transmission and storage systems with multidimensional data structures that suffer from multidimensional clusters of errors. Correction of two- and three-dimensional error clusters is required in holographic storage. Two-dimensional and three-dimensional array codes are very suitable for correcting cluster errors in such data structures. In this paper we consider three-dimensional array codes that can correct three-dimensional $2 \times 2 \times 2$-clusters of errors.

For integers n_l, $l = 1,2,3$ we consider the linear space $V(n_1,n_2,n_3)$ of all binary three-dimensional $n_1 \times n_2 \times n_3$ arrays. A linear k-dimensional $(k \leq n_1n_2n_3)$ subspace $C(n_1,n_2,n_3)$ of the space $V(n_1,n_2,n_3)$ is called a linear binary three-dimensional array $[n_1 \times n_2 \times n_3,\ k]$ code of size (or area) $n_1 \times n_2 \times n_3$ with k information symbols and $r = n_1n_2n_3 - k$ parity-check symbols. Thus, a codeword of the binary linear three-dimensional array code $C(n_1,n_2,n_3)$ is a three-dimensional array $c = (c_{i,j,h})$ where $c_{i,j,h} = 0,1$ for $i = 0,1,\ldots,n_1-1$; $j = 0,1,\ldots,n_2-1$; $h = 0,1,\ldots,n_3-1$.

A three-dimensional array $e = (e_{i,j,h})$ of size $n_1 \times n_2 \times n_3$ is called a rectangular cluster of size $b_1 \times b_2 \times b_3$ ($b_1 \times b_2 \times b_3$-cluster) if nonzero components of $e = (e_{i,j,h})$ are confined to a rectangular parallelepiped of size $b_1 \times b_2 \times b_3$.

By analogy with two-dimensional array codes ([1]) for $b_1 \times b_2 \times b_3$-cluster-error-correcting array codes whose sizes $n_1 \times n_2 \times n_3$ are much larger than $b_1 \times b_2 \times b_3$ the criterion of the excess redundancy can be used. We define the excess redundancy of the $b_1 \times b_2 \times b_3$-cluster-error-correcting array $[n_1 \times n_2 \times n_3,\ k]$ code $C(n_1,n_2,n_3)$ as

$$\tilde{r}_{n_1,n_2,n_3}(b_1,b_2,b_3) = \lceil r - \log_2 n_1n_2n_3 \rceil, \tag{1}$$

where $r = n_1n_2n_3 - k$ is the redundancy of the code $C(n_1, n_2, n_3)$ and $\lceil x \rceil$ is the least integer equal or more than $x \geq 0$. Now if $n = n_1n_2n_3$ and $n \to \infty$ we can define for a class $\tilde{C}$ of codes $C(n_1, n_2, n_3)$ the excess redundancy

$$\tilde{r}_C(b_1, b_2, b_3) = \lim_{n\to\infty} (r - \log_2 n_1n_2n_3), \tag{2}$$

if such limit exists. If this function is unbounded, we take $\tilde{r}_C(b_1, b_2, b_3) = \infty$.

For any class of three-dimensional $b_1 \times b_2 \times b_3$-cluster-error-correcting array codes

$$\tilde{r}_C(b_1, b_2, b_3) \geq b_1b_2b_3 - 1. \tag{3}$$

For $b_i = 2,\ i = 1, 2, 3$ the excess redundancy

$$\tilde{r}_C(2, 2, 2) > 7. \tag{4}$$

In [2] we had shown that there exist linear three-dimensional $b_1 \times b_2 \times b_3$-cluster-error-correcting array codes with small excess redundancy $\tilde{r}_C(b_1, b_2, b_3) > 3b_1b_2b_3 - 5$ for all b_1, b_2, b_3. Constructions of the codes are based on the constructions [3] and used the approaches [4], [5].

In this paper we give new constructions of linear three-dimensional $2 \times 2 \times 2$-cluster-error-correcting array codes with the excess redundancy $\tilde{r}_C(2, 2, 2) = 17$.

2 Three-dimensional interleaved array codes

For constructing linear binary three-dimensional array $2 \times 2 \times 2$-cluster-error-correcting codes with small excess redundancy we can use one-dimensional 2-burst-error-correcting codes with the same property.

Let $C(n_1, n_2, n_3)$ be a binary linear three-dimensional array code such that for any code array word $c = (c_{i,j,h})$ we have

$$\sum_{j=0}^{n_2-1}\sum_{h=0}^{n_3-1} c_{i,j,h} = 0, \quad \sum_{i=0}^{n_1-1}\sum_{j=0}^{n_2-1} c_{i,j,h} = 0, \quad \sum_{i=0}^{n_1-1}\sum_{h=0}^{n_3-1} c_{i,j,h} = 0.$$

The code $C(n_1, n_2, n_3)$ is the single-error-correcting-double-error-detecting (SEC-DED) array $[n_1 \times n_2 \times n_3, k]$ code with $k = n_1n_2n_3 - n_1 - n_2 - n_3 + 2$ information symbols [6].

The parity-check symbols of the code $C(n_1, n_2, n_3)$ are $c_{0,j,h}$, $c_{i,0,h}$ and $c_{i,j,0}$ where $i = 0, \ldots, n_1 - 1$, $j = 0, \ldots, n_2 - 1$, $h = 0, \ldots, n_3 - 1$.

Given the three-dimensional SEC–DED array $[n_1 \times n_2 \times n_3, k]$ code $C(n_1, n_2, n_3)$ we can construct the three-dimensional single-cluster-error-correcting-double-cluster-error-detecting
(SCEC-DCED) array $[b_1n_1 \times b_2n_2 \times b_3n_3, b_1b_2b_3k]$ code $Z(b_1n_1, b_2n_2, b_3n_3)$ by the rectangular three-dimensional interleaving.

The parity-check symbols of the code $Z(b_1n_1, b_2n_2, b_3n_3)$ are $z_{i',j,h}, z_{i,j',h}$ and $z_{i,j,h'}$ where $i = 0, \ldots, b_1(n_1 - 1)$, $j = 0, \ldots, b_2(n_2 - 1)$, $h = 0, \ldots, b_3(n_3 - 1)$, $i' = 0, \ldots, b_1 - 1$, $j' = 0, \ldots, b_2 - 1$, $h' = 0, \ldots, b_3 - 1$. The sets of parity-check symbols $z_{i',j,h}$, $z_{i,j',h}$ and $z_{i,j,h'}$ are confined to rectangular parallelepipeds $(z_{i',j,h})$, $(z_{i,j',h})$ and $(z_{i,j,h'})$ of size $b_1 \times n_2 \times n_3$, $n_1 \times b_2 \times n_3$, $n_1 \times n_2 \times b_3$, respectively. The intersection of these rectangular parallelepipeds is the rectangular parallelepiped $(z_{i',j',h'})$ of size $b_1 \times b_2 \times b_3$.

3 Constructions of linear array codes correcting 2×2- and $2 \times 2 \times 2$-clusters of errors

At first we consider the two-dimensional construction.

Let

$$c = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,2n_2-2} & c_{0,2n_2-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,2n_2-2} & c_{1,2n_2-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{2n_1-2,0} & c_{2n_1-2,1} & \cdots & c_{2n_1-2,2n_2-2} & c_{2n_1-2,2n_2-1} \\ c_{2n_1-1,0} & c_{2n_1-1,1} & \cdots & c_{2n_1-1,2n_2-2} & c_{2n_1-1,2n_2-1} \end{bmatrix}$$

be a codeword of the binary two-dimensional array code $C(2n_1, 2n_2)$ constructed by the rectangular three-dimensional interleaving of the two-dimensional array code $C(n_1, n_2)$. If the code $C(n_1, n_2)$ is the direct product of two simple parity-check codes C_1 and C_2, then the code $C(l_1, l_2)$, where $l_1 = 2n_1 - 1$ or $2n_1$, $l_2 = 2n_2 - 1$ or $2n_2$, can correct 2×2-clusters of errors.

Let U and V be one-dimensional cyclic (or shortened cyclic) 2-burst-error-correcting $[l_1, k_1]$ and $[l_2, k_2]$ codes over $GF(2^2)$, respectively.

Let $u = (u_0, u_1, \ldots, u_{l_1-1})$ and $v = (v_0, v_1, \ldots, v_{l_2-1})$ be code words of codes U and V, $u_i = (u_i^{(1)}, u_i^{(2)})$ and $v_j = (v_j^{(1)}, v_j^{(2)})$ be binary representations of elements $u_i, v_j \in GF(2^2)$. Suppose that $\sum_{i=0}^{l_1-1} u_i^{(1)} = \sum_{i=0}^{l_1-1} u_i^{(2)} = 0$ for all $u \in U$.

Let $z = (z_{i,j})$ be the $l_1 \times l_2$ array such that
$z_{i,j} = c_{i,j}$ for $i = 0, \ldots, l_1 - 3$, $j = 1, \ldots, l_2 - 3$,
$z_{i,l_2-2} = c_{i,l_2-2} + u_i^{(1)}$ for $i = 1, \ldots, l_1 - 3$,

$$\begin{aligned}
&z_{i,l_2-1} = c_{i,l_2-2} + u_i^{(2)} \text{ for } i = 1,\ldots,l_1-3,\\
&z_{l_1-2,j} = c_{l_1-2,j} + v_j^{(1)} \text{ for } j = 1,\ldots,l_2-3,\\
&z_{l_1-1,j} = c_{l_1-1,j} + v_j^{(2)} \text{ for } j = 1,\ldots,l_2-3,\\
&z_{l_1-2,l_2-2} = c_{l_1-2,l_2-2} + u^{(1)}_{l_1-2} + v^{(1)}_{l_2-2},\quad z_{l_1-1,l_2-2} = c_{l_1-1,l_2-2} + u^{(2)}_{l_1-2} + v^{(1)}_{l_2-2}\\
&z_{l_1-2,l_2-1} = c_{l_1-2,l_2-1} + u^{(2)}_{l_1-2} + v^{(2)}_{l_2-1},\quad z_{l_1-1,l_2-1} = c_{l_1-1,l_2-1} + u^{(2)}_{l_1-2} + v^{(2)}_{l_2-1}.
\end{aligned}$$

Lemma 1 *The set of arrays* $(z_{i,j})$ *is a binary linear two-dimensional array* 2×2*-cluster-error-correcting* $[l_1 \times l_2, l_1 l_2 - 2(l_1 - k_1) - 2(l_2 - k_2) + 4]$ *code* $Z(l_1, l_2)$.

We can use one-dimensional cyclic 2–burst–error–correcting Fire codes over $GF(2^2)$ as codes U and V. Furthermore we can use the following lemma.

Lemma 2 *Let* $p(x)$ *be a irreducible polynomial of degree* m *over* $GF(2^c)$. *Then the polynomial* $g(x) = (x^2+1)p(x)$ *generate the one-dimensional cyclic 2 burst error correcting code over* $GF(2^c)$ *of length* $\frac{2(2^{cm}-1)}{2^c-1}$.

Example. Let m_1, m_2 be positive integers and $l_1 = \frac{2(2^{2m_1}-1)}{3}$, $l_2 = \frac{2(2^{2m_2}-1)}{3}$. Let U and V be cyclic $[l_1, k_1]$ and $[l_2, k_2]$ codes over $GF(2^2)$, satisfying lemma 2. Then the binary two-dimensional array code $Z(l_1, l_2)$ of size $l_1 \times l_2$ with $r = 2m_1 + 2m_2 + 4$ parity–check symbols can correct single error clusters of size 2×2. The excess redundancy of the code $Z(l_1, l_2)$ is

$$\tilde{r}_{n_1,n_2}(2,2) = 6.$$

Now we consider the construction of three-dimensional array 2×2×2-cluster-error-correcting codes. For constructing the linear binary three-dimensional array codes with small excess redundancy we use one-dimensional cyclic (or shortened cyclic) 2-burst-error-correcting codes over $GF(2^4)$ with the same property.

Let α be a primitive element of the Galois field $GF(2^4)$. There is a one-to-one correspondence between elements α^i and binary arrays of size $GF(2^4)$. Therefore we can represent the cyclic 2-burst-error-correcting $[l, k]$ code L over $GF(2^4)$ as the binary three-dimensional array $[l \times 2 \times 2, 4k]$ code U_L correcting $2 \times 2 \times 2$-clusters of errors. A code word of the code U_L is the the rectangular parallelepiped $(u_{i,j,h})$ of size $l \times 2 \times 2$.

Theorem 1 *Let* $C(l_1, l_2, l_3)$ *be the binary linear three-dimensional twice interleaved array code correcting* $2 \times 2 \times 2$*-clusters of errors. Let* L_i, L_j *and* L_h *be the cyclic 2-burst-error-correcting* $[l_{j,h}, k_{j,h}]$, $[l_{h,i}, k_{h,i}]$ *and* $[l_{i,j}, k_{i,j}]$ *codes over*

$GF(2^4)$, satisfying Lemma 2, respectively. Then the sum of the codes $C(l_1,l_2,l_3)$ and U_{L_i}, U_{L_j}, U_{L_h} is the binary linear three-dimensional array code $Z(l_1,l_2,l_3)$ of size $l_1 \times l_2 \times l_3$ with $r = 4(l_{i,j} - k_{i,j}) + 4(l_{j,h} - k_{j,h}) + 4(l_{h,i} - k_{h,i}) - 16$ parity-check symbols that can correct single error clusters of size $2 \times 2 \times 2$.

The excess redundancy of the code $Z(l_1,l_2,l_3)$ is

$$\bar{r}_{l_1,l_2,l_3}(2,2,2) = 17.$$

References

[1] K. A. S. Abdel-Ghaffar, R. J. McEliece, H. C. A. van Tilborg, Two-dimensional burst identification codes and their use in burst correction, *IEEE Trans. Inform. Theory* 34, 1988, 494-504.

[2] I. M. Boyarinov, Three-dimensional array SCEC codes with small excess redundancy, *Proc. Tenth Intern. Workshop ACCT*, Zvenigorod, Russia, 2006.

[3] I. M. Boyarinov, Three-dimensional cluster-error-correcting array codes, *Proc. Intern. Symp. Inform. Theory*, Lausanne, Switzerland, 2002, 118.

[4] I. M. Boyarinov, Two-dimensional array SCEC codes with small excess redundancy, *Proc. Ninth Intern. Workshop ACCT*, Kranevo, Bulgaria, 2004, 80-85.

[5] M. Blaum, P. G. Farrell, Array codes for cluster-error correction, *Electr. Letters* 30, 1994, 1752-1753.

[6] I. M. Boyarinov, G. A. Kabatianski, Products of binary arithmetic codes correcting multiple random errors, *Proc. VI Conf. Coding Inform. Theory*, part III, 1975, 54-59, Moscow-Tomsk (in Russian).

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 51–56*

New 5-ary and 7-ary linear codes [1]

Rumen Daskalov daskalov@tugab.bg
Elena Metodieva metodieva@tugab.bg
Department of Mathematics, Technical University of Gabrovo,
5300 Gabrovo, BULGARIA

Abstract. Let $[n, k, d]_q$ code be a linear code of length n, dimension k and Hamming minimum distance d over GF(q). In this paper record-breaking codes with parameters $[30, 10, 15]_5$, $[33, 11, 16]_5$, $[41, 10, 22]_5$, $[24, 14, 8]_7$, $[40, 11, 22]_7$, $[60, 10, 38]_7$, $[60, 13, 34]_7$, $[88, 8, 63]_7$, $[96, 11, 64]_7$, $[96, 13, 61]_7$ and $[96, 15, 58]_7$ are constructed.

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 weight is d. For linear codes, the minimum distance is equal to the minimum weight of the nonzero codewords. The orthogonal code $C^\perp$ of C is the set of words of length n that are orthogonal to all codewords in C, w.r.t. the ordinary inner product.

A $k \times n$ matrix G_C having as rows the vectors of a basis of a linear code C is called a generator matrix for C.

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 a smallest one. The codes with such parameters are called optimal.

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 $A_0 = 1$, $A_d = \alpha, \dots, A_n = \gamma$ is expressed as $0^1 d^\alpha \dots n^\gamma$ also.

In the last years many good linear codes over GF(5) and GF(7) were constructed. In [2] Daskalov and Gulliver constructed 44 good codes and presented a table with lower and bounds on the minimum distances for $1 \le k \le 8, 1 \le n \le 100$. In [3] Daskalov, Hristov and Metodieva constructed 32 QC and QT codes. Grassl and White presented 28 new codes in [4] and 55 in [5]. Maruta

[1] This work was partially supported by the Ministry of Education and Science under contract in TU-Gabrovo.

et al. constructed in [6], [7] and [8] eighteen, twenty four and twenty six new codes respectively. Six new codes were constructed in [9]. Fifty eight new linear codes over $GF(7)$ are constructed and a table for the minimum distances ($k \leq 7$, $n \leq 100$) is presented in [10]. Thirty tree linear codes over $GF(7)$ are constructed in [11]. New linear codes ($n \leq 50$) over $GF(7)$ are constructed in [12], [13], [14]. Good linear codes, including and some high-rate codes, are presented also in [15] and [16].

In the presented paper we continue our investigation from [15] and [16]. In the time of construction the codes presented in this paper improved the respective lower bounds on the minimum distances in Grassl's tables [17] and now are the best-known such codes.

2 Quasi-cyclic codes

The basic object in our considerations is the class of quasi-cyclic codes. A code C is said to be quasi-cyclic (QC or p-QC) if a cyclic shift of a codeword by p positions results in another codeword. The length, n, of a p-QC code is a multiple of p, so hat $n = pm$ [18]. With a suitable permutation of coordinates [19] a class of QC codes can be constructed from $m \times m$ circulant matrices. In this case, C has a generator matrix of the following form

$$G = [B_1, B_2, \ldots, B_p], \tag{1}$$

where B_i are circulant matrices.

The algebra of $m \times m$ circulant matrices over $\mathrm{GF}(q)$ is isomorphic to the algebra of polynomials in the ring $\mathrm{GF}(q)[x]/(x^m - 1)$ if B is mapped onto the polynomial, $b(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{m-1}x^{m-1}$, formed from the entries in the first row of B [1]. The polynomials $b_i(x)$, associated with a QC code are called the *defining polynomials* [18].

The dimension k of the QC code is equal to the degree of $h(x)$ [20], where

$$h(x) = \frac{x^m - 1}{\gcd\{x^m - 1, b_0(x), b_1(x), \cdots, b_{p-1}(x)\}}.$$

If $\deg h(x) = m$, then the dimension of the code is m, and (1) is a generator matrix. If $\deg h(x) = k < m$, then the matrices B_i in (1) are near circulant matrices i.e. circulant matrix with $m - k$ rows deleted. In this case the QC code is called *degenerate* [18].

3 The new codes over GF(5) and GF(7)

Theorem 3.1 *There exist QC codes with parameters* $[30, 10, 15]_5$, $[33, 11, 16]_5$.

Proof: The coefficients of the defining polynomials and the weight distributions of the codes are:

A $[30, 10, 15]_5$ **code:**
0000000001, 1124204402, 1121400003;
$0^1\ 15^{2080}\ 16^{7020}\ 17^{22520}\ 18^{65280}\ 19^{149880}\ 20^{346292}\ 21^{660040}\ 22^{1083800}\ 23^{1508400}\ 24^{1746660}$
$25^{1679120}\ 26^{1291840}\ 27^{772280}\ 28^{329420}\ 29^{88360}\ 30^{12632}$

A $[33, 11, 16]_5$ **code:**
01222012441, 10100324344, 00000000001;
$0^1\ 16^{2420}\ 17^{9460}\ 18^{30844}\ 19^{93764}\ 20^{261800}\ 21^{651068}\ 22^{1439108}\ 23^{2717748}\ 24^{4568872}\ 25^{6545924}$
$26^{8075364}\ 27^{8360044}\ 28^{7185992}\ 29^{4938164}\ 30^{2636700}\ 31^{1025948}\ 32^{254188}\ 33^{30716}$

Theorem 3.2 *There exist a code with parameters* $[41, 10, 22]_5$.

Proof: The generator matrix and the weight distribution of a code are:

$$G = \begin{pmatrix}
10000000001124204402112140000311231314300 \\
01000000002112420440311214000001123131434 \\
00100000000211242044031121400030112313140 \\
00010000004021124204003112140043011231314 \\
00001000004402112420000311214014301123134 \\
00000100000440211242000031121431430112311 \\
00000010002044021124400003112113143011234 \\
00000001004204402112140000311231314301120 \\
00000000102420440211214000031123131430113 \\
00000000011242044021121400003112313143011
\end{pmatrix}$$

$0^1\ 22^{744}\ 23^{3464}\ 24^{9868}\ 25^{25452}\ 26^{61344}\ 27^{134916}\ 28^{273636}\ 29^{486300}\ 30^{782716}\ 31^{1110920}$
$32^{1388616}\ 33^{1515936}\ 34^{1423136}\ 35^{1140784}\ 36^{758828}\ 37^{410424}\ 38^{173964}\ 39^{52796}\ 40^{10672}\ 41^{1108}$

Theorem 3.3 *There exist high rate code with parameters* $[24, 14, 8]_7$.

Proof: The generator matrix and the weight distribution of a code are:

$$G = \begin{pmatrix}
423610423610000000000000 \\
465201465201000000000000 \\
026330435400100000000000 \\
564363535200010000000000 \\
610616214000001000000000 \\
061061621400000100000000 \\
561536654500000010000000 \\
125603034200000001000000 \\
643040234600000000100000 \\
410214446300000000010000 \\
256361652200000000001000 \\
656116026400000000000100 \\
550341464300000000000010 \\
263304354000000000000001
\end{pmatrix}.$$

$0^1\ 8^{3654}\ 9^{50832}\ 10^{425052}\ 1^{3186288}\ 12^{20822340}\ 13^{115439688}\ 14^{544238244}\ 15^{2176066200}$
$16^{7345453896}\ 17^{20738832048}\ 18^{48394140564}\ 19^{91686811104}\ 20^{137538690156}\ 21^{157181317512}$
$22^{128605224636}\ 23^{67097856600}\ 24^{16774514034}$

Theorem 3.4 *There exist QC codes with parameters* $[40, 11, 22]_7$, $[60, 10, 38]_7$, $[60, 13, 34]_7$, $[88, 8, 63]_7$, $[96, 11, 64]_7$, $[96, 13, 61]_7$ *and* $[96, 15, 58]_7$.

Proof: The coefficients of the defining polynomials and the weight distributions of the codes are:

A $[40, 11, 22]_7$ **code**: 15564400610000000000, 35414452362321136401;
$0^{1}\ 22^{4560}\ 23^{21600}\ 24^{96180}\ 25^{364080}\ 26^{1217880}\ 27^{3810000}\ 28^{10696560}\ 29^{26406360}\ 30^{58171548}$
$31^{112771200}\ 32^{189940560}\ 33^{276569400}\ 34^{341670240}\ 35^{351016728}\ 36^{292879380}\ 37^{189980400}\ 38^{89863260}$
$39^{27694680}\ 40^{4152126}$

A $[60, 10, 38]_7$ **code**: 331000000000, 250210261351, 403105264111, 514042322401, 560440523051;
$0^{1}\ 38^{2376}\ 39^{9912}\ 40^{30276}\ 41^{96336}\ 42^{249240}\ 43^{623880}\ 44^{1450476}\ 45^{3045432}\ 46^{6031620}\ 47^{10754712}$
$48^{17446698}\ 49^{25622640}\ 50^{33956676}\ 51^{39919032}\ 52^{41353470}\ 53^{37437768}\ 54^{29186076}\ 55^{19092096}$
$56^{10229562}\ 57^{4301448}\ 58^{1338948}\ 59^{267696}\ 60^{28878}$

A $[60, 13, 34]_7$ **code**: 12344321000000000000, 42613561501564230031, 05661452241504230031;
$0^{1}\ 34^{3720}\ 35^{15600}\ 36^{74280}\ 37^{274920}\ 38^{1017480}\ 39^{3373200}\ 40^{10644720}\ 41^{31248600}\ 42^{85083120}$
$43^{213281760}\ 44^{494367840}\ 45^{1054182360}\ 46^{2063203560}\ 47^{3687216960}\ 48^{5992297590}\ 49^{8804553960}$
$50^{11621292600}\ 51^{13672393920}\ 52^{14198102910}\ 53^{12859026600}\ 54^{10002538320}\ 55^{6547491720}\ 56^{3507203700}$
$57^{1476258960}\ 58^{458310840}\ 59^{93175320}\ 60^{9375846}$

A $[88, 8, 63]_7$ **code**: 12306036, 14510603, 00012525, 00106412, 00001432, 13513142, 15511022, 11123235, 12025240,
11011353, 00000001;
$0^{1}\ 63^{1632}\ 64^{4512}\ 65^{8256}\ 66^{16656}\ 67^{34128}\ 68^{65160}\ 69^{106704}\ 70^{174480}\ 71^{268656}\ 72^{381294}\ 73^{503328}$
$74^{604464}\ 75^{672144}\ 76^{701844}\ 77^{654720}\ 78^{550776}\ 79^{427056}\ 80^{281670}\ 81^{165792}\ 82^{86736}\ 83^{37248}\ 84^{12576}$
$85^{1128}\ 86^{840}$

A $[96, 11, 64]_7$ **code**: 25251541063106140014605501530611260541000000000,
36004114320130111352520061523224563525221162226 1;

$0^{1}\ 64^{2754}\ 65^{6912}\ 66^{18576}\ 67^{67392}\ 68^{153216}\ 69^{387456}\ 70^{916560}\ 71^{1975968}\ 72^{4000140}\ 73^{8071200}$
$74^{15012432}\ 75^{26381184}\ 76^{43720848}\ 77^{68309856}\ 78^{99434448}\ 79^{136235232}\ 80^{173901780}\ 81^{205788480}$
$82^{225847152}\ 83^{228783744}\ 84^{212396592}\ 85^{179650368}\ 86^{138030048}\ 87^{95439168}\ 88^{58289796}\ 89^{31515552}$
$90^{14662464}\ 91^{5829408}\ 92^{1894032}\ 93^{497376}\ 94^{92592}\ 95^{12960}\ 96^{1056}$

A $[96, 13, 61]_7$ **code**: 24103222461356613616643353663625345100000000000,
02462451302351623306255564406650343535145145036 1;
$0^{1}\ 61^{6048}\ 62^{15552}\ 63^{45792}\ 64^{147348}\ 65^{443232}\ 66^{1188480}\ 67^{3094560}\ 68^{7801200}\ 69^{18779040}$
$70^{43454160}\ 71^{95385600}\ 72^{198956064}\ 73^{393827040}\ 74^{734367600}\ 75^{1294196352}\ 76^{2144382192}$
$78^{590103968}\ 81^{1491214240}\ 84^{1816395520}\ 85^{228415168}\ 87^{367422368}\ 89^{1542674592}\ 90^{719256960}$
$91^{285332256}\ 92^{93211056}\ 93^{24076704}\ 94^{4692384}\ 95^{611136}\ 96^{42486}$

A $[96, 15, 58]_7$ **code**: 60314406523426013545321530234002110000000000000,
64031356110540451234350266156116532641126112465 1;
$0^{1}\ 58^{4320}\ 59^{11808}\ 60^{57744}\ 61^{203328}\ 62^{656928}\ 63^{2152320}\ 64^{6641874}\ 65^{19658304}\ 66^{55177152}$
$67^{148230720}\ 68^{379245744}\ 69^{923984160}\ 70^{2137170528}\ 71^{4403042496}\ 72^{1194511828}\ 73^{2125657280}$

$74^{1643946448}$ $76^{1965003456}$ $77^{496025632}$ $79^{680604032}$ $80^{424402100}$ $81^{367055072}$ $82^{1333042272}$ $86^{703990672}$ $87^{929886976}$ $90^{937797808}$ $91^{1080566592}$ $92^{260736944}$ $93^{1174700736}$ $94^{224626032}$ $95^{28479744}$ $96^{1770216}$

References

[1] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.

[2] R. Daskalov, T. Gulliver, Bounds on minimum distance for linear codes over GF(5), *AAECC* 9, 1999, 521-546.

[3] R. Daskalov, P. Hristov, E. Metodieva, New minimum distance bounds for linear codes over GF(5), *Discr. Math.* 275, 2004, 97-110.

[4] M. Grassl, G. White, New goog linear codes by special puncturings, *Proc. ISIT2004*, Chicago, USA, 2004, 454.

[5] M. Grassl, G. White, New codes from chains of quasi-cyclic codes, *Proc. ISIT2005*, Adelaide, Australia, 2005, 2095-2099.

[6] T. Maruta, M. Takenaka, M. Shinohara, Y. Shobara, Constructing new linear codes from pseudo-cyclic codes, *Proc. Ninth Intern. Workshop ACCT*, Kranevo, Bulgaria, 2004, 292-298.

[7] T. Maruta, M. Shinohara, F. Yamane, K. Tsuji, E. Takanaka, H. Miki. R. Fujiwara, New linear codes from cyclic and generalized cyclic codes by puncturing, *Proc. Tenth Intern. Workshop ACCT*, Zvenigorod, Rissia, 2006, 194-197.

[8] T. Maruta, M. Shinohara, M. Takenaka, Constructing linear codes from some orbits of projectivities, *Discr. Math.*, to appear.

[9] M. Braun, A. Kohnert, A. Wassermann, Optimal linear codes from matrix groups, *IEEE Trans. Inform. Theory* 51, 2005, 4247-4251.

[10] R. N. Daskalov, T. A. Gulliver, Minimum distance bounds for linear codes over GF(7), *JCMCC* 36, 2001, 175-191.

[11] R. N. Daskalov, T. A. Gulliver, New minimum distance bounds for linear codes over small fields, *Probl. Pered. Inform.* 37, 3, 2001, 24-33.

[12] R. N. Daskalov, P. Hristov. New one-generator quasi-cyclic codes over GF(7), *Probl. Pered. Inform.* 38, 1 , 2002, 59-63.

[13] T. Rehfinger, N. S. Babu, K. Zimmermann, New good codes via CQuest – a system for the silicon search of linear codes, *Algebr. Combin. Appl.*, A. Betten et al., eds, Springer, 2001, 294-306.

[14] E. Metodieva, Six new linear codes over GF(7), *Math. Educ. Math.*, Sofia, 2004, 158-161.

[15] R. Daskalov, Some high-rate linear codes over GF(5) and GF(7), *Probl. Pered. Inform.* 43, 2, 2007, 65-73.

[16] R. Daskalov, E. Metodieva, Minimum distance bounds for 7-ary linear codes, *Proc. Fifth Intern. Workshop OCRT*, White Lagoon, Bulgaria, 2007, 62-67.

[17] M. Grassl, Bounds on the minimum distance of linear codes [electronic table; online], `http://www.codetables.de`.

[18] P. P. Greenough, R. Hill, Optimal ternary quasi-cyclic codes, *Des., Codes Crypt.* 2, 1992, 81-91.

[19] T. Koshy, Polynomial approach to quasi-cyclic codes, *Bul. Cal. Math. Soc.* 69, 1977, 51-59.

[20] G. E. Séguin, G. Drolet, The theory of 1-generator quasi-cyclic codes, Technical Report, Royal Military College of Canada, Kingston, ON, 1991.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 57-62*

On the spectrum of sizes of complete caps in projective spaces $PG(n,q)$ of small dimension

Alexander Davydov adav@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences, Bol'shoi Karetnyi per. 19, GSP-4, Moscow, 127994, RUSSIA

Giorgio Faina, Stefano Marcugini, Fernanda Pambianco
faina, gino, fernanda@dipmat.unipg.it
Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, Perugia, 06123, ITALY

Dedicated to the memory of Giuseppe Pellegrino (1926-2007)

Abstract. For projective spaces $PG(n,q)$ of small dimension, new sizes of complete caps including small these are obtained. The corresponding tables are given. A generalization of Segre's construction of complete caps in $PG(3,2^h)$ is described. In $PG(2,q)$, for $q=17$, $\delta=4$, and $q=19,27$, $\delta=3$, we give complete $(\frac{1}{2}(q+3)+\delta)$-arcs other than conics and sharing $\frac{1}{2}(q+3)$ points with an irreducible conic. We have proven they are unique up to collineations.

1 Introduction

Let $PG(n,q)$ be the projective space of dimension n over the Galois field F_q of q elements. A k-cap in $PG(n,q)$ is a set of k points, no three of which are collinear. A k-cap in $PG(n,q)$ is complete if it is not contained in a $(k+1)$-cap of $PG(n,q)$. If $n=2$, then a k-cap is called a k-arc. We use the following notations for $PG(n,q)$: $m_2(n,q)$ is the size of the largest complete cap, $m_2'(n,q)$ is the size of the second largest complete cap, and $t_2(n,q)$ is the size of the smallest complete cap. The corresponding *best known* values are denoted by $\overline{m}_2(n,q)$, $\overline{m}_2'(n,q)$, and $\bar{t}_2(n,q)$.

In all tables new bounds and sizes of complete caps obtained in this work are marked by the asterisk $\star$ and are written by the bold font.

For the spectrum of possible sizes of complete caps in the spaces of small dimension, the known results are collected in [3],[6]. Using recent results from literature and computer search done in this work we obtained new upper bounds on $t_2(n,q)$ and new sizes of complete caps. As result, we essentially updated tables of [3],[6], see Tables 1-4 below.

Also we generalize Segre's construction [19] of complete caps in $PG(3,2^h)$ basing on ideas of unpublished manuscript [16], see Theorem 4.

2 Complete arcs in planes $PG(2,q)$

The smallest known sizes $\bar{t}_2 = \bar{t}_2(2,q)$ are given in Table 1. For $q = 2^7$, see [12].

Table 1: The smallest known sizes $\bar{t}_2 = \bar{t}_2(2,q) < 4\sqrt{q}$ of complete arcs in planes $PG(2,q)$. $A_q = \lfloor 4\sqrt{q} - \bar{t}_2(2,q) \rfloor$

q	$\bar{t}_2$	A_q	q	$\bar{t}_2$	A_q	q	$\bar{t}_2$	A_q	q	$\bar{t}_2$	A_q	q	$\bar{t}_2$	A_q
3	4.	2	101	30	10	256	56	8	439	78	5	641	**99**	2
4	6.	2	103	31	9	257	56	8	443	79	5	643	**99**	2
5	6.	2	107	32	9	263	**56**	8	449	80	4	647	99	2
7	6.	4	109	32	9	269	57	8	457	81	4	653	**100**	2
8	6.	5	113	33	9	271	58	7	461	81	4	659	100	2
9	6.	6	121	34	10	277	59	7	463	82	4	661	**100**	2
11	7.	6	125	35	9	281	59	8	467	82	4	673	**102**	1
13	8.	6	127	35	10	283	**59**	8	479	83	4	677	103	1
16	9.	7	128	34	11	289	60	8	487	84	4	683	103	1
17	10.	6	131	36	9	293	61	7	491	84	4	691	104	1
19	10.	7	137	37	9	307	**62**	8	499	85	4	701	**104**	1
23	10.	9	139	37	10	311	63	7	503	85	4	709	105	1
25	12.	8	149	39	9	313	63	7	509	85	5	719	106	1
27	12.	8	151	39	10	317	**63**	8	512	86	4	727	106	1
29	13.	8	157	40	10	331	**65**	7	521	**86**	5	729	104	4
31	14	8	163	41	10	337	66	7	523	**86**	5	733	107	1
32	14	8	167	42	9	343	67	7	529	88	4	739	**107**	1
37	15	9	169	**42**	10	347	67	7	541	89	4	743	**108**	1
41	16	9	173	44	8	349	**67**	7	547	**89**	4	751	108	1
43	16	10	179	44	9	353	68	7	557	**90**	4	757	109	1
47	18	9	181	45	8	359	69	6	563	92	2	761	109	1
49	18	10	191	46	9	361	69	7	569	93	2	769	110	0
53	18	11	193	47	8	367	70	6	571	93	2	773	111	0
59	20	10	197	47	9	373	71	6	577	93	3	787	112	0
61	20	11	199	47	9	379	71	6	587	94	2	797	112	0
64	22	10	211	49	9	383	71	7	593	95	2	809	113	0
67	23	9	223	51	8	389	**72**	6	599	95	2	811	**113**	0
71	22	11	227	51	9	397	73	6	601	96	2	821	114	0
73	24	10	229	52	8	401	74	6	607	**96**	2	823	**114**	0
79	26	9	233	52	9	409	75	5	613	97	2	827	**115**	0
81	26	10	239	53	8	419	76	5	617	97	2	829	**115**	0
83	27	9	241	53	9	421	**76**	6	619	**97**	2	839	**115**	0
89	28	9	243	54	8	431	77	6	625	96	4	841	112	4
97	30	9	251	55	8	433	**77**	6	631	**98**	2			

Through the paper for new computer results we used the randomized greedy algorithms [3, Sec. 2],[6, Sec. 2], the back-tracking algorithms [3, Sec. 2], the breadth-first algorithm, algorithms combining orbits of groups, and other geometrical algorithms.

Theorem 1 *In $PG(2,q)$ we have $t_2(2,q) < 4\sqrt{q}$ for $3 \leq q \leq 841$. In addition,*

$$\begin{array}{lll}
t_2(2,q) \leq 4\sqrt{q}-8 & \mathit{for} & 23 \leq q \leq 269, \quad q = 281, 283, 289, 307, 317; \\
t_2(2,q) \leq 4\sqrt{q}-7 & \mathit{for} & 19 \leq q \leq 353, \quad q = 361, 383; \\
t_2(2,q) \leq 4\sqrt{q}-6 & \mathit{for} & 9 \leq q \leq 401, \quad q = 421, 431, 433; \\
t_2(2,q) \leq 4\sqrt{q}-5 & \mathit{for} & 8 \leq q \leq 443, \quad q = 509, 521, 523; \\
t_2(2,q) \leq 4\sqrt{q}-4 & \mathit{for} & 7 \leq q \leq 557, \quad q = 625, 729, 841; \\
t_2(2,q) \leq 4\sqrt{q}-2 & \mathit{for} & 3 \leq q \leq 661; \\
t_2(2,q) \leq 4\sqrt{q}-1 & \mathit{for} & 3 \leq q \leq 761.
\end{array}$$

Theorem 2 *There exist the following complete k-arcs in $PG(2,q)$ with $k \leq m'_2(2,q)$:*

$PG(2,64)$: $22 \leq k \leq 35$, $k = 42$ [5] , $k = 57$.
$PG(2,128)$: $34 \leq k \leq 67$. $PG(2,163)$: $41 \leq k \leq 85$.
$PG(2,167)$: $42 \leq k \leq 87$.

Theorem 3 *In $PG(2,q)$, q odd, we denote by $K_q(\delta)$ a complete $(\frac{1}{2}(q+3)+\delta)$-arc other than conic and sharing $\frac{1}{2}(q+3)$ points with an irreducible conic. Let Δ_q be the maximal possible value of δ. Then $\Delta_{17} = 4$, $\Delta_{19} = \Delta_{27} = 3$, $\Delta_{11} = \Delta_{23} = \Delta_{29} = \Delta_{31} = 2$. There is no any arc $K_{17}(3)$. For $\delta = 3, 4$ and $q \leq 27$, the arcs $K_q(\delta)$ are unique up to collineations.*

Theorem 3 is proved by an exhaustive computer search. Note also that $\Delta_{25} = 2$ [14], as $25^2 \equiv 1 \pmod{16}$, and $\Delta_{13} = 4$ [1]. One can compare the results of [17],[18] with Theorem 3 and the arcs $K_q(\Delta_q)$ written below. The unique 14-arc $K_{17}(4)$ is a counterexample to [18]. The 14-arc $K_{19}(3)$ is obtained in [18] but we have proven that it is unique. Finally, the unique 18-arc $K_{27}(3)$ is new.

Points of the unique 14-arc $K_{17}(4)$ are given in [7]: { (1,10,12),(1,6,8); (1,0,6),(1,0,11), (1,1,4),(1,1,13),(1,6,9),(1,10,5),(1,14,3),(1,3,14); (0,1,3),(0,1,0), (1,5,1),(1,14,10)}. The first ten points lie on the conic $3x_1^2 + x_2^2 = 2x_0^2$ and the last four are placed outside it. By semicolon we separate orbits of the stabilizer group. For $K_{17}(4)$ the stabilizer is the dihedral group $\mathbf{D}_4$ of order eight.

The unique 14-arc $K_{19}(3)$ obtained in this work may be represented with the following coordinates: {(1,5,6), (1,2,4); (1,0,0), (1,7,11),(1,13,17); (1,1,1),(1,3,9), (1,4,16),(1,6,17), (1,9,5),(1,17,4); (1,13,6), (1,1,11),(1,6,8)}. The first 11 points lie on the conic $x_1^2 = x_0x_2$.

The unique 18-arc $K_{27}(3)$ may be represented with the following coordinates: {(1,14,1), (1,12,23),(1,10,19); (1,0,0), (0,0,1),(1,2,3),(1,22,17),(1,13,25), (1,11,21); (1,8,15),(1,20,13), (1,19,11),(1,16,5),(1,4,7),(1,5,9); (0,1,0),(1,6,8), (1,21,12)}. The field F_{27} is generated by the polynomial $x^3 - x^2 - 2$. Elements of

F_{27} are represented as follows: $0 = 0$, $\alpha^i = i+1$, where α is a primitive element of the field. The first 15 points of $K_{27}(3)$ belong to the conic $x_1^2 = x_0x_2$.

The stabilizer group of the arcs $K_{19}(3)$ and $K_{27}(3)$ is the symmetric group $\mathbf{S}_3$ of order six. In $K_{17}(4), K_{19}(3)$, and $K_{27}(3)$ the points outside the conic lie on the same stabilizer group orbit.

3 Small caps in $PG(n,q)$, $n \geq 3$

In [19] Segre constructed complete $(3q+2)$-caps in $PG(3,2^h)$. In unpublished manuscript [16], connected with the paper [15] and cited in [13, Table 4.8], it is remarked $t_2(3,q) \leq 2q + t_2(2,q)$, $q = 2^h \geq 4$. We generalize ideas of [19],[16].

Theorem 4 *Let $q \geq 4$ be even. For every complete k_2-arc in the plane $PG(2,q)$ there is a complete $(2q+k_2)$-cap in the space $PG(3,q)$.*

Table 2 : The sizes $\bar{t}_2(n,q)$ of the known small complete caps in $PG(n,q)$

n	q	$t_2(n,q)$	$\bar{t}_2(n,q)$	new	n	q	$t_2(n,q)$	$\bar{t}_2(n,q)$	new
4	4	$19 \leq$	20	[2]	5	4	$31 \leq$	50	
4	5	$21 \leq$	31		5	5	$36 \leq$	**82**	⋆
4	7	$29 \leq$	**56**	⋆	5	7	$70 \leq$	**174**	⋆
4	8	$33 \leq$	**53**	⋆	5	8	$91 \leq$	**181**	⋆
4	9	$39 \leq$	87		5	9	$115 \leq$	**302**	⋆
4	11	$52 \leq$	**121**	⋆	6	3	$34 \leq$	44	
4	13	$67 \leq$	**162**	⋆	6	4	$61 \leq$	**114**	⋆
4	16	$91 \leq$	153	[4]	6	5	$80 \leq$	131	
4	17	$100 \leq$	**255**	⋆	6	7	$121 \leq$	349	
5	3	$20 \leq$	22		6	8	$256 \leq$	**437**	⋆

Table 3 : The sizes $\bar{t}_2(3,q)$ of the known small complete caps in $PG(3,q)$

q	$t_2(3,q)$	$\bar{t}_2(3,q)$	new	q	$t_2(3,q)$	$\bar{t}_2(3,q)$	new
7	17	$3q-4=17$		43	$63 \leq$	$3q+25=\mathbf{153}$	⋆
8	$14 \leq$	$3q-4=20$		47	$69 \leq$	$3q+28=\mathbf{169}$	⋆
9	$15 \leq$	$3q-3=24$		49	$72 \leq$	$3q+33=\mathbf{180}$	⋆
11	$18 \leq$	$3q-3=30$		53	$77 \leq$	$3q+36=\mathbf{195}$	⋆
13	$21 \leq$	$3q-3=36$		59	$86 \leq$	$3q+43=\mathbf{220}$	⋆
16	$25 \leq$	$2q+9=41$		61	$89 \leq$	$3q+47=\mathbf{230}$	⋆
17	$26 \leq$	$3q=51$		64	$93 \leq$	$2q+22=\mathbf{150}$	Th. 4
19	$29 \leq$	$3q+1=58$		67	$97 \leq$	$3q+56=\mathbf{257}$	⋆
23	$35 \leq$	$3q+3=72$		71	$103 \leq$	$3q+62=\mathbf{275}$	⋆
25	$38 \leq$	$3q+6=\mathbf{81}$	⋆	73	$106 \leq$	$3q+68=\mathbf{287}$	⋆
27	$41 \leq$	$3q+8=\mathbf{89}$	⋆	79	$114 \leq$	$4q-4=\mathbf{312}$	⋆
29	$43 \leq$	$3q+9=\mathbf{96}$	⋆	81	$117 \leq$	$4q-3=\mathbf{321}$	⋆
31	$46 \leq$	$3q+11=\mathbf{104}$	⋆	83	$120 \leq$	$4q-2=\mathbf{330}$	⋆
32	$48 \leq$	$2q+14=\mathbf{78}$	Th. 4	89	$128 \leq$	$4q-1=\mathbf{355}$	⋆
37	$55 \leq$	$3q+17=\mathbf{128}$	⋆	97	$140 \leq$	$4q+6=\mathbf{394}$	⋆
41	$60 \leq$	$3q+22=\mathbf{145}$	⋆	128		$2q+34=\mathbf{290}$	[12], Th. 4

Theorem 5 *It holds that* $t_2(3,q) \leq 3q$ *if* $2 \leq q \leq 17$; $t_2(3,q) < 4q$ *if* $2 \leq q \leq 89$.

4 On the complete cap sizes in $PG(n,q)$, $n \geq 4$

Theorem 6 *The upper bound on the smallest size* $t_2(n,3)$ *of a complete cap in the ternary projective space* $PG(n,3)$ *has the form* $t_2(n,3) \leq 11 \cdot 2^{n-4}$, $n \geq 4$.

Theorem 7 *There are the following bounds:* $m_2(5,4) \leq 153$, $m_2(6,4) \leq 607$, $m_2(7,3) \leq 404$, $m_2(8,3) \leq 1208$, $m_2(9,3) \leq 3247$, $t_2(4,7) \leq 56$, $t_2(6,4) \leq 114$, $534 \leq m_2(8,3)$.

Table 4 gives sizes of the known complete caps in $PG(n,q)$, $n \geq 4$, $q \geq 3$. We used sizes and bounds from [2],[6, Table 2],[8]-[11],[13, Table 4.5],[15, Table I]. The result $19 \leq t_2(4,4)$ and a complete 21-cap in $PG(4,4)$ are obtained in [2].

Table 4: The sizes of the known complete k-caps in $PG(n,q)$, $n \geq 4$, $q \geq 3$

n	q	$t_2(n,q)$	Sizes k of the known complete caps with $t_2(n,q) \leq k \leq m'_2(n,q)$	$m'_2(n,q)$	$m_2(n,q)$	new
4	3.	11	$k = 11$ and $16 \leq k \leq 19$	19	20	
4	4	$19 \leq$	$20 \leq k \leq 40$	40	41	[2]
4	5	$21 \leq$	$31 \leq k \leq 66$		≤ 88	$\star$
4	7	$29 \leq$	$\mathbf{56} \leq k \leq 124$ and $k = 126, 132$		≤ 238	$\star$
5	3	$20 \leq$	$k = 22$ and $26 \leq k \leq 48$	48	56	$\star$
5	4	$31 \leq$	$50 \leq k \leq 108$ and $k = 112, 126$		$\leq \mathbf{153}$	$\star$
6	3	$34 \leq$	$k = 44$ and $46 \leq k \leq 103$, $k = 112$		≤ 136	$\star$
6	4	$61 \leq$	$\mathbf{114} \leq k \leq 288$		$\leq \mathbf{607}$	$\star$
7	3	$58 \leq$	$88 \leq k \leq 238$ and $243 \leq k \leq 248$		$\leq \mathbf{404}$	$\star$
8	3	$100 \leq$	$176 \leq k \leq 532$ and $k = \mathbf{534}$		$\leq \mathbf{1208}$	$\star$
9	3	$172 \leq$	$352 \leq k \leq 1214$ and $k = 1216$		$\leq \mathbf{3247}$	$\star$

References

[1] A. H. Ali, J. W. P. Hirschfeld, H. Kaneta, The automorphism group of a complete $(q-1)$-arc in $PG(2,q)$, *J. Combin. Des.* 2, 1994, 131-145.

[2] D. Bartoli, A. A. Davydov, S. Marcugini, F. Pambianco, On the complete spectrum of the sizes for complete caps in $PG(4,4)$, in preparation.

[3] A. A. Davydov, G. Faina, S. Marcugini, F. Pambianco, Computer search in projective planes for the sizes of complete arcs, *J. Geom.* 82, 2005, 50-62.

[4] A. A. Davydov, M. Giulietti, S. Marcugini, F. Pambianco, New constructions of small complete caps in $PG(N,q)$, q even, *Proc. XI Intern. Symp. Probl. Redund. Inf. Contr. Syst.*, S.-Petersburg, Russia, 2007, 212-216, available at http://k36.org/redundancy2007.

[5] A. A. Davydov, M. Giulietti, S. Marcugini, F. Pambianco, On sharply transitive sets in $PG(2, q)$, submitted.

[6] A. A. Davydov, S. Marcugini, F. Pambianco, Complete caps in projective spaces $PG(n, q)$, *J. Geom.* 80, 2004, 23-30.

[7] A. A. Davydov, S. Marcugini, F. Pambianco, Complete $(q^2 + q + 8)/2$-caps in the spaces $PG(3, q)$, $q \equiv 2 \pmod 3$ odd prime, and the 20-cap in $PG(3, 5)$, submitted.

[8] Y. Edel, Extensions of generalized product caps, *Des. Codes Crypt.* 31, 2004, 5-14.

[9] Y. Edel, J. Bierbrauer, Recursive constructions for large caps, *Bull. Belgian Math. Soc. - Simon Stevin* 6, 1999, 249-258.

[10] Y. Edel, J. Bierbrauer, Large caps in small spaces, *Des. Codes Crypt.* 23, 2001, 197-212.

[11] Y. Edel, L. Storme, P. Sziklai, New upper bounds on the sizes of caps in $PG(N, 5)$ and $PG(N, 7)$, *J. Combin. Math. Combin. Comput.* 60, 2007, 7-32.

[12] M. Giulietti, Small complete caps in $PG(N, q)$, q even, *J. Comb. Des.* 15, 2007, 420-436.

[13] J. W. P. Hirschfeld, L. Storme, The packing problem in statistics, coding theory, and finite projective spaces: update 2001, *Finite Geom.*, Proc. 4-th Isle of Thorns Conf., A. Blokhuis, J. W. P. Hirschfeld, D. Jungnickel, J. A. Thas, Eds., *Devel. Math.* 3, Kluwer Acad. Publ., Boston, 2000, 201-246.

[14] G. Korchmáros, A. Sonnino, Complete arcs arising from conics, *Discr. Math.* 267, 2003, 181-187.

[15] F. Pambianco, L. Storme, Small complete caps in spaces of even characteristic, *J. Combin. Theory Ser. A* 75, 1996, 70-84.

[16] F. Pambianco, L. Storme, unpublished manuscript (1995).

[17] G. Pellegrino, Un'osservazione sul problema dei k-archi completi in $S_{2,q}$, con $q \equiv 1 \pmod 4$), *Atti Accad. Naz. Lincei Rend.* 63, 1977, 33-44.

[18] G. Pellegrino, Sugli archi completi dei piani $PG(2, q)$, con q dispari, contenenti $(q + 3)/2$ punti di una conica, *Rend. Mat.* 12, 1992, 649-674.

[19] B. Segre, On complete caps and ovaloids in three-dimensional Galois spaces of characteristic two, *Acta Arith.* 5, 1959, 315-332.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria pp. 63-69

Symmetric configurations for bipartite-graph codes

ALEXANDER DAVYDOV adav@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences, Bol'shoi Karetnyi per. 19, GSP-4, Moscow, 127994, RUSSIA

MASSIMO GIULIETTI, STEFANO MARCUGINI, FERNANDA PAMBIANCO
giuliet, gino, fernanda@dipmat.unipg.it
Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, Perugia, 06123, ITALY

Abstract. We propose geometrical methods for constructing square 01-matrices with the same number n of units in every row and column, and such that any two rows of the matrix have at most one unit in the same position. In terms of Design Theory, such a matrix is an incidence matrix of a symmetric configuration. Also, it gives rise to an n-regular bipartite graphs without 4-cycles, which can be used for constructing bipartite-graph codes so that both the classes of their vertices are associated with local constraints (constituent codes). We essentially extend the region of parameters of such matrices by using some results from Galois Geometries. Many new matrices are either circulant or consist of circulant submatrices: this provides code parity-check matrices consisting of circulant submatrices, and hence quasi-cyclic bipartite-graph codes with simple implementation.

1 Introduction

Bipartite-graph codes are studied in the context of low-density parity check (LDPC) codes, see [1],[2],[4],[6]-[8], and the references therein.

In [7] Tanner proposed to associate a bipartite graph T to an $[N, K]$ code C in the following way. Fix a positive integer U, and for any $i = 1, \ldots, U$ choose a set of n_i distinct positions of codewords of C, that is a subset $j_1, \ldots, j_{n_i}$ of $\{1, \ldots, N\}$. One class of vertices $\{V'_1, \ldots, V'_N\}$ of T correspond to the positions of the codewords of C. Let $\{V''_1, V''_2, \ldots, V''_U\}$ be the other class. A vertex V''_i has degree n_i and is adjacent with $V'_{j_1}, \ldots, V'_{j_{n_i}}$. The $[n_i, k_i]$ subcode C_i obtained from C by projection to the positions corresponding to $j_1, \ldots, j_{n_i}$ is called a *local constraint on variables*, while the vertices $V'_1, \ldots, V'_N$ are said to be the *variable vertices* of T. If $n_i - k_i = 1$ holds for all subcodes C_i, one can build the graph T directly using the $U \times N$ parity-check matrix H of the code C : the jth column (ith row) of H is identified with a vertex V'_j (V''_i) and every nonzero entry into H implies an edge of T. Usually, such variant of T is called the *Tanner graph* of the code $\mathcal{C}$ [8].

We consider the following modification of the construction of [7] , see [2],[1]. Let G be an n-regular bipartite graph with two classes of vertices $\{V_1, \ldots, V_m\}$ and $\{V_{m+1}, \ldots, V_{2m}\}$ (i.e. any vertex is adjacent to exactly n vertices, but any two vertices from the same class are not adjacent). Let $\mathcal{C}_t$ be an $[n, k_t]$ *constituent code*, $t = 1, 2, \ldots, 2m$. A *bipartite-graph code* $\mathcal{C} = \mathcal{C}(G; \mathcal{C}_1, \ldots, \mathcal{C}_{2m})$ is a linear $[N, K]$ code with length equal to the number of edges of G, that is $N = mn$. Coordinates of $\mathcal{C}$ are in one-to-one correspondence with the edges of G. In addition, the projection of a codeword of $\mathcal{C}$ to the positions corresponding to the n edges incident to the vertex V_t must be a codeword of the constituent code $\mathcal{C}_t$. We call G a *supporting graph* of the bipartite-graph code $\mathcal{C}$.

To a supporting graph G it can be naturally associated a square 01-matrix $M(m, n)$ of order m with n units in every row and column. The ith row (jth column) of $M(m, n)$ corresponds to the vertex V_i (V_{m+j}). The entry in position (i, j) is 1 if and only if V_i and V_{m+j} are adjacent. It is easily seen that the graph G is 4-cycle free if and only if the matrix $M(m, n)$ does not contain the 2×2 submatrix J_4 consisting of all units. A matrix without submatrix J_4 is called a *J_4-free matrix*.

In order to improve the performance of the code, it is desirable to increase the girth of the graph [7],[8]. We study supporting graphs with girth at least six (i.e. with no 4-cycles). It should be noted that if *the supporting graph* of a bipartite-graph code has girth at least *six*, then the girth of the *Tanner graph* of this code is at least *ten*.

Parameters of the bipartite-graph codes depend on the values of m and n. The goal of this work is *to construct J_4-free matrices $M(m, n)$ with* ***distinct parameters*** *m, n.*

J_4-free matrices for LDPC codes are considered in many papers, see e.g. [1],[4],[8] and the references therein. Mainly, non-square matrices are investigated. It is also known that both symmetric and resolvable non-symmetric 2-$(v, k, 1)$ designs [3] can be used for obtaining J_4-free matrices $M(v, k)$. The reason is that in a 2-$(v, k, 1)$ design every pair of elements is contained in *exactly* one block. Actually, in order to obtain a J_4 -free matrix $M(m, n)$ it is enough that every pair of elements is contained in *at most* one block. An incidence structure with this property is said to be a *configuration* [3, Sec. IV.6]. If a configuration is *symmetric*, then its incidence matrix is a J_4 -free matrix $M(m, n)$.

Even though J_4-free matrices $M(m, n)$ have already been studied in literature, the region of parameters of the constructed matrices is not wide enough if compared to the permanently growing needs of practice, when often exact values of m, n are necessary. Also, it should be considered that distinct constructions of matrices have distinct properties, and clearly some choice can be useful.

In this work we propose a number of constructions of both square and non-square J_4-free matrices based on incidence structures in projective spaces $PG(v,q)$ over Galois fields F_q (see [3],[5] for basic facts on Galois Geometries).

We essentially extend the region of parameters of J_4-free square 01-matrices with the same number of units in every row and column. The obtained matrices have new structures that gives wide choice for code implementation. Many of them either are circulant or consist of circulant submatrices: this provides code parity-check matrices consisting of circulant submatrices which give rise to quasi-cyclic (QC) bipartite-graph codes. QC codes can be encoded with complexity linearly proportional to code length [4],[6].

2 Construction A: a single orbit of a collineation group

Construction A. Take any point orbit $\mathcal{P}$ under the action of a collineation group in an affine or projective space of order q. Choose an integer $n \leq q+1$ such that the set $\mathcal{L}(\mathcal{P},n)$ of lines meeting $\mathcal{P}$ in precisely n points is not empty. Define the following incidence structure: the points are the points of $\mathcal{P}$, the lines are the lines of $\mathcal{L}(\mathcal{P},n)$, the incidence is that of the starting space. Let M be the incidence matrix of such a structure.

Theorem 1 *In Construction A the number of lines of $\mathcal{L}(\mathcal{P},n)$ through a point of $\mathcal{P}$ is a constant r_n. If $n = r_n$, the matrix M in Construction A is a J_4-free matrix $M(|\mathcal{P}|,n)$.*

Example 2 i) *We consider a conic $\mathcal{K}$ in $PG(2,q)$, q odd [5, Sec. 8.2]. Let $\mathcal{P}$ be the set of $\frac{1}{2}q(q-1)$* internal points *to $\mathcal{K}$. It is an orbit under the collineation group $G_\mathcal{K}$ fixing the conic. Let $n=\frac{1}{2}(q+1)$. Then $\mathcal{L}(\mathcal{P},n)$ is the set of lines external to $\mathcal{K}$. We obtain*

$$M(m,n): m=\tfrac{1}{2}q(q-1),\ n=\tfrac{1}{2}(q+1),\ q \text{ odd.}$$

Another orbit $\mathcal{P}_2$ of the group $G_\mathcal{K}$ is the set of $\frac{1}{2}q(q+1)$ external points *to $\mathcal{K}$. We form the set $\mathcal{L}(\mathcal{P}_2,\frac{1}{2}(q-1))$ from $\frac{1}{2}q(q+1)$ bisecants. As a result, we obtain a matrix*

$$M(m,n): m=\tfrac{1}{2}q(q+1),\ n=\tfrac{1}{2}(q-1),\ q \text{ odd.}$$

ii) *Let $\mathcal{P}$ be the complement of a Baer subplane π of $PG(2,q)$, q a square. It is an orbit of the collineation group fixing π. The set $\mathcal{L}(\mathcal{P},q)$ is the set of tangents to π. We obtain*

$$M(m,n): m=q^2-\sqrt{q},\ n=q,\ q \text{ square.}$$

iii) *In $PG(2,q)$, q a square, let $\mathcal{P}$ be the complement of the Hermitian curve [5, Sec. 7.3]. It is an orbit of the group $PGU(3,q)$ fixing the point $(0,0,1)$. We obtain*

$$M(m,n) : m = q^2 + q - q\sqrt{q},\ n = q - \sqrt{q},\ q \text{ square}.$$

It should be noted that Construction A works for any 2-$(v,k,1)$ design D and for any group of automorphisms of D. The role of $q+1$ is played by the size of any block in D.

3 Construction B: an union of orbits of a Singer subgroup

We treat points of $PG(2,q)$ as nonzero elements of F_{q^3}. Elements a,b of F_{q^3} correspond to the same point if and only if $a = xb$, $x \in F_q$. Let α be a primitive element of F_{q^3}. The point represented by α^i is denoted by P_i. Then $PG(2,q) = \{P_0, P_1, \ldots, P_{q^2+q}\}$. The map $\sigma : P_i \mapsto P_{i+1 \pmod{q^2+q+1}}$ is a projectivity of $PG(2,q)$. The group S of order q^2+q+1 generated by σ is called the Singer group of $PG(2,q)$ [5, Sec. 4.2]. Clearly, $P_i = \sigma^i(P_0)$.

For any divisor d of q^2+q+1, the group S has a unique cyclic subgroup $\widehat{S}_d$ of order d, namely the group generated by σ^t, $t = (q^2+q+1)/d$. It is well known that under the action of a cyclic collineation group *the point set and the line set of a projective plane have the same cyclic structure.*

Let $O_0, O_1, \ldots, O_{t-1}$ be the orbits of points of $PG(2,q)$ under the action of the subgroup $\widehat{S}_d$. Clearly, $|O_i| = d$. We arrange indexes so that $P_0 \in O_0$. $O_v = \sigma^v(O_0)$. Then

$$O_i = \{P_i, \sigma^t(P_i), \sigma^{2t}(P_i), \ldots, \sigma^{(d-1)t}(P_i)\},\ i = 0,1,\ldots,t-1. \qquad (1)$$

Let ℓ_0 be a fixed line of $PG(2,q)$ and let $\ell_i = \sigma^i(\ell_0)$. Then the set of lines of $PG(2,q)$ is $L = \{\ell_0, \ell_1, \ell_2, \ldots, \ell_{q^2+q}\}$. Let $L_0, \ldots, L_{t-1}$ be the orbits of the set L under the action of $\widehat{S}_d$. Clearly, $|L_i| = d$. We arrange indexes in such a way that $\ell_0 \in L_0$, $L_v = \sigma^v(L_0)$. Then

$$L_i = \{\ell_i, \sigma^t(\ell_i), \sigma^{2t}(\ell_i), \ldots, \sigma^{(d-1)t}(\ell_i)\},\ i = 0,1,\ldots,t-1. \qquad (2)$$

Theorem 3 *Let $t = (q^2+q+1)/d$ and let $O_0, \ldots, O_{t-1}$ (resp. $L_0, \ldots, L_{t-1}$) be the point (resp. line) orbits under the action of the Singer subgroup $\widehat{S}_d$ of order d. Assume that for points, lines, and orbits, indexes are arranged as in (1) and (2). Then for any i and j, every line of the orbit L_i meets the orbit O_j in the same number of points $w_{j-i \pmod t}$, where $w_u = |\ell_0 \cap O_u|$, $u = 0,1,\ldots,t-1$.*

Corollary 4 *Let d and t be as in Theorem 3. The J_4-free incidence $(q^2+q+1)\times(q^2+q+1)$ matrix V of the plane $PG(2,q)$ can be represented as follows:*

$$V=\begin{bmatrix} C_{0,0} & C_{0,1} & C_{0,2} & \dots & C_{0,t-1}\\ C_{1,0} & C_{1,1} & C_{1,2} & \dots & C_{1,t-1}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ C_{t-1,0} & C_{t-1,1} & C_{t-1,2} & \dots & C_{t-1,t-1}\end{bmatrix}$$

where $C_{i,j}$ is a J_4-free binary **circulant** *$d\times d$ matrix of weight $w_{j-i \pmod t}$.*

Weights w_u of the submatrices $C_{i,j}$ can be written as the circulant $t\times t$ matrix

$$W(V)=\begin{bmatrix} w_0 & w_1 & w_2 & w_3 & \dots & w_{t-2} & w_{t-1}\\ w_{t-1} & w_0 & w_1 & w_2 & \dots & w_{t-3} & w_{t-2}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ w_1 & w_2 & w_3 & w_4 & \dots & w_{t-1} & w_0\end{bmatrix}.$$

Remark 5 *We use in the sequel the following natural* decomposition of square circulant *01*-matrices, *cf. [8, Sec. IV, B]. From now on, we assume that in circulant matrices rows are shifted to the right. A binary circulant $d\times d$ matrix C of weight w is defined by the vector $s(C)=(s_1,s_2,\dots,s_w)$ where the s_i 's are the positions of the units in the first row of C, arranged in such a way that $s_1<s_2<\dots<s_w$. Let $I_d=I_d(0)$ be the identity matrix of order d and let $I_d(v)$ be the circulant permutation $d\times d$ matrix obtained from I_d by shifting of every row by v positions. The matrix C can be treated as the superposition of w matrices $I_d(s_i)$, $i=1,\dots,w$. From the matrix C one can obtain a circulant matrix $C^{(\delta)}$ of weight $w-\delta$ using the superposition of any $w-\delta$ distinct matrices $I_d(s_i)$. It should also be noted that* if the starting matrix C is J_4 -free then any matrix $C^{(\delta)}$ is J_4-free too.

Construction B. Fix some integers $u_1,\dots,u_r$, $0\le u_i\le t-1$. Let V' be a matrix obtained from V by replacing the circulant submatrices $C_{i,j}$ such that $j-i=u_k \pmod t$ with $d\times d$ matrices $C_{i,j}^{(\delta_{u_k})}$ as in Remark 5. Here, and in the rest of the paper, the subscript difference $j-i$ is calculated modulo t. Let $W(V')$ be the matrix $W(V)$ in which corresponding elements w_{j-i} are changed by $w'_{j-i}=w_{j-i}-\delta_{j-i}$. If an $\frac{m}{d}\times\frac{m}{d}$ submatrix of $W(V')$ is such that the sum of elements of every row and every column is equal to the same number n, then the corresponding submatrix of V' is a J_4-free matrix $M(m,n)$.

Example 6 *The matrix $C_{i,j}^{(\delta)}$, obtained from the submatrix $C_{i,j}$ of V as in Remark 5, is a circulant matrix $M(d,w_{j-i}-\delta)$. So, we can form a family of J_4-free circulant matrices.*

$$M(m,n): m=d,\ n=w_u-\delta,\ u=0,1,\dots,t-1,\ \delta=0,1\dots,w_u-1. \quad (3)$$

Example 7 *By Remark 5, from the matrix V several families of J_4-free matrices $M(m,n)$ can be obtained. Significantly, every such matrix consists of circulant submatrices. Sometimes some conditions on weights w'_u of submatrices $C_{i,j}^{(\delta_{j-i})}$ are needed. Here, we provide a list of parameters m, n of some of these families of J_4-free matrices $M(m,n)$.*

i) $m = q^2+q+1$, $n = \sum_{u=0}^{t-1}(w_u - \delta_u) = q+1-\sum_{u=0}^{t-1}\delta_u$, $\delta_u = 0, 1\ldots, w_u$.

ii) $m = cd$, $n = (c-h)w$, $c = 1, 2, \ldots, \left[\frac{k}{2}\right]$, $h = 0, 1, \ldots, c-1$.

(for $w'_0 = w'_1 = \ldots = w'_{k-1} = w$, $k \geq 2$);

iii) $m = cd$, $n = w_0 - \delta_0 + (c-h)w$, $\delta_0 = 0, 1, \ldots, w_0$, $c = 2, 3, \ldots, t-1$, $h = 1, \ldots, c$.

(for $w'_0 = w_0 - \delta_0 \neq w$, $w'_1 = \ldots = w'_{t-1} = w$);

iv) $m = 2d$, $n = 2w$

(for $w'_i = w'_{i+m} = w'_{i+m+k} = w'_{i+2m+k} = w$, $k \geq 1$, $m \geq 1$);

v) $m = (k+1)d$, $n = w'_0 + w'_1 + ... + w'_k$.

(for $w'_{k+1} = w'_0$, $w'_{k+2} = w'_1, \ldots, w'_{2k} = w'_{k-1}$, $k \geq 1$).

Remark 8 *Assume that $M(m,n)$ is a circulant matrix, see e.g. Example 6. Let $M(m,n)$ be defined by the vector $s(M(m,n)) = (s_1, s_2, \ldots, s_n)$, see Remark 5. We consider $M(m,n)$ as a superposition of n circulant permutation $m \times m$ matrices $I_m(s_i)$, $i = 1, \ldots, n$. Assume that for constituent $[n, k_t]$ codes C_t we have $C_1 = \ldots = C_m$, $C_{m+1} = \ldots = C_{2m}$. Let $r_t = n - k_t$. Let also $[c_{j,1}^{(t)} c_{j,2}^{(t)} \ldots c_{j,r_t}^{(t)}]$ be the jth column of a parity check matrix H_t of the q-ary code C_t. Finally, let $H_1 = \ldots = H_m$, $H_{m+1} = \ldots = H_{2m}$. Then the parity check matrix H corresponding to the code associated to the matrix $M(m,n)$ has the form*

$$H = \begin{bmatrix} c_{1,1}^{(1)} I_m & c_{2,1}^{(1)} I_m & \ldots & c_{n,1}^{(1)} I_m \\ \vdots & \vdots & \vdots & \vdots \\ c_{1,r_1}^{(1)} I_m & c_{2,r_1}^{(1)} I_m & \ldots & c_{n,r_1}^{(1)} I_m \\ c_{1,1}^{(m+1)} I_m(s_1) & c_{2,1}^{(m+1)} I_m(s_2) & \ldots & c_{n,1}^{(m+1)} I_m(s_n) \\ \vdots & \vdots & \vdots & \vdots \\ c_{1,r_{m+1}}^{(m+1)} I_m(s_1) & c_{2,r_{m+1}}^{(m+1)} I_m(s_2) & \ldots & c_{n,r_{m+1}}^{(m+1)} I_m(s_n) \end{bmatrix}.$$

The matrix H consists of circulant submatrices, and therefore it defines a QC code, cf. [4],[8]. QC codes can be implemented with relatively small complexity [6].

References

[1] V. B. Afanassiev, A. A. Davydov, V. V. Zyablov, Low density concatenated codes with Reed-Solomon component codes, *Proc. XI Intern. Symp. Problems Redund. Inf. Contr. Syst.*, S.-Petersburg, Russia, 2007, 47-51.

[2] A. Barg, G. Zémor, Distances properties of expander codes, *IEEE Trans. Inform. Theory* 52, 2006, 78-90.

[3] C. J. Colbourn, J. Dinitz, Eds., *The CRC Handbook of Combinatorial Designs*, 2nd edition, Boca Raton, FL: CRC Press, 2006.

[4] E. Gabidulin, A. Moinian, B. Honary, Generalized construction of quasi-cyclic regular LDPC codes based on permutation matrices, *Proc. Intern. Symp. ISIT 2006*, Seattle, USA, 2006, 679-683.

[5] J. W. P. Hirschfeld, *Projective Geometries over Finite Fields*, 2nd ed., Oxford, U.K.: Oxford Science, 1998.

[6] Z.-W. Li, L. Chen, L. Zeng, S. Lin, W. H. Fong, Efficient encoding of quasi-cyclic low-density parity-check codes, *IEEE Trans. Commun.* 54, 2006, 71-81.

[7] R. M. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory* 27, 1981, 533-547.

[8] J. Xu, L. Chen, I. Djurdjevic, K. Abdel-Ghaffar, Construction of regular and irregular LDPC codes: geometry decomposition and masking, *IEEE Trans. Inform. Theory* 53, 2007, 121-134.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 70-75*

Linear covering codes over nonbinary finite fields

Alexander Davydov adav@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences,
Bol'shoi Karetnyi per. 19, GSP-4, Moscow, 127994, RUSSIA

Massimo Giulietti, Stefano Marcugini, Fernanda Pambianco
giuliet, gino, fernanda@dipmat.unipg.it
Dipartimento di Matematica e Informatica, Università degli Studi di Perugia,
Via Vanvitelli 1, Perugia, 06123, ITALY

Abstract. For a prime power q and for integers R, η with $R > 0$, $0 \le \eta \le R-1$, let $\mathcal{A}_{R,q}^{(\eta)} = (\mathcal{C}_{n_i})_i$ denote an infinite sequence of q-ary linear $[n_i, n_i - r_i]_q R$ codes $\mathcal{C}_{n_i}$ with covering radius R and such that the following two properties hold: (a) the codimension $r_i = Rt_i + \eta$, where $(t_i)_i$ is an increasing sequence of integers; (b) the length n_i of $\mathcal{C}_i$ coincides with $f_q^{(\eta)}(r_i)$, where $f_q^{(\eta)}$ is an increasing function. In this paper, sequences $\mathcal{A}_{R,q}^{(\eta)}$ with asymptotic covering density bounded from above by a constant independent of q are constructed for an arbitrary R, and for each value of $\eta \in \{0, 1, \dots, R-1\}$, under the condition that $q = (q')^R$. The key tool is the description of new small saturating sets in projective spaces over finite fields, which are the starting point for the q^m-concatenating constructions of covering codes. A new concept of N-fold strong blocking set is introduced. Several upper bounds on the length function of covering codes and on the smallest sizes of saturating sets are improved.

1 Introduction

Denote by F_q the Galois field with q elements. Let F_q^n be the n-dimensional vector space over F_q. Denote by $[n, n-r]_q$ a q-ary *linear code* of length n and codimension r. The *covering radius* of an $[n, n-r]_q$ code is the least integer R such that F_q^n is covered by spheres of radius R centered on codewords. An $[n, n-r]_q R$ code is an $[n, n-r]_q$ code with covering radius R. For an introduction to coverings of vector spaces over finite fields, see [1] .

The covering quality of an $[n, n-r(\mathcal{C})]_q R$ code $\mathcal{C}$ can be measured by its *covering density*

$$\mu_q(n, R, \mathcal{C}) = q^{-r(\mathcal{C})} \sum_{i=0}^{R} (q-1)^i \binom{n}{i} \ge 1. \tag{1}$$

From the point of view of the covering problem, the best codes are those with small covering density.

For given integers R, η with $R > 0$, $0 \leq \eta \leq R-1$, and for a fixed prime power q, let $\mathcal{A}_{R,q}^{(\eta)} = (\mathcal{C}_{n_i})_i$ denote an infinite sequence of q-ary linear $[n_i, n_i - r_i]_q R$ codes $\mathcal{C}_{n_i}$ with covering radius R and such that the following two properties hold:

(a) the codimension $r_i = Rt_i + \eta$, where $(t_i)_i$ is an increasing sequence of integers;

(b) the length n_i of $\mathcal{C}_i$ coincides with $f_q^{(\eta)}(r_i)$, where $f_q^{(\eta)}$ is an increasing function.

We call $\mathcal{A}_{R,q}^{(\eta)}$ an *infinite family of covering codes* or an *infinite code family,* or simply *infinite family.*

Considering families of type $\mathcal{A}_{R,q}^{(\eta)}$ is a standard method of investigation of *linear* covering codes, see [1]-[5], and the references therein. In particular, it is related to the fact that families with distinct values of η often have distinct properties. Throughout the paper, distinct families $\mathcal{A}_{R,q}^{(\eta)}$ with the same parameters η, R, q will be denoted as follows: $\mathcal{A}_{R,q,1}^{(\eta)}$, $\mathcal{A}_{R,q,2}^{(\eta)}$, and so on.

For an infinite code family $\mathcal{A}_{R,q}^{(\eta)}$, its *asymptotic covering density* is defined as follows:

$$\overline{\mu}_q(R, \mathcal{A}_{R,q}^{(\eta)}) = \liminf_{i\to\infty} \mu_q(n_i, R, \mathcal{C}_{n_i}). \tag{2}$$

The size q of the base field F_q is fixed for a given family, but, when an infinite set of families is considered, the value of q can infinitely grow. A central problem for covering codes is the following: for fixed R and η *find a set of sequences* $\mathcal{A}_{R,q}^{(\eta)}$ *of* q*-ary codes with* q *running over an infinite set of prime power, such that the asymptotic covering density of every sequence is bounded from above by a constant independent of* q. Each sequence of such a set is said to be *good*. Accordingly, an $[n, n-r]_q R$ covering code is called *good* or *short* if $n = O(q^{\frac{r-R}{R}})$. By (1) and (2), a sequence $\mathcal{A}_{R,q}^{(\eta)}$ consisting of good codes is good. So far, the problem has been solved only for $\eta = 0$ and arbitrary R and q, for $R = 2, \eta = 1$ and q a square [3, formula (33)], and for $R = 3$, $\eta = 1$ and q a cube [4, p. 540].

The main result of the paper is the construction of good infinite families $\mathcal{A}_{R,q}^{(\eta)}$ for arbitrary R and all $\eta = 0, 1, 2, \ldots, R-1$, under the condition $q = (q')^R$. A key tool in our investigation is the connection between linear covering codes and *saturating sets* in projective spaces over finite fields.

Let $PG(v, q)$ be the v-dimensional projective space over F_q. We say that a set of points $S \subseteq PG(v, q)$ is ϱ*-saturating* if for any point $x \in PG(v, q)$ there exist $\varrho + 1$ points in S generating a subspace of $PG(v, q)$ containing x, and ϱ is the smallest value with such property [2, Definition 1.1], [6]. In the literature *saturating sets* are also called *saturated sets* [2],[3], *spanning sets*, and *dense sets*.

Points of an $(R-1)$-saturating set K of size n in $PG(r-1, q)$ can be viewed as columns of a *parity check matrix* of an $[n, n-r]_q R$ *related* covering code $\mathcal{C}_K$ [2]-[6]. A saturating set K will be said to be *small* if the related covering code $\mathcal{C}_K$ if short.

A basic tool to obtain an infinite family of codes with good covering properties from a covering code are the so-called q^m-concatenating constructions [1, Section 5.4]-[5].

The good infinite families of covering codes provided in this paper are obtained by applying the q^m-concatenating constructions to covering codes related to new small saturating sets. The construction of such sets relies on a new notion of N-fold *strong* blocking set.

The *length function* $\ell_q(r, R)$ is the smallest length of a q -ary linear code with codimension r and covering radius R [1]. Existence of an $[n, n-r]_q R$ code or, equivalently, of an $(R-1)$ -saturating n-set in $PG(r-1, q)$, implies the upper bounds $\ell_q(r, R) \leq n$. Denote by $k_q(v, \varrho)$ the smallest possible size of a ϱ-saturating set in the space $PG(v, q)$. Clearly, $\ell_q(r, R) = k_q(r-1, R-1)$.

The small saturating sets and the infinite code families obtained in this paper provide an improvement on the previously known upper bounds on the length function $\ell_q(r, R)$, and on the corresponding value of $k_q(v, \varrho)$.

2 Infinite families $\mathcal{A}_{R,q}^{(0)}$ of $[n, n-Rt]_q R$ codes

The best known families $\mathcal{A}_{2,q}^{(0)}$ and $\mathcal{A}_{3,q}^{(0)}$ are given in [5]. By using them in the direct sum construction [1], we obtain an infinite family $\mathcal{A}_{R,q}^{(0)}$ of $[n, n-r]_q R$ codes with parameters

$$\mathcal{A}_{R,q}^{(0)} \; : \; R \geq 4, \; r = Rt \geq 5R, \; q \geq 7, \; q \neq 9, \; n = Rq^{\frac{r-R}{R}} + \left\lceil \frac{R}{3} \right\rceil q^{\frac{r-2R}{R}}, \; r \neq 6R.$$

The main term of the asymptotic density $\overline{\mu}_q(R, \mathcal{A}_{R,q}^{(0)})$ is $\frac{R^R}{R!}$ and it does not depend of q.

The codes of the family $\mathcal{A}_{R,q}^{(0)}$ are shorter than those of the family arising from the direct sum of the $[\frac{q^m-1}{q-1}, \frac{q^m-1}{q-1} - m]_q 1$ perfect Hamming codes, see, e.g., [2, formula (5)].

3 Small ρ-saturating sets in the spaces $PG(\rho+1, q)$

We introduce a new concept of N-fold *strong* blocking set.

Definition 3.1 *A subset B of a projective space $PG(N, q)$ is an N-*fold strong blocking set* if every hyperplane of $PG(N, q)$ is spanned by N points in B.*

Theorem 3.2 *Let $q = (q')^{\rho+1}$. Any $(\rho+1)$-fold strong blocking set in a subspace*
$PG(\rho+1, q') \subset PG(\rho+1, q)$ is a ρ-saturating set in the space $PG(\rho+1, q)$.

Theorem 3.3 *Let $q = (q')^4$. In $PG(2, q)$ there is a 1 -saturating set of size $2\sqrt{q} + 2\sqrt[4]{q} + 2$.*

Theorem 3.4 *Let $q = (q')^6$, q' prime, $q' \le 73$. In $PG(2, q)$ there is a 1-saturating set of size $2\sqrt{q} + 2\sqrt[3]{q} + 2\sqrt[6]{q} + 2$.*

Let x_0, x_1, x_2, x_3 be homogenous coordinates for the points in $PG(3, q)$ and let l_1, l_2, l_3 be lines in $PG(3, q)$ with equations $l_1 : x_0 = x_2 = 0$; $l_2 : x_1 = x_3 = 0$; $l_3 : x_0 = x_3$, $x_1 = x_2$. The lines are contained in the hyperbolic quadric $\mathcal{Q}$: $x_0x_1 = x_2x_3$. Let g be any line disjoint from $\mathcal{Q}$. We denote $B = l_1 \cup l_2 \cup l_3 \cup g$. The following can be proved.

Theorem 3.5 *The set B of size $4q + 4$, is a 3-fold strong blocking set in $PG(3, q)$.*

The following result shows that N-fold strong blocking sets can be obtained by an *inductive construction* . Each inductive steps consists of embedding the blocking set in a higher dimensional space, and then adding the union of some properly chosen lines.

Theorem 3.6 *Assume that there exists an N-fold strong blocking set in $PG(N, q)$ of size k. Then there exists an $(N+1)$-fold strong blocking set in $PG(N+1, q)$ of size*
$k + 1 + (N+1)(q-1)$.

Corollary 3.7 *In $PG(N, q)$, $N \ge 3$, there exists an N-fold strong blocking set of size*

$$(q-1)\left(\frac{N(N+1)}{2} - 2\right) + N + 5.$$

Corollary 3.8 *Let $q = (q')^{\rho+1}$, $\rho > 1$. Then there exists a ρ-saturating set in $PG(\rho+1, q)$ of size*

$$(\sqrt[\rho+1]{q} - 1)\left(\frac{(\rho+1)(\rho+2)}{2} - 2\right) + \rho + 6.$$

4 Infinite families $\mathcal{A}^{(1)}_{R,q}$ of $[n, n-(Rt+1)]_q R$ codes

We use ρ-saturating sets in the spaces $PG(\rho+1, q)$, obtained in the previous section, as starting points for the q^m-concatenating constructions of [2]-[5]. To this end, it is useful that the set B described in Section 3 and the ρ-saturating set of Corollary 3.8 consist of lines.

Theorem 4.1 *There exist infinite families $\mathcal{A}^{(1)}_{R,q}$ of $[n, n-r]_q R$ codes with the following parameters:*

$$\mathcal{A}^{(1)}_{2,q,1} \ : \ R = 2,\ r = 2t+1 \geq 3,\ q = (q')^4,\ n = 2(\sqrt{q} + \sqrt[4]{q} + 1)q^{\frac{r-3}{2}} + \left\lfloor q^{\frac{r-5}{2}} \right\rfloor,$$
$$\overline{\mu}_q(2, \mathcal{A}^{(1)}_{2,q,1}) \approx 2 + \frac{4}{\sqrt[4]{q}} + \frac{6}{\sqrt{q}} + \frac{4}{\sqrt[4]{q^3}} - \frac{4}{q}.$$

$$\mathcal{A}^{(1)}_{2,q,2} \ : \ R = 2,\ r = 2t+1 \geq 3,\ q = (q')^6,\ q' \text{ prime},\ q' \leq 73,\ r \neq 9, 13,$$
$$n = 2(\sqrt{q} + \sqrt[3]{q} + \sqrt[6]{q} + 1)q^{\frac{r-3}{2}} + 2\lfloor q^{\frac{r-5}{2}} \rfloor.$$

$$\mathcal{A}^{(1)}_{3,q} \ : \ R = 3,\ r = 3t+1 \geq 7,\ q = (q')^3 \geq 64,\ n = 4(\sqrt[3]{q} + 1)q^{\frac{r-4}{3}},$$
$$\overline{\mu}_q(3, \mathcal{A}^{(1)}_{3,q}) \approx \frac{32}{3} - \frac{96}{\sqrt[3]{q}} + \frac{96}{\sqrt[3]{q^2}} - \frac{64}{3q}.$$

$$\mathcal{A}^{(1)}_{R,q} \ : \ R \geq 4,\ r = Rt+1,\ q = (q')^R,\ n = n_{R,q} q^{\frac{r-(R+1)}{R}} + (R-3)\frac{q^{\frac{r-(R+1)}{R}} - 1}{q-1},$$
$$n_{R,q} = (\sqrt[R]{q} - 1)\left(\frac{R(R+1)}{2} - 2\right) + R + 5,\ t = 1 \text{ and } t \geq t_0,\ q^{t_0 - 1} \geq n_{R,q}.$$

The main term of the asymptotic density $\overline{\mu}_q(R, \mathcal{A}^{(1)}_{R,q})$ is $\frac{(R^2+R)^R}{2^R R!}$. Significantly, it does not depend on q.

5 Infinite families $\mathcal{A}^{(\eta)}_{R,q}$ of $[n, n-(Rt+\eta)]_q R$ codes, $\eta = 2, 3, \ldots, R-1$

We construct small ρ-saturating sets in $PG(N, (q')^{\rho+1})$, $N = \rho+2, \rho+3, \ldots, 2\rho-1$.

Lemma 5.1 *Fix $1 \leq k < N$. Let B_k be the subset of $PG(N, q)$ consisting of points whose weight is at most $N-k+1$, i.e. B_k is the union of the $(N-k)$-dimensional subspaces of equation $x_{i_1} = \ldots = x_{i_k} = 0$. Then every k-dimensional subspace of $PG(N, q)$ is generated by $k+1$ independent points in B_k.*

Theorem 5.2 *Let ρ be any positive integer. Let $q = (q')^{\rho+1}$. Let $N > \rho+1$. Then in $PG(N, q)$ there exists a ρ-saturating set of size*

$$\frac{V_{q'}(N+1, N-\rho+1) - 1}{q'-1} \sim \binom{N+1}{\rho} q^{\frac{N-\rho}{\rho+1}}, \text{ where } V_{q'}(a, b) = \sum_{i=0}^{b} (q'-1)^i \binom{a}{i}.$$

For a parameter $\eta \in \{2, 3, \ldots, \rho\}$ we take $N = \rho + \eta$. Then the length of the $[\overline{n}_{R,q,\eta}, \overline{n}_{R,q,\eta} - (R+\eta)]_q R$ code related to the ρ-saturating set of Theorem 5.2 is equal to

$$\overline{n}_{R,q,\eta} = \frac{\left(\sum_{i=0}^{\eta+1} (\sqrt[R]{q}-1)^i \binom{R+\eta}{i}\right) - 1}{\sqrt[R]{q}-1} \sim \binom{R+\eta}{R-1} q^{\frac{\eta}{R}}.$$

The code is an (R, ℓ)-object with $\ell \geq 3$, see [2, Section II] for definitions of (R, ℓ)-objects and (R, ℓ)-partitions. We use it as the starting code of the q^m-concatenating constructions of [2, Th. 3.1, Condition A2] with the trivial (R, ℓ) -partition.

Theorem 5.3 *Let $q = (q')^R$ and let $R \geq 4$. We fix the parameter $\eta \in \{2, 3, \ldots, R-1\}$. Then there is an infinite family $\mathcal{A}_{R,q}^{(\eta)}$ of $[n, n-r]_q R$ codes with the following parameters*

$$\mathcal{A}_{R,q}^{(\eta)} : R \geq 4,\ r = Rt + \eta.\ q = (q')^R,\ n = \overline{n}_{R,q,\eta} q^{\frac{r-(R+\eta)}{R}} + (R-3)\frac{q^{\frac{r-(R+\eta)}{R}} - 1}{q-1},$$

$$t = 1 \text{ and } t \geq t_0,\ q^{t_0 - 1} \geq \overline{n}_{R,q,\eta}.$$

The main term of the asymptotic covering density $\overline{\mu}_q(R, \mathcal{A}_{R,q}^{(\eta)})$ is $\frac{(R+\eta)^{R^2-R}}{((R-1)!)^R R!}$, which does not depend of q.

References

[1] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering Codes*, Amsterdam, The Netherlands: North-Holland, 1997.

[2] A. A. Davydov, Constructions and families of covering codes and saturated sets of points in projective geometry, *IEEE Trans. Inform. Theory* 41, 1995, 2071-2080.

[3] A. A. Davydov, Constructions and families of nonbinary linear codes with covering radius 2, *IEEE Trans. Inform. Theory* 45, 1999, 1679-1686.

[4] A. A. Davydov, S. Marcugini, F. Pambianco, Linear codes with covering radius 2, 3 and saturating sets in projective geometry, *IEEE Trans. Inform. Theory* 50, 2004, 537-541.

[5] A. A. Davydov, P. R. J. Östergård, Linear codes with covering radius $R =$ 2, 3 and codimension tR, *IEEE Trans. Inform. Theory* 47, 2001, 416-421.

[6] A. A. Davydov, P. R. J. Östergård, On saturating sets in small projective geometries, *Europ. J. Combin.* 21, 2000 563-570.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 76-81*

On the properness of some optimal binary linear codes and their dual codes

Rossitza Dodunekova[1] rossitza@math.chalmers.se
Chalmers University of Technology and the University of Gothenburg
412 96 Gothenburg, SWEDEN

S. M.Xiaolei Hu
Chalmers University of Technology, 412 96 Gothenburg, SWEDEN

1 Introduction

A linear code is said to be proper in error detection over a symmetric memoryless channel if its undetected error probability is an increasing function of the channel symbol error probability. A proper code performs well in error detection in the sense that the better the channel, the better the performance, which makes the code appropriate for use in channels where the symbol error probability is not known exactly.

A q-ary linear code may be optimal in different ways. Of most interest are codes whose parameters are in some sense extremal. For example, Maximum Distance Separable (MDS) codes are distance-optimal among the q-ary linear codes of the same length and dimension. Codes may be also length-optimal and size-optimal.

Studies have shown that many linear codes which are optimal in some sense, or close to optimal, are also proper, and most often their dual codes are proper, too. For example, proper are the Perfect codes over finite fields, MDS codes and some Near MDS codes, many Griesmer codes, and Maximum Minimum Distance codes and their duals. Could it be the case that properness and optimality are closely related? What kind of relation would this be?

It is most natural to start the study of these questions by looking for optimal codes which are not proper. In this work we present some preliminary results in this direction. We have studied some binary linear codes of optimal length which cannot be obtained by shortening or puncturing other binary linear codes. The codes turn out to be proper, together with their dual codes. Moreover, like most of the codes listed above, these binary codes satisfy certain conditions that imply properness. These conditions are expressed in terms of the so called extended binomial moments, which are just linear combinations of the elements of the weight distribution of the codes. One interesting observation based on

[1] Partially supported by Research Platform MP2 of the University of Gothenburg.

computer graphs is that the extended binomial moments of these binary proper codes are rather close to a certain general lower bound.

2 Preliminaries

Error detection with linear codes. Let C be a linear $[n,k,d]_q$ code over the finite field $GF(q)$ of q elements, i.e., a k-dimensional subspace of the n-dimensional vector space $GF(q)^n$ over $GF(q)$, with minimum Hamming weight d. Suppose C is used to detect transmission errors on a q-ary discrete memoryless channel. In such a channel, any symbol transmitted has a probability $1-\varepsilon$ of being received correctly and a probability $\frac{\varepsilon}{q-1}$ of being transformed into each of the $q-1$ other symbols. Naturally, it should be more likely for a symbol to remain unchanged during the transmission than to change into another symbol, which leads to the restriction $0 \le \varepsilon \le \frac{q-1}{q}$.

Let $x \in C$ be the code word transmitted and $y \in GF(q)^n$ be the vector received. In error detection, when y is not a codeword the decoder makes the correct decision that a transmission error has occurred, and asks for a retransmission. When y is a codeword, the decoder decides that y was sent. Such a decision is of course incorrect when y and x are different, thus a transmission error for which the error vector $y-x$ is a non-zero codeword remains undetected. The probability $P_{ue}(C,\varepsilon)$ that an undetected error occurs depends on ε, the basic parameters n, k, d, and q of C, and its weight distribution $\{A_i,\ 0 \le i \le n\}$, where A_i is the number of code words in C with weight i. The formula is given by [7]

$$P_{ue}(C,\varepsilon) = \sum_{i=1}^{n} A_i\Big(\frac{\varepsilon}{q-1}\Big)^i (1-\varepsilon)^{n-i}, \quad 0 \le \varepsilon \le \frac{q-1}{q}. \tag{2.1}$$

Proper error detecting codes. In error detection over a particular channel, codes with the smallest probability of undetected error would be the best. However, in order to find such a code, one has to use exhaustive search since presently we don't have any efficient general method for such a search. But even if we would have such a method, this would not solve the problem, since most often ε is not known exactly, and a best code for some ε' may be very inappropriate for the channel, even if its symbol error probability is close to ε'. For this reason the concept of a proper code has been introduced [8, 6, 7].

A linear code is *proper*, if its undetected error probability is an increasing function of ε. Thus the error detecting performance of a proper code is better on better channels, i.e., channels with smaller symbol error probability, which makes the code appropriate for channels where ε is not known exactly.

Another view to properness is gained by comparing the function $P_{ue}(C,\varepsilon)$ of a proper $[n,k,d]_q$ code C to the function $P_{ue}(\varepsilon)$ obtained by averaging the undetected error probability in some set of $[n,k]_q$ codes. In the set of systematic $[n,k]_q$ codes, the averaging procedure gives [10, 11]

$$P_{ue}(\varepsilon) = q^{-(n-k)}[1-(1-\varepsilon)^k],$$

which is an increasing function of ε. Also in the set of binary $[n,k]$ codes the average undetected error probability is an increasing function [8]:

$$P_{ue}(\varepsilon) = \frac{2^k-1}{2^n-1}[1-(1-\varepsilon)^n].$$

Hence a hypothetical "average" code in the class would be proper. In this sense a proper code is similar to an "average" code, which makes the code a reasonable choice in situations where we cannot do better.

Codes, which are optimal or close to optimal in some sense, are prevailing in the list of proper codes [4]. The question we want to address is if properness and optimality are closely related. As a first step, we have studied some length-optimal binary codes from [1].

Discrete sufficient conditions for properness. Let C be an $[n,k,d]_q$ linear code with weight distribution $\{A_0, A_1, \ldots, A_n\}$. The *extended binomial moments* A_ℓ^* of C are defined as [2]

$$A_\ell^* = \sum_{i=d}^{\ell} \frac{\ell(\ell-1)\ldots(\ell-i+1)}{n(n-1)\ldots(n-i+1)} A_i, \quad d \le \ell \le n, \qquad (2.2)$$
$$A_0^* = 0, \quad 0 \le \ell \le d-1.$$

Let B_ℓ^* be the extended binomial moments of the dual code. It holds [2]

$$B_\ell^* + 1 = q^{\ell-k}(A_{n-\ell}^* + 1), \quad \ell = 0, \ldots, n. \qquad (2.3)$$

Denote by $d^\perp$ the minimum Hamming distance of the dual code. The following results have been derived in [3, 5, 2].

Theorem 1 *If*

$$A_\ell^* \ge q A_{\ell-1}^*, \quad \ell = d+1, \ldots n - d^\perp + 1, \qquad (2.4)$$

then C is proper.

Theorem 2 *Suppose C is a binary code. If*

$$d \ge \left\lceil \frac{n}{2} \right\rceil$$

or

$$\left\lceil \frac{n}{3} \right\rceil + 1 \le d^\perp \le \left\lfloor \frac{n}{2} \right\rfloor \quad \text{and} \quad n(n+1-2d^\perp) \le d(n-d^\perp),$$

then C is proper.

Theorem 3 *The extended binomial moments satisfy*

$$\max\{0,\, q^{\ell-n+k}-1\} < A_\ell^* < q^{\min(\ell+1-d,\ k+1-d^\perp)}-1,\quad \ell = d,\ldots,n-d^\perp$$

$$A_\ell^* = q^{\ell-n+k}-1,\quad \ell = n-d^\perp+1,\ldots,n.$$

Linear binary codes of dimension at most 7. Following [1], we say that an $[n,k,d]$ code is *distance-optimal* if no $[n,k,d-1]$ code exists; it is *length-optimal* (which is a stronger condition) if no $[n-1,k,d]$ code exists, and *optimal*, if no $[n+1,k+1,d]$ or $[n+1,k,d+1]$ code exists. An optimal code cannot be obtained by shortening or puncturing other binary linear codes.

Summary of optimal binary codes with $k \le 7$, $n \le 2^k$ [1].

$[n,k,d]$	# codes	$[n,k,d]$	# codes	$[n,k,d]$	# codes
	$(form.equiv.)$		$(form.equiv.)$		$(form.equiv.)$
$[8,4,4]$	1	$[12,4,6]$	1	$[16,5,8]$	1
$[21,5,10]^*$	2	$[24,5,12]$	1	$[28,5,14]$	1
$[32,6,16]$	1	$[38,6,18]$	1	$[45,6,22]$	1
$[48,6,24]$	1	$[53,6,26]$	2	$[56,6,28]$	1
$[60,6,30]$	1	$[24,7,10]^*$	6(5)	$[27,7,12]$	1
$[40,7,18]$	172(46)	$[43,7,20]$	7(3)	$[56,7,26]^*$	> 19000
$[59,7,28]$	143(38)	$[64,7,32]$	1	$[71,7,34]$	1
$[75,7,36]^*$	3603	$[79,7,38]$	216(22)	$[82,7,40]$	11(7)
$[87,7,42]$	55(36)	$[90,7,44]$	6(6)	$[93,7,46]$	1
$[96,7,48]$	1	$[102,7,50]^*$	3	$[105,7,52]$	1
$[109,7,54]$	1	$[112,7,56]$	1	$[117,7,58]$	2
$[120,7,60]$	1	$[124,7,62]$	1		

Recall that two codes are formally equivalent if they have the same basic parameters and weight distribution. Clearly, the undetected error probability function (2.1) of such codes is the same. In the above table, the even columns show the number of non-isomorphic codes with the given basic parameters and, in parentheses, the number of classes of formal equivalence.

3 The result

Theorem 4 *All codes in the above table and their duals are proper, except those marked by an asterisk.*

The proof is based on theorems 1 and 2 above. We have used Matlab for computing the extended binomial moments of the codes and their duals, given in (2.2) and (2.3), and for checking the conditions of the theorems. Information about weight enumerators and dual code distances has been taken from [1] and also from the Internet based data bases http://www.codetables.de/ and www.math.unl.edu/∼djaffe2/codes/webcodes/binary/codes.cgi?n=28&k=5.

The codes $[8,4,4], [12,4,6], [16,5,81], [24,5,12], [28,5,14], [60,6,30]$, and $[56,6,28]$ have minimum distance $n/2$ and are proper by the first part of Theorem 2. The dual codes have minimum distance at least 3, and are proper by the second part of the theorem. In fact the codes achieve the Grisemer bound. It has been noticed earlier [5] that Theorem 2 is quite efficient for the study of such codes.

We end by noting the following. The extended binomial moments have shown to be a useful tool in the study of the undetected error probability function. We plotted the extended binomial moments of the above codes together with their bounds from Theorem 3. It turns out that the extended binomial moments of these optimal proper codes almost lie on the lower bound.

References

[1] I. Bouykliev, D. B. Jaffe, Optimal binary linear codes of dimension at most seven, *Discr. Math.* 226, 2001, 51-70.

[2] R. Dodunekova, The extended binomial moments of a linear code and the undetected error probability, *Probl. Pered. Inform.* 39, 2003, 28-39, English transl. *Probl. Inform. Transm.* 39, 2003, 255-265.

[3] R. Dodunekova, S. M. Dodunekov, Sufficient conditions for good and proper error detecting codes, *IEEE Trans. Inform. Theory* 43, 1997, 2023-2026.

[4] R. Dodunekova, S. Dodunekov, E. Nikolova, A survey on proper codes. *Disc. Appl. Math.* 156, 2008, 1499-1509.

[5] R. Dodunekova, E. Nikolova, Sufficient conditions for the monotonicity of the undetected error probability for large channel error probabilities. *Probl. Pered. Inform.* 41, 2005, 3-16.

[6] T. Kasami, S. Lin, On the probability of undetected error for the maximum distance separable codes. *IEEE Trans. Commun.* 32, 1984, 998-1006.

[7] T. Kløve, V. Korzhik, *Error detecting codes, General Theory and their Application in Feedback Communication Systems.* Kluwer, Boston, MA 1995.

[8] S. K. Leung-Yan-Cheong, E. R. Barnes, D. U. Friedman, On some properties of the undetected error probability of linear codes. *IEEE Trans. Inform. Theory* 25, 1979, 110-112.

[9] S. K. Leung-Yan-Cheong, M. E. Hellman, Concerning a bound on undetected error probability. *IEEE Trans. Inform. Theory*, 22, 1976, 235-237.

[10] J. Massey, Coding techniques for digital data networks. In *Proc. Int. Conf. Inform. Theory Syst.*, NTG-Fachberichte, Berlin, Germany, 65, 1978.

[11] J. K. Wolf, A. M. Michelson, A. H. Levesque, On the probability of undetected error for linear block codes. *IEEE Trans. Commun.* 30, 1982, 317-324.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 82-85*

On complexity of decoding Reed-Muller codes within their code distance

Ilya Dumer dumer@ee.ucr.edu
University of California, Riverside, CA, USA
Grigory Kabatiansky kaba@iitp.ru
Institute for Information Transmission Problems, Moscow, RUSSIA
Cédric Tavernier cedric.tavernier@c-s.fr
Communications and Systems Le Plessis Robinson, FRANCE

Abstract. Recently Gopalan, Klivans, and Zuckerman proved that any binary Reed-Muller (RM) code RM(s,m) can be list-decoded up to its minimum distance d with a polynomial complexity of order n^3 in blocklength n. The GKZ algorithm employs a new upper bound that is substantially tighter for RM codes of fixed order s than the universal Johnson bound, and yields a constant number of codewords in a sphere of radius less than d. In this note, we modify the GKZ algorithm and show that full list decoding up to the code distance d can be performed with a lower complexity order of at most $n \ln^{s-1} n$. We also show that our former algorithm yields the same complexity order $n \ln^{s-1} n$ if combined with the new GKZ bound on the list size.

1 Introduction

Binary Reed-Muller (RM) codes RM(s,m) of order s have length $n = n(m)$, dimension $k = k(s,m)$, and distance $d = d(s,m)$ as follows

$$n = 2^m, \quad k = \sum_{i=0}^{s} \binom{m}{i}, \quad d = 2^{m-s}.$$

The renowned majority decoding algorithm of [1] provides bounded-distance decoding (BDD) for any code $RM(s,m)$ and corrects all errors of weight less than $d/2$ with complexity order of kn. Even a lower complexity order of $n \min(s, m-s)$ is required for various recursive techniques of [2], [3], and [4]. Both recursive and majority algorithms correct many error patterns beyond the BDD radius $d/2$; however, they fall short of complete error-free decoding within any given decoding radius $T \geq d/2$. Therefore, below we address *list decoding* [5] algorithms that output the list

$$L_T(\mathbf{y}) = \{\mathbf{c} \in \text{RM}(s,m) : d(\mathbf{y},\mathbf{c}) \leq T\}$$

of *all* vectors $\mathbf{c}$ of a code RM(s,m) located within the distance T from any received vector $\mathbf{y}$.

Our study will be based on the recent algorithm obtained in [6] by Gopalan, Klivans, and Zuckerman (GKZ). The GKZ algorithm list-decodes any binary Reed-Muller (RM) code RM(s,m) up to its minimum distance d with a polynomial complexity of order n^3 in blocklength n. Another important advance is a new upper bound on the list size that is substantially tighter than the universal Johnson bound for codes RM(s,m), and yields a constant number of RM-codewords in any sphere of radius less than d. More precisely, let

$$\delta_s = \frac{d(s,m)}{n(m)} = 2^{-s},\ T(s,m,\epsilon) = n(\delta_s - \epsilon)$$

be the relative distance of RM(s,m) and the decoding radius of interest. Here we take any $\epsilon \in (0,\delta_s)$. Also, let $\chi(s,m,\epsilon)$ be the maximum number of binary operations required by GKZ algorithm to design the list $L_T(\mathbf{y})$ and let

$$l(s,m,\epsilon) = \max_{\mathbf{y}} |L_T(\mathbf{y})| \tag{1}$$

be the largest possible number of codewords in a sphere of radius $T(s,m,\epsilon)$. We will use the new upper bound

$$l(s,m,\epsilon) \le 2(2^{s+5}\epsilon^{-2})^{4s} \tag{2}$$

discovered in [6]. This bound also leads to a new list decoding algorithm [6] that outputs the list $L_T(\mathbf{y})$ with complexity

$$\chi(s,m,\epsilon) = O(n^3 l^s(s,m,\epsilon)) = O(\epsilon^{-8s^2} n^3)$$

In the following, we simplify the GKZ algorithm and prove

Theorem 1 *For any received vector* $\mathbf{y}$, *RM codes RM*(s,m) *can be list-decoded within the decoding radius* $(2^{-s}-\epsilon)n$ *with complexity*

$$\chi^{(1)}(s,m,\epsilon) = O(\epsilon^{-18} n \ln^{s-1} n) + O(\epsilon^{8-16s} n \ln n) \tag{3}$$

Also, consider our former recursive algorithm [7] that has the same complexity order $n \ln^{s-1} n$ *in blocklength* n *but was used in [7] to decode within the Johnson bound. In fact, this algorithm is restricted only by the correspomding list size. Namely, it is shown in [7] that complexity* $\chi^{(2)}(s,m,\epsilon)$ *of the algorithm* $\Psi_{s,m,\epsilon}$ *satisfies recursion*

$$\chi^{(2)}(s,m,\epsilon) \le m(\chi^{(2)}(s-1,m-1,\epsilon) + cn\epsilon^{-1} l(s,m,\epsilon/2) l(s-1,m-1,\epsilon)) \tag{4}$$

Thus, we can now extend the decoding radius to code distance d using the GKZ bound (2). As initial step of our recursion (4), we can also use the list decoding algorithm [8] of RM$(1,m)$ *codes, which has linear complexity* $O(n \ln^2(\epsilon^{-1}))$

within radius $T(1.m,\epsilon)$. This combination of estimates (2) and (4) shows that the former algorithm $\Psi_{s,m,\epsilon}$ decodes within the radius $(2^{-s}-\epsilon)n$ with complexity

$$\chi^{(2)}(s,m,\epsilon)=O(\chi^{(1)}\epsilon^{-1})$$

In the next section, we briefly outline a modification of the GKZ algorithm that gives Theorem 1.

2 Error-free list decoding of RM codes

We shall use the well known Plotkin construction of RM-codes [9] which represents any codeword $\mathbf{f}\in \mathrm{RM}(s,m)$ as the vector $\mathbf{u},\mathbf{u}+\mathbf{v}$, where $\mathbf{u}\in \mathrm{RM}(s,m-1)$ and $\mathbf{v}\in \mathrm{RM}(s-1,m-1)$. Let a received vector $\mathbf{y}$ be decomposed into two halves $\mathbf{y}'$ and $\mathbf{y}''$, which can be considered as the corrupted versions of some vectors $\mathbf{u}$ and $\mathbf{u}+\mathbf{v}$ correspondingly.

Algorithm. Given ϵ and any received vector $\mathbf{y}$, we consider below an algorithm $\Phi(s,m,\epsilon)$ that decodes $\mathbf{y}$ into the list $L_T(\mathbf{y})$ within the radius $T(s,m,\epsilon)=n(\delta_s-\epsilon)$.

Step 1. Decode the vector $\mathbf{y}^v=\mathbf{y}'+\mathbf{y}''$ within the radius $T(s,m,\epsilon)=T(s-1,m-1,2\epsilon)$, using the algorithm $\Phi(s-1,m-1,2\epsilon)$. The resulting list of codewords L^v belongs to $\mathrm{RM}(s-1,m-1)$.

Step 2. Decode both vectors $\mathbf{y}'$ and $\mathbf{y}''$ within the radius $T(s,m,\epsilon)/2=T(s,m-1,\epsilon)$ using the algorithm $\Phi(s,m-1,\epsilon)$. The resulting lists of codewords $L'$and L'' belong to $\mathrm{RM}(s,m-1)$.

3. Consider the two lists of vectors

$$A=\{(\mathbf{u}'.\mathbf{u}'+\mathbf{v}) : \mathbf{u}'\in L',\mathbf{v}\in L^v\}$$
$$B=\{(\mathbf{u}''+\mathbf{v},\mathbf{u}'') : \mathbf{u}''\in L'',\mathbf{v}\in L^v\}$$

Calculate the distance from $\mathbf{y}$ to each vector of the two lists. Leave the vectors located within distance $T(s,m,\epsilon)$.

The above algorithm gives complete list $L_{T(s,m,\epsilon)}(\mathbf{y})$ and thus performs the required decoding. This is due to the following:

1. Vector $\mathbf{y}^v$ has no more errors than $\mathbf{y}$;

2. Either $\mathbf{y}'$ or $\mathbf{y}''$ has at most $T(s,m,\epsilon)/2$ errors.

Complexity. Algorithm $\Phi(s,m,\epsilon)$ includes one decoding $\Phi(s-1,m-1,2\epsilon)$, two decodings $\Phi(s,m-1,\epsilon)$ plus requires the order of $2nl(s,m-1,\epsilon)l(s-1,m-1,2\epsilon)$ operations to verify the distance from vector of lists A and B to the vector $\mathbf{y}$. Thus, algorithm $\Phi(s.m,\epsilon)$ has complexity

$$\begin{aligned}\chi(s.m.\epsilon)\leq{}&\chi(s-1,m-1,2\epsilon)+2\chi(s,m-1,\epsilon)\\&+2nl(s,m-1,\epsilon)l(s-1,m-1,2\epsilon).\end{aligned}\qquad(5)$$

Now we proceed, for $s = 2, 3, ..$ using complexity $\chi(1, m, \epsilon) = 2^m \ln^2 \epsilon^{-1}$ in step $s = 1$, the Johnson bound $l(1, m, \epsilon) \leq (2\epsilon)^{-2}$ for $RM-1$ codes and the upper bound (2) for $s > 1$. Then

$$\chi(2, m, \epsilon) = O(m2^m \left[\ln^2 \epsilon^{-1} + \epsilon^{-18}\right]) = O(m2^m \epsilon^{-18})$$

and for any $s > 2$ we obtain the estimate

$$\chi(s, m, \epsilon) = O(m^{s-1} 2^m \epsilon^{-18}) + \sum_{i=3}^{s} O\left(m^{s-i+1} 2^m \epsilon^{8-16i}\right)$$

$$= O(\epsilon^{-18} n \ln^{s-1} n) + O(\epsilon^{8-16s} n \ln n)$$

which proves Theorem 1.

References

[1] I. S. Reed, A class of multiple error correcting codes and the decoding scheme, *IEEE Trans. Inform. Theory* 4, 1954, 38-49.

[2] S. Litsyn, On complexity of decoding low rate Reed-Muller codes, *Proc. 9th All Union Conf. Coding Theory Inform. Transm.* 1, 1988, 202-204 (in Russian).

[3] G. A. Kabatianskii, On decoding of Reed-Muller codes in semicontinuous channels, *Proc. Second Intern. Workshop ACCT*, Leningrad, USSR, 1990, 87-91.

[4] I. Dumer, Recursive decoding and its performance for low-rate Reed-Muller codes, *IEEE Trans. Inform. Theory* 50, 2004, 811-823.

[5] P. Elias, List decoding for noisy channels, *1957-IRE WESCON Conven. Record* 2, 1957, 94-104.

[6] P. Gopalan, A. R. Klivans, D. Zuckerman, List-decoding Reed-Muller codes over small fields, STOC 2008.

[7] I. Dumer, G. Kabatiansky, C. Tavernier, List decoding of Reed-Muller codes up to the Johnson bound with almost linear complexity, *IEEE Symp. Inform. Theory*, 2006, Seattle, WA, USA, 138-142.

[8] I. Dumer, G. Kabatiansky, C. Tavernier, List decoding for binary Reed-Muller codes of the first order, *Probl. Inform. Transm.* 43, 3, 66-74, 2007.

[9] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1981.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 86-91*

DNA codes based on stem Hamming similarity

A.G. D'yachkov dyachkov@mech.math.msu.su
A.N. Voronina vorronina@gmail.com
Department of Probability Theory, Faculty of Mechanics and Mathematics,
Moscow State University, Moscow 119992, RUSSIA

Abstract. For q-ary n-sequences, we continue the development [1, 2] of similarity functions that can be used (for $q = 4$) to model a thermodynamic similarity of DNA sequences. Codes based on similarity functions are called DNA codes [1]. In this paper, we discuss a biologically motivated [2] additive similarity function called a *stem Hamming similarity* and defined as the total number of common 2-blocks containing adjacent symbols in the longest common Hamming subsequence between two q-ary n-sequences. Conventional lower and upper bounds called the Gilbert-Varshamov, Plotkin and Elias bounds [3] on the rate of corresponding DNA codes are obtained.

1 Notations and definitions

Symbol $\triangleq$ denotes definitional equalities and symbol $[n] \triangleq \{1, 2, \ldots, n\}$ denotes the set of integers from 1 to n. Let $q = 2, 4, \ldots$ be an arbitrary even integer, $\mathcal{A} \triangleq \{0, 1, \ldots, q-1\}$ is the standard alphabet of size $|\mathcal{A}| = q$ and $\lfloor u \rfloor$ ($\lceil u \rceil$) denotes the largest (smallest) integer $\leq u$ ($\geq u$).

For any letter $x \in \mathcal{A}$, we define $\bar{x} \triangleq (q-1) - x \in \mathcal{A}$, which is called a *complement* of the letter x. For any q-ary n-sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_{n-1}, x_n) \in \mathcal{A}^n$, we define its *reverse complement* $\widetilde{\boldsymbol{x}} \triangleq (\bar{x}_n, \bar{x}_{n-1}, \ldots, \bar{x}_2, \bar{x}_1) \in \mathcal{A}^n$. If $\boldsymbol{y} \triangleq \widetilde{\boldsymbol{x}}$, then $\boldsymbol{x} = \widetilde{\boldsymbol{y}}$ for any $\boldsymbol{x} \in \mathcal{A}^n$.

Consider two arbitrary q-ary n-sequences

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{A}^n \quad \text{and} \quad \boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in \mathcal{A}^n.$$

The number

$$H_{st}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{i=1}^{n-1} s_i(\boldsymbol{x}, \boldsymbol{y}), \quad \text{where}$$

$$s_i(\boldsymbol{x}, \boldsymbol{y}) \triangleq \begin{cases} 1 & \text{if } x_i = y_i,\ x_{i+1} = y_{i+1}, \\ 0 & \text{otherwise}, \end{cases} \quad i = 1, 2 \ldots, n-1, \tag{1}$$

is called a *stem Hamming similarity* between $\boldsymbol{x}$ and $\boldsymbol{y}$. Evidently, $H_{st}(\boldsymbol{x}, \boldsymbol{y})$ can be defined as the total number of common 2-blocks containing adjacent symbols in the longest common Hamming subsequence between sequences $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{A}^n$.

In addition, $0 \le H_{st}(\boldsymbol{x}, \boldsymbol{y}) \le n-1$ and $H_{st}(\boldsymbol{x}, \boldsymbol{y}) = n-1$ if and only if $\boldsymbol{x} = \boldsymbol{y}$. Therefore, the difference $\mathcal{D}_{st}(\boldsymbol{x}, \boldsymbol{y}) \triangleq (n-1) - H_{st}(\boldsymbol{x}, \boldsymbol{y}) \ge 0$, $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{A}^n$, can be called a *stem Hamming distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Let $\boldsymbol{x}(1), \boldsymbol{x}(2), \dots, \boldsymbol{x}(N)$, where $\boldsymbol{x}(j) \triangleq (x_1(j), \dots, x_n(j)) \in \mathcal{A}^n$, $j \in [N]$, be *codewords* of a q-ary *code* $X = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \dots, \boldsymbol{x}(N)\}$ of *length* n and *size* N, where $N = 2, 4, \dots$ is an *even* number. Let D, $0 < D < n-1$, be an arbitrary number.

A code X is called a *DNA (n, D)-code* [1] based on the stem Hamming similarity if the following two conditions are fulfilled: (i). For any $j \in [N]$, there exists $j' \in [N]$, $j' \ne j$, such that $\boldsymbol{x}(j') = \widetilde{\overline{\boldsymbol{x}(j)}} \ne \boldsymbol{x}(j)$. In other words, X is a collection of $N/2$ pairs of mutually reverse complementary sequences. (ii). For any $j \ne j'$, distance $\mathcal{D}_{st}(\boldsymbol{x}(j), \boldsymbol{x}(j')) \ge D$, i.e., similarity

$$H_{st}\left(\boldsymbol{x}(j), \boldsymbol{x}(j')\right) \le (n-1) - D, \quad j \ne j', \qquad 0 < D < n-1. \tag{2}$$

Let $N_{st}(n, D)$ be the maximal size of DNA (n, D)-codes. If d, $0 < d < 1$, is a fixed number, then

$$R_{st}(d) \triangleq \varlimsup_{n \to \infty} \frac{\log_q N_{st}(n, dn)}{n}, \qquad 0 < d < 1, \tag{3}$$

is called a *rate* of DNA codes based on the stem Hamming similarity.

2 Lower bound on $R_{st}(d)$

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be *independent identically distributed* random sequences having the *uniform* distribution on $\mathcal{A}^n$. Introduce binary random variables

$$\eta_i \triangleq \begin{cases} 0 & \text{if } x_i = y_i,\ x_{i+1} = y_{i+1}, \\ 1 & \text{otherwise, } i = 1, 2 \dots, n-1 \end{cases} \tag{4}$$

and their sum

$$S_n \triangleq \sum_{i=1}^{n-1} \eta_i = (n-1) - H_{st}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{D}_{st}(\boldsymbol{x}, \boldsymbol{y}). \tag{5}$$

Denote by $\overline{\xi}$, the average value of random variable ξ. From definition (4) it follows

$$\overline{\eta_i} = \frac{q^2-1}{q^2}, \quad \overline{S_n} = \sum_{i=1}^{n-1} \eta_i = (n-1)\frac{q^2-1}{q^2}.$$

Let d, $0 < d < \frac{q^2-1}{q^2}$, be a fixed parameter. Introduce function

$$\underline{R}_{st}(d) \triangleq \varlimsup_{n \to \infty} \frac{-\log_q \Pr\{H_{st}(\boldsymbol{x}, \boldsymbol{y}) \ge (1-d)n\}}{n} =$$

$$= \overline{\lim_{n\to\infty}} \frac{-\log_q \Pr\{S_n \le dn\}}{n}. \tag{6}$$

The random coding method for DNA codes [1] leads to

Proposition 1 *If $0 < d \le \frac{q^2-1}{q^2}$, then the rate $R_{st}(d)$ of DNA codes based on stem Hamming similarity satisfies inequality $R_{st}(d) \ge \underline{R}_{st}(d)$, where $\underline{R}_{st}(d)$ is defined by* (6).

Function $\underline{R}_{st}(d)$ is called a *random coding bound* (or the Gilbert-Varshamov bound) on the rate (3) of DNA codes identified by inequality (2).

2.1 Calculation of random coding bound

For the sum (5), introduce the *generating function*

$$G_n(u) \triangleq \sum_{a=0}^{n-1} \Pr\{S_n = a\}\, q^{ua} = \overline{q^{u\,S_n}}, \quad -\infty < u < \infty, \tag{7}$$

and the *semi-invariant generating function*

$$\mu_n(u) \triangleq \log_q G_n(u), \qquad -\infty < u < \infty. \tag{8}$$

Define independent identically distributed random variables

$$\xi_i \triangleq \begin{cases} 1 & \text{if } x_i = y_i, \\ 0 & \text{otherwise}, \end{cases} \quad \Pr\{\xi_i = a\} = \begin{cases} \frac{1}{q} & \text{if } a = 1, \\ \frac{q-1}{q} & \text{if } a = 0\,. \end{cases} \tag{9}$$

One can easily see that the vector sequence $\underline{\xi}_i \triangleq (\xi_i, \xi_{i+1})$, $i = 1, \ldots, n-1$, is a stationary Markov chain with transition probabilities:

$$\Pr\left\{ \underline{\xi}_i = (a_1, a_2) \middle| \underline{\xi}_{i-1} = (a_3, a_4) \right\} = \begin{cases} \Pr\{\xi_{i+1} = a_2\} & \text{if } a_1 = a_4, \\ 0 & \text{if } a_1 \ne a_4, \end{cases} =$$

$$= \begin{cases} \frac{q-1}{q} & \text{if } a_1 = a_4,\ a_2 = 0, \\ \frac{1}{q} & \text{if } a_1 = a_4,\ a_2 = 1, \\ 0 & \text{if } a_1 \ne a_4. \end{cases} \tag{10}$$

In addition, η_i, $i = 1, \ldots, n-1$, defined by (4) can be written in the form: $\eta_i = f(\underline{\xi}_i) \triangleq 1 - \xi_i \xi_{i+1}$, i.e., the given sequence is a *deterministic* function of Markov chain (10)[1]. Hence, using the standard Markov arguments [4],

[1] Note that η_i, $i = 1, \ldots, n-1$, is not a Markov chain because for any i, $3 \le i \le n-1$, the conditional probability

$$\Pr\{\eta_i = 0 | \eta_{i-1} = 1, \eta_{i-2} = 0\} = 0 \quad \text{and} \quad \Pr\{\eta_i = 0 | \eta_{i-1} = 1\} = \frac{1}{q(q+1)}.$$

pp. 230-232, we can calculate the generating functions (7)-(8) and obtain the following asymptotic ($n \to \infty$) formula:

$$\mu_n(u) = \log_q G_n(u) = n\,\mu(u) + O(1), \qquad \mu(u) \triangleq \log_q \lambda(u)$$

$$\lambda(u) \triangleq \frac{1}{2q}\left[1 + (q-1)q^u + \sqrt{[1+(q-1)q^u]^2 - 4(q-1)q^u(1-q^u)}\right]. \qquad (11)$$

Finally, applying the Large Deviations Principle [5] to S_n, we get

Theorem 1 *Random coding bound $\underline{R}_{st}(d)$ defined by (6) has the form*

$$\underline{R}_{st}(d) = L_\mu(d) \triangleq \max_{u \le 0}\{ud - \mu(u)\}, \qquad 0 < d < \frac{q^2-1}{q^2}, \qquad (12)$$

where $L_\mu(d)$, $0 \le d \le \frac{q^2-1}{q^2}$, is a decreasing $\bigcup$-convex function and

$$L_\mu(0) = 1, \quad L_\mu\left(\frac{q^2-1}{q^2}\right) = 0, \qquad L_\mu(d) > 0, \quad 0 < d < \frac{q^2-1}{q^2}. \qquad (13)$$

3 Upper bounds on $N_{st}(n, D)$ and $R_{st}(d)$

3.1 The Plotkin upper bound on $R_{st}(d)$

A standard upper bound on the rate $R_{st}(d)$ is given by

Proposition 2 . *If* $\frac{q^2-1}{q^2} \le d < 1$, *then* $R_{st}(d) = 0$ *and*

$$R_{st}(d) \le 1 - \frac{q^2}{q^2-1}\,d \quad \textit{if} \quad 0 < d < \frac{q^2-1}{q^2}. \qquad (14)$$

3.2 On sphere size for stem Hamming similarity

For $q \ge 2$, introduce three recurrent Fibonacci-type sequences [6] of numbers $F_q^1(t)$, $F_q^2(t)$, $F_q^3(t)$, $t = 1, 2, \ldots$, where

$$F_q^i(t) \triangleq (q-1)F_q^i(t-1) + (q-1)F_q^i(t-2), \quad i = 1, 2, 3, \quad t \ge 3, \qquad (15)$$

and $F_q^1(1) \triangleq q$, $F_q^1(2) \triangleq q^2 - 1$; $F_q^2(1) \triangleq q - 1$, $F_q^2(2) \triangleq (q-1)^2$; $F_q^3(1) \triangleq q - 1$, $F_q^3(2) \triangleq q(q-1)$. One can prove, that $F_q^1(t)$ $(F_q^2(t)/F_q^3(t))$ is the number of q-ary sequences $\boldsymbol{x} \in \mathcal{A}^n$ which do not contain 2-stems of the form $(0, 0)$ (and do not start and end/do not start or do not end with 0, correspondingly).

Let $\mathbf{t}^{(k)} \triangleq (t_1, t_2, \dots, t_k)$, $k = 1, 2, \dots$, denote an ordered collection of k integers. For fixed integers s, $1 \le s \le n-1$, and k, $1 \le k \le \min\left\{s; \lceil\frac{n-s}{2}\rceil\right\}$, define set

$$T(s,k) \triangleq \left\{ \mathbf{t}^{(k+1)} : t_1 \ge 0,\ t_{k+1} \ge 0,\ \ t_i \ge 1,\ i = 2, 3, \dots k, \right. \\ \left. \sum_{i=1}^{k+1} t_i = n - (s+k) \right\}. \tag{16}$$

Proposition 3 *For any s, $0 \le s \le n-1$, the sphere size $S_{st}(n,s) \triangleq |\{\boldsymbol{y} : H_{st}(\boldsymbol{x}, \boldsymbol{y}) = s\}|$ does not depend on its center $\boldsymbol{x} \in \mathcal{A}^n$. If $s = 0$, then*

$$\mathcal{S}_{st}(n,0) \triangleq |\{\boldsymbol{y} : H_{st}(\boldsymbol{x}, \boldsymbol{y}) = 0\}| = F_q^1(n). \tag{17}$$

If $1 \le s \le n-1$, then

$$S_{st}(n,s) = \sum_{k=1}^{\min\{s;\lceil\frac{n-s}{2}\rceil\}} \binom{s-1}{k-1} \sum_{T(s,k)} \left\{ F_q^3(t_1) \prod_{i=2}^{k} F_q^2(t_i)\, F_q^3(t_{k+1}) \right\}. \tag{18}$$

For the case $q \ge 2$, Proposition 3 means that the random coding bound $\underline{R}_{st}(d) = L_\mu(d)$, $0 < d < \frac{q^2-1}{q^2}$, (defined by (6) and calculated in Theorem 1) can be also written as

$$\underline{R}_{st}(d) = L_\mu(d) = 1 - \overline{\lim_{n\to\infty}}\ \frac{\log_q \mathcal{S}(n, (1-d)n)}{n}, \qquad 0 < d < \frac{q^2-1}{q^2}. \tag{19}$$

3.3 The Elias upper bound on $R_{st}(d)$

The standard Elias arguments [3] and asymptotic formula (19) yield

Theorem 2 *For any d, $0 < d < \frac{q^2-1}{q^2}$, the rate $R_{st}(d) \le U_\mu(d)$, and upper bound $U_\mu(d)$ is presented by parametric equations*

$$U_\mu(d) = u\mu'(u) - \mu(u), \qquad d = \mu'(u)\left[2 - \mu'(u)\frac{q^2}{q^2-1}\right], \quad u \le 0, \tag{20}$$

where function $\mu(u)$, $u \le 0$, is defined in Theorem 1.

Upper bound $U_\mu(d)$ can be called the Elias bound [3]. The given bound improves the Plotkin bound (14) for small values of d, $0 < d < d_q$. We calculated $d_2 \approx 0.60$ and $d_4 \approx 0.13$.

Acknowledgement. The authors are grateful to L. A. Bassalygo, V. M. Blinovsky, B. M. Gurevich and S. A. Pirogov for discussions and valuable info about Large Deviations Principle [5].

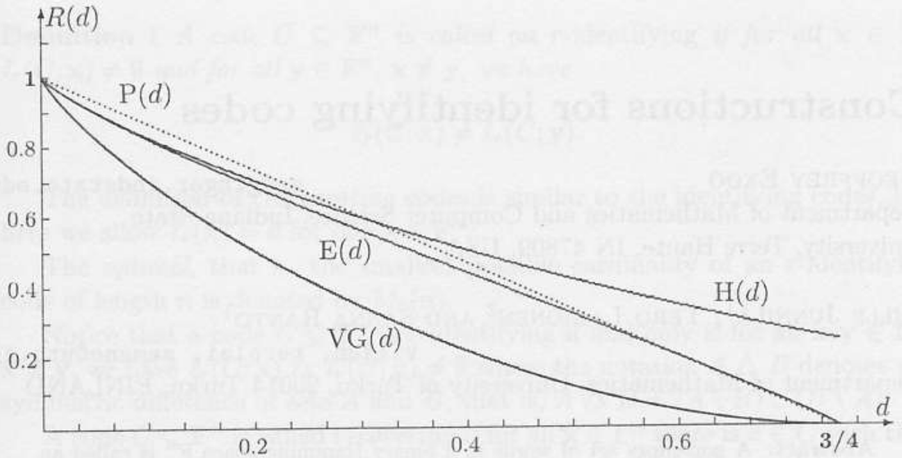

Figure 1: Upper and lower bounds for the rate of DNA-codes for $q = 2$: H(d) – Hamming bound, P(d) – Plotkin bound. E(d) – Elias bound, VG(d) – Varshamov-Gilbert bound.

References

[1] A. G. D'yachkov, A. J. Macula, D. C. Torney, P. A. Vilenkin, P. S. White, I. K. Ismagilov, R. S. Sarbayev, On DNA codes, *Probl. Pered. Inform.* 41, 2005, 57-77 (in Russian). English transl.: *Probl. Inform. Transm.* 41, 2005, 349-367.

[2] M. A. Bishop, A. G. D'yachkov, A. J. Macula, T. E. Renz, V. V. Rykov, Free energy gap and statistical thermodynamic fidelity of DNA codes *J. Comput. Biol.* 14, 2007, 1088-1104.

[3] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-correcting Codes*, Amsterdam, The Netherlands: North Holland, 1977.

[4] V. N. Tutubalin, *The Theory of Probability and Random Processes*, Moscow: Publishing House of Moscow State University, 1992 (in Russian).

[5] A. Dembo, O. Zeitouni, *Large Deviations Techniques and Applications*, Boston, MA: Jones and Bartlett, 1993.

[6] P. J. Cameron, *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, 1994.

Constructions for identifying codes

GEOFFREY EXOO ge@ginger.indstate.edu
Department of Mathematics and Computer Science, Indiana State University, Terre Haute, IN 47809, USA

VILLE JUNNILA[1], TERO LAIHONEN[2] AND SANNA RANTO[1]
viljun, terolai, samano@utu.fi
Department of Mathematics, University of Turku, 20014 Turku, FINLAND

Abstract. A nonempty set of words in a binary Hamming space $\mathbf{F}^n$ is called an r-identifying code if for every word the set of codewords within distance r from it is unique and nonempty. The smallest possible cardinality of an r-identifying code is denoted by $M_r(n)$. In this paper, we consider questions closely related to the open problem whether $M_{t+r}(n+m) \leq M_t(m)M_r(n)$ is true. For example, we show results like $M_{1+r}(n+m) \leq 4M_1(m)M_r(n)$, which improve previously known bounds. We also obtain a result $M_1(n+1) \leq (2+\varepsilon_n)M_1(n)$ where $\varepsilon_n \to 0$ when $n \to \infty$. This bound is related to the conjecture $M_1(n+1) \leq 2M_1(n)$. Moreover, we give constructions for the best known 1-identifying codes of certain lengths.

1 Introduction

Karpovsky, Chakrabarty and Levitin introduced identifying codes in [6] for locating malfunctioning processors in multiprocessor architectures. The research of identifying codes is also inspired by applications to sensor networks and alarm systems. Nowadays identifying codes are an actively studied topic of its own; the updated bibliography of identifying codes can be found from [7]. Identifying codes have been considered in many different graphs; in this paper we consider the binary Hamming spaces (i.e. binary hypercubes).

We denote by $\mathbf{F}^n$ the binary Hamming space of dimension n. The (Hamming) *distance* between two vectors (called words) $\mathbf{x}$ and $\mathbf{y}$ in $\mathbf{F}^n$ is denoted by $d(\mathbf{x}, \mathbf{y})$. The (Hamming) *weight* of a word $\mathbf{x}$, is denoted by $w(\mathbf{x})$. The (Hamming) *ball* of radius r centered at $\mathbf{x} \in \mathbf{F}^n$ is $B_r(\mathbf{x}) = \{\mathbf{y} \in \mathbf{F}^n \mid d(\mathbf{x}, \mathbf{y}) \leq r\}$.

A code of length n is a nonempty subset of $\mathbf{F}^n$. Let $C \subseteq \mathbf{F}^n$ be a code. The *I-set* of a word $\mathbf{x} \in \mathbf{F}^n$ (with respect to the code C) is defined to be

$$I_r(\mathbf{x}) = I_r(C; \mathbf{x}) = B_r(\mathbf{x}) \cap C.$$

[1] Research supported by the Academy of Finland under grant 210280.
[2] Research supported by the Academy of Finland under grant 111940.

Definition 1 *A code $C \subseteq \mathbf{F}^n$ is called an r-identifying if for all $\mathbf{x} \in \mathbf{F}^n$ $I_r(C;\mathbf{x}) \neq \emptyset$ and for all $\mathbf{y} \in \mathbf{F}^n$, $\mathbf{x} \neq \mathbf{y}$, we have*

$$I_r(C;\mathbf{x}) \neq I_r(C;\mathbf{y}).$$

The definition of *r-separating* codes is similar to the identifying codes, but here we allow $I_r(\mathbf{x}) = \emptyset$ for one $\mathbf{x} \in \mathbf{F}^n$.

The *optimal*, that is, the smallest possible cardinality of an r-identifying code of length n is denoted by $M_r(n)$.

Notice that a code $C \subseteq \mathbf{F}^n$ is r-identifying if and only if for all $\mathbf{x}, \mathbf{y} \in \mathbf{F}^n$, $\mathbf{x} \neq \mathbf{y}$, we have $I_r(C;\mathbf{x}) \bigtriangleup I_r(C;\mathbf{y}) \neq \emptyset$ where the notation $A \bigtriangleup B$ denotes the symmetric difference of sets A and B, that is, $A \bigtriangleup B = (A \setminus B) \cup (B \setminus A)$.

A code $C \subseteq \mathbf{F}^n$ is called *r-covering* if for all $\mathbf{x} \in \mathbf{F}^n$ there is $\mathbf{c} \in C$ such that $d(\mathbf{x},\mathbf{c}) \leq r$ (i.e., $|I_r(C;\mathbf{x})| \geq 1$). Moreover, if a code $C \subseteq \mathbf{F}^n$ has the property that for all $\mathbf{x} \in \mathbf{F}^n$ $|I_r(C;\mathbf{x})| \geq \mu$, then the code is called *μ-fold r-covering*. The optimal cardinality of an r-covering is denoted by $K(n,r)$. The vast topic of covering codes have been considered, for instance, in [3].

Let $C_1 \subseteq \mathbf{F}^n$ and $C_2 \subseteq \mathbf{F}^m$ be two codes, then their *direct sum*

$$C_1 \oplus C_2 = \{(\mathbf{a},\mathbf{b}) \mid \mathbf{a} \in C_1, \mathbf{b} \in C_2\} \subseteq \mathbf{F}^{n+m}.$$

In [1], the question whether

$$M_{r+t}(n+m) \leq M_r(n)M_t(m) \tag{1}$$

holds is mentioned as an open problem. In [5] the result is proved for $r = t = 1$. In Section 2 of this paper, we consider the problems closely related to the conjecture (1) in a general case. In particular, we show that $M_{r+1}(n+m) \leq 4M_r(n)M_1(m)$ and also present some numerical improvements on known bounds on $M_r(n)$. In [1], it is also asked whether $M_1(n+1) \leq 2M_1(n)$ is true. In the last section, we show that $M_1(n+1) \leq (2+\varepsilon_n)M_1(n)$ where $\varepsilon_n \to 0$ as $n \to \infty$.

The proofs omitted in this paper are in [4].

2 New code constructions for r-identifying codes

In this section, we will present some direct sum constructions for $(r+t)$-identifying codes. The motivation for this comes from the conjecture (1).

Lemma 1 *Let $C \subseteq \mathbf{F}^n$ be an r-identifying code. Then for all $\mathbf{x} \in \mathbf{F}^n$ there exists $\mathbf{c} \in C$ such that $d(\mathbf{c},\mathbf{x}) = r$ or $r+1$.*

In the subsequent considerations we refer to the following condition for a given code C:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbf{F}^n : I_t(C; \mathbf{x}) \setminus I_{t-1}(C; \mathbf{y}) \neq \emptyset. \tag{2}$$

We will use the following notations:

- The optimal cardinality of a t-identifying code satisfying the condition (2) is denoted by $\overline{M}_t(n)$.
- The optimal cardinality of a t-identifying code which is also $(t-1)$-separating and satisfy the condition (2) is denoted by $\overline{M}_{t,t-1}(n)$.
- The optimal cardinality of a t-identifying code such that for every $\mathbf{x} \in \mathbf{F}^n$ there exists a codeword *exactly* at distance t from $\mathbf{x}$ is denoted by $M_t'(n)$.
- We denote by $M_1''(n)$ the optimal cardinality of a 1-identifying and 2-fold 1-covering code. It is clear that $M_1''(n) \leq 2M_1(n)$.

Theorem 1 *We have*

$$M_{r+t}(n+m) \leq \begin{cases} M_r(n)\overline{M}_{t,t-1}(m), \\ M_r'(n)\overline{M}_t(m) \end{cases} \tag{3}$$

and

$$M_{r+1}(n+m) \leq M_r'(n)M_1''(m). \tag{4}$$

Moreover, $M_r'(n) \leq 2M_r(n)$. *Especially,*

$$M_{r+t}(n+m) \leq 2M_r(n)\overline{M}_t(n) \tag{5}$$

$$M_{r+1}(n+m) \leq 4M_r(n)M_1(m). \tag{6}$$

Proof. Let us first prove the inequalities (3). Let $C_1 \subseteq \mathbf{F}^n$ be an r-identifying code and $C_2 \subseteq \mathbf{F}^m$ be a t-identifying and $(t-1)$-separating code satisfying the condition (2). We will first show that $C = C_1 \oplus C_2 \subseteq \mathbf{F}^{n+m}$ is an $(r+t)$-identifying code. It is easy to see that C is an $(r+t)$-covering code, this implies that $I_r(X) = \emptyset$ if and only if $X = \emptyset$. Therefore, in order to prove that C is $(r+t)$-identifying, it is enough to show that $I_{r+t}(\mathbf{x}) \triangle I_{r+t}(\mathbf{y}) \neq \emptyset$ for all $\mathbf{x}, \mathbf{y} \in \mathbf{F}^{n+m}$ $(\mathbf{x} \neq \mathbf{y})$. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \in \mathbf{F}^{n+m}$, where $\mathbf{x}_1, \mathbf{y}_1 \in \mathbf{F}^n$ and $\mathbf{x}_2, \mathbf{y}_2 \in \mathbf{F}^m$, moreover $\mathbf{x} \neq \mathbf{y}$.

1) Suppose first $\mathbf{x}_1 \neq \mathbf{y}_1$. Then there exists $\mathbf{c}_1 \in I_r(C_1; \mathbf{x}_1) \triangle I_r(C_1; \mathbf{y}_1)$. Without loss of generality we may assume that $\mathbf{c}_1 \in I_r(C_1; \mathbf{x}_1) \setminus I_r(C_1; \mathbf{y}_1)$. Since the code C_2 satisfies the condition (2), there exists a codeword $\mathbf{c}_2 \in C_2$ such that $\mathbf{c}_2 \in I_t(C_2; \mathbf{x}_2) \setminus I_{t-1}(C_2; \mathbf{y}_2)$. Hence, $(\mathbf{c}_1, \mathbf{c}_2) \in I_{r+t}(C; \mathbf{x}) \setminus I_{r+t}(C; \mathbf{y})$.

2) Suppose then $\mathbf{x}_1 = \mathbf{y}_1$. By Lemma 1, there exists $\mathbf{c}_1 \in C_1$ such that $d(\mathbf{c}_1, \mathbf{x}_1) = r$ or $r+1$. Assume first that $d(\mathbf{c}_1, \mathbf{x}_1) = r$. Since C_2 is a t-identifying code and $\mathbf{x}_2 \neq \mathbf{y}_2$, there exists a codeword $\mathbf{c}_2 \in C_2$ such that $\mathbf{c}_2 \in I_t(\mathbf{x}_2) \triangle I_t(\mathbf{y}_2)$. Therefore, $(\mathbf{c}_1, \mathbf{c}_2) \in I_{r+t}(\mathbf{x}) \triangle I_{r+t}(\mathbf{y})$. Assume then that $d(\mathbf{c}_1, \mathbf{x}_1) = r+1$. Since C_2 is also a $(t-1)$-separating code and $\mathbf{x}_2 \neq \mathbf{y}_2$, there exists a codeword $\mathbf{c}_2 \in C_2$ such that $\mathbf{c}_2 \in I_{t-1}(\mathbf{x}_2) \triangle I_{t-1}(\mathbf{y}_2)$. Hence, $(\mathbf{c}_1, \mathbf{c}_2) \in I_{r+t}(\mathbf{x}) \triangle I_{r+t}(\mathbf{y})$. Thus, we have proved that $C_1 \oplus C_2$ is an $(r+t)$-identifying code.

Let $C_3 \subseteq \mathbf{F}^n$ be an r-identifying code such that for every $\mathbf{x} \in \mathbf{F}^n$ there exists a codeword exactly at distance r from it and $C_4 \subseteq \mathbf{F}^m$ a t-identifying code satisfying the condition (2). Showing that $C_3 \oplus C_4 \subseteq \mathbf{F}^{n+m}$ is an $(r+t)$-identifying code is similar to the proof described above. However, in the second part of the proof we can assume that there always exists a codeword $\mathbf{c}_1 \in \mathbf{F}^n$ such that $d(\mathbf{x}_1, \mathbf{c}_1) = r$.

Let us now move on to the inequality (4). It is easy to see that 1-identifying and 2-fold 1-covering code satisfies the condition (2) for $t = 1$. Therefore, the result immediately follows from (3).

For the estimate $M'_r(n) \le 2M_r(n)$, see [4]. □

In [2, Theorem 3] it is proved that when $1 \le t < m \le r$ we have

$$M_{r+t}(n+m) \le 2^m M_r(n). \tag{7}$$

Assume first $t = 1$. Since $C = \mathbf{F}^m \setminus \{1^m\}$ is clearly a 1-identifying and 0-separating code satisfying the condition (2), we have, by (3), that $M_{r+1}(n+m) \le (2^m - 1)M_r(n)$. Using (6) we obtain further improvements to (7). Namely, we know that $M_1(m) \le \frac{9}{2} \cdot \frac{2^m}{m+1} < 2^{m-2} - 1$ when $m \ge 18$ and, by the tables of [2], this also holds for $m \ge 8$.

In the next theorem we improve (7) using (5) when $t \ge 2$ and $m \ge 2t$. We give an upper bound for $\overline{M}_t(m)$ using a method inspired by Delsarte and Piret [3, p. 320].

Theorem 2 *Let $m \ge 2t$.*

$$M_{r+t}(n+m) \le 2\left\lceil \frac{2^m}{\min\{\binom{m}{t}, 2\binom{m-1}{t}\}} 2m \ln 2 \right\rceil M_r(n).$$

In what follows, we develop further the direct sum approach with the aid of k-locating-dominating codes. It is a class of codes introduced by Slater (see [8]) closely related to identifying codes; a code $C \subseteq \mathbf{F}^n$ is *k-locating-dominating* if $I_r(C; \mathbf{x})$ is nonempty and $I_r(C; \mathbf{x}) \neq I_r(C; \mathbf{y})$ for all non-codewords $\mathbf{x}, \mathbf{y} \in \mathbf{F}^n \setminus C$.

Theorem 3 *Let $C_1 \subseteq \mathbf{F}^n$ be a 1-identifying code which is also a 2-fold 1-covering and has the property that it is k-locating-dominating for all $1 \leq k \leq r+1 \leq n-2$. Let $C_2 \subseteq \mathbf{F}^m$ be an r-identifying code. Then $C_1 \oplus C_2 \subseteq \mathbf{F}^{n+m}$ is an $(r+1)$-identifying code.*

The condition that the identifying code C_1 is a 2-fold 1-covering increases the cardinality only slightly (see [5]). The extra requirement that C_1 is also k-locating-dominating for $1 \leq k \leq n-2$ is not demanding cardinalitywise either. Indeed, the best 1-identifying 2-fold 1-covering codes which were found (Theorem 4), are immediately k-locating-dominating for all $1 \leq k \leq n-2$ as well.

Theorem 4 $M_1''(7) \leq 38$, $M_1''(8) \leq 70$, *and* $M_1''(10) \leq 249$.

It can also be checked that the best known 1-identifying and 2-fold 1-covering code of length 9 and of cardinality 128 [5] is k-locating-dominating for all $1 \leq k \leq 7$.

Corollary 1 $M_4(n) \leq 38M_3(n-7)$, $M_5(n) \leq 70M_4(n-8)$, $M_6(n) \leq 128M_5(n-9)$ *and* $M_7(n) \leq 249M_6(n-10)$.

The codes of Theorem 4 are also useful for bounding $M_1(n)$ from above. Namely, it has been proved in [5] that if a code $C \subseteq \mathbf{F}^n$ is 1-identifying and 2-fold 1-covering then the code $D = \{(\pi(\mathbf{u}), \mathbf{u}, \mathbf{u}+\mathbf{v}) \mid \mathbf{u} \in \mathbf{F}^n, \mathbf{v} \in C\} \subseteq \mathbf{F}^{2n+1}$ is 1-identifying and 2-fold 1-covering ($\pi(\cdot)$ is the parity check bit). Hence, we have the following theorem where the previous records are given in the parenthesis [2].

Theorem 5 $M_1(17) \leq 17920$ (18558) *and* $M_1(21) \leq 254976$ (262144).

A natural generalization of r-identifying codes are codes which identify sets of words, see [6]. A code $C \subseteq \mathbf{F}^n$ is called an *$(r, \leq \ell)$-identifying* if for all $X, Y \subseteq \mathbf{F}^n$, $|X|, |Y| \leq \ell$, $X \neq Y$, we have

$$\bigcup_{\mathbf{x} \in X} I_r(C; \mathbf{x}) \neq \bigcup_{\mathbf{y} \in Y} I_r(C; \mathbf{y}).$$

The smallest cardinality of such codes in $\mathbf{F}^n$ is denoted by $M_r^{(\leq \ell)}(n)$.

Theorem 6 *Let r be a positive integer and suppose $\ell \geq r+3$. Let $C_1 \subseteq \mathbf{F}^{n_1}$ be a $(1, \leq \ell)$-identifying code and $C_2 \subseteq \mathbf{F}^{n_2}$ be an $(r, \leq \ell)$-identifying code. Then $C_1 \oplus C_2 \subseteq \mathbf{F}^{n_1+n_2}$ is an $(r+1, \leq \ell)$-identifying code.*

Corollary 2 *When $r \geq 1$ and $\ell \geq r+3$ we have*

$$M_{1+r}^{(\leq \ell)}(n+m) \leq M_1^{(\leq \ell)}(n) M_r^{(\leq \ell)}(m).$$

3 A direct sum of 1-identifying code and F

In [1] it has been stated as an open problem whether $M_1(n+1) \leq 2M_1(n)$ holds, from there it also follows that $M_1(n+1) \leq 3M_1(n)$. The next theorem shows that $M_1(n+1) \leq (2+\varepsilon_n)M_1(n)$ where $\varepsilon_n \to 0$ as $n \to \infty$.

Theorem 7 *Assume $n \geq 2$. Then we have*

$$M_1(n+1) \leq (2 + \frac{1}{n+1})M_1(n).$$

Proof. Let $C \subseteq \mathbf{F}^n$ be an optimal 1-identifying code attaining $M_1(n)$. Define

$$C_1 = \{\mathbf{x} \mid \mathbf{x} \in C\ ,\ |I_1(\mathbf{x})| = 1\} \text{ and}$$
$$N_1 = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{F}^n,\ \mathbf{x} \notin C,\ |I_1(\mathbf{x})| = 1\}.$$

Clearly, $|C_1 \cup N_1| \leq M_1(n)$. Assume first $|C_1| \leq M_1(n)/(n+1)$. Let $D_1 = C \oplus \mathbf{F} \subseteq \mathbf{F}^{n+1}$. Denote $O_l = \mathbf{F}^n \oplus \{l\}$ where $l \in \mathbf{F}$. Assume $\mathbf{x} = (\mathbf{x}', a) \in \mathbf{F}^{n+1}$ with $\mathbf{x}' \in \mathbf{F}^n$ and $a \in \mathbf{F}$. Since C is 1-identifying, the set $I_1(D_1;\mathbf{x})$ can coincide only with the I-sets of words in O_{a+1}. If $|I_1(C;\mathbf{x}')| \geq 2$, then the word $\mathbf{x}$ is uniquely identified by its I-set $I_1(D_1;\mathbf{x})$ since each word in O_{a+1} 1-covers a unique word in O_a. It can now be assumed that $|I_1(C;\mathbf{x}')| = 1$.

Assume $\mathbf{x}' \in N_1$, i.e. $I_1(C;\mathbf{x}') = \{\mathbf{x}' + \mathbf{e}\}$, where $\mathbf{e} \in \mathbf{F}^n$ is a word of weight 1. The only word in O_{a+1} which 1-covers the codeword $(\mathbf{x}' + \mathbf{e}, a)$ is the word $(\mathbf{x}' + \mathbf{e}, a+1)$. However, $|I_1(C;\mathbf{x}' + \mathbf{e})| \geq 2$ and therefore, as above, it can be said that $\mathbf{x}$ is uniquely identified. If $\mathbf{x}' \in C_1$, then clearly, $I_1(D_1;(\mathbf{x}', a)) = I_1(D_1;(\mathbf{x}', a+1))$. But such a problematic case can be solved by adding one codeword to the code D_1. Thus, we have the claim in this case.

Assume then $|C_1| > M_1(n)/(n+1)$. Let $\mathbf{z} \in \mathbf{F}^n$ be a word of weight 1. Consider then a code $D_2 \subseteq \mathbf{F}^{n+1}$ defined as

$$D_2 = (C \oplus \{0\}) \cup ((C + \mathbf{z}) \oplus \{1\}).$$

Assume $\mathbf{x} = (\mathbf{x}', a) \in \mathbf{F}^{n+1}$ with $\mathbf{x}' \in \mathbf{F}^n$ and $a \in \mathbf{F}$. If $|I_1(C;\mathbf{x}')| \geq 2$, then, as above, the word $\mathbf{x}$ is uniquely identified by its I-set $I_1(D_2;\mathbf{x})$.

Assume now that $\mathbf{x}' \in C_1$, i.e. $I_1(C;\mathbf{x}') = \{\mathbf{x}'\}$. The only word in O_{a+1} which 1-covers the codeword $(\mathbf{x}', a)$ is the word $(\mathbf{x}', a+1)$. However, $|I_1(D_2;(\mathbf{x}', a+1)) \cap O_{a+1}| \geq 2$ since $|I_1(D_2;(\mathbf{x}' + \mathbf{z}, a+1)) \cap O_{a+1}| = 1$ and the underlying code C is 1-identifying. Therefore, as before, it can be deduced that $\mathbf{x}$ is uniquely identified by its I-set $I_1(D_2;\mathbf{x})$.

Assume then $\mathbf{x}' \in N_1$, i.e. $I_1(C;\mathbf{x}') = \{\mathbf{x}' + \mathbf{e}\}$, where $w(\mathbf{e}) = 1$. Again it suffices to consider the word $(\mathbf{x}' + \mathbf{e}, a+1)$. If $I_1(D_2;(\mathbf{x}', a)) = I_1(D_2;(\mathbf{x}' + \mathbf{e}, a+$

1)), then $I_1(D_2; (\mathbf{x}'+\mathbf{e}, a+1)) \cap O_{a+1} = \{(\mathbf{x}', a+1)\}$. Since $I_1(D_2; (\mathbf{x}', a)) \cap O_a = \{(\mathbf{x}'+\mathbf{e}, a)\}$, we have $d((\mathbf{x}', a), (\mathbf{x}'+\mathbf{e}+\mathbf{z}, a+1)) = 1$. Thus, $I_1(D_2; (\mathbf{x}', a)) = I_1(D_2; (\mathbf{x}'+\mathbf{e}, a+1))$ if and only if $\mathbf{e} = \mathbf{z}$. The code D_2 can clearly be made 1-identifying by adding a codeword to the set for each one of these problematic cases. Moreover, there exists a word $\mathbf{e}' \in \mathbf{F}^n$ of weight 1 such that

$$|\{\mathbf{x} \in \mathbf{F}^n | \ \mathbf{x} \notin C, \ I_1(\mathbf{x}) = \{\mathbf{x}+\mathbf{e}'\}\}| \leq \frac{|N_1|}{n}.$$

If we now choose $\mathbf{z} = \mathbf{e}'$, then we have, by the previous considerations, that

$$M_1(n+1) \leq 2M_1(n) + \frac{|N_1|}{n} \leq (2 + \frac{1}{n+1})M_1(n).$$

□

References

[1] U. Blass, I. Honkala, S.Litsyn, Bounds on identifying codes, *Discr. Math.* 241, 2001, 119-128.

[2] I. Charon, G. Cohen, O. Hudry, A. Lobstein, New identifying codes in the binary Hamming space, *Europ. J. Combin.*, to appear.

[3] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering Codes*, Elsevier, Amsterdam, 1997.

[4] G. Exoo, V. Junnila, T. Laihonen, S. Ranto, Upper bounds for binary identifying codes, submitted.

[5] G. Exoo, T. Laihonen, S. Ranto, Improved upper bounds on binary identifying codes, *IEEE Trans. Inform. Theory* 53, 2007, 4255-4260.

[6] M. G. Karpovsky, K. Chakrabarty, L. B. Levitin, On a new class of codes for identifying vertices in graphs, *IEEE Trans. Inform. Theory* 44, 1998, 599-611.

[7] A. Lobstein, Identifying and locating-dominating codes in graphs, a bibliography, Published electronically at `http://perso.enst.fr/~lobstein/debutBIBidetlocdom.pdf`.

[8] P. J. Slater, Dominating and reference sets in a graph, *J. Math. Phys. Sci.* 22, 1988, 445-455.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 99-107*

Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes

Cédric Faure cedric.faure@inria.fr
INRIA Rocquencourt

Lorenz Minder lorenz@eecs.Berkeley.edu
EECS Berkeley

Abstract. We present a practical expected usually quartic time algorithm to recover the structure of an algebraic geometry code defined over a hyperelliptic code of genus $g \leq 2$. Its main application is an attack of the McEliece cryptosystem based on algebraic geometry codes defined over curves of small genus. Our algorithm is a adaptation of the well-known Sidelnikov-Shestakov algorithm [6].

1 Introduction

In 1978, R. J. McEliece presented the first version of the cryptosystem which was to become the reference in public key cryptography based on coding theory [2]. The main version of McEliece's scheme uses Goppa codes. However, many other codes families have been studied to fit in McEliece's system.

The choice of a different code was often motivated by the goal to provide better security for a given key size. The most basic measure for security of that type of cryptosystem is the cost to decode the code with (a refined version of) information set decoding to the decoding bound with a given fixed key size (which is the amount of memory needed to store a generator matrix).

It is, however, important to be aware of the fact that this direct-decoding measure does not take into account the possibility that an attacker may attempt to recover the structure of the code instead of trying to break the system by attempting to decode an unstructured linear code. The possibility of doing so depends on the code that has been used in the construction. Efficient structural attacks have been developed for example against Reed Solomon codes by Sidelnikov and Shestakov [6], then against concatenated codes by Sendrier [5], and against Reed-Muller codes by Minder and Shokrollahi [4].

Since structural attacks tend to be very effective if applicable, a difficult task faced by the designer of McEliece-type cryptosystems is to chose a family of codes which has a good tradeoff between rate and correction capability (and is thus resistant against direct decoding attacks) while also being structurally secure.

When it comes to correction capability at a fixed rate, algebraic geometry codes are often the best known choice for a given set of parameters.[1]

The superiority of geometric codes makes them thus a seemingly excellent building block for McEliece type cryptosystems, and it is therefore important to the researchers in the field to know whether and to what extent these codes lead to secure cryptosystems.

In this paper, we show that curves of very low genus g (more precisely, $g \leq 2$) are a bad choice, and that they can be broken with very high probability in heuristic expected polynomial (usually quartic) time. Specifically, this breaks some of the Janwa-Moreno parameters [1]. A predecessor of this attack which worked for $g = 1$ was presented in [3] and was partly based on earlier, unpublished ideas by Bleichenbacher, Melnik and Shokrollahi. Since the case $g = 0$ (Reed Solomon codes) was taken care of by Sidelnikov and Shestakov [6], we restrict our attention in this paper to the case $g = 2$ and we present an algorithm which attacks the cryptosystem in time $O(n^4)$ binary operations, where n is the length of the ciphertext block.

For our attack to work, a few additional assumption on the code have to be made, such as the requirement that the blocklength n be reasonably close to maximal for the given curve. Since this covers the most interesting cases, it does not appear to be a severe restriction.

It is in principle possible to run our attack on hyperelliptic curves of genus larger than two, but in its current form only at a large cost in both running time and success probability. We have not investigated the question closely, and it may well be that already for hyperelliptic curves of genus 3, our attack is not all that interesting in its current form. Our preliminary opinion on the matter is that without substantial improvements, this kind of attack is not applicable to codes defined over sufficiently complicated curves.

In section 2, we recall mathematical concepts and definitions. In section 3, we present the McEliece cryptosystem in the setting of hyperelliptic codes, and in section 4, our attack will be exposed. We will then present our conclusions in section 5.

2 Definitions and notations

2.1 Notions of algebraic geometry

Let $\mathcal{X}$ be a hyperelliptic curve of genus $g = 2$ over $\mathbb{A}_2(\mathbb{F}_q)$, defined by the equation:

$$y^2 + G(x)y = F(x), \text{ with } \deg(F) = 2g + 1, \text{ and } \deg(G) \leq g.$$

[1] We ignore graph based codes here, because they are structurally weak.

A divisor Δ over $\mathcal{X}$ is a formal finite sum of points of $\mathcal{X}$ with positive and negative multiplicities:

$$\Delta = \sum_{P \in \mathcal{X}} n_P \langle P \rangle,\ n_P \in \mathbb{Z}.$$

The degree of a divisor is the sum of the multiplicities of the points in the divisor:

$$\deg(\Delta) = \sum_{P \in \mathcal{X}} n_P.$$

Any rational function f over $\mathcal{X}$ has an associated divisor $\operatorname{div}(f)$ which is obtained by adding every zero of f and subtracting every pole (counted with multiplicity in either case), i.e.,

$$\operatorname{div}(f) = \sum_{P \in \mathcal{X}} \operatorname{ord}_P(f) \langle P \rangle.$$

For every rational function, we have $\deg(\operatorname{div}(f)) = 0$, but the converse is not true in general. The Jacobian group of $\mathcal{X}$ is defined as the group:

$$\operatorname{Jac}(\mathcal{X}) = \text{Divisors of degree } 0/\text{divisors of rational functions.}$$

The Generalized Hasse-Weil theorem states that

$$\operatorname{Jac}(\mathcal{X}) \simeq \mathcal{G} = \frac{\mathbb{Z}}{d_1\mathbb{Z}} \times \cdots \times \frac{\mathbb{Z}}{d_{2g}\mathbb{Z}},\ \text{with } d_1 | \ldots | d_{2g},\ d_1 | q-1,$$

and that

$$(\sqrt{q}-1)^{2g} \leq \|\sharp\mathcal{G}\| \leq (\sqrt{q}+1)^{2g}.$$

2.2 Geometric codes

Let Δ be a divisor of degree $k+g-1$ over $\mathcal{X}$. We define the associated linear space $\mathcal{L}(\Delta)$:

$$\mathcal{L}(\Delta) = \{f \in \mathbb{F}_q(\mathcal{X}) | \operatorname{div}(f) + \Delta \geq 0\} \cup \{0\}$$

The Riemann-Roch theorem states that $\mathcal{L}(\Delta)$ is a vector space of dimension k if $k \geq g-1$. (We shall always assume $k \geq g-1$ in the sequel, the other case being of no interest for the problem at hand.)

Given a set $(P_1, \ldots, P_n)$ of distinct rational points on $\mathcal{X}$, we can now define the associated geometric code $\operatorname{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n))$ as:

$$\operatorname{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n)) = \{(f(P_1), \ldots, f(P_n)) | f \in \mathcal{L}(\Delta)\}$$

This is a linear code of length n, of dimension k, and minimal distance $d \geq n-k-g+1$. Furthermore, this code can be decoded in polynomial time up to its correction capability $t = \frac{n-k-g}{2}$. See, e.g., [7].

3 The McEliece cryptosystem on geometric codes

A McEliece type cryptosystem on geometric codes can be defined as follows: Fix blocklength n and dimension k. Select a random curve $\mathcal{X}$ of genus 2 having at least $n+1$ rational points, and randomly select distinct rational points $P_1, \ldots, P_n$ on $\mathcal{X}$.

Let Δ be a divisor of degree $k+g-1$, and whose support is disjoint from the points P_i. This defines a code

$$\mathcal{C} := \mathrm{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n)).$$

Now let G be a random generator matrix of $\mathcal{C}$, computed by multiplying a canonical generator matrix for this code on the left with an invertible $k \times k$-matrix with coefficients in $\mathbb{F}_q$.

This generator matrix G then serves as public key. The code parameters $\mathcal{X}, \Delta, (P_1, \ldots, P_n)$ are the private key. A message $\mathbf{x} \in \mathbb{F}_q^k$ is encrypted by computing $\mathbf{y} := \mathbf{x}G + \mathbf{e}$, where $\mathbf{e}$ is a random weight $t = (n-k-g)/2$ error vector. The legitimate receiver who knows the secret parameters $\mathcal{X}$, Δ, $P_1, \ldots, P_n$ can recover $\mathbf{x}$ by applying a decoding algorithm to $\mathcal{C}$, and thus computing $\mathbf{x}G$. Given $\mathbf{x}G$, the value of $\mathbf{x}$ can be recovered by solving a system of linear equations.

4 An attack against geometric codes of genus 2

Our goal is to recover a private key given the public key. In our setting this means the following: The attacker is given a generator matrix G of a hyperelliptic code $\mathcal{C}'$ of unknown parameters $\mathcal{X}', \Delta', (P_1', \ldots, P_n')$. Inspecting G, he then finds a (typically different) set of parameters $\mathcal{X}, \Delta, (P_1, \ldots, P_n)$, such that the code

$$\mathcal{C} := \mathrm{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n))$$

is just a directional scaling of $\mathcal{C}'$, i.e., there are nonzero constants $c_1, \ldots, c_n$ such that any codeword $(y_1, \ldots, y_n) \in \mathcal{C}$ corresponds to a codeword $(c_1 y_1, \ldots, c_n y_n) \in \mathcal{C}'$.

He then finds the scaling coefficients $(c_1, \ldots, c_n)$, and this enables him to use the decoder for $\mathcal{C}$ to decode codewords for $\mathcal{C}'$, thus breaking the system.

4.1 Outline of the attack

Our algorithm works in four steps:

1. Recovering the group structure. In this stage we sample minimum weight codewords in order to collect linear equations on elements of the Jacobian. Given enough such relations, we can retrieve the finite group structure of the Jacobian $\mathcal{G}$.

2. Recovering the curve equation. We then use the Jacobian structure, along with a few additional carefully chosen minimum weight codewords, in order to get conditions on the coordinates of a few points of the curve.

   We then guess the coordinates of three points, and compute from this guess the coordinates of a larger set of points using the precomputed conditions. If the guess is correct, we can draw a hyperelliptic curve passing through our points. Otherwise we know that the guess is wrong, and we retry with another guess.

3. Recovering the coordinates of all the evaluation points. It is now possible, from the Jacobian structure, and the curve equation, to retrieve the coordinates of all points in the evaluation set.

4. Computing the scaling coefficients.

For all the steps to work, we will need additional assumptions. First, we assume that we have many evaluation points, i.e., that n is close to the number of rational points on $\mathcal{X}$.

Second, we will assume that $\gcd(k+g-1, |\mathcal{G}|) = 1$. Notice that we can force this latter condition to hold by working on a shortened version of the code, if necessary.

Third, we assume that the true minimum distance of the code is indeed $n-k-g+1$, and that many such codewords exist. Empirical evidence shows that this is virtually always true in our setting.

4.2 Preliminaries: code invariance

The parameters of a given geometric code are not unique : It is actually possible to generate the same code using a different evaluation set and a different divisor. This is very useful to the cryptanalyst, because it means that we can arbitrarily select some of the parameters we seek, and focus our search on the other ones.

In particular, we have $\mathrm{AGC}(\mathcal{X}', \Delta', (P'_1, \ldots, P'_n)) = \mathrm{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n))$ if there exists a curve isomorphism from $\mathcal{X}'$ to $\mathcal{X}$ which maps P'_i to P_i and Δ' to Δ.

Let $u, v, a, b, c \in \mathbb{F}_q$. If $g = 2$, then the mapping

$$(x, y) \mapsto (u^2 x + v, u^{2g+1} y + ax^2 + bx + c)$$

is a curve isomorphism. If we arbitrarily select 2 points, and the Y-coordinate of a third, for example x_1, x_2, y_1, y_2, y_3 of P_1, P_2, P_3, then with probability 1/2 there exists such an isomorphism which maps P'_i to P_i.

So we can arbitrarily fix P_1, P_2, y_3, and still have

$$\mathrm{AGC}(\mathcal{X}', \Delta', (P'_1, \ldots, P'_n)) = \mathrm{AGC}(\mathcal{X}, \Delta, (P_1, \ldots, P_n)).$$

for some $x_3, P_4, \ldots, P_n, \mathcal{X}, \Delta$ with probability 1/2.

Furthermore, because $\gcd(k+g-1, |\mathcal{G}|) = 1$, we can assume $\Delta = (k+g-1)\Delta_0$, where Δ_0 is a divisor of degree 1.

4.3 First step: recovering the Jacobian structure

We know that $\mathrm{Jac}(\mathcal{X}) \simeq \mathcal{G} = \frac{\mathbb{Z}}{d_1\mathbb{Z}} \times \cdots \times \frac{\mathbb{Z}}{d_{2g}\mathbb{Z}}$. In this step, we will recover the values of $d_1, \ldots, d_{2g}$, along with the images in $\mathcal{G}$ of some particular elements of the Jacobian by an unknown isomorphism φ.

We generate minimum weight codewords, i.e., we compute $\mathbf{x} \in \mathcal{C}$ such that $|\mathbf{x}| = n-k-g+1$. For general codes, finding minimum weight words is hard, but in our specific case we need only $O(n^2)$ operations in $\mathbb{F}_q$ on the average to find a single word with the desired property.

Let $\mathbf{x}$ be such a codeword, and write $f \in \mathcal{L}(\Delta)$ the associated rational function (i.e., the function such that $\mathbf{x}_i = f(P_i)$ for $1 \le i \le n$).

If $x_{i_1} = \cdots = x_{i_{k+g-1}} = 0$, then

$$f(P_{i_1}) = \cdots = f(P_{i_k+g-1}) = 0,$$

and so

$$\mathrm{div}(f) = \langle P_{i_1} \rangle + \cdots + \langle P_{i_{k+g-1}} \rangle - (k+g-1)\Delta_0. \qquad (1)$$

Notice that the minimality of $\mathbf{x}$ implies equality in (1) rather than just the greater-than relation that holds for any word and its associated function.

If we set $\tilde{z}_i = \varphi(\langle P_i \rangle - \Delta_0)$ in $\mathcal{G}$, we have

$$\sum_{j=0}^{k+g-1} \tilde{z}_{i_j} = 0.$$

With many (slightly more than n) equations of this form, we are able to recover the structure of the group $\mathcal{G}$, i.e., the values of $d_1, \ldots, d_{2g}$. This is true because a random system of overdetermined linear equations does not have any solution in $\mathbb{Z}/m\mathbb{Z}$ for arbitrary m; but our system has solutions in $\mathbb{Z}/d_i\mathbb{Z}$.

One technique to recover the d_i is thus to select several systems with only n equations, and compute the determinant of the associated matrix; it will always be a multiple of $d_1 \ldots d_{2g}$, and so we can find the d_i by taking the gcd of several such determinants. Only few determinants are needed in practice, but the computation of a determinant is usually $O(n^4)$ binary operations.

We can also solve the system to get the values of all the $\tilde{z}_i$. However, the map φ is still unknown at this point.

We now want to determine in $\mathcal{G}$ the value of $\delta_0 = \varphi(\Delta_0 - \langle \mathcal{O} \rangle)$. We use a statistical test to do this. Since most rational points are in the set $\{P_1, \ldots, P_n\}$, the probability that for a random index i, there is an index j, such that P_i and P_j are opposites of one another is $1-\varepsilon$, where ε is upper bounded by twice the number of points not in $\{P_1, \dots, P_n\}$ over the total number of rational points of $\mathcal{X}$. If P_i and P_j are indeed opposites, then $\langle P_i \rangle + \langle P_j \rangle = 2\langle \mathcal{O} \rangle$. It follows that $\tilde{z}_i + \tilde{z}_j = -2\delta_0$ in this case. For random indices i, j, the probability that $\tilde{z}_i + \tilde{z}_j = -2\delta_0$ is thus at least $(1-\varepsilon)/n$. At most $n/(1-\varepsilon)$ values can have that large a probability to be the equal to of $\tilde{z}_i + \tilde{z}_j$, so in the worst case we have $n/(1-\varepsilon)$ candidates for $-2\delta_0$, but we can expect that the set of sums behaves more randomly, and that we are able to extract the unique correct value of $-2\delta_0$ from the set of sums of all pairs. So we now know the value of δ_0, and we can compute all the values $z_i = \varphi(\langle P_i \rangle - \langle \mathcal{O} \rangle) = \tilde{z}_i + \delta_0$.

This test to recover the z_i usually runs in $O(n^2)$ multiplications over the base field.

4.4 Second step: Recovering the curve equation

We now generate two codewords $\mathbf{v}$ and $\mathbf{w}$ of weight $(n - k - g + 1)$, such that $\mathbf{v}$ and $\mathbf{w}$ have exactly $k + g - 3$ zero positions in common.

In order to do this, we first build a set $I \subset [1, n]$ of size $k + g - 3$ such that $\sum_{i \in I} z_i = (k + g - 1)\delta_0$. We now select two couples of opposite points, i.e., $(i_1, i_2, j_1, j_2) \in [1, n]$ so that $z_{i_1} + z_{i_2} = z_{j_1} + z_{j_2} = 0$.

Then we know that there exists a codeword $\mathbf{v} \in C$ with zero positions on $I \cup \{i_1, i_2\}$. We can easily compute $\mathbf{v}$ from the generator matrix G. By the same method, we compute the codeword $\mathbf{w} \in C$ with zero positions on $I \cup \{j_1, j_2\}$.

Now that we have two such codewords $\mathbf{v}$ and $\mathbf{w}$, if we call f_1 and f_2, their respective (unknown) associated rational functions in $\mathcal{L}((k + g - 1)\Delta_0)$, then there exists $a, b, c, d \in \mathbb{F}_q$ so that $\frac{f_1}{f_2} = \frac{ax+b}{cx+d}$.

The functions f_1 and f_2 are unknown but, by definition of f_1 and f_2, for every i such that $w_i \neq 0$, we have $\frac{f_1}{f_2}(P_i) = \frac{v_i}{w_i} = \frac{ax_i+b}{cx_i+d}$.

All the v_i and w_i are known. So, if we know the coordinates (x_i, y_i) of three points (say, $P_{k_1}, P_{k_2}, P_{k_3}$), then we can first recover the constants a, b, c, d from the preceding equation on indexes k_1, k_2, k_3. We can then use those constants to recover the X-coordinates of many other P_i.

The Y-coordinate of P_i can be recovered by collinearity conditions: If, for example, we have $z_{k_1} + z_{k_2} + z_i + z_{i'} + z_{i''} = 0$ with a curve of genus $g = 2$, we can deduce that a straight line passes through P_{k_1}, P_{k_2}, P_i, $P_{i'}$ and $P_{i''}$. Then, if we know the coordinates of $P_{k_1}, P_{k_2}, P_{k_3}$, we can recover x_i from the preceding equation, and thereafter, y_i is deduced from the alignment of P_{k_1}, P_{k_2}, P_i.

So the indices k_1, k_2, k_3 must be chosen carefully. We will need $z_{k_1}, z_{k_2}, z_{k_3}$ to define three different 5-points collinearity equations, involving indexes which are non-zero positions of the word $\mathbf{w}$. We will also have to check that the set of 12 points involved in those collinearity equations generates the group $\mathcal{G}$.

Once these indices are chosen, we can guess and try the coordinates (x, y) of points $P_{k_1}, P_{k_2}, P_{k_3}$: We arbitrarily choose the values of their 6 coordinates. Then, with this set of values, we determine the constants a, b, c, d from our evaluation equation. Then, by the use of the evaluation equation and collinearity equations, we are able to determine the coordinates of 9 points P_i on the curve. We now try to build a hyperelliptic curve of genus 2 passing through our 12 points (the 3 we guessed, and the 9 we computed). If such a curve exists, then with great probability, our coordinates guess is correct and we proceed to the step 3 of our cryptanalysis. If such a curve does not exist, it means that our coordinates guess is wrong, and we try a new set of values $(x_{k_1}, y_{k_1}, x_{k_2}, y_{k_2}, x_{k_3}, y_{k_3})$.

Actually, we don't have many guesses to make in order to recover a valid curve. As we have seen precedently, if we arbitrarily choose $(x_{k_1}, y_{k_1}, x_{k_2}, y_{k_2}, y_{k_3})$ and try all the values for $x_{k_3} \in \mathbb{F}_q$, we have a probability $1/2$ to obtain three points (P_1, P_2, P_3) such that there exists a curve isomorphism from $\mathcal{X}$ to $\mathcal{X}'$ which maps (P_1, P_2, P_3) to (P'_1, P'_2, P'_3). So, with q guesses, we have probability $1/2$ to find a curve and a set of points so that $\mathcal{C} = \mathrm{AGC}(\mathcal{X}, (k+g-1)\Delta_0, (P_1, \ldots, P_n), (c_1, \ldots, c_n))$.

Since $q \approx n$, and processing one guess takes constant time, the total cost of the guessing step without preprocessing is $O(n)$ multiplications over the base field. The preprocessing, which consists of finding the words **v**, **w** is $O(n^3)$ multiplications if a naive algorithm to find I is used.

4.5 Third and fourth step: recovering the remaining evaluation points and the distortion coefficients

We now know the equation of the hyperelliptic curve $\mathcal{X}$, along with the coordinates of a dozen points P_i on $\mathcal{X}$. We also know the values of all the $z_i = \varphi(\langle P_i \rangle)$ where φ is an unknown isomorphism.

The third step is then quite easy. For each P_i whose coordinates are still unknown, we write z_i as a sum of z_j corresponding to points whose coordinates are known. Computing the same sum with couples of points, we will find the coordinates of P_i. The value of the divisor Δ_0 will be computed the same way from the value of $\delta_0 = \varphi(\Delta_0 - \langle \mathcal{O} \rangle)$. The cost for computing the coordinates of one point is a constant, so the cost of this step is $O(n)$ multiplications over the base field.

When everything else is known, computing the distortion coefficients c_i is a simple linear algebra problem, which can be solved by a matrix inversion. The cost of this step is $O(n^3)$ multiplications, if we use a basic matrix inversion algortihm.

5 Conclusion

We have presented a polynomial time attack against a version of McEliece cryptosystem based on hyperelliptic codes of genus 2. As the first step of our algorithm has complexity $O(n^4)$ in the usual case, the complexity of the presented attack is $O(n^4)$ binary operations. Our attack is based on many probabilistic but reasonable assumptions, for example that the collected linear relations in step 1 behave like random relations modulo arbitrary integers.

Our attack is also restricted to the case where n is close to the number of rational points on the curve. We do not believe this to be a serious restriction. Ultimately, chosing small n is just one of many ways for a designer of a cryptosystem to trade efficiency for structural security, and quite possibly not the best one.

As it stands, the attack does not scale well with the genus. Indeed, for genus 3 the probability that the same attack works on a given instance is already quite low, even though certainly non-negligible. The fact, many of the steps of the attack work just fine also on these curves suggests that it is likely possible to devise almost-always working versions for these curves as well. It would be interesting to have a more thorough understanding of the fundamental limits of this kind of attack.

References

[1] H. Janwa, O. Moreno, McEliece public key cryptosystems using algebraic-geometric codes, *Des., Codes Crypt.* 8, 1996, 293-307.

[2] R. J. McEliece, A public key cryptosystem based on algebraic coding theory, DSN progress report 42-44, 1978, 114-116.

[3] L. Minder, Cryptography Based on Error Correcting codes Phd Thesis 3846, EPFL 2007,
`http://www.eecs.berkeley.edu/~lorenz/thesis.html`

[4] L. Minder, A. Shokrollahi, Cryptanalysis of the Sidelnikov cryptosystem, *Adv. Cryptology: Proc. EUROCRYPT 2007, LNCS.*

[5] N. Sendrier, On the structure of a randomly permuted concatenated code, *EUROCODE94*, October 1994.

[6] V. M. Sidelnikov, S. O. Shestakov, On insecurity of cryptosystems based on generalized Reed-Solomon codes, *Discr. Math. Appl.* 2, 1992, 439-444.

[7] S. G. Vlăduţ, On the decoding of algebraic-geometric codes over $\mathbb{F}_q$ for $q \geq 16$, *IEEE Trans. Inform. Theory* 36, 1990.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 108-118*

New systematic easy decoding symmetric rank codes

ERNST GABIDULIN[1] gab@mail.mipt.ru
Moscow Institute of Physics and Technology (State University), RUSSIA

Abstract. A family of rank-metric codes over binary fields with lengths $N_s = 2^s$, $s = 0, 1, \dots$, is constructed. Codes of length N_s are designed recursively from codes of length N_{s-1}. This provides very high degree of symmetry of code matrices. In turn, it allows to decode corrupted received matrices recursively starting with small lengths. The construction allows to use many simple algorithms for decoding in rank metric such as majority rules and similar.

1 Introduction

Rank-metric codes are of interest to communications, cryptography, space-time coding, network coding, etc., [1, 2, 4, 5, 6]. Symmetric rank-metric codes were introduced in [7] and investigated in [8]-[14]. Symmetry allows to simplify decoding and to correct some rank errors beyond the error capability bound. In this paper, we propose a recursive construction of rank codes over *binary* fields starting with length 2. The length is doubled at each step and is equal to $N_s = 2^s$ after step s. In matrix representation, code words are $N_s \times N_s$ matrices. They are constructed by means of $N_{s-1} \times N_{s-1}$ code matrices obtained at the previous step. This leads to very high degree of symmetry of code matrices. First, each code matrix of size $2^s \times 2^s$ is element wise symmetric. Second, if this matrix is represented as a $2^{s-1} \times 2^{s-1}$ block matrix consisting of blocks of size 2×2, then the matrix will be block wise symmetric for these blocks and all blocks are element wise symmetric. Further, if the original code matrix is represented as a $2^{s-2} \times 2^{s-2}$ block matrix with blocks of size $2^2 \times 2^2$, then the matrix will be block wise symmetric for these blocks and all $2^2 \times 2^2$ blocks are both element wise symmetric and 2×2 subblocks wise symmetric. Finally, represent the $2^s \times 2^s$ code matrix as 2×2 block matrix with four blocks of size $2^{s-1} \times 2^{s-1}$. Then the matrix will be block wise symmetric for these blocks. Moreover, each block element of the code matrix is in turn a symmetric matrix with the same properties.

For example, the binary code matrix for length $N_1 = 2$ has the form

$$V_1(x_1, x_2) = \begin{pmatrix} x_1 & x_2 \\ x_2 & x_1 + x_2 \end{pmatrix}, \tag{1.1}$$

where x_1 and x_2 are information bits. Each nonzero 2×2 code matrix has rank 2 and is symmetric.

[1] This work was partially supported under Grant 05-01-39017 GFEN-a

Code matrices of length $N_2 = 4$ constructed by our approach have the form

$$V_2(x_1, x_2, x_3, x_4) = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_1 + x_2 & x_4 & x_3 + x_4 \\ x_3 & x_4 & x_1 + x_4 & x_2 + x_3 + x_4 \\ x_4 & x_3 + x_4 & x_2 + x_3 + x_4 & x_1 + x_2 + x_3 \end{pmatrix} \quad (1.2)$$

$$= \begin{pmatrix} V_1(x_1, x_2) & V_1(x_3, x_4) \\ V_1(x_3, x_4) & V_1(x_1, x_2) + \Gamma_1 V_1(x_3, x_4) \end{pmatrix},$$

where x_1, x_2, x_3, x_4 are information bits. The matrix $\Gamma_1 = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right)$ provides a property that each nonzero 4×4 code matrix has rank 4 and is symmetric. It can be represented as a 2×2 block matrix with symmetric blocks $V_1(\cdot, \cdot)$ of size 2×2.

In general, if code matrices $V_{s-1}(x_1, \dots, x_{2^{s-1}})$ of length $N_{s-1} = 2^{s-1}$ are constructed, then code matrices $V_s(x_1, \dots, x_{2^{s-1}}, x_{2^{s-1}+1}, \dots, x_{2^s})$ of length $N_s = 2^s$ will have the form

$$V_s(x_1, \dots, x_{N_s}) = \begin{pmatrix} V_{s-1}(x_1, \dots, x_{N_{s-1}}) & V_{s-1}(x_{N_{s-1}+1}, \dots, x_{N_s}) \\ V_{s-1}(x_{N_{s-1}+1}, \dots, x_{N_s}) & V_{s-1}(x_1, \dots, x_{N_{s-1}}) + \Gamma_{s-1} V_{s-1}(x_{N_{s-1}+1}, \dots, x_{N_s}) \end{pmatrix}, \quad (1.3)$$

where $x_1, \dots, x_{N_{s-1}}, x_{N_{s-1}+1}, \dots, x_{N_s}$ are information bits. The matrix Γ_{s-1} of size $N_{s-1} \times N_{s-1}$ is calculated using the previous matrix Γ_{s-2}. It provides a property that each nonzero $N_s \times N_s$ code matrix has rank N_s and is symmetric.

We will exploit super symmetry to construct new decoding algorithms to correct rank and array errors.

2 Auxiliary results

2.1 Notations and definitions

Let F_2 be a base field and let F_{2^n} be an extension of degree n of F_2. Let $F_{2^n}^n$ be a normalized vector space of dimension n over F_{2^n}.

The *rank* norm of a vector $\mathbf{g} = (g_1, g_2, \dots, g_n)$, $\mathbf{g} \in F_{2^n}^n$, is defined as the *maximal number* of coordinates g_j which are linearly independent over F_2. We denote the rank norm of $\mathbf{g}$ by $r(\mathbf{g})$.

A *vector* code $\mathcal{V} \subset F_{q^n}^n$ is any set of vectors. A *linear* vector code $\mathcal{V}$ is a subspace of $F_{2^n}^n$.

Let $F_2^{n\times n}$ be a normalized space of square matrices of order n over F_q. The *rank* norm of a matrix $M \in F_2^{n\times n}$ is defined as ordinary rank of this matrix, i.e., the *maximal number* of rows (or, columns) which are linearly independent over F_2. We denote the rank norm of M as $\operatorname{rank}(M)$.

A *matrix* code $\mathcal{M} \subset F_2^{n\times n}$ is any set of binary matrices. A code $\mathcal{M}$ is said to be linear if $\mathcal{M}$ is subspace of $F_2^{n\times n}$. Given a code $\mathcal{M}$ one can construct a code $\mathcal{M}^T = \{M^T : M \in \mathcal{M}\}$ where M^T means the transpose of M. A code $\mathcal{M}$ is said to be symmetric if $\mathcal{M} = \mathcal{M}^T$.

2.2 Relations between vector rank-metric codes and matrix rank-metric codes

Let $\mathbf{g} = (g_1, g_2, \ldots, g_n)$, $g_j \in F_{2^n}$, be a basis of F_{2^n} over F_2. Then any vector $\mathbf{m} = (m_1, m_2, \ldots, m_n) \in F_n^n$ can be uniquely represented as

$$\mathbf{m} = (m_1, m_2, \ldots, m_n) = \mathbf{g}M = (g_1, g_2, \ldots, g_n)M,$$

where M is the $n \times n$-matrix in F_q. One refers to the matrix M as the matrix $\mathbf{g}$-representation of the vector $\mathbf{m}$. Note that $r(\mathbf{m}) = \operatorname{rank}(M)$.

Given a vector code $\mathcal{V}$ and a basis $\mathbf{g}$, one can get a corresponding matrix code $\mathcal{M}$ in $\mathbf{g}$-representation as $\mathcal{V} = \mathbf{g}\mathcal{M}$, and vice versa.

2.3 Self-orthogonal bases

Let

$$\mathbf{g} = (g_1, g_2, \ldots, g_n), \quad g_j \in F_{2^n}, \tag{2.4}$$

be a basis of F_{2^n} over F_2. Associate with the vector $\mathbf{g}$ the $n \times n$-matrix

$$\mathbf{G} = \begin{bmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^{[1]} & g_2^{[1]} & \cdots & g_n^{[1]} \\ g_1^{[2]} & g_2^{[2]} & \cdots & g_n^{[2]} \\ \ldots & \ldots & \ldots & \ldots \\ g_1^{[n-1]} & g_2^{[n-1]} & \cdots & g_n^{[n-1]} \end{bmatrix}. \tag{2.5}$$

We use the notation $[i] := 2^i$, if $i \geq 0$ and $[i] := 2^{n+i}$, if $i < 0$. It is known [15] that the matrix $\mathbf{G}_n$ is non singular.

Definition 1 *A basis $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ is called a **self-dual** basis if $Tr(g_i g_j) = \delta_{ij}$, where $Tr(\cdot)$ is the trace function of F_{2^n} into F_2 defined as $Tr(g) = g + g^{[1]} + g^{[2]} + \cdots + g^{[n-1]} \in F_2$, $g \in F_{2^n}$.*

Definition 2 (Equivalent) *A basis $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ is called a **self-dual** basis if*

$$\mathbf{G}^T\mathbf{G} = \mathbf{I}_n,$$

where $\mathbf{G}^T$ is the transpose of $\mathbf{G}$ and $\mathbf{I}_n$ is the identity matrix of order n.

Definition 3 *A basis* $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ *is called a* ***self-orthogonal*** *basis if*

$$\mathbf{G}\mathbf{G}^T = \mathbf{I}_n.$$

It is clear that a self-dual basis is also a self-orthogonal basis, and vice versa.

Definition 4 *A basis* $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ *is called a* ***weak self-orthogonal*** *basis if*

$$\mathbf{G}\mathbf{G}^T = \mathbf{B},$$

where $\mathbf{B}$ *is a diagonal matrix in* F_{2^n}*, but not multiple of the identity matrix* $\mathbf{I}_n$.

Note that a *weak* self-orthogonal basis is not a self-dual basis. For example, let $\mathbf{G} = \left(\begin{smallmatrix} 1 & \gamma \\ 1 & \gamma^2 \end{smallmatrix}\right)$, where γ is a primitive element of F_{2^2}. Then $\mathbf{G}\mathbf{G}^T = \left(\begin{smallmatrix} \gamma & 0 \\ 0 & \gamma^2 \end{smallmatrix}\right)$. Hence the basis $(1\ \gamma)$ is the weak self-orthogonal one. On the other hand we have $\mathbf{G}^T\mathbf{G} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. Hence the basis $(1\ \gamma)$ is *not* self-dual.

2.4 One-dimensional rank codes

Let $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ be a basis of F_{2^n} over F_2. We shall use this vector in two manner. First, it will be used to represent elements of the field F_{2^n}. An element $\gamma \in F_{2^n}$ is represented as $\gamma = x_1 g_1 + x_2 g_2 + \cdots + x_n g_n$, where coefficients $x_j \in F_2$ are called information bits of γ.

On the other hand, the vector $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ will be used as the generator vector of a linear $[n, 1, d = n]$ rank-metric vector code $\mathcal{V}_1$. The code $\mathcal{V}_1$ consists of the all zero vector $\mathbf{0} = (0, 0, \ldots, 0)$ and code vectors $\{\mathbf{g}_s = \alpha^s (g_1, g_2, \ldots, g_n),\ s = 0, 1, \ldots, 2^n - 2\}$, where α is a primitive element of F_{2^n}. In terms of the primitive element α the vector $\mathbf{g}$ can be rewritten as $\mathbf{g} = (\alpha^{i_1}, \alpha^{i_2}, \ldots, \alpha^{i_n})$, where $i_1, i_2, \ldots, i_n$ are some integers.

Find the matrix representation $\mathcal{M}_1$ of the vector code $\mathcal{V}_1$. Consider the matrix representation of the vector $\alpha\mathbf{g}$:

$$\alpha\mathbf{g} = \mathbf{g}A, \tag{2.6}$$

where A is the $(n \times n)$-matrix in F_2. It follows, that α is an eigenvalue and $\mathbf{g}$ is an eigenvector of A. Hence, A has as the characteristic polynomial a monic primitive polynomial of degree n over F_2. Moreover, all non-zero code vectors are given by

$$\alpha^s\mathbf{g} = \mathbf{g}A^s, \quad s = 0, 1, \ldots, 2^n - 2. \tag{2.7}$$

Therefore the rank-metric matrix code $\mathcal{M}_1$ consists of the all zero matrix $\mathbf{O}$ and code matrices $\{A^s\ s = 0, 1, \ldots, 2^n - 2\}$.

If an element $\gamma = x_1 g_1 + x_2 g_2 + \cdots + x_n g_n$, then the corresponding code matrix is

$$M(\gamma) = x_1 A^{i_1} + x_2 A^{i_2} + \cdots + x_n A^{i_n}.$$

Let $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ be a (*weak*) *self-orthogonal basis* of F_{2^n} over F_2. Then the matrix A defined above is the symmetric matrix (see, [12]).

2.5 A recursive construction of a weak self-orthogonal basis – the vector representation

As mentioned before, a weak self-orthogonal basis provides the symmetry of the matrix A. Let $N_s = 2^s$, $q_s = 2^{N_s}$, $s = 1, 2, \ldots$. We construct sequentially bases for the fields $F_{q_2} \subset F_{q_3} \subset \cdots \subset F_{q_s}$. Assume that the weak self-orthogonal basis is already constructed for the field F_{q_s}:

$$\mathbf{g}(N_s) = (g_1, g_2, \ldots, g_{N_s}) \tag{2.8}$$

Choose in the superfield $F_{q_{s+1}}$ an element f_{N_s+1} of order $q_s + 1$. Construct the vector

$$\mathbf{g}(N_{s+1}) = (g_1, g_2, \ldots, g_{N_s}, g_{N_s+1}, g_{N_s+2}, \ldots, g_{N_{s+1}}), \tag{2.9}$$

where $(g_{N_s+1}, g_{N_s+2}, \ldots, g_{N_{s+1}}) = (f_{N_s+1} g_1, f_{N_s+1} g_2, \ldots, f_{N_s+1} g_{N_s})$.

Lemma 1 *The vector* $\mathbf{g}(N_{s+1})$ *is a weak self-orthogonal basis for the field* $F_{q_{s+1}}$.

Proof. Let $\mathbf{G}(N_s)$ be the associated matrix of the vector $\mathbf{g}(N_s)$:

$$\mathbf{G}(N_s) = \begin{bmatrix} g_1 & g_2 & \cdots & g_{N_s} \\ g_1^{[1]} & g_2^{[1]} & \cdots & g_{N_s}^{[1]} \\ g_1^{[2]} & g_2^{[2]} & \cdots & g_{N_s}^{[2]} \\ \cdots & \cdots & \cdots & \cdots \\ g_1^{[N_s-1]} & g_2^{[N_s-1]} & \cdots & g_{N_s}^{[N_s-1]} \end{bmatrix}.$$

We have $\mathbf{G}(N_s)\mathbf{G}(N_s)^T = \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix.

It is easy to show that the associated matrix $\mathbf{G}(N_{s+1})$ of the vector $\mathbf{g}(N_{s+1})$ is of the form

$$\mathbf{G}(N_{s+1}) = \begin{bmatrix} \mathbf{G}(N_s) & \mathbf{F}\mathbf{G}(N_s) \\ \mathbf{G}(N_s) & \mathbf{F}^{q_s}\mathbf{G}(N_s) \end{bmatrix}, \tag{2.10}$$

where $\mathbf{F} = \operatorname{diag}[f_{N_s+1}, f_{N_s+1}^{[1]}, \ldots, f_{N_s+1}^{[N_s-1]}]$ is the diagonal matrix. Note that $\mathbf{F}^{q_s+1} = \mathbf{I}_{N_s}$.

Calculate the product

$$\begin{aligned}
\mathbf{G}(N_{s+1})\mathbf{G}(N_{s+1})^T &= \begin{bmatrix} \mathbf{G}(N_s)\mathbf{G}(N_s)^T+\mathbf{F}\mathbf{G}(N_s)\mathbf{G}(N_s)^T\mathbf{F} & \mathbf{G}(N_s)\mathbf{G}(N_s)^T+\mathbf{F}\mathbf{G}(N_s)\mathbf{G}(N_s)^T\mathbf{F}^{q_s} \\ \mathbf{G}(N_s)\mathbf{G}(N_s)^T+\mathbf{F}^{q_s}\mathbf{G}(N_s)\mathbf{G}(N_s)^T\mathbf{F} & \mathbf{G}(N_s)\mathbf{G}(N_s)^T+\mathbf{F}^{q_s}\mathbf{G}(N_s)\mathbf{G}(N_s)^T\mathbf{F}^{2^{2^s}} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{\Lambda}(\mathbf{I}_{N_s}+\mathbf{F}^2) & \mathbf{\Lambda}(\mathbf{I}_{N_s}+\mathbf{F}^{q_s+1}) \\ \mathbf{\Lambda}(\mathbf{I}_{N_s}+\mathbf{F}^{q_s+1}) & \mathbf{\Lambda}(\mathbf{I}_{N_s}+\mathbf{F}^{q_s+1}) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{\Lambda}(\mathbf{I}_{N_s}+\mathbf{F}^2) & \mathbf{O}_{N_s} \\ \mathbf{O}_{N_s} & \mathbf{\Lambda}(\mathbf{I}_{N_s}+\mathbf{F}^{q_s+1}) \end{bmatrix}.
\end{aligned} \tag{2.11}$$

This matrix is diagonal. Therefore the basis $\mathbf{g}(N_{s+1})$ is the weak self-orthogonal basis. □

We have to choose an element $f_{N_s+1} \in F_{q_{s+1}}$ of order q_s+1. Consider the last component $g_{N_s} \in F_{q_s} \subset F_{q_{s+1}}$ of the basis $\mathbf{g}(N_s)$. Assume that $\mathrm{Tr}_{F_{q_s}}(g_{N_s}) = 1$. Consider the polynomial $f_s(x) = x^2 + xg_{N_s}^m + 1$, where $m = 2^{N_s-1} - 1$.

Lemma 2 *The polynomial $f_s(x)$ is irreducible over the field F_{q_s}. Hence its roots belong to the field $F_{q_{s+1}}$. Moreover, the order of roots is q_s+1.*

Proof. Consider the polynomial $r(x) = f_s(xg_{N_s}^m) = g_{N_s}^{2m}(x^2 + x + g_{N_s}^{-2m}) = g_{N_s}^{2m}(x^2+x+g_{N_s})$. This polynomial is irreducible over F_{q_s} because $\mathrm{Tr}_{F_{q_s}}(g_{N_s}) = 1$. So is the polynomial $f_s(x)$. Further, by f_{N_s+1} denote a root of $f_s(x)$. Another root is $f_{N_s+1}^{q_s}$. We have by Viète theorem $f_{N_s+1} \cdot f_{N_s+1}^{q_s} = f_{N_s+1}^{q_s+1} = 1$, or, $\mathrm{ord}(f_{N_s+1}) = q_s + 1$. □

By construction, the last component of the basis $\mathbf{g}(N_{s+1})$ is $g_{N_{s+1}} = f_{N_s+1}g_{N_s}$.

Lemma 3 $\mathrm{Tr}_{F_{q_{s+1}}}(g_{N_{s+1}}) = 1$.

Proof. By definition, we have

$$f_{N_s+1}^2 + f_{N_s+1}g_{N_s}^m + 1 = 0, \tag{2.12}$$

where $m = 2^{N_s-1} - 1$. Multiply this equation by $g_{N_s}^2$. We obtain

$$g_{N_{s+1}}^2 + g_{N_{s+1}}g_{N_s}^{2^{N_s-1}} + g_{N_s}^2 = 0. \tag{2.13}$$

By Viète theorem, $g_{N_{s+1}} + g_{N_{s+1}}^{q_s} = g_{N_s}^{2^{N_s-1}}$. Hence $\mathrm{Tr}_{F_{q_s}}(g_{N_{s+1}} + g_{N_{s+1}}^{q_s}) = \mathrm{Tr}_{F_{q_s}}(g_{N_s}^{2^{N_s-1}}) = \mathrm{Tr}_{F_{q_s}}(g_{N_s}) = 1$. On the other hand,

$$\mathrm{Tr}_{F_{q_s}}(g_{N_{s+1}} + g_{N_{s+1}}^{q_s}) = \sum_{i=0}^{N_s-1}(g_{N_{s+1}} + g_{N_{s+1}}^{q_s})^{2^i} = \sum_{i=0}^{N_{s+1}-1} g_{N_{s+1}}^{2^i} = \mathrm{Tr}_{F_{q_{s+1}}}(g_{N_{s+1}}).$$

□

Example 1 *For $s = 1,\ N_1 = 2$, a weak self-orthogonal basis is*

$$\mathbf{g}(N_1) = (g_1, g_2) = (1, g_2), \tag{2.14}$$

where g_2 is a root of the polynomial $f(x) = x^2 + x + 1$.

For $s = 2,\ N_2 = 4$, a weak self-orthogonal basis is

$$\mathbf{g}(N_2) = (g_1, g_2, g_3, g_4) = (1, g_2, f_3, f_3 g_2), \tag{2.15}$$

where $g_3 = f_3$ is a root of the polynomial $f_1(x) = x^2 + xg_2 + 1$ and $\mathrm{Tr}_{F_{q_2}}(g_4) = \mathrm{Tr}_{F_{q_2}}(f_3 g_2) = 1$.

For $s = 3,\ N_3 = 8$, a weak self-orthogonal basis is

$$\mathbf{g}(N_3) = (g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8) = (1, g_2, g_3, g_4, f_5, f_5 g_2, f_5 g_3, f_5 g_4), \tag{2.16}$$

where $g_5 = f_5$ is a root of the polynomial $f_2(x) = x^2 + xg_4^7 + 1$ and $\mathrm{Tr}_{F_{q_3}}(g_8) = \mathrm{Tr}_{F_{q_3}}(f_5 g_4) = 1$.

2.6 A recursive construction of a weak self-orthogonal basis – the matrix representation

The matrix representation can be obtained from the vector representation if we replace elements g_j in the basis by suitable matrices. Note that if an element $\beta \in F_{q_s}$ is represented as a $N_s \times N_s$ matrix B over the base field F_2, then being considered as an element of the superfield $F_{q_{s+1}}$ its representation will be a block-diagonal $N_{s+1} \times N_{s+1}$ matrix $\left[\begin{smallmatrix} B & O \\ O & B \end{smallmatrix}\right]$.

Example 2 *For $s = 1,\ N_1 = 2$, the vector basis (2.14) is replaced by the matrix basis*

$$I_{N_1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, G_2(N_1) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}. \tag{2.17}$$

The corresponding code matrix is given by Eq. (1.1).

For $s = 2,\ N_2 = 4$, the vector basis (2.15) is replaced by the matrix basis

$$\begin{aligned} I_{N_2} &= \left[\begin{smallmatrix} I_{N_1} & O_{N_1} \\ O_{N_1} & I_{N_1} \end{smallmatrix}\right], G_2(N_2) = \left[\begin{smallmatrix} G_2(N_1) & O_{N_1} \\ O_{N_1} & G_2(N_1) \end{smallmatrix}\right], \\ G_3(N_2) &= \left[\begin{smallmatrix} O_{N_1} & I_{N_1} \\ I_{N_1} & G_2(N_1) \end{smallmatrix}\right], G_4(N_2) = \left[\begin{smallmatrix} O_{N_1} & G_2(N_1) \\ G_2(N_1) & G_2(N_1)^2 \end{smallmatrix}\right]. \end{aligned} \tag{2.18}$$

The corresponding code matrix is given by (1.2).

For $s = 3$, $N_3 = 8$, the vector basis (2.16) is replaced by the matrix basis

$$\begin{array}{l} I_{N_3} = \left[\begin{smallmatrix} I_{N_2} & O_{N_2} \\ O_{N_2} & I_{N_2} \end{smallmatrix}\right], G_2(N_3) = \left[\begin{smallmatrix} G_2(N_2) & O_{N_2} \\ O_{N_2} & G_2(N_2) \end{smallmatrix}\right], \\ G_3(N_3) = \left[\begin{smallmatrix} G_3(N_2) & O_{N_2} \\ O_{N_2} & G_3(N_2) \end{smallmatrix}\right], G_4(N_3) = \left[\begin{smallmatrix} G_4(N_2) & O_{N_2} \\ O_{N_2} & G_4(N_2) \end{smallmatrix}\right], \\ G_5(N_3) = \left[\begin{smallmatrix} O_{N_2} & I_{N_2} \\ I_{N_2} & G_4(N_2)^7 \end{smallmatrix}\right], G_6(N_3) = \left[\begin{smallmatrix} O_{N_2} & G_2(N_2) \\ G_2(N_2) & G_4(N_2)^7 G_2(N_2) \end{smallmatrix}\right], \\ G_7(N_3) = \left[\begin{smallmatrix} O_{N_2} & G_3(N_2) \\ G_3(N_2) & G_4(N_2)^7 G_3(N_2) \end{smallmatrix}\right], G_8(N_3) = \left[\begin{smallmatrix} O_{N_2} & G_4(N_2) \\ G_4(N_2) & G_4(N_2)^8 \end{smallmatrix}\right] \end{array} . \tag{2.19}$$

The corresponding code matrix is given by

$$V_3(x_1, \ldots, x_8) = \begin{pmatrix} V_2(x_1, \ldots, x_4) & V_2(x_5, \ldots, x_8) \\ V_2(x_5, \ldots, x_8) & V_2(x_1, \ldots, x_4) + \Gamma_2 V_2(x_5, \ldots, x_8) \end{pmatrix}, \tag{2.20}$$

where

$$\Gamma = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

3 Decoding super-symmetric rank-metric codes

Here we consider decoding one-dimensional rank-metric matrix codes. Let $V_s(x_1, \ldots, x_{N_s})$ be a code matrix of rank N_s and E is an error matrix of size $N_s \times N_s$ over F_2. If a received matrix is $Y = V_s(x_1, \ldots, x_{N_s}) + E$ and $\text{rank}(E) = t \leq N_{s-1} - 1$, then standard methods (see, [1] and others) allow to correct all such errors.

On the other hand, use of Eq. (1.3) and represent E as $\left(\begin{smallmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{smallmatrix}\right)$. Then

$$.Y = \begin{pmatrix} V_{s-1}(x_1 \ldots x_{N_{s-1}}) + E_{11} & V_{s-1}(x_{N_{s-1}+1} \ldots x_{N_s}) + E_{12} \\ V_{s-1}(x_{N_{s-1}+1} \ldots x_{N_s}) + E_{21} & V_{s-1}(x_1 \ldots x_{N_{s-1}}) + \Gamma_{s-1} V_{s-1}(x_{N_{s-1}+1} \ldots x_{N_s}) + E_{22} \end{pmatrix}. \tag{3.21}$$

One can see that decoding the $N_s \times N_s$ code matrices can be reduced to decoding several code matrices of order $N_{s-1} = N_s/2$. Namely, we have to decode the code submatrix $V_{s-1}(x_1 \ldots x_{N_{s-1}})$ depending only on half information variables $x_1, \ldots, x_{N_{s-1}}$. It satisfies conditions from Eq. (3.21):

$$\begin{aligned} V_{s-1}(x_1 \ldots x_{N_{s-1}}) + E_{11} &= Y_{11}, \\ V_{s-1}(x_1 \ldots x_{N_{s-1}}) + E_{22} + \Gamma_{s-1} E_{12} &= Y_{22} + \Gamma_{s-1} Y_{12}, \\ V_{s-1}(x_1 \ldots x_{N_{s-1}}) + E_{22} + \Gamma_{s-1} E_{21} &= Y_{22} + \Gamma_{s-1} Y_{21}. \end{aligned} \tag{3.22}$$

Similarly, the code submatrix $V_{s-1}(x_{N_{s-1}+1}\dots x_{N_s})$ satisfies conditions

$$\begin{array}{rcl} V_{s-1}(x_{N_{s-1}+1}\dots x_{N_s})+E_{12} & = & Y_{12}, \\ V_{s-1}(x_{N_{s-1}+1}\dots x_{N_s})+E_{21} & = & Y_{21}, \\ V_{s-1}(x_{N_{s-1}+1}\dots x_{N_s})+E_{11}+\Gamma_{s-1}^{-1}E_{22} & = & Y_{11}+\Gamma_{s-1}^{-1}Y_{22}. \end{array} \tag{3.23}$$

If $\min\{\mathrm{rank}(E_{12}),\mathrm{rank}(E_{21}),\mathrm{rank}(E_{22}+\Gamma_{s-1}E_{21})\}\le N_{s-2}-1$ and $\min\{\mathrm{rank}(E_{11}),\mathrm{rank}(E_{22}+\Gamma_{s-1}E_{12}),\mathrm{rank}(E_{11}+\Gamma_{s-1}^{-1}E_{22})\}\le N_{s-2}-1$, then decoding will be successful.

Note that $\mathrm{rank}(E)$ of the original error matrix may be greater than $N_{s-1}-1$. Hence the symmetry of a code matrix $V_s(x_1\dots x_{N_s})$ allows to correct many rank errors beyond the one half distance bound. For example, the code $V_3(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8)$ has rank distance 8 and can correct all rank errors up to rank 3. The error matrix

$$E=\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&1&0&0&0&0&1&0\\ 0&0&1&0&0&0&0&1\\ 0&0&0&1&0&0&0&0 \end{pmatrix}$$

has rank 6 and can not be corrected by general fast algorithms. But Eq.'s (3.22) and (3.23) allow to correct this error. On the other hand, the error matrix

$$E=\begin{pmatrix} 1&0&0&0&1&0&0&0\\ 0&1&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 1&0&0&0&1&0&0&0\\ 0&1&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0 \end{pmatrix}$$

has rank 3 and can be corrected by general fast algorithms. But Eq.'s (3.22) and (3.23) do not allow to correct this error. Therefore general algorithms and symmetry algorithms should be used in common: first a general algorithm but if it failes use a symmetry algorithm.

The proposed approach can be iterated until we get the best conditions from the point of view of complexity.

4 Conclusion

We proposed one-dimensional rank-metric matrix codes generated by weak self-orthogonal bases. These codes allow to correct not only all errors of rank not greater than $\lfloor (d-1)/2 \rfloor$ but also many specific (namely, symmetric) errors beyond this bound.

References

[1] E. M. Gabidulin, Theory of codes with maximum rank distance, *Probl. Inform. Transm.* 21, 1985, 3-14.

[2] E. M. Gabidulin, A. V. Paramonov, O. V. Tretjakov, Ideals over a non-commutative ring and their application in cryptology, *Lect. Notes Comp. Sci.* 547, Adv. Crypt., Proc. Eurocrypt91, Brighton, UK, 1991, 482-489.

[3] E. M. Gabidulin, A fast matrix decoding algorithm for rank-error-correcting codes, (Eds G. Cohen, S. Litsyn, A. Lobstein, G. Zemor), *Lect. Notes Comp. Sci.* 573, Alg. Coding, Springer-Verlag, Berlin, 1992, 126-132.

[4] E. M. Gabidulin, M. Bossert, P. Lusina, Space-time codes based on rank codes, *Proc. IEEE Intern. Symp. Inform. Theory*, 2000, Sorrento, Italy, 283.

[5] R. Koetter, F. R. Kschischang, Coding for errors and erasures in random network coding, *Proc. IEEE Intern. Symp. Inform. Theory*, Nice, France, 2007, 791-795.

[6] E. M. Gabidulin, N. I. Pilipchuk, Error and erasure correcting algorithms for rank codes, *Des., Codes Crypt.*, Springer Netherlands, DOI 10.1007/s10623-008-9185-7. Online: 11 March 2008.

[7] E. M. Gabidulin, N. I. Pilipchuk, Representation of a finite field by symmetric matrices and applications, *Proc. Eighth Intern. Workshop ACCT*, 2002, Tsarskoe Selo, Russia, 120-123.

[8] E. M. Gabidulin, N. I. Pilipchuk, Transposed rank codes based on symmetric matrices, *Proc. WCC2003*, 2003, Versailles (France), 203-211.

[9] E. M. Gabidulin, N. I. Pilipchuk, A new method of erasure correction by rank codes, *Proc. IEEE Intern. Symp. Inform. Theory*, Yokohama, Japan, 2003, 423.

[10] E. M. Gabidulin, N. I. Pilipchuk, Symmetric rank codes, *Probl. Inform. Transm.* 40, 2004, 3-17.

[11] E. M. Gabidulin, N. I. Pilipchuk, Correcting of rank erasures by symmetrization and information sets, *Proc. Ninth Intern. Workshop ACCT*, 2004, Kranevo, Bulgaria, 333-337.

[12] E. M. Gabidulin, N. I. Pilipchuk, Symmetric matrices and codes correcting rank errors beyond the $\lfloor \frac{d-1}{2} \rfloor$ bound, *Discr. Appl. Math.* 154, 2006, 305-312.

[13] A. Kshevetskiy, Information set decoding for codes in rank metric, *Proc. Ninth Intern. Workshop ACCT*, 2004, Kranevo, Bulgaria, 254-259.

[14] N. I. Pilipchuk, E. M. Gabidulin, Decoding of symmetric rank codes by information sets, *Proc. Tenth Intern. Workshop ACCT*, 2006, Zvenigorod, Russia, 214-219.

[15] F. J. MacWilliams, N. J. A. Sloane, The Theory of Error Correcting Codes, 8th ed, North Holland Press, Amsterdam, 1993.

On permutation automorphism groups of q-ary Hamming codes

Evgeny V. Gorkunov evgumin@gmail.com
Novosibirsk State University, RUSSIA

Abstract. It is established that for any $q > 2$ the permutation automorphism group of a q-ary Hamming code of length $n = (q^m - 1)/(q - 1)$ is isomorphic to the unitriangular group $\mathbf{UT}_m(q)$.

1 Introduction

Let $\mathbb{F}_q^n$ be a vector space of dimension n over $GF(q)$ where q is a prime power. In contrast to the traditional code automorphism group definitions considered in [1,2], all transformations of the space $\mathbb{F}_q^n$ are taken into consideration in the papers [3–8]. In this paper, following the approach started in [3–8] we prove that the permutation automorphism group of a q-ary Hamming code of length $n = (q^m - 1)/(q - 1)$ is isomorphic to the unitriangular group $\mathbf{UT}_m(q)$.

The study of codes automorphism groups is an important topic in the theory of error-correcting codes. Almost all obtained results on the topic concern binary codes. Phelps in [9] established that every finite group is isomorphic to the full permutation automorphism group of some perfect binary code. Unfortunately, the result does not elucidate the structure of the full automorphism group of the code. It is proved in [4,5] that there exist perfect binary codes with trivial automorphism groups. The permutation automorphism group of well-known Vasil'ev code was investigated in the paper [8].

It is well known (see [1]) that the permutation automorphism group of the binary Hamming code $\mathcal{H}_2^n$ of length $n = 2^m - 1$ is isomorphic to the general linear group $\mathbf{GL}_m(2)$. Solov'eva and Topalova (see [6]) showed that the order of the automorphism group of an arbitrary perfect binary code is not greater than the order of the automorphism group of the Hamming code with the same length. In addition, these authors in [7] established that the only perfect binary code that has an automorphism group of maximal order within all perfect binary codes of the same length is the Hamming code. A similar result was independently obtained by Malyugin in [10]. Semilinear automorphisms of a q-ary Hamming code that preserve the Hamming weight are investigated in [2, Sec. 7].

The Hamming distance $d(x, y)$ between vectors $x, y \in \mathbb{F}_q^n$ is the number of coordinates where x and y differ. Any subset C of the space $\mathbb{F}_q^n$ is a *q-ary code* of length n. If for some $e \geq 0$ every $x \in \mathbb{F}_q^n$ is within the distance e from exactly one codeword of C, then the code C is called *e-perfect* (in the sequel simply *perfect*). It is well known (see [1]) that nontrivial perfect codes over $\mathbb{F}_q$ must have length $n = (q^m - 1)/(q - 1)$ for some integer $m \geq 2$ and cardinality q^{n-m}.

A code is *linear* if it is a subspace of $\mathbb{F}_q^n$. The Hamming codes are the only linear perfect codes. However Lindström (see [11]) presented group perfect codes nonequivalent to any linear code.

2 Definitions of codes automorphism groups

A mapping $\varphi\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$ is called an *isometry* of the space $\mathbb{F}_q^n$ if for any two vectors $x, y \in \mathbb{F}_q^n$ the following equality holds: $d(x, y) = d(\varphi(x), \varphi(y))$.

Suppose $\pi \in S_n$, where S_n is the symmetric group on n elements of the ground set $\{1, 2, \ldots, n\}$. The action of the permutation π on any vector $x = (x_1, \ldots, x_n)$ from $\mathbb{F}_q^n$ is defined by

$$\pi(x) = \left(x_{\pi^{-1}(1)}, \ldots, x_{\pi^{-1}(n)}\right).$$

Following [3] by a *configuration* we call an isometry $\sigma\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$ such that

$$\sigma(x) = (\sigma_1(x_1), \ldots, \sigma_n(x_n)),$$

where σ_i are permutations from the symmetric group S_q acting on the field $\mathbb{F}_q$.

It is widely known (see, e.g., [12–14]) that the automorphism group of the space $\mathbb{F}_q^n$ is a semidirect product of the group S_n on the group S_q^n of all configurations, i.e.

$$\mathrm{Aut}(\mathbb{F}_q^n) = S_n \leftthreetimes S_q^n = \{(\pi; \sigma)\colon \pi \in S_n,\ \sigma = (\sigma_1, \ldots, \sigma_n) \in S_q^n\}.$$

The group of all isometries of $\mathbb{F}_q^n$ mapping a code C into itself is called the *automorphism group* of the code C:

$$\mathrm{Aut}(C) = \{(\pi; \sigma) \in \mathrm{Aut}(\mathbb{F}_q^n)\colon (\pi; \sigma)(C) = C\}.$$

It is should be noted that the q-ary code automorphism group definition given in [2] takes into account the only semilinear mappings preserving the Hamming weight of codewords.

Multiplying all elements of the field $\mathbb{F}_q$ by some nonzero element $\beta \in \mathbb{F}_q$ we get the permutation τ_β from S_q:

$$\tau_\beta = \begin{pmatrix} 0 & \alpha^0 & \alpha^1 & \ldots & \alpha^{q-2} \\ 0 & \alpha^0\beta & \alpha^1\beta & \ldots & \alpha^{q-2}\beta \end{pmatrix}.$$

By S_q^* we denote the set of all $q-1$ such permutations. Define the *monomial automorphism group* of a code C as

$$\mathrm{MAut}(C) = \{(\pi;\sigma) \in \mathrm{Aut}(C) \colon \sigma \in (S_q^*)^n\}.$$

Let ε be the identity configuration, i.e. all its components are the identity permutations. It is natural to identify the isometry $(\pi;\varepsilon)$ with the permutation π. Define the *permutation automorphism group* of a code C as

$$\mathrm{PAut}(C) = \{\pi \in \mathrm{Aut}(C)\}.$$

3 The group $\mathrm{PAut}(\mathcal{H}_q^n)$

In this section we are going to prove that for any $q > 2$ the permutation automorphism group of a q-ary Hamming code of length n is isomorphic to the unitriangular group $\mathbf{UT}_m(q)$ where $n = (q^m-1)/(q-1)$. Let us start with the definitions of some groups of matrices over $\mathbb{F}_q$. The *general linear group* consists of all nonsingular $m \times m$ matrices and is denoted by $\mathbf{GL}_m(q)$. The set of $m \times m$ matrices with units on the main diagonal and zeros above (under) the diagonal is called the *lower (upper) unitriangular group.* Both these groups are isomorphic to each other. The map taking each lower unitriangular matrix L to the upper unitriangular matrix $R = L^{-\mathrm{T}}$ is an isomorphism between these two groups. Taking that into account we will further denote the groups by $\mathbf{UT}_m(q)$.

The parity check matrix H_m of the q-ary Hamming code $\mathcal{H}_q^n$ of length $n = (q^m-1)/(q-1)$ consists of n pairwise linear independent column vectors from $\mathbb{F}_q^m$. In the sequel we will use the parity check matrix H_m given in the following way. Consider all nonzero vectors of length m that have 1 as their first nonzero coordinate. Let α be a primitive element of $\mathbb{F}_q$. In the case $m=2$ we have

$$H_2 = \begin{bmatrix} 0 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & \alpha^0 & \alpha^1 & \dots & \alpha^{q-2} \end{bmatrix}.$$

Let for any m we have $H_m = \begin{bmatrix} h_1 & h_2 & \dots & h_n \end{bmatrix}$. Then H_{m+1} can be defined by

$$H_{m+1} = \begin{bmatrix} \mathbf{0} & h_1 & h_1 & h_1 & \dots & h_1 & \dots & h_n & h_n & h_n & \dots & h_n \\ 1 & 0 & \alpha^0 & \alpha^1 & \dots & \alpha^{q-2} & \dots & 0 & \alpha^0 & \alpha^1 & \dots & \alpha^{q-2} \end{bmatrix},$$

here $\mathbf{0}$ is the all-zero vector of length m. Let T_m denote the column set of the matrix H_m. If $K \in \mathbf{GL}_m(q)$, then the multiplication $y = Kx$ gives a linear mapping on $\mathbb{F}_q^m$. It is not difficult to prove the following

Lemma 1. *Any matrix $L \in \mathbf{UT}_m(q)$ gives a bijection on the set T_m.*

Note that the linear map mentioned above is a bijection on T_m if the matrix L is *lower* unitriangular (in opposite to an upper unitriangular matrix U in the rule $y = xU$). In the following lemma we will show that in the group $\mathbf{GL}_m(q)$ there are no bijections acting on the set T_m besides those described in Lemma 1.

Lemma 2. *If a matrix U belongs to $\mathbf{GL}_m(q) \setminus \mathbf{UT}_m(q)$, where $m \geq 1, q > 2$, then in the set T_m there is a vector h such that $Uh \notin T_m$.*

Proof. We prove the statement by induction on m. Consider the Hamming code parity check matrix H_m multiplied on the left by a matrix U. For $m = 1$, there is nothing to prove since $UH_1 = [u_{11}][1] = [u_{11}]$, where $u_{11} \neq 0$ and $u_{11} \neq 1$.

Suppose the statement is true for matrices of order m. Now we prove it for a matrix U of order $m + 1$. A matrix U can be represented as follows

$$U = \begin{bmatrix} \widetilde{U} & b \\ c & \beta \end{bmatrix},$$

where $\widetilde{U}$ is a $m \times m$ submatrix, a column vector b and a row vector c have length m and $\beta \in \mathbb{F}_q$. We have

$$UH_{m+1} = \begin{bmatrix} b & \widetilde{U}h_1 & \widetilde{U}h_1 + \alpha^0 b & \dots & \widetilde{U}h_1 + \alpha^{q-2} b & \dots & \widetilde{U}h_n & \dots & \widetilde{U}h_n + \alpha^{q-2} b \\ \beta & ch_1 & ch_1 + \alpha^0 \beta & \dots & ch_1 + \alpha^{q-2}\beta & \dots & ch_n & \dots & ch_n + \alpha^{q-2}\beta \end{bmatrix}.$$

There are the following four possible cases to check.

1. If $\det \widetilde{U} \neq 0$ and $\widetilde{U} \notin \mathbf{UT}_m(q)$, then, by induction hypothesis, there is a vector $h_j \in T_m$ such that $\widetilde{U}h_j \notin T_m$. Hence,

$$U \begin{bmatrix} h_j \\ 0 \end{bmatrix} = \begin{bmatrix} \widetilde{U}h_j \\ ch_j \end{bmatrix} \notin T_{m+1} \quad \text{and therefore} \quad h = \begin{bmatrix} h_j \\ 0 \end{bmatrix}.$$

2. Let either $\det \widetilde{U} = 0$ or $\widetilde{U} \in \mathbf{UT}_m(q)$, and at the same time $b \neq \mathbf{0}$. In this case, the vector b is collinear with some vector of the set T_m. Hence we have $b = \gamma h_k$ for some $\gamma \in \mathbb{F}_q$ and $h_k \in T_m$.

If $\det \widetilde{U} = 0$, then there is a vector h_j in T_m such that $\widetilde{U}h_j = \mathbf{0}$.

On the other hand, if $\widetilde{U} \in \mathbf{UT}_m(q)$, then we can apply Lemma 1. Namely, in the set T_m there is a vector h_j that is assigned the vector h_k under the action of the matrix $\widetilde{U}$. So we have $\widetilde{U}h_j = h_k$.

Combining these two subcases we can conclude that the matrix UH_{m+1} has a submatrix of the form

$$\begin{bmatrix} \delta h_k & (\delta + \alpha^0\gamma)h_k & (\delta + \alpha^1\gamma)h_k & \dots & (\delta + \alpha^{q-2}\gamma)h_k \\ ch_j & ch_j + \alpha^0\beta & ch_j + \alpha^1\beta & \dots & ch_j + \alpha^{q-2}\beta \end{bmatrix},$$

where δ equals either 0 or 1 in accordance with the subcases considered above. Since the set $\{\delta,\, \delta+\alpha^0\gamma,\, \delta+\alpha^1\gamma, \dots, \delta+\alpha^{q-2}\gamma\}$ coincides with the set of all field elements, then for $q > 2$ one can find an integer l from $[0,\, q-2]$ such that $\delta+\alpha^l\gamma \neq 0$ and $\delta+\alpha^l\gamma \neq 1$. Hence,

$$U\begin{bmatrix} h_j \\ \alpha^l \end{bmatrix} = \begin{bmatrix} (\delta+\alpha^l\gamma)h_k \\ ch_j+\alpha^l\beta \end{bmatrix} \notin T_{m+1} \quad \text{and} \quad h = \begin{bmatrix} h_j \\ \alpha^l \end{bmatrix}.$$

3. If $\widetilde{U} \in \mathbf{UT}_m(q)$ and $b = \mathbf{0}$, then we have $\beta \neq 0$ for $\det U \neq 0$. In addition, we obtain $\beta \neq 1$ for $U \notin \mathbf{UT}_{m+1}(q)$. This implies that

$$U\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \beta \end{bmatrix} \notin T_{m+1} \quad \text{and therefore} \quad h = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}.$$

4. It should be noted that the conditions $\det \widetilde{U} = 0$ and $b = \mathbf{0}$ are not compatible since $\det U \neq 0$. □

Theorem 1. *For any $n = (q^m-1)/(q-1)$, where $m \geq 2, q > 2$, it is true that*

$$\mathrm{PAut}(\mathcal{H}_q^n) \cong \mathbf{UT}_m(q).$$

Proof. It is known (see, e.g., [2]) that the Hamming code monomial automorphism group is isomorphic to the general linear group, namely $\mathrm{MAut}(\mathcal{H}_q^n) \cong \mathbf{GL}_m(q)$. The isomorphism $\theta\colon \mathrm{MAut}(\mathcal{H}_q^n) \to \mathbf{GL}_m(q)$ can be defined by

$$\theta\colon M \mapsto K, \quad \text{where} \quad K^{\mathsf{T}} H_m = H_m M^{\mathsf{T}}.$$

Here H_m is the parity check matrix of the Hamming code $\mathcal{H}_q^n$, the matrix M is a monomial $n \times n$ matrix and $K \in \mathbf{GL}_m(q)$.

By Lemmas 1 and 2 we have $\theta(\mathrm{PAut}(\mathcal{H}_q^n)) = \mathbf{UT}_m(q)$. Therefore a restriction of the isomorphism θ on the permutation automorphism group $\varphi = \theta|_{\mathrm{PAut}(\mathcal{H}_q^n)}$ is an isomorphism between $\mathrm{PAut}(\mathcal{H}_q^n)$ and $\mathbf{UT}_m(q)$. This proves the theorem. □

The author is very grateful to professor Faina I. Solov'eva for constant attention to this work, useful discussions and significant improvements in appearance of the paper.

References

[1] F. J. MacWilliams, N. J. A. Sloane, *The THeory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.

[2] W. C. Huffman, Codes and groups, V. S. Pless, W. C. Huffman, eds., Handbook of coding theory. Amsterdam – New York: Elsevier Science 17,1998, 1345-1440.

[3] F. I. Solov'eva, S. V. Avgustinovich, T. Honold, W. Heise, On the extendability of code isometries, *J. Geom.* 61, 1998, 3-16.

[4] S. V. Avgustinovich, F. I. Solov'eva, Perfect binary codes with trivial automorphism group, *Proc. Intern. Workshop Inform. Theory*, Killarney, Ireland, 1998, 114-115.

[5] S. A. Malyugin, Perfect codes with trivial automorphism group, *Proc. Second Intern. Workshop OCRT*, Sozopol, Bulgaria, 1998, 163-167.

[6] F. I. Solov'eva, S. T. Topalova, On the automorphism groups of perfect binary codes and Steiner triple systems, *Probl. Inform. Transm.* 36, 2000, 331-335.

[7] F. I. Solov'eva, S. T. Topalova, Perfect binary codes and Steiner triple systems with automorphism groups of maximal order, *Discr. Analysis Oper. Res.* 7, 2000, 101-110 (in Russian).

[8] S. V. Avgustinovich, F. I. Solov'eva, O. Heden, On the structure of symmetry groups of Vasil'ev codes, *Probl. Inform. Transm.* 41, 2005, 105-112.

[9] K. T. Phelps, Every finite group is the automorphism group of some perfect code, *J. Combin. Theory*, Ser. A, 43, 1986, 45-51.

[10] S. A. Malyugin, On the order of the automorphism group of perfect binary codes, *Discr. Analysis Oper. Res.*, 7, 2000, 91-100 (in Russian).

[11] B. Lindström, On group and nongroup perfect codes in q symbols, *Math. Scand.* 25, 1969, 149-158.

[12] A. V. Babash, M. M. Gluhov, G. P. Shankin, On transformations of sets of words over a finite alphabet which do not propagate errors, *Discr. Math. Appl.* 7, 1997, 437-454.

[13] I. Constantinescu, W. Heise, On the concept of code-isomorphy, *J. Geom.* 57, 1996, 63-69.

[14] P. M. Winkler, Isometric embeddings in products of complete graphs, *Discr. Appl. Math.* 7, 1984, 221-225.

Construction of a self-dual [94, 47, 16] code

Masaaki Harada
Department of Mathematical Sciences, Yamagata University,
Yamagata 990–8560, JAPAN

Radinka Yorgova[1] radinka@ii.uib.no
Department of Informatics, University of Bergen
Thormøhlensgate 55, N-5008, Bergen, NORWAY

Abstract. The existences of an extremal doubly even self-dual [96, 48, 20] code and a self-dual [94, 47, 18] code are equivalent. The largest minimum weight among self-dual codes of length 94 was previously known as 14, 16 or 18. In this note, a self-dual [94, 47, 16] code is constructed for the first time.

1 Introduction

A (binary) $[n, k]$ code C is a k-dimensional vector subspace of $\mathbb{F}_2^n$, where $\mathbb{F}_2$ is the field of two elements. An $[n, k, d]$ code is an $[n, k]$ code with minimum weight d. A code C is *self-dual* if $C = C^\perp$ where $C^\perp$ is the dual code of C. A self-dual code C is *doubly even* if all codewords of C have weight divisible by four, and *singly even* if there is at least one codeword of weight $\equiv 2 \pmod 4$. Note that a doubly even self-dual code of length n exists if and only if n is divisible by eight. It was shown in [9] that the minimum weight d of a doubly even self-dual code of length n is bounded by $d \leq 4[n/24] + 4$. In [10] it is proved that the same bound is valid also for the minimum weight d of a singly even self-dual code of length n unless $n \equiv 22 \pmod{24}$ when $d \leq 4[n/24] + 6$ or $n \equiv 0 \pmod{24}$ when $d \leq 4[n/24] + 2$.

An extremal doubly even self-dual $[24k, 12k, 4k + 4]$ code is known for only $k = 1, 2$, namely, the extended Golay [24, 12, 8] code and the extended quadratic residue [48, 24, 12] code. It is not known if there exist other extremal doubly even self-dual codes of length $24k$. It was shown in [10] that the existences of an extremal doubly even self-dual $[24k, 12k, 4k + 4]$ code and a self-dual $[24k - 2, 12k - 1, 4k + 2]$ code are equivalent. From this viewpoint, it would be interesting to determine the largest minimum weight among self-dual codes of length $24k - 2$. The largest minimum weight among self-dual codes of length 70 is known as 12 or 14, and the largest minimum weight among self-dual codes of length 94 was previously known as 14, 16 or 18 (see [4, Table VI], [6, Table 2]).

In this note, a self-dual [94, 47, 16] code is constructed for the first time. Hence the largest minimum weight among self-dual codes of length 94 is 16 or 18.

2 A self-dual [94, 47, 16] code

2.1 Construction

An automorphism of C is a permutation of the coordinates of C which preserves C and the set consisting of all automorphisms of C forms a group called the automorphism group of C. Extremal doubly even self-dual codes with automorphisms of a fixed odd prime order have been widely investigated (see e.g., [8], [11]).

Suppose that σ is an automorphism of order 23 of a self-dual $[94, 47, 16]$ code. By [11, Theorem 1], one can show that σ consists of four 23-cycles together with two fixed points. Using the technique developed by Huffman [8] and Yorgov [11], we have found a self-dual $[94, 47, 16]$ code C_{94} with an automorphism of order 23. The code C_{94} has the following generator matrix:

$$\left(\begin{array}{cccc|cc}
\boldsymbol{a} & & \boldsymbol{a} & & & \\
 & \boldsymbol{a} & & & 1 & \\
 & & & \boldsymbol{a} & & 1 \\
\hline
e_1 & & e_2 & e_2 & & \\
 & e_1 & e_3 & e_4 & & \\
f_2 & f_3 & f_1 & & & \\
f_2 & f_4 & & f_1 & &
\end{array}\right),$$

where $\boldsymbol{a}$ is the all-one's vector of length 23, e_i $(i = 1, 2, 3, 4)$ and f_j $(j = 1, 2, 3, 4)$ are the 11×23 circulant matrices M with first rows r:

M	r	M	r
e_1	(10000101001100110101111)	e_2	(11010001001111110100100)
e_3	(10001110110000111010101)	e_4	(10001000010001010011100)
f_1	(11111010110011001010000)	f_2	(10010010111111001000101)
f_3	(11010101110000110111000)	f_4	(10011100101000100001000)

and the blanks are filled up with zero's.

Hence we have the following:

Proposition 1 *There is a self-dual* $[94, 47, 16]$ *code. The largest minimum weight among self-dual codes of length* 94 *is* 16 *or* 18.

Remark 2 *The largest minimum weight among known linear* $[94, 47]$ *codes is currently* 16 *(see [7]).*

2.2 Weight enumerators

Let C be a singly even self-dual code and let C_0 denote the subcode of codewords having weight $\equiv 0 \pmod 4$. Then C_0 is a subcode of codimension 1. The *shadow* S of C is defined to be $C_0^\perp \setminus C$ [2]. There are cosets C_1, C_2, C_3 of C_0 such that $C_0^\perp = C_0 \cup C_1 \cup C_2 \cup C_3$ where $C = C_0 \cup C_2$ and $S = C_1 \cup C_3$. Shadows are often used to provide restrictions on the weight enumerators of singly even self-dual codes.

By Theorem 5, 4) in [2], a self-dual $[94, 47, 16]$ code C and its shadow S have the following possible weight enumerators:

$$\begin{aligned}
W_C =& 1 + 2\alpha y^{16} + (134044 - 2\alpha + 128\beta) y^{18} \\
&+ (2010660 - 30\alpha - 896\beta + 8192\gamma) y^{20} \\
&+ (22385348 + 30\alpha + 1280\beta - 106496\gamma - 524288\delta) y^{22} \\
&+ (207307788 + 210\alpha + 5376\beta + 581632\gamma + 9961472\delta) y^{24} \\
&+ (1545393276 - 210\alpha - 18048\beta - 1597440\gamma - 88080384\delta) y^{26} + \cdots, \\
W_S =& \delta y^3 + (\gamma - 22\delta) y^7 + (-\beta - 20\gamma + 231\delta) y^{11} \\
&+ (\alpha + 18\beta + 190\gamma - 1540\delta) y^{15} \\
&+ (1072352 - 16\alpha - 153\beta - 1140\gamma + 7315\delta) y^{19} \\
&+ (140151744 + 120\alpha + 816\beta + 4845\gamma - 26334\delta) y^{23} + \cdots,
\end{aligned}$$

respectively, where $\alpha, \beta, \gamma, \delta$ are integers. By Theorem 5, 3) in [2], we have the restrictions $(\delta, \gamma) = (0,0), (0,1), (1,22)$. In the case $(\delta, \gamma) = (1, 22)$, we have $\beta = -209$ since the sum of two vectors in the shadow is a codeword. To save space, we do not list the possible weight enumerators for each of the three cases.

We have verified that the number of codewords of weight 16 in C_{94} is 6072 and that the minimum weight of the shadow is 15. Hence the weight enumerator of the code C_{94} corresponds to $(\alpha, \beta, \gamma, \delta) = (3036, 0, 0, 0)$. We have verified by Magma that C_{94} has automorphism group of order 23.

2.3 A related self-dual code of length 96

Let C be a singly even self-dual code of length $n \equiv 6 \pmod 8$. Let C^* be the code of length $n+2$ obtained by extending $C_0^\perp$ as follows:

$$(0, 0, C_0) \cup (1, 1, C_2) \cup (1, 0, C_1) \cup (0, 1, C_3)$$

where (x, y, C_i) denotes the set $\{(x, y, z) \in \mathbb{F}_2^{n+2} | z \in C_i\}$. Then C^* is a doubly even self-dual code [1]. In our case, C_{94}^* is a doubly even self-dual $[96, 48, 16]$ code since C_{94} has shadow of minimum weight 15. The code C_{94}^* has the following weight enumerator:

$$\begin{aligned}
&1 + 9108y^{16} + 3071328y^{20} + 370937840y^{24} + 18637739040y^{28} \\
&+ 422086556775y^{32} + 4552826872672y^{36} + 24292762502544y^{40} \\
&+ 65726907444000y^{44} + 91447786444040y^{48} + \cdots + y^{96}.
\end{aligned}$$

There are 30 known inequivalent doubly even self-dual $[96, 48, 16]$ codes [3], [4] and [5]. Since C_{94}^* and the 30 known codes have different weight enumerators, C_{94}^* is inequivalent to any of the known codes. We have verified by MAGMA that C_{94}^* has automorphism group of order 23.

References

[1] R. Brualdi, V. Pless, Weight enumerators of self-dual codes, *IEEE Trans. Inform. Theory* 37, 1991, 1222-1225.

[2] J. H. Conway, N. J. A. Sloane, A new upper bound on the minimal distance of self-dual codes, *IEEE Trans. Inform. Theory* 36, 1990, 1319-1333.

[3] R. Dontcheva, On the doubly-even self-dual codes of length 96, *IEEE Trans. Inform. Theory* 48, 2002, 557-561.

[4] S. T. Dougherty, T. A. Gulliver, M. Harada, Extremal binary self-dual codes, *IEEE Trans. Inform. Theory* 43, 1997, 2036-2047.

[5] W. Feit, A self-dual even $(96, 48, 16)$ code, *IEEE Trans. Inform. Theory* 20, 1974, 136-138.

[6] P. Gaborit, A. Otmani, Experimental constructions of self-dual codes, *Finite Fields Appl.* 9, 2003, 372-394.

[7] M. Grassl, Code tables: Bounds on the parameters of various types of codes, Available online at `http://www.codetables.de/`.

[8] W. C. Huffman, Automorphisms of codes with application to extremal doubly-even codes of length 48, *IEEE Trans. Inform. Theory* 28, 1982, 511-521.

[9] C. L. Mallows, N. J. A. Sloane, An upper bound for self-dual codes, *Inform. Control* 22, 1973, 188-200.

[10] E. M. Rains, Shadow bounds for self-dual codes, *IEEE Trans. Inform. Theory* 44, 1998, 134-139.

[11] V. Y. Yorgov, Binary self-dual codes with automorphisms of odd order, *Probl. Pered. Inform.* 19, 1983, 11-24 (in Russian); English transl. *Probl. Inform. Transm.* 19, 1983, 260-270.

New linear codes over GF(8)[1]

Plamen Hristov plhristov@tugab.bg
Department of Mathematics, Technical University of Gabrovo,
5300 Gabrovo, BULGARIA

Abstract. Let $[n, k, d]_q$-code be a linear code of length n, dimension k and minimum Hamming distance d over $GF(q)$. One of the most important problems in coding theory is to construct codes with best possible minimum distances. Recently, the class of quasi-cyclic (QC) codes has been proven to contain many such codes. In this paper, thirty two codes over $GF(8)$ are constructed (among them one optimal code), which improve the best known lower bounds on minimum distance.

1 Introduction

Let $GF(q)$ denote the Galois field of q elements. A linear code C over $GF(q)$ of length n, dimension k and minimum Hamming distance d is called an $[n, k, d]_q$-code.

A code C is said to be quasi-cyclic (QC or p-QC) if a cyclic shift of a codeword by p positions results in another codeword. A cyclic shift of an m-tuple $(x_0, x_1, \ldots, x_{m-1})$ is the m-tuple $(x_{m-1}, x_0, \ldots, x_{m-2})$. The blocklength, n, of a p-QC code is a multiple of p, so that $n = pm$.

A matrix B of the form

$$B = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{m-2} & b_{m-1} \\ b_{m-1} & b_0 & b_1 & \cdots & b_{m-3} & b_{m-2} \\ b_{m-2} & b_{m-1} & b_0 & \cdots & b_{m-4} & b_{m-3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ b_1 & b_2 & b_3 & \cdots & b_{m-1} & b_0 \end{bmatrix}, \tag{1}$$

is called a *circulant matrix*. A class of QC codes can be constructed from $m \times m$ circulant matrices. In this case, the generator matrix, G, can be represented as

$$G = [B_1,\ B_2,\ \ldots,\ B_p], \tag{2}$$

where B_i is a circulant matrix.

The algebra of $m \times m$ circulant matrices over $GF(q)$ is isomorphic to the algebra of polynomials in the ring $GF(q)[x]/(x^m - 1)$ if B is mapped onto the polynomial, $b(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{m-1}x^{m-1}$, formed from the entries in the first row of B. The $b_i(x)$ associated with a QC code are called the *defining polynomials*.

[1] This work was partially supported by the Bulgarian National Science Fund under Contract in TU–Gabrovo.

If the defining polynomials $b_i(x)$ contain a common factor which is also a factor of $x^m - 1$, then the QC code is called *degenerate*.

The dimension k of the QC code is equal to the degree of $h(x)$, where [4]

$$h(x) = \frac{x^m - 1}{\gcd\{x^m - 1, b_0(x), b_1(x), \cdots, b_{p-1}(x)\}}. \tag{3}$$

If the polynomial $h(x)$ has degree m, the dimension of the code is m, and (2) is a generator matrix. If $\deg(h(x)) = k < m$, a generator matrix for the code can be constructed by deleting $m - k$ rows of (2).

Let the defining polynomials of the code C be in the next form

$$d_1(x) = g(x),\ d_2(x) = f_2(x)g(x),\ \cdots\ ,\ d_p(x) = f_p(x)g(x), \tag{4}$$

where $g(x)|(x^m-1), g(x), f_i(x) \in GF(q)[x]/(x^m-1),\ (f_i(x), (x^m-1)/g(x)) = 1$ and $\deg f_i(x) < m - \deg g(x)$ for all $1 \leq i \leq p$. Then C is a degenerate QC code, which is one-generator QC code (see [4],[2]) and for this code $n = mp$, and $k = m - \deg g(x)$.

Similarly to the case of cyclic codes, an p-QC code over $GF(q)$ of length $n = pm$ can be viewed as an $GF(q)[x]/(x^m - 1)$ submodule of $(GF(q)[x]/(x^m - 1))^p$ [4],[2]. Then an r-generator QC code is spanned by r elements of $(GF(q)[x]/(x^m - 1))^p$.

In this paper we consider one-generator QC codes. A well-known results regarding the one-generator QC codes are as follows.

Theorem 1 [4],[2]: Let C be an one-generator QC code over $GF(q)$ of length $n = pm$. Then, a generator $\mathbf{g(x)} \in (GF(q)[x]/(x^m - 1))^p$ of C has the following form

$$\mathbf{g(x)} = (f_1(x)g_1(x), f_2(x)g_2(x), \cdots, f_p(x)g_p(x))$$

where $g_i(x)|(x^m - 1)$ and $(f_i(x), (x^m - 1)/g_i(x)) = 1$ for all $1 \leq i \leq p$.

Theorem 2 [2]: Let C be an one-generator QC code over $GF(q)$ of length $n = pm$ with a generator of the form

$$\mathbf{g(x)} = (f_1(x)g(x), f_2(x)g(x), \cdots, f_p(x)g(x))$$

where $g(x)|(x^m - 1), g(x), f_i(x) \in GF(q)[x]/(x^m - 1)$ and $(f_i(x), (x^m - 1)/g(x)) = 1$ for all $1 \leq i \leq p$. Then

$$p.((\#\text{ of consecutive roots of } g(x)) + 1) \leq d_{\ \min}(C)$$

and the dimension of C is equal to $m - \deg g(x)$.

Theorem 3 (construction X) Let $C_2 = [n, k - l, d + s]_q$ code be a subcode of the code $C_1 = [n, k, d]_q$ and let $C_3 = [a, l, s]_q$ be a third code. Then there exists an $C = [n + a, k, d + s]_q$ code.

Table 1: Minimum distances of the $[17p, 8, d]_8$ quasi-cyclic codes

p	$17p$	f_p	d	d_{gr}	p	$17p$	f_p	d	d_{gr}
2	34	1025246	21	21	5	85	1003347	62	61
3	51	1536	34	35	6	102	1237534	76	75
4	68	147711	48	49	7	119	1014524	90	89

Quasi-cyclic codes form an important class of linear codes. A large number of record breaking (and optimal codes) are QC codes [1]. In this paper, new one-generator QC codes ($p \geq 2$) are constructed using a algebraic-combinatorial computer search, similar to that in [3]. For convenience, the elements of $GF(8)$ are given as integers: $2 = \beta, 4 = \beta^2, 3 = \beta^3, 6 = \beta^4, 7 = \beta^5, 5 = \beta^6$, where β is a root of the binary primitive polynomial $y^3 + y + 1$. The codes presented here (Table 2) improve the respective lower bounds on the minimum distance in [1].

2 The new QC codes

We have restricted our search to one-generator QC codes with a generator of the form as in Theorem 2 and $f_1(x) = 1$. The main aim in our search is to find good $g(x)$, i.e. $g(x)$ which gives better minimum distance for $p = 2$ due to Theorem 2. When choosing $g(x)$ we calculate the minimum distance of the respective quasi-cyclic code D. After that we have compared the $d_{\min}(D)$ with the minimum distance of the best known codes [1] and with the given m and $g(x)$ we search for $f_p(x), p = 3, 4, \ldots$. Depending of the degree of $g(x)$, we obtain improvements on minimum distances for some dimensions.

We illustrate the search method in the following example. Let $m = 17$ and $q = 8$. Then the $\gcd(m, q) = 1$ and the splitting field of $x^m - 1$ is $GF(q^l)$ where l is the smallest integer such that $m|(q^l - 1)$. In our case $l = 8$ and so our splitting field is $GF(8^8)$. One of the generating polynomial for $GF(8^8)$ is a primitive polynomial $p(x) = x^8 + 2x^7 + 6x^6 + x^5 + x^4 + x^3 + 4^2 + 3x + 6$ and let α be a root of $p(x)$. Then

$$x^{17} - 1 = \prod_{j=0}^{16} (x - \alpha^j)$$

Let now $k = 8$. There are two possibilities to obtain $g(x)$ of degree nine. By this reason, we can use exhaustive search. Taken $g(x) = x^9 + x^8 + x^6 + x^3 + x + 1$, we obtain $f_2(x) = x^6 + 2x^4 + 5x^3 + 2x^2 + 4x + 6$ and quasi-cyclic code $D = [34, 8, 21]_8$, the best known. After that we make search for $f_p(x), p = 3, 4 \ldots, 7$. This is a sequence of six quasi-cyclic codes. The results are given in Table 1.

It seems, that there are three new results: $[85,8,62]_8$, $[102,8,76]_8$ and $[119,8,90]_8$ codes. We present the new quasi-cyclic codes.

Theorem 1: There exist one-generator quasi-cyclic codes with parameters:

$[28,5,20]_8$ $[35,5,26]_8$ $[42,5,32]_8$ $[49,5,38]_8$ $[78,5,63]_8$ $[81,5,65]_8$
$[90,5,73]_8$ $[105,5,86]_8$ $[120,5,100]_8$ $[38,6,28]_8$ $[42,6,30]_8$ $[84,6,66]_8$
$[95,6,75]_8$ $[42,7,29]_8$ $[84,7,63]_8$ $[90,7,68]_8$ $[95,7,72]_8$ $[105,7,81]_8$
$[36,8,23]_8$ $[42,8,28]_8$ $[85,8,62]_8$ $[91,8,67]_8$ $[102,8,76]_8$ $[105,8,78]_8$
$[119,8,90]_8$ $[39,9,24]_8$ $[91,9,65]_8$ $[102,9,74]_8$ $[105,9,76]_8$ $[93,11,62]_8$

Proof. The coefficients of the defining polynomials of the codes are as follows:

A $[28,5,20]_8$**-code:** 2310000,7712210,4343110,1642100; Adding the columns $(63421)^t$, $(25641)^t$, $(47261)^t$ and $(52371)^t$ to the generator matrix, the above code can be extended to a $[32,5,24]_8$ code.

A $[35,5,26]_8$**-code:** 2310000,4575210,1612510,5131710,1201310; Adding the columns $(63421)^t$, $(25641)^t$ and $(52371)^t$, the above code can be extended to a $[38,5,29]_8$ code.

A $[49,5,38]_8$**-code:** 2310000,6722100,4556310,2644510,5473410,3265310,3415210; Adding the columns $(74531)^t$ and $(52371)^t$, the above code can be extended to a $[51,5,40]_8$ code.

A $[38,6,28]_8$**-code:** 1301247742103100000,6333647125776166100; Adding the columns $(130100)^t$ and $(164361)^t$, the above code can be extended to a $[40,6,29]_8$ code.

A $[42,6,30]_8$**-code:** 6432343617331251000000,5377215221334555427l0; Adding the columns $(630210)^t$, $(520710)^t$, $(602301)^t$, $(703401)^t$, $(063021)^t$ and $(052071)^t$, the above code can be extended to a $[48,6,36]_8$ code.

A $[42,7,29]_8$**-code:** 2553561507027510000000,506312404625072547100 ; Adding the column $(3657521)^t$, the above code can be extended to a $[43,7,30]_8$ code.

A $[84,7,63]_8$**-code:** 2553561507027510000000,506312404625072547100, 442406377267775621000,354174272601230173510; Adding the columns $(0630210)^t$, $(5703401)^t$ and $(5063021)^t$, the above code can be extended to an $[87,7,66]_8$ code.

A $[95,7,72]_8$**-code:** 1223152513221000000,6454574176233563710,3251455612372474710, 3737472772015457210, 1207412747214702100 ; Adding the columns $(3273010)^t$ and $(5536010)^t$, the above code can be extended to a $[97,7,74]_8$ code.

A $[105,8,78]_8$**-code:** 55356150702751000000,506312404625072547100; Adding the column $(11326073)^t$, the above code can be extended to an $[106,8,79]_8$ code.

Remark: The defining polynomials of the some codes, which are missing in Theorem 1,are given in [1]. All defining polynomials, generator matrices and weight enumerators are available on request from the author.

Theorem 2: There exist $[45,8,30]_8$ code.

Proof. There exist quasi-cyclic $[42,8,28]_8$code with defining polynomials: 126716642762710000000, 316544405114436465310. This code as a subcode a $[42,6,30]_8$ code with defining polynomials: 1431256103657132000000, 106500266260354044710. Using auxiliary $[3,2,2]_8$ code and applying construction X, we obtain a $[45,8,30]_8$ code. The following generator matrix yields a

Table 2: Minimum distances of the new linear codes over GF(8)

code	d	d_{gr}	code	d	d_{gr}	code	d	d_{gr}	code	d	d_{gr}
[32,5]	24	23	[120,5]	100	98	[91,7]	69	68	[102,8]	76	75
[38,5]	29	28	[20,6]	13	12	[97,7]	74	73	[106,8]	79	78
[43,5]	33	32	[40,6]	29	28	[105,7]	81	80	[119,8]	90	89
[51,5]	40	39	[48,6]	36	35	[36,8]	23	22	[39,9]	24	23
[78,5]	63	62	[84,6]	66	65	[42,8]	28	27	[91,9]	65	64
[82,5]	66	65	[95,6]	75	74	[45,8]	30	29	[102,9]	74	73
[91,5]	74	73	[43,7]	30	29	[85,8]	62	61	[106,9]	77	76
[107,5]	88	87	[87,7]	66	65	[91,8]	67	66	[93,11]	62	61

$[45,8,30]_8$ code:

$$\begin{pmatrix} G \mid 000 \\ \hdashline 065633052407003200000210044663374762421400201 \\ 600516152040504700000647072502075311376325510 \end{pmatrix},$$

where G denotes the generator matrix of the $[42,6,30]_8$ code.

Theorem 3. There exist optimal $[20,6,13]_8$ code.

Proof. There exist quasi-cyclic $[18,6,11]_8$ code with defining polynomials: 232701, 213171, 510661. Adding the columns $(414141)^t$ and $(717171)^t$, this code can be extended to an optimal $[20,6,13]_8$ code with weight enumerator $0^1 13^{2898} 14^{6363} 15^{13860} 16^{39060} 17^{59010} 18^{71757} 19^{50792} 20^{18403}$.

References

[1] M. Grassl, Linear code bound [electronic table; online], http://www.codetables.de.

[2] K. Lally, P. Fitzpatrick, Construction and classification of quasi-cyclic codes, *Proc. Intern. Workshop WCC1999*, Paris, France, 1999, 11-20.

[3] I. Siap, N. Aydin, D. Ray-Chaudhury, New ternary quasi-cyclic codes with better minimum distances, *IEEE Trans. Inform. Theory* 46, 2000, 1554-1558.

[4] G.E. Séguin, G. Drolet, The theory of 1-generator quasi-cyclic codes, Technical Report, Royal Military College of Canada, Kingston, ON, 1991.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 134-139*

The least known length of ordered basis of symmetric group

S. A. Kalinchuk
Yu. L. Sagalovich sagal@iitp.ru
Institute for Information Transmission Problems,
Russian Academy of Sciences, Moscow, RUSSIA

Abstract. The recurrent algorithm for construction of ordered basis of symmetric group with degree $n = 2^k$ is given. It is shown that the number of transpositions constituting such basis is equal to $O(n \log_2^2 n)$. This value exceeds the order of lower bound estimation only in coefficient $\log_2 n$.

1 Introduction

Let S_X be a symmetric group with degree $|X|$ on a set of numbers X. By S_n denote group S_X if $X = \{1, \ldots, n\}$.

Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_r$ be an ordered set of transpositions of S_X, where $r \leqslant C^2_{|X|}$. We shall denote such *ordered system of transpositions* by Ψ and represent as:

$$\Psi = \mathcal{T}_1 \; \mathcal{T}_2 \; \ldots \; \mathcal{T}_r \, ,$$

where the transpositions' number r will be denoted by $|\Psi|$.

Definition 1. *The system* Ψ *is called* ordered basis *of symmetric group* S_X *if any permutation* $\mathcal{P}_X \in S_X$ *can be represented as*

$$\mathcal{P}_X = \mathcal{T}_1^{\gamma_1} \cdot \mathcal{T}_2^{\gamma_2} \cdot \ldots \cdot \mathcal{T}_r^{\gamma_r} \, ,$$

where $\gamma_j \in \{0, 1\}, j = 1, 2, \ldots, r$. *Note that there can exist several vectors* $(\gamma_1, \ldots, \gamma_r)$ *representing the same permutation* $\mathcal{P}_X$.

In [1], we announced a result that can be easily used to show the existence of algorithms for constructions of ordered bases with the transpositions' number of order $\frac{3}{4} C_n^2$. Also there it was supposed that r should be close to value $n \log_2 n$. This assumption corresponds well to the rough upper bound of factorial

$$n! \leqslant n^n = 2^{n \log_2 n}.$$

The obtained result is based on that the degree n of symmetric group S_n is chosen to be equal to $n = 2^k$, $k \geqslant 3$. Such choice allows successively partitioning set of permutated objects in two equal-sized subsets. At each stage of partition, "mixing" among objects is introduced, for example, by permutation (7). The main results are formed by relations (3) – (6).

2 Main results

2.1 Part 1

Consider a symmetric group S_X at $|X| = 4m$, where $m \geqslant 2$. Partition the set $X = \{x_1, \dots, x_{4m}\}$ into two subsets, $\mathbb{O}$ and $\mathbb{E}$:

$$\mathbb{O} \cup \mathbb{E} = X,\ \mathbb{O} \cap \mathbb{E} = \varnothing,\ |\mathbb{O}| = |\mathbb{E}| = 2m\,. \tag{1}$$

Let $\mathcal{P}_X \triangleq \mathcal{P}_{\mathbb{O}\cup\mathbb{E}}$ be any permutation of group $S_X \triangleq S_{\mathbb{O}\cup\mathbb{E}}$. It is evident that

$$\mathcal{P}_{\mathbb{O}\cup\mathbb{E}} = \begin{pmatrix} \mathbb{O}' & \mathbb{E}' & \mathbb{O}'' & \mathbb{E}'' \\ \widetilde{\mathbb{O}}' & \widetilde{\mathbb{E}}' & \widetilde{\mathbb{E}}'' & \widetilde{\mathbb{O}}'' \end{pmatrix} = \begin{pmatrix} \mathbb{O}' & \mathbb{O}'' & \mathbb{E}' & \mathbb{E}'' \\ \widetilde{\mathbb{O}}' & \widetilde{\mathbb{O}}'' & \widetilde{\mathbb{E}}' & \widetilde{\mathbb{E}}'' \end{pmatrix} \cdot \begin{pmatrix} \widetilde{\mathbb{O}}'' & \widetilde{\mathbb{E}}'' \\ \widetilde{\mathbb{E}}'' & \widetilde{\mathbb{O}}'' \end{pmatrix},$$

where $\mathbb{O} = \mathbb{O}' \cup \mathbb{O}'' = \widetilde{\mathbb{O}}' \cup \widetilde{\mathbb{O}}''$, $\mathbb{E} = \mathbb{E}' \cup \mathbb{E}'' = \widetilde{\mathbb{E}}' \cup \widetilde{\mathbb{E}}''$ and notation $\begin{matrix} \mathbb{A} \\ \mathbb{B} \end{matrix} \triangleq \begin{matrix} a_1 & a_2 & \dots & a_{|\mathbb{A}|} \\ b_1 & b_2 & \dots & b_{|\mathbb{B}|} \end{matrix}$, $\mathbb{A} = \{a_1, a_2, \dots, a_{|\mathbb{A}|}\}$, $\mathbb{B} = \{b_1, b_2, \dots, b_{|\mathbb{B}|}\}$, $|\mathbb{A}| = |\mathbb{B}|$. Therefore,

Proposition 1. *Any permutation $\mathcal{P}_{\mathbb{O}\cup\mathbb{E}}$ of group $S_{\mathbb{O}\cup\mathbb{E}}$ can be factored as*

$$\mathcal{P}_{\mathbb{O}\cup\mathbb{E}} = \mathcal{P}_{\mathbb{O}} \cdot \mathcal{P}_{\mathbb{E}} \cdot \mathcal{T}_{\mathbb{O},\mathbb{E}}\,, \tag{2}$$

where $\mathcal{P}_{\mathbb{O}}$ and $\mathcal{P}_{\mathbb{E}}$ are some permutations belonging to symmetric groups $S_{\mathbb{O}}$ and $S_{\mathbb{E}}$ correspondingly, and a permutation $\mathcal{T}_{\mathbb{O},\mathbb{E}}$ of group $S_{\mathbb{O}\cup\mathbb{E}}$ has the form as

$$\begin{pmatrix} \mathbb{O}^* & \mathbb{E}^* \\ \mathbb{E}^* & \mathbb{O}^* \end{pmatrix} \triangleq (\mathbb{O}^*, \mathbb{E}^*),\ \textit{where } \mathbb{O}^* \subseteq \mathbb{O},\ \mathbb{E}^* \subseteq \mathbb{E}. \tag{3}$$

Definition 2. *An ordered system of transpositions of group $S_{\mathbb{O}\cup\mathbb{E}}$ is called* system generating permutations of the form $\mathfrak{S}_{\mathbb{O}}\mathfrak{S}_{\mathbb{E}}\mathcal{T}_{\mathbb{O},\mathbb{E}}$, *if $\mathcal{T}_{\mathbb{O},\mathbb{E}}$ can be* any *permutation of the form* (3), *and $\mathfrak{S}_{\mathbb{O}}$, $\mathfrak{S}_{\mathbb{E}}$ are some permutations of groups $S_{\mathbb{O}}$, $S_{\mathbb{E}}$ correspondingly.*

Proposition 2. *Let $\Psi_{\mathbb{O}}$ and $\Psi_{\mathbb{E}}$ be ordered bases of groups $S_{\mathbb{O}}$ and $S_{\mathbb{E}}$ correspondingly. Let $\Psi_{\mathbb{O},\mathbb{E}}$ be an ordered system of transpositions of group $S_{\mathbb{O}\cup\mathbb{E}}$, and this system generates permutations of the form $\mathfrak{S}_{\mathbb{O}}\mathfrak{S}_{\mathbb{E}}\mathcal{T}_{\mathbb{O},\mathbb{E}}$. Then the system*

$$\Psi_{\mathbb{O}\cup\mathbb{E}} = \Psi_{\mathbb{O}}\,\Psi_{\mathbb{E}}\,\Psi_{\mathbb{O},\mathbb{E}} \tag{4}$$

is the ordered basis of group $S_{\mathbb{O}\cup\mathbb{E}}$.

Proof Follows directly from the factorization (2) and that

$$\mathcal{P}_{\mathbb{O}} \cdot \mathcal{P}_{\mathbb{E}} \cdot \mathcal{T}_{\mathbb{O},\mathbb{E}} = \underbrace{\mathcal{P}_{\mathbb{O}}\mathfrak{S}_{\mathbb{O}}{}^{-1}}_{\Psi_{\mathbb{O}}} \cdot \underbrace{\mathcal{P}_{\mathbb{E}}\mathfrak{S}_{\mathbb{E}}{}^{-1}}_{\Psi_{\mathbb{E}}} \cdot \underbrace{\mathfrak{S}_{\mathbb{O}}\mathfrak{S}_{\mathbb{E}}\mathcal{T}_{\mathbb{O},\mathbb{E}}}_{\Psi_{\mathbb{O},\mathbb{E}}}\,.$$

2.2 Part 2

Partition the set $\mathbb{O}$ into subsets $\mathbb{O}_1$, $\mathbb{O}_2$ and the set $\mathbb{E}$ into subsets $\mathbb{E}_1$, $\mathbb{E}_2$ by the same way as in (1). Thus,

$$\mathbb{O}_1 \cup \mathbb{O}_2 = \mathbb{O},\ \mathbb{O}_1 \cap \mathbb{O}_2 = \varnothing, \qquad \mathbb{E}_1 \cup \mathbb{E}_2 = \mathbb{E},\ \mathbb{E}_1 \cap \mathbb{E}_2 = \varnothing,$$

where $|\mathbb{O}_1| = |\mathbb{O}_2| = |\mathbb{E}_1| = |\mathbb{E}_2| = \frac{1}{4}|X| = m$.

Let $\mathbb{O}_1 = \{o_1^1, o_2^1, \ldots, o_m^1\}$, $\mathbb{O}_2 = \{o_1^2, o_2^2, \ldots, o_m^2\}$, $\mathbb{E}_1 = \{e_1^1, e_2^1, \ldots, e_m^1\}$, $\mathbb{E}_2 = \{e_1^2, e_2^2, \ldots, e_m^2\}$.

Consider an ordered system of transpositions $\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1}$ consisting of m transpositions of the form $(o_i^1, e^2_{\pi_1(i)})$ and m transpositions of the form $(o_j^2, e^1_{\pi_2(j)})$, where $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant m$, and π_1, π_2 are some permutations defined on the set $\{1, 2, \ldots, m\}$. In expanded form such system is represented as:

$$\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1} = \left(o_1^1, e^2_{\pi_1(1)}\right) \ldots \left(o_m^1, e^2_{\pi_1(m)}\right) \left(o_1^2, e^1_{\pi_2(1)}\right) \ldots \left(o_m^2, e^1_{\pi_2(m)}\right)$$

Definition 3. *Consider* $\widetilde{\mathbb{O}} \subseteq \mathbb{O}$, $\widetilde{\mathbb{E}} \subseteq \mathbb{E}$.

Let $\widetilde{\mathbb{O}} \succ\overset{\pi_1;\pi_2}{\circ}\prec \widetilde{\mathbb{E}}$ *denote that at any* $\tilde{o} \in \widetilde{\mathbb{O}}$ *and* $\tilde{e} \in \widetilde{\mathbb{E}}$ *transposition* $(\tilde{o}, \tilde{e})$ *does not belong to the system* $\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1}$.

If $\widetilde{\mathbb{O}} = \{\tilde{o}_1, \tilde{o}_2, \ldots, \tilde{o}_v\}$, $\widetilde{\mathbb{E}} = \{\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_v\}$, $|\widetilde{\mathbb{O}}| = |\widetilde{\mathbb{E}}| = v$ *then let* $\widetilde{\mathbb{O}} \succ\overset{\pi_1;\pi_2}{\bullet}\prec \widetilde{\mathbb{E}}$ *denote that all transpositions* $(\tilde{o}_i, \tilde{e}_i)$, $1 \leqslant i \leqslant v$, *belong to the system* $\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1}$.

Proposition 3. *Let* $\Psi_{\mathbb{O}_1,\mathbb{E}_1}$ *and* $\Psi_{\mathbb{O}_2,\mathbb{E}_2}$ *be some ordered systems of transpositions generating permutations of the forms* $\mathfrak{S}_{\mathbb{O}_1}\mathfrak{S}_{\mathbb{E}_1}\mathcal{T}_{\mathbb{O}_1,\mathbb{E}_1}$ *and* $\mathfrak{S}_{\mathbb{O}_2}\mathfrak{S}_{\mathbb{E}_2}\mathcal{T}_{\mathbb{O}_2,\mathbb{E}_2}$ *correspondingly. Then the system*

$$\Psi_{\mathbb{O},\mathbb{E}} = \Psi_{\mathbb{O}_1,\mathbb{E}_1}\Psi_{\mathbb{O}_2,\mathbb{E}_2}\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1} \tag{5}$$

generates permutations of the form $\mathfrak{S}_{\mathbb{O}}\mathfrak{S}_{\mathbb{E}}\mathcal{T}_{\mathbb{O},\mathbb{E}}$ *at any* π_1 *and* π_2.

Proof. Consider any permutation $\mathcal{T}_{\mathbb{O},\mathbb{E}} = (\mathbb{O}^*, \mathbb{E}^*)$, where $\mathbb{O}^* \subseteq \mathbb{O}$, $\mathbb{E}^* \subseteq \mathbb{E}$. Suppose $\mathbb{O}^* = \mathbb{O}_1^* \cup \mathbb{O}_2^*$ and $\mathbb{E}^* = \mathbb{E}_1^* \cup \mathbb{E}_2^*$, where $\mathbb{O}_1^* \subseteq \mathbb{O}_1$, $\mathbb{O}_2^* \subseteq \mathbb{O}_2$, $\mathbb{E}_1^* \subseteq \mathbb{E}_1$, $\mathbb{E}_2^* \subseteq \mathbb{E}_2$.

Let $\mathbb{O}^* = \{o_1, o_2, \ldots, o_t\}$, $\mathbb{E}^* = \{e_1, e_2, \ldots, e_t\}$, and let $\mathbb{O}_\alpha^* = \{o_1^\alpha, o_2^\alpha, \ldots, o_t^\alpha\}$, $\mathbb{E}_\beta^* = \{e_1^\beta, e_2^\beta, \ldots, e_t^\beta\}$ be the sets obtained by renumbering elements of the corresponding sets $\mathbb{O}^*$, $\mathbb{E}^*$ by means of permutations α, β defined on the set $\{1, 2, \ldots, t\}$: $o_i^\alpha = o_{\alpha(i)}$, $e_i^\beta = e_{\beta(i)}$, $1 \leqslant i \leqslant t$. It is obvious that at any α, β there exist such permutations $\widetilde{\mathfrak{S}}_{\mathbb{O}}$, $\widetilde{\mathfrak{S}}_{\mathbb{E}}$ of groups $S_{\mathbb{O}}$, $S_{\mathbb{E}}$ correspondingly that $(\mathbb{O}^*, \mathbb{E}^*) = \widetilde{\mathfrak{S}}_{\mathbb{O}}\widetilde{\mathfrak{S}}_{\mathbb{E}} \cdot (\mathbb{O}_\alpha^*, \mathbb{E}_\beta^*)$.

The sets $\mathbb{O}_1^*$, $\mathbb{O}_2^*$, $\mathbb{E}_1^*$, $\mathbb{E}_2^*$ can be partitioned into the following subsets:
$O_1' \overset{\pi_1;\pi_2}{\succ\circ\prec} E_2'$; $O_1'' \overset{\pi_1;\pi_2}{\succ\bullet\prec} E_2''$; $O_1' \cup O_1'' = \mathbb{O}_1^*$, $E_2' \cup E_2'' = \mathbb{E}_2^*$, $O_1' \cap O_1'' = \varnothing$, $E_2' \cap E_2'' = \varnothing$;
$O_2' \overset{\pi_1;\pi_2}{\succ\circ\prec} E_1'$; $O_2'' \overset{\pi_1;\pi_2}{\succ\bullet\prec} E_1''$; $O_2' \cup O_2'' = \mathbb{O}_2^*$, $E_1' \cup E_1'' = \mathbb{E}_1^*$, $O_2' \cap O_2'' = \varnothing$, $E_1' \cap E_1'' = \varnothing$;
$O' = O_1' \cup O_2'$; $E' = E_1' \cup E_2'$; $O' \overset{\pi_1;\pi_2}{\succ\circ\prec} E'$, $|O'| = |E'|$.

There exists such renumbering of elements for each of the sets $\mathbb{O}^*$, $\mathbb{E}^*$ that

$$\mathcal{T}_{\mathbb{O},\mathbb{E}} = (\mathbb{O}^*, \mathbb{E}^*) = \widetilde{\mathfrak{S}}'_{\mathbb{O}} \widetilde{\mathfrak{S}}'_{\mathbb{E}} \cdot (O', E')(O_1'', E_2'')(O_2'', E_1'') .$$

Whereas $|O'| = |O_1'| + |O_2'| = |E_1'| + |E_2'| = |E'|$, three cases are possible: 1) $|O_1'| = |E_1'|$, $|O_2'| = |E_2'|$; 2) $|O_1'| > |E_1'|$, $|O_2'| < |E_2'|$; 3) $|O_1'| < |E_1'|$, $|O_2'| > |E_2'|$.

Without loss of generality consider only case 2) : $|O_1'| > |E_1'|$, $|O_2'| < |E_2'|$.

Let $\mathcal{O}_1 \cup \widehat{\mathcal{O}}_1 = O_1'$, $\mathcal{O}_1 \cap \widehat{\mathcal{O}}_1 = \varnothing$, $\mathcal{E}_2 \cup \widehat{\mathcal{E}}_2 = E_2'$, $\mathcal{E}_2 \cap \widehat{\mathcal{E}}_2 = \varnothing$, $\widehat{\mathcal{E}}_1 = E_1'$, $\widehat{\mathcal{O}}_2 = O_2'$, where $|\mathcal{O}_1| = |\mathcal{E}_1|$, $|\mathcal{O}_2| = |\mathcal{E}_2|$, $|\widehat{\mathcal{O}}_1| = |\widehat{\mathcal{E}}_2|$. Also $\widehat{\mathcal{O}}_1 \overset{\pi_1;\pi_2}{\succ\circ\prec} \widehat{\mathcal{E}}_2$, since $O' \overset{\pi_1;\pi_2}{\succ\circ\prec} E'$.

There exists such renumbering of elements for each of the sets O', E' that

$$(O', E') = \widetilde{\mathfrak{S}}''_{\mathbb{O}} \widetilde{\mathfrak{S}}''_{\mathbb{E}} \cdot (\mathcal{O}_1, \mathcal{E}_1)(\mathcal{O}_2, \mathcal{E}_2)(\widehat{\mathcal{O}}_1, \widehat{\mathcal{E}}_2) .$$

It is clear, there exist such sets $\widehat{\mathcal{O}}_2 \in \mathbb{O}_2$, $\widehat{\mathcal{E}}_1 \in \mathbb{E}_1$ that

$$\widehat{\mathcal{O}}_2 \overset{\pi_1;\pi_2}{\succ\bullet\prec} \widehat{\mathcal{E}}_1,\ |\widehat{\mathcal{O}}_2| = |\widehat{\mathcal{E}}_1| = |\widehat{\mathcal{O}}_1| = |\widehat{\mathcal{E}}_2|;\ \mathcal{O}_2 \cap \widehat{\mathcal{O}}_2 = \varnothing,\ \mathcal{E}_1 \cap \widehat{\mathcal{E}}_1 = \varnothing .$$

It is also evident that $(\widehat{\mathcal{O}}_1, \widehat{\mathcal{E}}_2) = (\widehat{\mathcal{O}}_1, \widehat{\mathcal{O}}_2)(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \cdot (\widehat{\mathcal{O}}_1, \widehat{\mathcal{E}}_1)(\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_2) \cdot (\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_1)$.

This implies that

$$(O', E') = \widetilde{\mathfrak{S}}''_{\mathbb{O}} \widetilde{\mathfrak{S}}''_{\mathbb{E}} \cdot (\mathcal{O}_1, \mathcal{E}_1)(\mathcal{O}_2, \mathcal{E}_2) \cdot (\widehat{\mathcal{O}}_1, \widehat{\mathcal{O}}_2)(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \cdot (\widehat{\mathcal{O}}_1, \widehat{\mathcal{E}}_1)(\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_2) \cdot (\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_1) .$$

Since $\mathcal{O}_1 \cap \widehat{\mathcal{O}}_1 = \varnothing$, $\mathcal{O}_2 \cap \widehat{\mathcal{O}}_2 = \varnothing$, $\mathcal{E}_1 \cap \widehat{\mathcal{E}}_1 = \varnothing$, $\mathcal{E}_2 \cap \widehat{\mathcal{E}}_2 = \varnothing$, it follows that

$$\mathcal{T}_{\mathbb{O},\mathbb{E}} = \widetilde{\mathfrak{S}}'_{\mathbb{O}} \widetilde{\mathfrak{S}}''_{\mathbb{O}} (\widehat{\mathcal{O}}_1, \widehat{\mathcal{O}}_2) \cdot \widetilde{\mathfrak{S}}'_{\mathbb{E}} \widetilde{\mathfrak{S}}''_{\mathbb{E}} (\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \cdot (\mathcal{O}_1, \mathcal{E}_1)(\widehat{\mathcal{O}}_1, \widehat{\mathcal{E}}_1) \cdot (\mathcal{O}_2, \mathcal{E}_2)(\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_2) \cdot (O_1'', E_2'')(\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_1)(O_2'', E_1'') .$$

Each of the systems $\Psi_{\mathbb{O}_1,\mathbb{E}_1}$, $\Psi_{\mathbb{O}_2,\mathbb{E}_2}$ generates permutations of the forms $\mathfrak{S}_{\mathbb{O}_1} \mathfrak{S}_{\mathbb{E}_1} \mathcal{T}_{\mathbb{O}_1,\mathbb{E}_1}$, $\mathfrak{S}_{\mathbb{O}_2} \mathfrak{S}_{\mathbb{E}_2} \mathcal{T}_{\mathbb{O}_2,\mathbb{E}_2}$ correspondingly. Suppose $\mathcal{T}_{\mathbb{O}_1,\mathbb{E}_1} = (\mathcal{O}_1, \mathcal{E}_1)(\widehat{\mathcal{O}}_1, \widehat{\mathcal{E}}_1)$, $\mathcal{T}_{\mathbb{O}_2,\mathbb{E}_2} = (\mathcal{O}_2, \mathcal{E}_2)(\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_2)$. Then

$$\mathfrak{S}_{\mathbb{O}} \mathfrak{S}_{\mathbb{E}} \mathcal{T}_{\mathbb{O},\mathbb{E}} = \underbrace{\mathfrak{S}_{\mathbb{O}_1} \mathfrak{S}_{\mathbb{E}_1} \mathcal{T}_{\mathbb{O}_1,\mathbb{E}_1}}_{\Psi_{\mathbb{O}_1,\mathbb{E}_1}} \cdot \underbrace{\mathfrak{S}_{\mathbb{O}_2} \mathfrak{S}_{\mathbb{E}_2} \mathcal{T}_{\mathbb{O}_2,\mathbb{E}_2}}_{\Psi_{\mathbb{O}_2,\mathbb{E}_2}} \cdot \underbrace{(O_1'', E_2'')(\widehat{\mathcal{O}}_2, \widehat{\mathcal{E}}_1)(O_2'', E_1'')}_{\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1}} ,$$

where $\mathfrak{S}_{\mathbb{O}}^{-1} = \widetilde{\mathfrak{S}}'_{\mathbb{O}} \widetilde{\mathfrak{S}}''_{\mathbb{O}} (\widehat{\mathcal{O}}_1, \widehat{\mathcal{O}}_2) \cdot \mathfrak{S}_{\mathbb{O}_1}^{-1} \mathfrak{S}_{\mathbb{O}_2}^{-1}$, $\mathfrak{S}_{\mathbb{E}}^{-1} = \widetilde{\mathfrak{S}}'_{\mathbb{E}} \widetilde{\mathfrak{S}}''_{\mathbb{E}} (\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \cdot \mathfrak{S}_{\mathbb{E}_1}^{-1} \mathfrak{S}_{\mathbb{E}_2}^{-1}$. Each of three permutations marked out in previous expression is generated by corresponding ordered system of transpositions.

Based on that the permutation $\mathcal{T}_{\mathbb{O},\mathbb{E}}$ is any, it follows that the system $\Psi_{\mathbb{O},\mathbb{E}}$ generates permutations of the form $\mathfrak{S}_{\mathbb{O}} \mathfrak{S}_{\mathbb{E}} \mathcal{T}_{\mathbb{O},\mathbb{E}}$ at any π_1 and π_2 as they have been choosing at random. Proposition is proved.

2.3 Part 3

Using relations (4) and (5), we recurrently construct an ordered basis of symmetric group S_n at $n = 2^k$, $k \geqslant 3$.

At each step some sets are partitioned into two equal-sized subsets, that is, if $|\mathbb{A}| = 2t$ then $|\mathbb{A}_1| = |\mathbb{A}_2| = t$. By analogy we shall partition the original set $X = \{1, 2, 3, \dots, 2^k\}$ and apply (4) to being divided subsets till their minimal size is equal to 4. Let us use that if $\mathbb{A} = \{a_1, a_2, a_3, a_4\}$ then

$$\Psi_{\mathbb{A}} = (a_1, a_3)(a_1, a_4)(a_2, a_3)(a_1, a_2)(a_3, a_4) \tag{6}$$

is the ordered basis of group $S_{\mathbb{A}}$.

Suppose that in relation (5) for all subsets

$$\pi_1 = \pi_2 = \begin{pmatrix} 1 & 2 & \dots & m-1 & m \\ m & m-1 & \dots & 2 & 1 \end{pmatrix},\ 1 \leqslant m \leqslant 2^{k-2}\,. \tag{7}$$

We shall apply (5) until the minimal size of subsets is equal to 2.

Example. Consider $n = 2^3 = 8$, $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Let $X^0 = \{1, 3, 5, 7, 9, 11, 13, 15\}$, $X^1 = \{2, 4, 6, 8, 10, 12, 14, 16\}$, $X^{00} = \{1, 5, 9, 13\}$, $X^{01} = \{3, 7, 11, 15\}$, $X^{10} = \{2, 6, 10, 14\}$, $X^{11} = \{4, 8, 12, 16\}$. Then

$$\Psi_X = \Psi_{X^0}\Psi_{X^1}\ \Psi_{X^0,X^1} = \Psi_{X^{00}}\Psi_{X^{01}}\ \Psi_{X^{00},X^{01}}\ \ \Psi_{X^{10}}\Psi_{X^{11}}\ \Psi_{X^{10},X^{11}}\ \ \Psi_{X^0,X^1}\,.$$

Let ${X^{00}}_0 = \{1, 5\}$, ${X^{00}}_1 = \{9, 13\}$, ${X^{01}}_0 = \{3, 7\}$, ${X^{01}}_1 = \{11, 15\}$, ${X^{10}}_0 = \{2, 6\}$, ${X^{10}}_1 = \{10, 14\}$, ${X^{11}}_0 = \{4, 8\}$, ${X^{11}}_1 = \{12, 16\}$. Then

$$\begin{aligned} \Psi_{X^{00},X^{01}} &= \Psi_{{X^{00}}_0,{X^{01}}_0}\Psi_{{X^{00}}_1,{X^{01}}_1}\Psi^{\pi_1;\,\pi_2}_{{X^{00}}_0,{X^{01}}_1;\,{X^{00}}_1,{X^{01}}_0} \\ \Psi_{X^{10},X^{11}} &= \Psi_{{X^{10}}_0,{X^{11}}_0}\Psi_{{X^{10}}_1,{X^{11}}_1}\Psi^{\pi_1;\,\pi_2}_{{X^{10}}_0,{X^{11}}_1;\,{X^{10}}_1,{X^{11}}_0} \end{aligned}$$

Let ${X^0}_0 = \{1, 3, 5, 7\}$, ${X^0}_1 = \{9, 11, 13, 15\}$, ${X^1}_0 = \{2, 4, 6, 8\}$, ${X^1}_1 = \{10, 12, 14, 16\}$. ${X^0}_{00} = \{1, 3\}$, ${X^0}_{01} = \{5, 7\}$, ${X^0}_{10} = \{9, 11\}$, ${X^0}_{11} = \{13, 15\}$, ${X^1}_{00} = \{2, 4\}$, ${X^1}_{01} = \{6, 8\}$, ${X^1}_{10} = \{10, 12\}$, ${X^1}_{11} = \{14, 16\}$. Then

$$\begin{aligned} \Psi_{X^0,X^1} &= \Psi_{{X^0}_0,{X^1}_0}\Psi_{{X^0}_1,{X^1}_1}\Psi^{\pi_1;\,\pi_2}_{{X^0}_0,{X^1}_1;\,{X^0}_1,{X^1}_0}\,, \\ \Psi_{{X^0}_0,{X^1}_0} &= \Psi_{{X^0}_{00},{X^1}_{00}}\Psi_{{X^0}_{01},{X^1}_{01}}\Psi^{\pi_1;\,\pi_2}_{{X^0}_{00},{X^1}_{01};\,{X^0}_{01},{X^1}_{00}}\,, \\ \Psi_{{X^0}_1,{X^1}_1} &= \Psi_{{X^0}_{10},{X^1}_{10}}\Psi_{{X^0}_{11},{X^1}_{11}}\Psi^{\pi_1;\,\pi_2}_{{X^0}_{10},{X^1}_{11};\,{X^0}_{11},{X^1}_{10}} \\ . \end{aligned}$$

Whereas $|X^{00}| = |X^{01}| = |X^{10}| = |X^{11}| = 4$, then applying (6), we obtain

$$\begin{aligned}\Psi_X = {} & \underbrace{(1,9)(1.13)(5,9)(1,5)(9,13)}_{\Psi_{X^{00}}}\ \underbrace{(3,11)(3.15)(7,11)(3,7)(11,15)}_{\Psi_{X^{01}}} \\ & \underbrace{(1,3)(5,7)\ (1,7)(5,3)}_{\Psi_{{X^{00}}_0,{X^{01}}_0}}\ \underbrace{(9,11)(13,15)\ (9,15)(13,11)}_{\Psi_{{X^{00}}_1,{X^{01}}_1}}\ \underbrace{(1,15)(5,11)\ (9,7)(13,3)}_{\Psi^{\pi_1;\pi_2}_{{X^{00}}_0,{X^{01}}_1;{X^{00}}_1,{X^{01}}_0}} \\ & \underbrace{(2,10)(2,14)(6,10)(2,6)(10,14)}_{\Psi_{X^{10}}}\ \underbrace{(4,12)(4,16)(8,12)(4,8)(12,16)}_{\Psi_{X^{11}}} \\ & \underbrace{(2,4)(6,8)\ (2,8)(6,4)}_{\Psi_{{X^{10}}_0,{X^{11}}_0}}\ \underbrace{(10,12)(14,16)\ (10,16)(14,12)}_{\Psi_{{X^{10}}_1,{X^{11}}_1}}\ \underbrace{(2,16)(6,12)\ (10,8)(14,4)}_{\Psi^{\pi_1;\pi_2}_{{X^{10}}_0,{X^{11}}_1;{X^{10}}_1,{X^{11}}_0}} \\ & \underbrace{(1,2)(3,4)\ (1,4)(3,2)}_{\Psi_{{X^0}_{00},{X^1}_{00}}}\ \underbrace{(5,6)(7,8)\ (5,8)(7,6)}_{\Psi_{{X^0}_{01},{X^1}_{01}}}\ \underbrace{(1,8)(3,6)\ (5,4)(7,2)}_{\Psi^{\pi_1;\pi_2}_{{X^0}_{00},{X^1}_{01};{X^0}_{01},{X^1}_{00}}} \\ & \underbrace{(9,10)(11,12)\ (9,12)(11,10)}_{\Psi_{{X^0}_{10},{X^1}_{10}}}\ \underbrace{(13,14)(15,16)\ (13,16)(15,14)}_{\Psi_{{X^0}_{11},{X^1}_{11}}}\ \underbrace{(9,16)(11,14)\ (13,12)(15,10)}_{\Psi^{\pi_1;\pi_2}_{{X^0}_{10},{X^1}_{11};{X^0}_{11},{X^1}_{10}}} \\ & \underbrace{(1,16)(3,14)(5,12)(7,10)\ (9,8)(11,6)(13,4)(15,2)}_{\Psi^{\pi_1;\pi_2}_{{X^0}_0,{X^1}_1;{X^0}_1,{X^1}_0}}\end{aligned}$$

It is easy to see that such construction of ordered basis results in the following recurrent relations for the number of transpositions in ordered systems involved in construction.

Consider relation (5). Let $|\Psi_{\mathbb{O},\mathbb{E}}| = r(n)$, $|\Psi_{\mathbb{O}_1,\mathbb{E}_1}| = |\Psi_{\mathbb{O}_2,\mathbb{E}_2}| = r(\frac{n}{2})$. Since $|\Psi^{\pi_1;\pi_2}_{\mathbb{O}_1,\mathbb{E}_2;\mathbb{O}_2,\mathbb{E}_1}| = \frac{n}{2}$ then $r(n) = 2 \cdot r(\frac{n}{2}) + \frac{n}{2}$, and $r(2) = 1$. Therefore,

$$|\Psi_{\mathbb{O},\mathbb{E}}| = r(n) = \frac{n}{2}\log_2 n\,.$$

Consider relation (4). Let $|\Psi_{\mathbb{O}\cup\mathbb{E}}| = l(n)$, $|\Psi_{\mathbb{O}}| = |\Psi_{\mathbb{E}}| = l(\frac{n}{2})$. Then

$$l(n) = 2 \cdot l\Big(\frac{n}{2}\Big) + r(n)\,.$$

Since also $l(4) = 5$ (it follows from (6)) then

$$|\Psi_n| = l(n) = \frac{n}{4} \cdot (\log_2^2 n + \log_2 n - 1) = O(n \log_2^2 n)\,.$$

This implies that at $n = 2^k$ the ordered basis constructed by such recurrent way consists of $O(n \log_2^2 n)$ transpositions. Note that this number differs from the lower bound estimation for the number of transpositions in ordered bases, namely, differs from $\log_2 n!$ only in factor $O(\log_2 n)$.

References

[1] S. A. Kalinchuk, Yu. L. Sagalovich, The problem of minimal ordered basis of symmetric group, *Proc. Tenth Intern. Workshop ACCT*, Zvenigorod, Russia, Sept. 2006, 139-142.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 140-143*

The nonexistence of the (23,13,2,2) superimposed codes[1]

Stoyan Kapralov s.kapralov@gmail.com
Department of Mathematics, Technical University of Gabrovo,
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
BULGARIA
Mladen Manev ml.manev@gmail.com
Department of Mathematics, Technical University of Gabrovo,
Gabrovo, BULGARIA

Abstract. The nonexistence of (23,13,2,2) superimposed codes is proved.

1 Introduction

Definition 1 *A binary $N \times T$ matrix $C = (c_{ij})$ is called an (N, T, w, r) superimposed code (SIC) if for any pair of subsets $W, R \subset \{1, 2, \dots, T\}$ such that $|W| = w$, $|R| = r$ and $W \bigcap R = \emptyset$, there exists a coordinate $i \in \{1, 2, \dots, N\}$ such that $c_{ij} = 1$ for all $j \in W$ and $c_{ij} = 0$ for all $j \in R$.*

Let $N(T, w, r)$ be the minimum length N for which an (N, T, w, r) SIC exists for fixed values of T, w and r. The problem of determining the exact values of $N(T, w, r)$ is completely solved only for $w = r = 1$ [6].

The exact values of $N(T, 2, 2)$ are known only for $T \leq 12$ [1], [3], [2]:

T	4	5	6	7	8	9	10	11	12
$N(T,2,2)$	6	10	14	14	14	18	20	22	22

A (16,26,2,2) superimposed code is constructed in [3], hence $22 \leq N(T, 2, 2) \leq 26$ for $T = 13, 14, 15, 16$.

The main result of this article is that there is no (23,13,2,2) superimposed code. Consequently $24 \leq N(13, 2, 2) \leq 26$ for $T = 13, 14, 15, 16$.

2 Preliminaries

For a binary matrix or vector C denote by $wt(C)$ the number of 1's in C, and by $wt(\overline{C})$ the number of 0's in C.

[1] Partially supported by the Technical University of Gabrovo under Grant C-801/2008.

For a binary matrix C denote by $d(x, y)$ the Hamming distance between two columns x and y of C. Let $d_2 = \min\{d(x, y) \mid x, y \in C,\ x \neq y\}$ and $d(C) = \sum_{x,y \in C,\ x \neq y} d(x, y)$.

Lemma 2 *(Plotkin bound) [5]* $\binom{T}{2} d_2 \leq N \left\lfloor \frac{T}{2} \right\rfloor \left\lfloor \frac{T+1}{2} \right\rfloor.$

Definition 3 *Let x be a column in the superimposed code C. The residual code $Res(C, x = a)$ is the code obtained in the following way:*
1) take the i-th row $(i = 1, 2, \ldots, N)$ iff $c_{i.x} = a$;
2) delete the column x in the selected rows.

We will use the shorter notation $Res(C,\ x = a,\ y = b)$ instead of $Res(Res(C, x = a), y = b)$.

Lemma 4 *Suppose C is an (N, T, w, r) superimposed code with $w > 1,\ r > 1$. Then*
(a) $N(T-1, w-1, r) \leq wt(x) \leq N - N(T-1, w, r-1)$ for any column x;
(b) $d_2 \geq 2N(T-2, w-1, r-1)$.

Proof. (a) The residual code $Res(C, x = 1)$ is a $(wt(x), T-1, w-1, r)$ SIC, while $Res(C, x = 0)$ is a $(wt(\overline{x}), T-1, w, r-1)$ SIC. Hence $wt(x) \geq N(T-1, w-1, r)$ and $wt(\overline{x}) \geq N(T-1, w, r-1)$.
(b) Let x and y be an arbitrary pair of different columns of C. The residual codes $Res(C, x = 1, y = 0)$ and $Res(C, x = 0, y = 1)$ are $(N', T-2, w-1, r-1)$ and $(N'', T-2, w-1, r-1)$ SIC, respectively. Hence $d(x, y) = N' + N'' \geq 2N(T-2, w-1, r-1)$. □

Lemma 5 *Let x be a column of C and $A = Res(C, x = 1)$. Then*

$$d(A) + wt(\overline{A}) \geq \binom{T}{2} d_2 - (N - wt(x)) \left\lfloor \frac{T}{2} \right\rfloor \left\lfloor \frac{T+1}{2} \right\rfloor$$

Proof. Denote by C_1 the submatrix of C, containing all rows with value 1 in the column x, and by C_0 the remaining part of C. Then

$$\binom{T}{2} d_2 \leq d(C) = d(C_1) + d(C_0).$$

But $d(C_1) = d(A) + wt(\overline{A})$ and $d(C_0) \leq (N - wt(x)) \left\lfloor \frac{T}{2} \right\rfloor \left\lfloor \frac{T+1}{2} \right\rfloor$.
The result follows. □

Definition 6 *Two (N, T, w, r) superimposed codes are equivalent if one of them can be obtained from the other by a permutation of the rows and a permutation of the columns. In the case $w = r$ an inversion of the all code entries is also allowed.*

3 Main result

Lemma 7 *If C is a $(23,13,2,2)$ superimposed code then*
(a) $9 \leq wt(x) \leq 14$ for any column x of C;
(b) $d_2 = 12$.

Proof. (a) Follows from Lemma 4 and the known value $N(12,1,2) = 9$ [3];
(b) Follows from Lemma 2, Lemma 4 and the known value $N(11,1,1) = 6$ [6]. □

Theorem 8 *There is no (23,13,2,2) superimposed code.*

Proof. Suppose C is a (23,13,2,2) SIC. Up to equivalence we may assume that the code has the following form:

$$\left(\begin{array}{c|c} 1 & \\ \vdots & A \\ 1 & \\ \hline 0 & \\ \vdots & B \\ 0 & \end{array}\right)$$

where the matrix A is a $(N,12,1,2)$ SIC where $N \in \{9,10,11\}$, and the matrix B has to be chosen in such a way that the whole matrix to be a $(23,13,2,2)$ SIC. We may assume that the rows of B are sorted lexicographically.

Applying the method described in [4] we constructed all inequivalent $(9,12,1,2)$, $(10,12,1,2)$ and $(11,12,1,2)$ superimposed codes. Then we checked the condition of Lemma 5, which turned out to reduce the amount of computations approximately 6 times.

SIC parameters	(9,12,1,2)	(10,12,1,2)	(11,12,1,2)
number of inequivalent SIC	1	99	243709
number of inequivalent SIC, which satisfy Lemma 5	1	54	44509

Using an exhaustive computer search we tried to construct the matrix B column by column taking into account the restrictions of Lemma 7 and the sorted rows property. It turned out, however, that superimposed codes with parameters (23,13,2,2) do not exist. □

References

[1] S. Kapralov, The nonexistence of the (21,11,2,2) superimposed codes, *Proc. Fifth Intern. Workshop OCRT*, White Lagoon, Bulgaria, 2007, 101-105.

[2] S. Kapralov, M. Manev, The nonexistence of (19,10,2,2) superimposed codes, *Proc. Fourth Intern. Workshop OCRT*, Pamporovo, Bulgaria, 2005, 196-200.

[3] H. K. Kim, V. S. Lebedev, On optimal superimposed codes, *J. Combin. Designs* 12, 2004, 79-91.

[4] M. Manev, On some optimal (N,T,1,2) superimposed codes, *Proc. Fifth Intern. Workshop OCRT*, White Lagoon, Bulgaria, 2007, 178-182.

[5] M. Plotkin, Binary codes with specified minimum distance, *IRE Trans. Inform. Theory* 6, 1960, 445-450.

[6] E. Sperner, Ein Satz über Untermengen einer endlichen Menge, *Math. Zeitschrift* 27, 1928, 544-548.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 144-150*

Double and bordered α-circulant self-dual codes over finite commutative chain rings

MICHAEL KIERMAIER, ALFRED WASSERMANN
michael.kiermaier, alfred.wassermann@uni-bayreuth.de
Mathematical Department, University of Bayreuth, D-95440 Bayreuth, GERMANY

Abstract. In this paper we investigate codes over finite commutative rings R, whose generator matrices are built from α-circulant matrices. For a non-trivial ideal $I < R$ we give a method to lift such codes over R/I to codes over R, such that some isomorphic copies are avoided. For the case where I is the minimal ideal of a finite chain ring we refine this lifting method: We impose the additional restriction that lifting preserves self-duality. It will be shown that this can be achieved by solving a linear system of equations over a finite field. Finally we apply this technique to $\mathbb{Z}_4$-linear double nega-circulant and bordered circulant self-dual codes. We determine the best minimum Lee distance of these codes up to length 64.

1 α-circulant matrices

In this section, we give some basic facts on α-circulant matrices, compare with [4], where some theory of circulant matrices is given in chapter 16, and with [1], where α-circulant matrices are called $\{k\}$-circulant on page 84.

Definition 1.1 *Let R be a commutative ring, k a natural number and $\alpha \in R$. An $(k \times k)$-matrix A is called* α-circulant, *if A has the form*

$$\begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{k-2} & a_{k-1} \\ \alpha a_{k-1} & a_0 & a_1 & \cdots & a_{k-3} & a_{k-2} \\ \alpha a_{k-2} & \alpha a_{k-1} & a_0 & \cdots & a_{k-4} & a_{k-3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \alpha a_1 & \alpha a_2 & \alpha a_3 & \cdots & \alpha a_{k-1} & a_0 \end{pmatrix}$$

with $a_i \in R$ for $i \in \{0, \ldots, k-1\}$. For $\alpha = 1$, A is called circulant, *for $\alpha = -1$, A is called* nega-circulant *or* skew-circulant, *and for $\alpha = 0$, A is called* semi-circulant.

An α-circulant matrix A is completely determined by its first row $v = (a_0, a_1, \ldots, a_{k-1}) \in R^k$. We denote A by $\operatorname{circ}_\alpha(v)$ and say that A is the α-circulant matrix generated *by v.*

In the following, α usually will be a unit or even $\alpha^2 = 1$.

We define $T_\alpha = \operatorname{circ}_\alpha(0, 1, 0, \ldots, 0)$, that is

$$T_\alpha = \begin{pmatrix} & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \\ \alpha & & & & \end{pmatrix}$$

Using T_α, there is another characterization of an α-circulant matrix: A matrix $A \in R^{k \times k}$ is α-circulant iff $AT_\alpha = T_\alpha A$. This is seen directly by comparing the components of the two matrix products.

In the following it will be useful to identify the generating vectors $(a_0, a_1, \dots, a_{k-1}) \in R^n$ with the polynomials $\sum_{i=0}^{k-1} a_i x^i \in R[x]$ of degree at most $k-1$, which again can be seen as a set of representatives of the R-algebra $R[x]/(x^k - \alpha)$. Thus, we get an injective mapping $\operatorname{circ}_\alpha : R[x]/(x^k - \alpha) \to R^{k \times k}$.

Obviously $\operatorname{circ}_\alpha(1) = I_k$, which denotes the $(k \times k)$-unit matrix, $\operatorname{circ}_\alpha(\lambda f) = \lambda \operatorname{circ}_\alpha(f)$ and $\operatorname{circ}_\alpha(f + g) = \operatorname{circ}_\alpha(f) + \operatorname{circ}_\alpha(g)$ for all scalars $\lambda \in R$ and all f and g in $R[x]/(x^k - \alpha)$. Furthermore, it holds $\operatorname{circ}_\alpha(e_i) = \operatorname{circ}_\alpha(x^i) = T_\alpha^i$ for all $i \in \{0, \dots, k-1\}$ and $\operatorname{circ}_\alpha(x^k) = \operatorname{circ}_\alpha(\alpha) = \alpha I_k = T_\alpha^k$, where e_i denotes the ith[1] unit vector. So we have $\operatorname{circ}_\alpha(x^i x^j) = \operatorname{circ}_\alpha(x^i) \operatorname{circ}_\alpha(x^j)$ for all $\{i, j\} \subset \mathbb{N}$. By linear extension it follows that $\operatorname{circ}_\alpha$ is a monomorphism of R-algebras. Hence the image of $\operatorname{circ}_\alpha$, which is the set of the α-circulant $(k \times k)$-matrices over R, forms a commutative subalgebra of the R-algebra $R^{k \times k}$ and it is isomorphic to the R-algebra $R[x]/(x^k - \alpha)$. Especially, we get $\operatorname{circ}_\alpha(a_0, \dots, a_{k-1}) = \sum_{i=0}^{k-1} a_i T_\alpha^i$.

2 Double α-circulant and bordered α-circulant codes

Definition 2.1 *Let R be a commutative ring and $\alpha \in R$. Let A be an α-circulant matrix. A code generated by a generator matrix $(I_k \mid A)$ is called* double α-circulant code. *A code generated by a generator matrix*

$$\left(\begin{array}{c|c|c} & \beta & \gamma \cdots \gamma \\ & \delta & \\ I_k & \vdots & A \\ & \delta & \end{array} \right)$$

with $\{\beta, \gamma, \delta\} \subset R\}$ is called bordered α-circulant code. *The number of rows of such a generator matrix is denoted by k, and the number of columns is denoted by $n = 2k$.*

As usual, two codes C_1 and C_2 are called *equivalent* or *isomorphic*, if there is a monomial transformation that maps C_1 to C_2.

Definition 2.2 *Let R be a commutative ring and $k \in \mathbb{N}$. The symmetric group over the set $\{0, \dots, k-1\}$ is denoted by S_k. For a permutation $\sigma \in S_k$ the permutation matrix $S(\sigma)$ is defined as $S_{ij} = \delta_{i,\sigma(j)}$, where δ is the Kronecker delta. An invertible matrix $M \in \mathrm{GL}(k, R)$ is called* monomial, *if $M = S(\sigma)D$ for a permutation $\sigma \in S_k$ and an invertible diagonal matrix D. The decomposition of a monomial matrix into the permutational and the diagonal matrix part is unique.*

[1] Throughout this article, counting starts at 0. Accordingly, $\mathbb{N} = \{0, 1, 2, \dots\}$

Let $\mathfrak{M} = \mathfrak{M}(k, R, \alpha)$ be the set of all pairs (N, M) of monomial $(k \times k)$-matrices M and N over R, such that for each α-circulant matrix $A \in R^{k \times k}$, the matrix $N^{-1}AM$ is again α-circulant. An element (N, M) of $\mathfrak{M}$ can be interpreted as a mapping $R^{k \times k} \to R^{k \times k}$, $A \mapsto N^{-1}AM$. The composition of mappings implies a group structure on $\mathfrak{M}$, and $\mathfrak{M}$ operates on the set of all α-circulant matrices.

Now let $(N, M) \in \mathfrak{M}$. The codes generated by $(I \mid A)$ and by $(I \mid N^{-1}AM)$ are equivalent, since

$$N^{-1}(I \mid A) \begin{pmatrix} N & 0 \\ 0 & M \end{pmatrix} = (I \mid N^{-1}AM)$$

and the matrix $\begin{pmatrix} N & 0 \\ 0 & M \end{pmatrix}$ is monomial. Thus, $\mathfrak{M}$ also operates on the set of all double α-circulant generator matrices.

In general $\mathfrak{M}$-equivalence is weaker than the code equivalence: For example the vectors $v = (1111101011011010) \in \mathbb{Z}_2^{16}$ and $w = (1110010011100000) \in \mathbb{Z}_2^{16}$ generate two equivalent binary double circulant self-dual $[32, 16]$-codes. But since the number of zeros in v and w is different, the two circulant matrices generated by v and w cannot be in the same $\mathfrak{M}$-orbit.

3 Monomial transformations of α-circulant matrices

Let R be a commutative ring, $k \in \mathbb{N}$ and $\alpha \in R$ a unit. In this section we give some elements (N, M) of the group $\mathfrak{M} = \mathfrak{M}(R, k, \alpha)$ defined in the last section. In part they can be deduced from [4], chapter 16, §6, problem 7.

Quite obvious elements of $\mathfrak{M}$ are (I_k, T_α), (T_α, I_k), (I_k, D) and (D, I_k), where D denotes an invertible scalar matrix.

For certain α further elements of $\mathfrak{M}$ are given by the following lemma, which is checked by a calculation:

Lemma 3.1 *Let $\alpha \in R$ with $\alpha^2 = 1$ and $s \in \{0, \ldots, k-1\}$ with $\gcd(s, k) = 1$. Let $\sigma = (i \mapsto si \mod k) \in S_k$. We define D as the diagonal matrix which has $\alpha^{(s+1)i + \lfloor si/k \rfloor}$ as i-th diagonal entry, and we define the monomial matrix $M = S(\sigma)D$. Then*

$$(M, M) \in \mathfrak{M}$$

More specifically: Let $f \in R[x]/(x^k - \alpha)$. It holds:

$$M^{-1} \operatorname{circ}_\alpha(f) M = \operatorname{circ}_\alpha(f((\alpha x)^s))$$

Finally, there is an invertible transformation $A \mapsto M^{-1}AM$ that converts an α-circulant matrix into a β-circulant matrix for certain pairs (α, β):

Lemma 3.2 *Let R be a commutative ring, $\alpha \in R$ a unit and $\{i, j\} \subset \mathbb{N}$. Let A be an α^i-circulant $(k \times k)$-matrix over R and M the diagonal matrix with the diagonal vector $(1, \alpha^j, \alpha^{2j}, \ldots, \alpha^{(k-1)j})$. Then $M^{-1}AM$ is an α^{i-kj}-circulant matrix. For $\alpha^2 = 1$ the matrix M is orthogonal.*

4 The lift of an α-circulant matrix

If we want to construct all equivalence classes of double α-circulant codes over a commutative ring R, it is enough to consider orbit representatives of the group action of $\mathfrak{M}$ on the set of all double α-circulant generator matrices, or equivalently, on the set of all α-circulant matrices.

Furthermore, we can benefit from non-trivial ideals of R: Let I be an ideal of R with $\{0\} \neq I \neq R$, and $^- : R \to R/I$ the canonical projection of R onto R/I. We set $\mathfrak{M} = \mathfrak{M}(k, R, \alpha)$ and $\bar{\mathfrak{M}} = \{(\bar{N}, \bar{M}) : (N, M) \in \mathfrak{M}\}$. It holds $\bar{\mathfrak{M}} \subseteq \mathfrak{M}(k, R/I, \bar{\alpha})$. Let $e : R/I \to R$ be a mapping that maps each element $r + I$ of R/I to an representative element $r \in R$.

Definition 4.1 *Let $A = \mathrm{circ}_{\bar{\alpha}}(v)$ be an $\bar{\alpha}$-circulant matrix with generating vector $v \in R/I$. An α-circulant matrix B over R is called* lift *of A, if $\bar{B} = A$. In this case we also say that the code generated by $(I_k \mid B)$ is a lift of the code generated by $(I_k \mid A)$. The lifts of A are exactly the matrices of the form $\mathrm{circ}_\alpha(e(v)) + \mathrm{circ}_\alpha(w)$ with $w \in I^k$.*[2] *The vector w is called* lift vector.

To find all double α-circulant codes over R, we can run over all lifts of all double $\bar{\alpha}$-circulant codes over R/I. The crucial point now is that for finding at least one representative all equivalence classes of double α-circulant codes over R, it is enough to run over the lifts of a *set of representatives* of the group action of $\bar{\mathfrak{M}}$ on the set of all $\bar{\alpha}$-circulant codes over R/I:

Lemma 4.1 *Let A and B be two $\bar{\alpha}$-circulant matrices over R/I which are in the same $\bar{\mathfrak{M}}$-orbit. Then for each lift of A there is a lift of B which is in the same $\mathfrak{M}$-orbit.*

Proof. Because A and B are in the same $\bar{\mathfrak{M}}$-orbit, there is a pair of monomial matrices $(N, M) \in \mathfrak{M}$ such that $\bar{N}^{-1} A \bar{M} = B$. Let $a \in (R/I)^k$ be the generating vector of A and $b \in (R/I)^k$ the generating vector of B. Since $\overline{\mathrm{circ}_\alpha(e(a))} = A$ and $\overline{\mathrm{circ}_\alpha(e(b))} = B$ it holds $N^{-1}\,\mathrm{circ}_\alpha(e(a))M = \mathrm{circ}_\alpha(e(b)) + K$, where $K \in I^{k \times k}$. $\mathrm{circ}_\alpha(e(b))$ is of course α-circulant, and $N^{-1}\,\mathrm{circ}_\alpha(e(a))M$ is α-circulant because of $(N, M) \in \mathfrak{M}$. Thus, also K is α-circulant and therefore there is a $z \in I^k$ with $\mathrm{circ}_\alpha(z) = K$.

Now, let $w \in I^k$ be some lift vector. $N^{-1}\,\mathrm{circ}_\alpha(w)M \in I^{k \times k}$ is α-circulant and generated by a lift vector $w' \in I^k$. Then $N^{-1}(\mathrm{circ}_\alpha(e(a)) + \mathrm{circ}_\alpha(w))M = \mathrm{circ}_\alpha(e(b)) + \mathrm{circ}_\alpha(z + w')$, and $z + w' \in I^k$. Therefore, the lift of A by the lift vector w and the lift of B by the lift vector $z + w'$ are in the same $\mathfrak{M}$-orbit. □

It is not hard to adapt this approach to bordered α-circulant codes. One difference is an additional restriction on the appearing monomial matrices: Its diagonal part must be a scalar matrix. The reason for this is that otherwise the monomial transformations would destroy the border vectors $(\gamma \ldots \gamma)$ and $(\delta \ldots \delta)^t$.

Circulant matrices are often used to construct self-dual codes. Thus we are interested in a fast way to generate the lifts that lead to self-dual codes. The next section gives such an algorithm for the case that R is a finite chain ring and I is its minimal ideal.

[2] To avoid confusion, we point out that I^k denotes the k-fold Cartesian product $I \times \ldots \times I$ here.

5 Self-dual double α-circulant codes over finite commutative chain rings

We want to investigate self-dual double α-circulant codes. Here we need $\alpha^2 = 1$. This is seen by denoting the rows of a generator matrix G of such a code by $w_0 \dots w_{k-1}$, and by comparing the scalar products $\langle w_0, w_1\rangle$ and $\langle w_1, w_2\rangle$, which must be both zero. Furthermore, given $\alpha^2 = 1$, we see that $\langle w_0, w_i\rangle = \langle w_j, w_{i+j}\rangle$, where $i + j$ must be read modulo k. Thus G generates a self-dual code if $\langle w_0, w_0\rangle = 1$ and for all $j \in \{1, \dots, \lfloor k/2\rfloor\}$ the scalar products $\langle w_0, w_j\rangle$ are equal to 0.

Definition 5.1 *A ring R is called* chain ring, *if its left ideals are linearly ordered by inclusion.*

For the theory of finite chain rings and linear codes over finite chain rings see [2].

In this section R will be a finite commutative chain ring, which is not a finite field, and α an element of R with $\alpha^2 = 1$. There is a ring element $\theta \in R$ which generates the maximal ideal $R\theta$ of R. The number q is defined by $R/R\theta \cong \mathbb{F}_q$, and m is defined by $|R| = q^m$. Because R is not a field, we have $m \geq 2$. The minimal ideal of R is $R\theta^{m-1}$. $\mathfrak{M}$ is defined as in section 2, with with the difference that all monomial matrices M should be orthogonal, that is $MM^t = I_k$. Thus each $\mathfrak{M}$-image of a generator matrix of a self-dual code again generates a self-dual code.

Now let $I = R\theta^{m-1}$ be the minimal ideal of R. As in section 4 let $e : R/I \to R$ be a mapping that assigns each element of R/I to an representative in R, now with the additional condition $e(\bar{\alpha}) = \alpha$.

We mention that if $(I_k \mid B)$ generates an double α-circulant self-dual code over R, then $(I_k \mid \bar{B})$ generates a double $\bar{\alpha}$-circulant self-dual code over R/I. So B is among the lifts of all $\bar{\alpha}$-circulant matrices A over R/I such that $(I_k \mid A)$ generates a self-dual double $\bar{\alpha}$-circulant code.

Let $A = \mathrm{circ}_{\bar{\alpha}}(a)$ be an $\bar{\alpha}$-circulant matrix over R/I such that $(I_k \mid A)$ generates a self-dual code. So $AA^t = -I_k$, and therefore

$$c_0 := 1 + \sum_{i=0}^{k-1} e(a_i)^2 \in I \quad \text{and}$$

$$c_j := \sum_{i=0}^{j-1} \alpha e(a_i)e(a_{k-j+i}) + \sum_{i=j}^{k-1} e(a_i)e(a_{i-j}) \in I \quad \text{for all } j \in \{1, \dots, \lfloor k/2\rfloor\}$$

We want to find all lifts $B = \mathrm{circ}_\alpha(e(a)) + \mathrm{circ}_\alpha(w)$ of A with $w \in I^k$ such that $BB^t = -I_k$. As we have seen, this is equivalent to

$$0 = 1 + \sum_{i=0}^{k-1}(e(a_i) + w_i)^2 \quad \text{and}$$

$$0 = \sum_{i=0}^{j-1}(e(a_i) + w_i)(\alpha e(a_{k-j+i}) + w_{k-j+i}) + \sum_{i=j}^{k-1}(e(a_i) + w_i)(e(a_{i-j}) + w_{i-j})$$

where the second equation holds for all $j \in \{1, \ldots, \lfloor k/2 \rfloor\}$. Using $I \cdot I = 0$, we get

$$0 = c_0 + 2\sum_{i=0}^{k-1} e(a_i)w_i \quad \text{and}$$
$$0 = c_j + \sum_{i=0}^{j-1}(e(a_i)w_{k-j+i} + \alpha e(a_{k-j+i})w_i) + \sum_{i=j}^{k-1}(e(a_i)w_{i-j} + e(a_{i-j})w_i)$$

This is a R-linear system of equations for the components $w_i \in I$ of the lift vector. Using the fact that the R-modules $R/(R\theta)$ and I are isomorphic, and $R/(R\theta) \cong \mathbb{F}_q$, this can be reformulated as a linear system of equations over the finite field $\mathbb{F}_q$, which can be solved efficiently.

Since R/I is again a commutative chain ring, the lifting step can be applied repeatedly. Thus, starting with the codes over $\mathbb{F}_q$, the codes over R can be constructed by $m - 1$ nested lifting steps.

Again, this method can be adapted to bordered α-circulant matrices over commutative finite chain rings.

6 Application: Self-dual codes over $\mathbb{Z}_4$

For a fixed length n we want to find the highest minimum Lee distance d_{Lee} of double nega-circulant and bordered circulant self-dual codes over $\mathbb{Z}_4$. In [5] codes of the bordered circulant type of length up to 32 were investigated.

First we notice that the length n must be a multiple of 8: Let C be a bordered circulant or a double nega-circulant code of length n and c a codeword of C. We have $0 = \langle c, c \rangle = \sum_{i=0}^{n-1} c_i^2 \in \mathbb{Z}_4$. The last expression equals the number of units in c modulo 4, so the number of units of each codeword is a multiple of 4. It follows that the image $\bar{C}$ of C over $\mathbb{Z}_2$ is a doubly-even self-dual code of length n, which can only exist for lengths n divisible by 8.

Furthermore, it holds

$$d_{\text{Lee}}(C) \leq 2d_{\text{Ham}}(\bar{C}) \tag{1}$$

As a result, we only need to consider the lifts of codes $\bar{C}$ which have a sufficiently high minimum Hamming distance.

We explain the algorithm for the case of the nega-circulant codes: In a first step, for a given length n we generate all doubly-even double circulant self-dual codes over $\mathbb{Z}_2$. This is done by enumerating Lyndon words of length n which serve as generating vectors for the circulant matrix. Next, we filter out all duplicates with respect to the group action of $\overline{\mathfrak{M}}$, where $\mathfrak{M}$ is the group generated by the elements given in section 3 which consist of pairs of orthogonal monomial matrices.

A variable d will keep the best minimum Lee distance we already found. We initialize d with 0. Now we loop over all binary codes $C_{\mathbb{Z}_2}$ in our list, from the higher to the lower minimum Hamming distance of $C_{\mathbb{Z}_2}$: If $2d_{\text{Ham}}(C_{\mathbb{Z}_2}) \leq d$ we are finished because of (1). Otherwise, as explained in section 5, we solve a system of linear equations over $\mathbb{Z}_2$ and get all self-dual lifts of $C_{\mathbb{Z}_2}$. For these lifts we compute the minimum Lee distance and update d accordingly.

Most of the computation time gets used on the computation of the minimum Lee distance. Thus it was a crucial point to write a specialized algorithm for this purpose. It is described in [3].

The results of our search are displayed in the following table. For given length n, it lists the highest minimum Lee distance of a self-dual code of the respective type:

n	8	16	24	32	40	48	56	64
double nega-circulant	6	8	12	14	14	18	16	20
bordered circulant	6	8	12	14	14	18	18	20

We see that the results are identical for the two classes of codes, except for length 56. Using (1) there is a simple reason that for this length no double circulant self-dual code over $\mathbb{Z}_4$ with minimum Lee distance greater than 16 exists: The best doubly-even double circulant self-dual binary code has only minimum Hamming distance 8.

Acknowledgment. This research was supported in part by Deutsche Forschungsgemeinschaft **WA 1666/4-1**.

References

[1] P. J. Davis, *Circulant Matrices*, Chelsea publishing, New York, second edition, 1994.

[2] T. Holold, I. Landjev, Linear codes over finite chain rings, *Electr. J. Comb.* 7, 2000.

[3] M. Kiermaier, A. Wassermann, On the minimum Lee distance of quadratic residue codes over $\mathbb{Z}_4$, *Proc. Intern. Symp. Inform. Theory*, 2008, to appear.

[4] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[5] M. Harada T. A. Gulliver, Extremal double circulant Type II codes over $\mathbb{Z}_4$ and construction of 5-(24, 10, 36) designs, *Discr. Math.* 194, 1999, 129-137.

Semiovals in projective planes of small order

GYÖRGY KISS[1] kissgy@cs.elte.hu
Department of Geometry, Eötvös Loránd University,
H-1117 Budapest, Pázmány s. 1/c, HUNGARY and
Bolyai Institute, University of Szeged,
H-6720 Szeged, Aradi vértanúk tere 1, HUNGARY

S. MARCUGINI[†], F. PAMBIANCO[2] gino, fernanda@dipmat.unipg.it
Department of Mathematics and Informatics, University of Perugia,
I-06123 Perugia, via Vanvitelli 1, ITALY

Abstract. Semiovals in PG$(2, q)$ for $q \leq 13$ are investigated. New examples are constructed, some characterization theorems and non-existence results of semiovals with extra properties are proved.

1 Introduction

Let Π be a projective plane of order q. A *semioval* in Π is a non-empty pointset $\mathcal{S}$ with the property that for every point in $\mathcal{S}$ there exists a unique line t_P such that $\mathcal{S} \cap t_P = \{P\}$. This line is called the tangent to $\mathcal{S}$ at P. The classical examples of semiovals arise from polarities (ovals and unitals), and from the theory of blocking sets (the vertexless triangle). The semiovals are interesting objects in their own right, but the study of semiovals is also motivated by their applications to cryptography. Batten [1] constructed an effective message sending scenario which use determining sets. She proved that determining sets in projective planes correspond to blocking semiovals. A *blocking semioval* is a semioval $\mathcal{S}$ such that every line of Π contains at least one point of $\mathcal{S}$ and at least one point which is not in $\mathcal{S}$. A blocking semioval that can be constructed in every projective plane of order $q > 2$ is the vertexless triangle.

It is known that $q + 1 \leq |\mathcal{S}| \leq q\sqrt{q} + 1$ and both bounds are sharp [10], [6], the extremes occur when $\mathcal{S}$ is an oval or a unital. In Section 2 we give the complete spectrum of the sizes of semiovals for $q \leq 9$. Besides, we determine the number of distinct semiovals up to collineations for $q \leq 7$. We also present the classification of small size semiovals for $q = 8, 9$ and new examples for $q = 11$ and 13. These semiovals were found by computer search.

Blocking semiovals in PG$(2, 7)$ were classified by Ranson and Dover [9]. The plane of order 7 contains several interesting semiovals. In Section 3 some characterization theorems for these semiovals are given.

2 On the spectrum of size for $q \leq 13$

For planes of order $q \leq 5$ the complete spectrum of the sizes and the number of projectively non-isomorphic semiovals has been known.

Case $q = 2$. Because of the bounds of the size, each semioval consists of three points, and these points are not collinear, hence semiovals are ovals.

Case $q = 3$. If a semioval $\mathcal{S}$ is not an oval, then there is a line ℓ which contains three points of $\mathcal{S}$, say A, B and C. There are four lines through each of these points, one of them is the tangent, but the others must meet $\mathcal{S}$. Hence $\mathcal{S}$ contains at least two points not on ℓ. Let $D, E \in \mathcal{S} \setminus \ell$. If F is the fourth point of the line ℓ, then $t_D \cap \ell = t_E \cap \ell = F$, thus $DE \cap \ell \neq F$. Without loss of generality we may assume, that $DE \cap \ell = A$. This implies that $\mathcal{S}$ must contain a sixth point G, otherwise there would be two tangents through A. But 6 is an upper bound of the cardinality of $\mathcal{S}$ because $\lfloor 3\sqrt{3} + 1 \rfloor = 6$. If $G = BD \cap CE$, then it is easy to check that the set $\{A, B, C, D, E, G\}$ is a semioval. These points form the vertices of a complete quadrilateral. Hence there is only one projectively non-isomorphic class of semiovals of order six in PG$(2, 3)$.

Case $q = 4$. The possible sizes of $\mathcal{S}$ are $5, 6, 7, 8$ and 9. If $|\mathcal{S}| = 5$, then $\mathcal{S}$ is an oval. If $|\mathcal{S}| > 5$, then $\mathcal{S}$ contains three collinear points. Semiovals with large secants were investigated by Dover [4]. He proved that if $\mathcal{S}$ is a semioval in a projective plane of order $q > 3$, then $\mathcal{S}$ does not contain q collinear points, and if $|\mathcal{S}| = 2q-1$, then $\mathcal{S}$ has no $(q-1)$-secant. In our case $\mathcal{S}$ has $3 = q-1$ collinear points, hence $|\mathcal{S}| \neq 7 = 2q-1$. The cases $|\mathcal{S}| = 2q-2 = 6$ and $|\mathcal{S}| = 9 = 3q-3$ are also characterized by Dover [4], these are a triangle with its vertices and all points on one side removed, and the vertexless triangle, respectively. Let us remark that in PG$(2, 4)$ each unital is a vertexless triangle and vice versa. If $|\mathcal{S}| = 8$, then an exhaustive computer search shows that the only semiovals of this size are vertexless triangles with one point deleted.

For $q > 4$ the situation becomes more and more complicated. Semiovals of size $2(q-1) + k$ for all $0 \leq k \leq q-1$ and $k \neq 1$ can be constructed easily. If we delete any set of $q-1-k$ points from one side of a vertexless triangle, then the remaining points form a semioval $\mathcal{S}$ and $|\mathcal{S}| = 2(q-1) + k$. Hence the spectrum of sizes always contains $2q-2$ and all integers in the interval $[2q, 3q-3]$. For $q \leq 9$, by exhaustive computer search, we found the following sizes.

Theorem 2.1 *The spectrum of the sizes of semiovals in PG$(2, q)$ is the following:*

- *If $q = 2$ then $|\mathcal{S}| = 3$.*
- *If $q = 3$ then $|\mathcal{S}| \in \{4, 6\}$.*
- *If $q = 4$ then $|\mathcal{S}| \in \{5, 6, 8, 9\}$.*

- *If* $q = 5$ *then* $|\mathcal{S}| \in \{6, 8, 9, 10, 11, 12\}$.
- *If* $q = 7$ *then* $|\mathcal{S}| \in \{8, 9, 12, 13, 14, 15, 16, 17, 18, 19\}$.
- *If* $q = 8$ *then* $|\mathcal{S}| \in \{9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23\}$
- *If* $q = 9$ *then*
  $|\mathcal{S}| \in \{10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28\}$.

For $q \leq 7$ we have determined the number of non-equivalent semiovals up to collineations. For $q \leq 4$ there is only one class for each size, as follows from the previous description. For $q = 5, 7$ the results are summarized in Table 1.
For $q = 8, 9$ we have classified the examples of minimum order which are not ovals. In both cases the minimum order is twelve and there are, respectively, four and one classes. Besides, for $q = 8$, we have proven that there are only two classes of semiovals of size 13.

PG(2,5)	size of $\mathcal{S}$	6	8	9	10	11	12				
	# of distinct classes	1	1	2	3	2	1				
PG(2,7)	size of $\mathcal{S}$	8	9	12	13	14	15	16	17	18	19
	# of distinct classes	1	1	10	21	69	118	82	21	7	1

Table 1

We have also found examples of the following sizes:

Theorem 2.2

- *In* $PG(2, 11)$ *there are semiovals of size* $12, 15, 20, 22 - 34$.
- *In* $PG(2, 13)$ *there are semiovals of size* $14, 18, 24, 26 - 40$.

3 The exceptional semiovals in PG(2, 7)

There are some interesting semiovals in PG(2, 7). The first one has only $q + 2$ points. If $q = 7$, then $q + 2 = 3(q - 1)/2$, and the semioval belongs to an infinite class of semiovals which was described by Kiss and Ruff [8]. The following classification theorem is a consequence of a result of Blokhuis [3].

Theorem 3.1 *If* $|\mathcal{S}| = q + 2$, q *odd, then* $q = 7$. $\mathcal{S}$ *is projectively equivalent to the set of points* $\{(0, 1, s), (s, 0, 1), (1, s, 0) : s \text{ is a square in } \mathrm{GF}(7)\}$, *hence it is contained in a vertexless triangle.* □

PG(2, 7) contains a semioval of size $13 = 2 \cdot 7 - 1$. There is no known infinite class of semiovals of size $2q - 1$. There are only three known semiovals of this size, they exist on the planes of order 5, 7 and 9. The following theorem of Faina, Kiss, Marcugini and Pambianco [5] characterizes the case $q = 7$.

Theorem 3.2 *If $|\mathcal{S}| = 2q - 1$ and $\mathcal{S}$ has a $(q-2)$-secant, then $q = 7$ and $\mathcal{S}$ has exactly two $(q-2)$-secants.* □

Batten and Dover [2] found a cyclic semioval in PG(2, 7). It follows from our computer search, that this semioval is projectively unique. Hence we have the following theorem.

Theorem 3.3 *If $\mathcal{S}$ is a semioval in $PG(2,7)$ then $|\mathcal{S}| \leq 19$. If $|\mathcal{S}| = 19$, then $\mathcal{S}$ is cyclic.* □

Cyclic semiovals are rare objects. There are only two known examples. The other one can be found in PG(2, 81), it has 511 points, see [5]. The following nonexistence result was proved by Faina, Kiss, Marcugini and Pambianco [5].

Theorem 3.4 *There is no cyclic semioval in $PG(2,q)$ if $q \equiv 2 \pmod 3$.* □

They also prowed by exhaustive computer search, that $\mathrm{PG}(2, 3^r)$ does not contain a cyclic semioval if $r \leq 11$ and $r \neq 4$.

References

[1] L. M. Batten, Determining sets, *Australas. J. Combin.* 22, 2000, 167-176.

[2] L. M. Batten, J. M. Dover, Blocking semiovals of type $(1, m+1, n+1)$, *SIAM J. Discr. Math.* 14, 2001, 446-457.

[3] A. Blokhuis, Characterization of seminuclear sets in a finite projective plane, *J. Geom.* 40, 1991, 15-19.

[4] J. M. Dover, Semiovals with large collinear subsets, *J. Geom.* 69, 2000, 58-67.

[5] G. Faina, Gy. Kiss, S. Marcugini, F. Pambianco, On the spectrum of the sizes of semiovals in PG(2,q), q odd, submitted.

[6] X. Hubaut, Limitation du nombre de points d'un (k, n)-arc regulier d'un plan projectif fini, *Atti. Accad. Naz. Lincei Rend.* 8, 1970, 490-493.

[7] Gy. Kiss, A survey on semiovals, *Contrib. Discr. Math.* 3, 2008, 81-95 (electronic).

[8] Gy. Kiss, J. Ruff, Notes on small semiovals, *Ann. Univ. Sci. Budapest*, Eötvös Sect. Math. 47, 2004, 97-105.

[9] B. B. Ranson, J. M. Dover, Blocking semiovals in PG(2, 7) and beyond, *Europ. J. Combin.* 24, 2003, 183-193.

[10] J. A. Thas, On semiovals and semiovoids, *Geom. Dedic.* 3, 1974, 229-231.

Double ±1-error correctable codes and their applications to modulation schemes

HRISTO KOSTADINOV[1] hristo@math.bas.bg
Institute of Mathematics and Informatics, BAS and
University of Electro-Communication, Chofu, Tokyo 182-8585, JAPAN

NIKOLAI L. MANEV nlmanev@math.bas.bg
Institute of Mathematics and Informatics,Bulgarian Academy of Sciences,
8 G.Bonchev str., 1113, Sofia, BULGARIA

HIROYOSHI MORITA morita@is.uec.ac.jp
University of Electro-Communication, Chofu, Tokyo 182-8585, JAPAN

Abstract. Codes capable to correct two errors of value ±1 in a codeword are constructed and studied. Large number of experiments simulating the implementation of several double ±1-error correctable codes in QAM-modulation schemes have been carried out. The obtained results present in graphical form the performance of the coded modulation schemes based on the considered codes versus signal-to-noise ratio (SNR). The results confirm the good performance of integer coded modulation in comparison to the other schemes for coded modulation.

1 Introduction

Coded modulation is the collective term for all techniques which combine and jointly optimize channel coding and modulation for digital transmission. As a result of more than thirty years intensive investigation numerous and multifarious methods for coded modulation have been proposed. Despite their variety, the coded modulation schemes can be classified in the following three large groups:

- **Trellis coded modulation (TCM)**: It consists in an expanding the input bits by a binary convolutional code and partitioning the used signal constellation into smaller subsets with a larger intra-set distance. A part of coded bits are used to select one of these subsets and the remaining determine which of the signal point in the chosen subset to be transmitted. The Ungerboeck's concept requires a larger signal set than the one used in the case of uncoded modulation.

[1] This work was partially supported by the Japan Society for the Promotion Science (JSPS).

- **Integer coded modulation (ICM)**: A type of block coded modulation - each point of the signal constellation corresponds to a symbol of $\mathbb{Z}_A$ and coded by a code over $\mathbb{Z}_A$.

- **Others**: Coded modulation based on Gaussian and algebraic integers ([3], [6], and others).

Integer codes have proved themselves to be very effective for coded modulation, where errors usually have a given type (see [4, 5]), that is, in the case of modulation schemes where the error-vectors are not equally probable. In partial M-QAM modulation fall in this case.

In this talk we address codes over integer rings which are capable to correct up two errors with values ±1. We demonstrate their practical potential by numerous simulations and comparisons with one error correctable integers codes and other types of coded modulations.

2 General remarks

Let C be an $[n, k]$ code over the integer ring $\mathbb{Z}_A$. Recall that a t-multiple $(\pm e_1, \pm e_2, \ldots, \pm e_s)$-error correctable code is a code that can correct any up to t errors with values from the set $\{\pm e_i, | \, i = 1, \ldots, s\}$ occurred in a codeword ([1, 4]). Single error correctable codes are discussed in [4, 5]. Herein we restrict our consideration only to the double ±1-error correctable codes. These codes are interesting since they can be effectively applied to improving the performance of Quadrature Amplitude Modulation (QAM) schemes.

Proposition 1 *Let C be an $[n, k]$ code over the ring $\mathbb{Z}_A$. If C is a double ±1-error correctable code, then the cardinality, A, of the ring satisfies the inequalities:*
when $k = n - 1$

$$A \geq 2n^2 + 1;$$

when $k = n - 2$

$$A \geq \sqrt{2n^2 + 1}$$

Proof. More generally, the number of the different error vectors when up to t error with values ±1 occur per a codeword is

$$N_t = \sum_{j=0}^{t} \binom{n}{j} 2^j.$$

This number, N_t, has to be less or equal to the number of possible syndromes. In the case $k = n-1$ the syndromes are elements of $\mathbb{Z}_A$, that is, their number is A. When $k = n-2$, the syndromes $\mathbf{s} \in \mathbb{Z}_A \times \mathbb{Z}_A$, and their number is A^2. Hence we must have $A \geq N_2$, respectively $A^2 \geq N_2$, which give the statement.

Let C be a double ± 1-error correctable code with a parity-check matrix

$$\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n),$$

where the length of the columns is $n-k$, i.e. one ($k = n-1$) or two ($k = n-2$). Therefore $\mathbf{h}_i \neq \pm\mathbf{h}_j,\ i \neq j$, and $\mathbf{h}_i \pm \mathbf{h}_j \neq \pm(\mathbf{h}_l \pm \mathbf{h}_m)$. Also, the permutations and multiplications of columns by -1 transform C into an equivalent code. Hence, we may assume that the first row of $\mathbf{H}$ contains only elements $\leq A/2$, arranged in a nondecreasing order.

Also, the multiplication of a row of $\mathbf{H}$ by an invertible element of $\mathbb{Z}_A$ does not change the code. Hence if there exists an invertible entry of $\mathbf{H}$ we may assume that there is 1 in the first row. Otherwise there is an element that divide A, and all others have g.c.d. with A greater than 1.

Therefore we can assume that the parity check matrix of $[n, n-2]$ double ± 1-error correctable code has the form

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & h_{13} & \ldots & h_{1n} \\ 0 & 1 & h_{23} & \ldots & h_{2n} \end{pmatrix} \quad \text{or} \quad \mathbf{H} = \begin{pmatrix} 1 & h_{12} & h_{13} & \ldots & h_{1n} \\ 0 & a & h_{23} & \ldots & h_{2n} \end{pmatrix},$$

where $a \mid A$.

In partial an interesting case is the group of matrices of the form

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 2 & 3 & \ldots & n-1 \\ 0 & 1 & h_{23} & h_{24} & \ldots & h_{2n} \end{pmatrix}$$

over a ring $\mathbb{Z}_A$ with $A \geq 2n-2$.

Unfortunately, if a code with a given parity-check matrix is double ± 1-error correctable for a given alphabet $\mathbb{Z}_A$ it may not preserve this property as a code over a larger cardinality of the alphabet. For example, the code with a parity-check matrix

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 5 & 3 & 6 & 2 & 4 \end{pmatrix}$$

is a double ± 1-error correctable code over $\mathbb{Z}_{15}$, but does not preserve this property over $\mathbb{Z}_{16}$. On the other hand the code with a parity-check matrix given in Example 1 is such a code over both $\mathbb{Z}_8$ and $\mathbb{Z}_9$.

In the case $k = n-1$ the parity-check matrix is $1 \times n$ and has the form $\mathbf{H} = (1\ h_2\ \ldots\ h_n)$. But according Proposition 1 such codes require large cardinality, A, of the alphabet. Nevertheless a $[2,1]$ code over $\mathbb{Z}_9$ with $\mathbf{H} = (31)$ demonstrates very good performance for 64-QAM.

From practical point of view the codes over $\mathbb{Z}_{2^m}$ or $\mathbb{Z}_{2^m+1}$ are more interesting since they enable the standard 2^{2m}-QAM constellations to be used.

We have found many codes for small length and reasonable alphabet cardinality, but the full classification has not been completed yet.

3 Applications and simulation results

In this section we demonstrate how a double ±1-error correctable code over $\mathbb{Z}_{2^m+1}$ can be used in real applications for improving the performance of 2^{2m}-QAM.

Example 1. Consider $[4,2]$ code C over $\mathbb{Z}_9$ with a parity-check matrix $\mathbf{H}$ and the corresponding generator matrix $\mathbf{G}$:

$$\mathbf{H} = \begin{pmatrix} 5 & 3 & 1 & 0 \\ 2 & 3 & 0 & 1 \end{pmatrix} \qquad \mathbf{G} = \begin{pmatrix} 1 & 0 & 4 & 7 \\ 0 & 1 & 6 & 6 \end{pmatrix}.$$

The code is double ±1-error correctable and we apply it to 64-QAM modulation scheme in order to correct errors of type "big square" (see [5]). Recall that in such a coding scheme each point of the constellation is indexed by a pair (x, y) of nonzero elements of an integer ring (in this example Z_9) as shown in Fig.1.

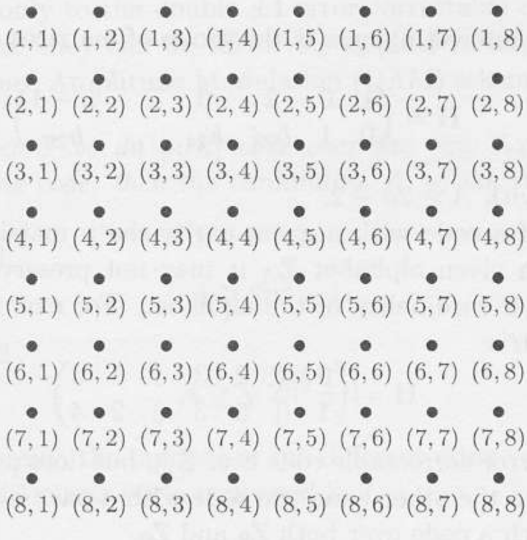

Figure 1: Indexing a 64-QAM constellation

Encoding and decoding procedures

The encoding and decoding procedure on each of the axes are separated. More detailed: Any incoming block of 6 bits is split into two 3-bit groups which are transformed into decimal integers. By adding 1 to each of them we obtain a pair (a, b) of nonzero elements of $\mathbb{Z}_9$. Each of the sequences $a_1, a_2, \ldots$, resp. $b_1, b_2, \ldots$, of the first, resp. the second, coordinates is encoded by the code C. Therefore

$$(a_1, a_2) \longrightarrow (a_1, a_2, 4a_1 + 6a_2, 7a_1 + 6a_2),$$

where the operations are in $\mathbb{Z}_9$. Note that $4a_1 + 6a_2 = 0 \Leftrightarrow 7a_1 + 6a_2 = 0 \Leftrightarrow a_1 = 3a_2$. Since the values of the check bits have to be also nonzero, we replace them with 1 when $a_1 = 3a_2$, that is, $(3a, a) \longrightarrow (3a, a, 1, 1)$.

At the receiver, for each of the axes, the detection procedure (hard or soft) gives as an output a vector $\mathbf{v} = (v_1, v_2, v_3, v_4)$, where $v_j \in \mathbb{Z}_9^*$, . The decoder proceeds both vectors in parallel following the standard syndrome decoding scheme giving at the output a pair (u_1, u_2). The only peculiarity is that after calculating the syndrome vector $\mathbf{s} = \mathbf{vH}$ the decoder uses the syndrome-error table two times: for $\mathbf{s}$ and for $\mathbf{s} - (1, 1)$. In the latter case if the output pair (u_1, u_2) does not satisfy $u_1 = 3u_2$, the result is discarded. Also, if $\mathbf{s}$ does not match to any vector in the table, the decoder gives $u_1 = v_1$ and $u_2 = v_2$. The correspondence between error frames and syndromes (error-syndrome table) is s given in Table 1.

Error vector	Syndrome	Error vector	Syndrome
1 0 0 0	(5,2)	-1 0 0 0	(4,7)
1 1 0 0	(8,5)	-1 -1 0 0	(1,4)
1 -1 0 0	(2,8)	-1 1 0 0	(7,1)
1 0 1 0	(6,2)	-1 0 -1 0	(3,7)
1 0 -1 0	(4,2)	-1 0 1 0	(5,7)
1 0 0 1	(5,3)	-1 0 0 -1	(4,6)
1 0 0 -1	(5,1)	-1 0 0 1	(4,8)
0 1 0 0	(3,3)	0 -1 0 0	(6,6)
0 1 1 0	(4,3)	0 -1 -1 0	(5,6)
0 1 -1 0	(2,3)	0 -1 1 0	(7,6)
0 1 0 1	(3,4)	0 -1 0 -1	(6,5)
0 1 0 -1	(3,2)	0 -1 0 1	(6,8)
0 0 1 0	(1,0)	0 0 -1 0	(8,0)
0 0 1 1	(1,1)	0 0 -1 -1	(8,8)
0 0 1 -1	(1,8)	0 0 -1 1	(8,1)
0 0 0 1	(0,1)	0 0 0 -1	(0,8)

Table 1: Error-syndrome table.

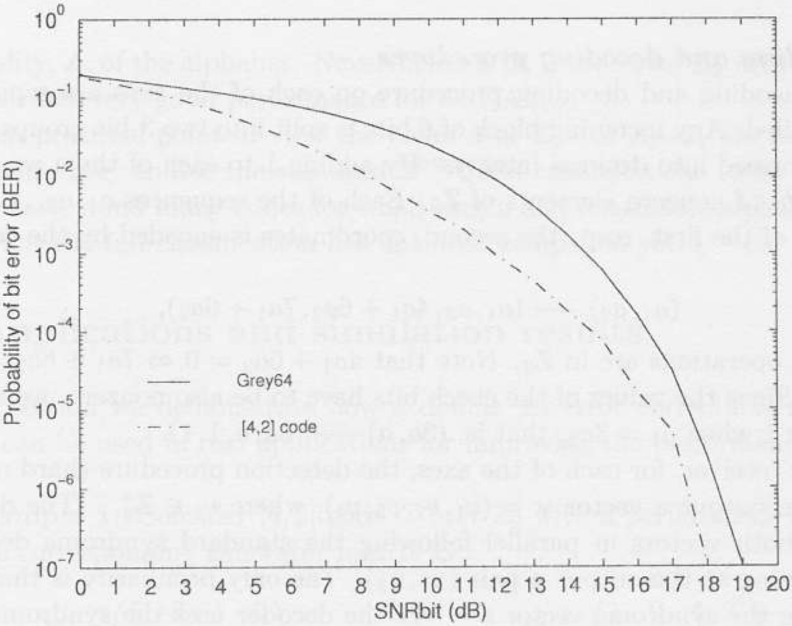

Figure 2: 64-QAM–Grey and $[4, 2]$ code over $\mathbb{Z}_9$. (Example 1).

References

[1] A. J. Han Vinck, H. Morita, Codes over the ring of integer modulo m, *IEICE Trans. Fundam.* E81-A, 1998, 2013-2018.

[2] G. Ungerboeck, Channel coding with multilevel/phase signals, *IEEE Trans. Inform. Theory* 28, 1982, 55-66.

[3] K. Huber, Codes over Gaussian integers, —it IEEE Trans. Inform. Theory 40, 1994, 207-216.

[4] H. Kostadinov, H. Morita, N. Manev, Integer codes correcting single errors of specific types $(\pm e_1, \pm e_2, \cdots, \pm e_s)$," *IEICE Trans. Fundam.* E86-A, 2003, 1843-1849.

[5] H. Kostadinov, H. Morita, N. Manev, Derivation on bit error probability of coded QAM using integer codes, *IEICE Trans. Fundam.* E87-A, 2004, 3397-3403.

[6] J. Rifa, Groups of complex integers used as QAM signals, *IEEE Trans. Inform. Theory* 41, 1995, 1512-1517.

Hard decision decoding of binary low-density codes based on extended Hamming codes

Alexey Kuznetsov ring0@list.ru
Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow 101447, RUSSIA

Abstract. In this paper we investigate decoding performance of binary low-density codes based on extended Hamming codes, using a hard decision decoding algorithm similar to Gallager's bit-flipping algorithm. The possibilities to implement the decoder in hardware are also discussed.

1 Introduction

Low-density parity-check codes were first introduced by Gallager [1]. In [2] asymptotical properties of low-density codes based on Hamming codes were investigated. The simulation results of the soft decision decoding of these codes were given in [3]. It has been also proven there that the minimum distance of such codes grows linearly with the code length. In this paper we study similar codes based on extended Hamming codes.

2 Code structure

The codes in question are defined by their parity-check matrix.

Let n_0 be a component code length. Let a matrix $\mathbf{H_1}$ be a parity-check matrix for the component code (extended Hamming code in our case). Let r_0 be a number of rows in this matrix. We can now construct a $n_0m \times r_0m$ -sized matrix $\mathbf{H_m}$ in the following way: every row $j = ir_0 + k$ ($k < r_0$, $i \leq m - 1$) is filled using k's row of $\mathbf{H_1}$ starting with $i * n0$ column. Other positions are filled with zeros. I. e.,

$$\mathbf{H_m} = \begin{bmatrix} \mathbf{H_1} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{H_1} & 0 & \dots & 0 \\ \dots & & & & \\ 0 & 0 & 0 & \dots & \mathbf{H_1} \end{bmatrix}$$

Now, a code's parity-check matrix can be defined as follows:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H_m P_1} \\ \mathbf{H_m P_2} \\ \dots \\ \mathbf{H_m P_l} \end{bmatrix}$$

Here $\mathbf{P_i}$ are permutation matrixes.

We will investigate a particular code class with the following parameters: $n_0 = 128$, $m = 256$, $r_0 = 8$, $l = 3$. We'll also consider P_1 to be a trivial transform, since it doesn't influence code BER performance.

3 Decoding algorithm

A simple hard decision decoding algorithm exists for these codes.

At first, one can decode every component code. There are three possibilities:

1. The syndrome equals **0**.
2. The syndrome doesn't equal **0** and corresponds to the particular error (a component code has 'detected the error at some position).
3. The syndrome doesn't equal **0** and doesn't correspond to any error. I. e., the decoder fails to decode this codeword.

Each bit in the codeword is contained in three component codes. For every component code word it is contained in, each bit is assigned one of four *states*:

1. This bit wasn't marked as erroneous and the syndrome of the component code word was **0**.
2. This bit was marked as erroneous by the component code decoder.
3. This bit wasn't marked as erroneous and the component code decoder failed to decode the corresponding code word.
4. This bit wasn't marked as erroneous, but the component code decoder marked other bit in the corresponding code word.

After that, for each bit the following decision process is invoked:

1. If the bit was corrected in one component codeword, and no component codeword containing this bit had zero syndrome, this bit is *inverted.*
2. if the bit was corrected in two component codewords, it is also *inverted*, regardless of the other codeword decoding result.

3. Otherwise, the bit is leaved as it is (i. e., *not inverted*).

The decoding process is now repeated until no component code can correct a error (i. e., every component codeword either has zero syndrome or the component code decoder fails to decode it). This condition will be called further a *stop condition*.

4 Simulation

The decoding algorithm described above was simulated for this code class. The permutations $\mathbf{P_2}$ and $\mathbf{P_3}$ were randomly generated. Binary symmetric channel (BSC) was used for simulation.

It was found that the most frequent error set leading to the stop condition consists of two errors, which are located in one codeword for every codeword group. This possibility can be eliminated if we arbitrarily choose $\mathbf{P_2}$ instead of choosing it randomly. This goal can be achieved, for example, in such a way:

$$i^* = 128(i \cdot \mathrm{mod} 128) + \left\lfloor \frac{i}{128} \right\rfloor,$$

for $i < 2^{14}$, and

$$i^* = 2^{14} + 128((i - 2^{14}) \cdot \mathrm{mod} 128) + \left\lfloor \frac{i - 2^{14}}{128} \right\rfloor,$$

for $i \geq 2^{14}$, where i is a column position and i^* is a new column position (counting from 0).

The simulation shows that the code with arbitrarily chosen $\mathbf{P_2}$ tends to have slightly better BER performance in the BSC, at least when the input error probability is considerably high.

The table below shows some simulation results. All probabilities here are bit error probabilities.

p_{in}	p_{out} (random $\mathbf{P_2}$)	p_{out} (arbitrarily chosen $\mathbf{P_2}$)
0.011	$6.9 \cdot 10^{-6}$	$4.6 \cdot 10^{-6}$
0.010	$8.2 \cdot 10^{-7}$	$3.2 \cdot 10^{-7}$
0.009	$4.3 \cdot 10^{-7}$	$9.8 \cdot 10^{-8}$

5 Hardware implementation

The decoding of this code requires a huge amount of data transfers, so parallel computing can hardly be benefited from. Instead, a simple sequential architecture can be proposed.

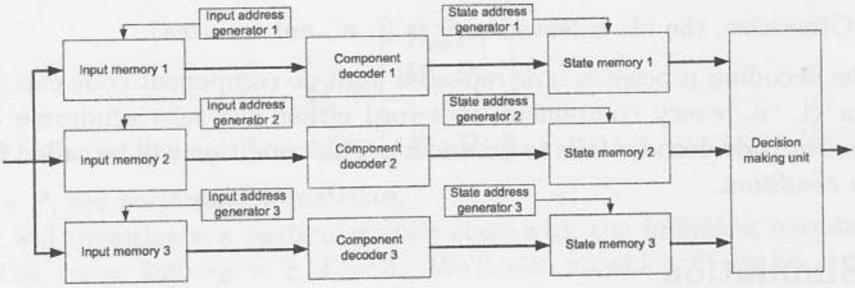

Figure 1: Decoder architecture

This architecture has quite a large memory requirements, but everything else will occupy quite a few logic cells.

The input memories are conventional 2-port static RAM blocks, configured for 1-bit data port width. This component should be available in any library. They are filled with identical data. Three blocks are needed to provide a possibility for simultaneous reading by the component code (extended Hamming) decoders. These three blocks can be, of course, replaced with one quad-port memory, if there's an appropriate component in the library.

Each of the component code decoders processes 1 bit per cycle. Each decoder contains a counter, which is used as a parity check matrix generator, and modulo 2 adder unit for syndrome calculation. The input data should be stored in internal memory until the syndrome is calculated. Then for each bit component code writes a bit value (unchanged) and a 2-bit state to the corresponding state memory.

Decision making unit then applies a decision process to every bit. The output data can be then iterated through the same hardware, at the cost of degraded processing performance. Alternatively, a long pipeline containing enough stages to make a decision in virtually any case can be used. Of course, this will lead to hardware duplication.

The memory requirements for the design can be estimated as follows:

- input memories - 32kbit $\cdot$ 3 = 96kbit.
- input address generators - $(32 \cdot 16) \cdot 3$kbit = 1536kbit (assuming 16-bit address width) in the general case. If the first permutation is trivial, and the second is arbitrarily chosen, the first two generators won't use memory, this will amount to 512kbit.
- component code decoders - negligible, about 128 bits each.
- stage memory address generators - like input address generators.

- stage memory blocks - $(32 \cdot 4) \cdot 3\text{kbit} = 384\text{kbit}$.

So, the whole design will use no less than 1504kbit of RAM, and 3552kbit in the general case.

The major drawback of the sequential architecture is that it can't process more than 1 bit per cycle. On modern programmable logic devices, such as Xilinx Virtex-5 chips, this amounts to about 500Mbit/s processing performance. Using a chip, designed specially for this problem (ASIC), one can reach about 1Gbit/s.

Getting processing performance considerably above 1Gbit/s is difficult due to the clock frequency limitations on semiconductor devices. To solve the problem, one can use several such decoders in parallel, possibly on different chips and with separate demultiplexer. This will lead to the increased latency, of course.

It should be also noted that in practical implementations the maximum number of iterations per codeword has to be limited. This will also lead to somewhat degraded BER performance.

6 Conclusion

The sequential architecture presented above has limited processing performance. In order to increase the performance further, one need to employ internal parallelism. It would be desirable to have a separate component code decoder for every component codeword. Unfortunately, this is connected with significant difficulties, since the problem requires a number of connections between the processing units, which seems to be beyond any practical limits.

References

[1] R. G. Gallager, *Low-density parity-check codes*, Cambridge: M.I.T. Press, 1963.

[2] S. Stiglmayr, V. V. Zyablov, Assymptotically good low-density codes based on Hamming codes, *Proc. Eleventh Intern. Symp. Probl. Redund. Inform. Contr. Syst.*, St. Petersburg, 2007.

[3] M. Lentmaier, K. S. Zigangirov, On generalized low-density parity-check codes based on Hamming component codes, *IEEE Commun. Lett.* 3, 1999.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 166-170*

Blocking sets of Rédei type in projective Hjelmslev planes

Ivan Landjev ivan@math.bas.bg
Institute of Mathematics and Informatics Bulgarian Academy of Sciences,
8 G. Bonchev str., 1113 Sofia, BULGARIA
New Bulgarian University, 21 Montevideo str., 1618 Sofia, BULGARIA

Stoyan Boev stoyan@nbu.bg
New Bulgarian University, 21 Montevideo str., 1618 Sofia, BULGARIA

1 Preliminary results

The aim of this paper is to generalize the notion of a Rédei type blocking set to projective Hjelmslev planes.

In what follows, we focus on Hjelmslev planes over chain rings of nilpotencey index 2, i.e. chain rings with $\operatorname{rad} R \neq (0)$ and $(\operatorname{rad} R)^2 = (0)$. Thus we have always $|R| = q^2$, where $R/\operatorname{rad} R \cong \mathbb{F}_q$. Chain rings with this property have been classified in [1, 6]. If $q = p^r$ there are exactly $r+1$ isomorphism classes of such rings. These are:

- for every $\sigma \in \operatorname{Aut} \mathbb{F}_q$ the ring $R_\sigma \cong \mathbb{F}_q[X;\sigma]/(X^2)$ of the so-called σ-dual numbers over $\mathbb{F}_q$ with underlying set $\mathbb{F}_q \times \mathbb{F}_q$, component-wise addition and multiplication given by $(x_0, x_1)(y_0, y_1) = (x_0y_0, x_0y_1 + x_1y_0^\sigma)$;
- the Galois ring $\mathrm{GR}(q^2, p^2) \cong \mathbb{Z}_{p^2}[X]/(f(X))$, where $f(X) \in \mathbb{Z}_{p^2}[X]$ is a monic polynomial of degree r, which is irreducible modulo p.

The rings R_σ with $\sigma \neq \mathrm{id}$ are noncommutative, while R_{id} is commutative. We have also that $\operatorname{char} R_\sigma = p$ for every σ. The Galois ring $\mathrm{GR}(q^2, p^2)$ is commutative and has characteristic p^2. From now on we denote by R a finite chain ring of nilpotency index 2.

In order to save space, we refer to [2, 3, 4] for the basic definitions and results about projective Hjelmslev planes over finite chain rings. We denote by $\mathrm{PHG}(R_R^3)$ the (right) projective Hjelmslev plane over the chain ring R. Similarly, $\mathrm{AHG}(R_R^2)$ denotes the (right) affine Hjelmslev plane over R.

Let $\Pi = (\mathcal{P}, \mathcal{L}, I)$ be a projective Hjelmslev plane. Any mapping from the pointset $\mathcal{P}$ to the nonnegative integers $\mathfrak{K} : \mathcal{P} \to \mathbb{N}_0$ is called a *multiset* in Π. The integer $\mathfrak{K}(P)$, $P \in \mathcal{P}$, is called the multiplicity of P. The mapping $\mathfrak{K}$ induces a mapping on the subsets of $\mathcal{P}$ by

$$\mathfrak{K}(\mathcal{Q}) = \sum_{P \in \mathcal{Q}} \mathfrak{K}(P), \quad \mathcal{Q} \subseteq \mathcal{P}.$$

The induced mapping is denoted again by $\mathfrak{K}$. The integer $|\mathfrak{K}| = \mathfrak{K}(\mathcal{P})$ is called the *cardinality* or the *size* of $\mathfrak{K}$. The support supp $\mathfrak{K}$ of a multiset $\mathfrak{K}$ is the set of points of positive multiplicity: supp $\mathfrak{K} = \{P \in \mathcal{P} \mid \mathfrak{K}(P) > 0\}$.

Two multisets $\mathfrak{K}'$ and $\mathfrak{K}''$ in the projective Hjelmslev plane Π are said to be equivalent if there exists a collineation σ in Π such that $\mathfrak{K}'(P) = \mathfrak{K}''(\sigma(P))$ for every point $P \in \mathcal{P}$.

Definition 1.1. *A multiset $\mathfrak{K}$ in $(\mathcal{P}, \mathcal{L}, I)$ is called a (k, n)*-blocking multiset *if*

(i) $\mathfrak{K}(P) = k$;

(ii) $\mathfrak{K}(\ell) \geq n$ *for every line* $\ell \in \mathcal{L}$;

(iii) there exists at least one line ℓ_0 *with* $\mathfrak{K}(\ell_0) = n$.

A (k, n)-blocking multiset $\mathfrak{K}$ is called *reducible* if there exists (k', n)-blocking multiset $\mathfrak{K}'$ with $k' < k$ and $\mathfrak{K}'(P) \leq \mathfrak{K}(P)$ for every point $P \in \mathcal{P}$. A blocking multiset that is not reducible is called *irreducible*.

A major problem is to determine the possible sizes of the irreducible blocking sets in the planes $\mathrm{PHG}(R_R^3)$, where R is a chain ring of nilpotency index 2. It is known that the minimal size of a blocking set in $\mathrm{PHG}(R_R^3)$, $|R| = q^2$, is $q^2 + q$.

2 Blocking sets of Rédei type in projective Hjelmslev planes

Until the end of the paper R will be a chain ring of nilpotency index 2, i.e. $|R| = q^2$, $R/\operatorname{rad} R \cong \mathbb{F}_q$, where q is a prime power. We denote by $\Gamma = \{\gamma_0 = 0, \gamma_1 = 1, \gamma_2, \ldots, \gamma_{q-1}\}$ a set of q elements of R no two of which are congruent modulo rad R. By θ we denote an arbitrary element of rad $R \setminus (0)$. The points of the affine plane $\mathrm{AHG}(R_R^2)$ are identified with the pairs (x, y), where $x, y \in R$. The lines of $AHG(R_R^2)$ have equations $Y = aX + b$ or $X = c$, $a, b, c \in R$. We say that the lines of the first type have slope a. A line with equation $X = c$ is said to have slope ∞_j, if $c = \gamma_i + \gamma_j\theta$, $j = 0, 1, \ldots, q-1$.

The infinite points on a fixed line ℓ from the neighbor class of infinite lines can be identified with the slopes. So, (a) (resp (∞_j)) will denote the infinite point from ℓ of the lines with slope (a) (resp (∞_j)).

Definition 2.1. *Let T be a set of q^2 points in* $\mathrm{AHG}(R_R^2)$. *We say that the infinite point (a) is determined by T if there exist different points $P, Q \in T$ such that P, Q and (a) are collinear in* $\mathrm{PHG}(R_R^3)$.

Theorem 2.2. *Assume T is a set of q^2 points in* $\mathrm{AHG}(R_R^2)$. *Denote by D the set of infinite points determined by T. If $|D| < q^2 + q$ then $B = T \cup D$ is an irreducible blocking set in* $\mathrm{PHG}(R_R^3)$.

The proof of this theorem is obvious.

The construction given by this theorem yields blocking sets of size at most $2q^2 + q - 1$. It is straightforward that every irreducible blocking set of size at most $2q^2 + q - 1$ with a line ℓ with $|B \setminus \ell| = q^2$ can be obtained by this construction.

Definition 2.3. *A blocking set of size* $q^2 + m$ *in* $\mathrm{PHG}(R_R^3)$ *is said to be of Rédei type if it has an* m*-secant. Such a line is called a Rédei line.*

We are interested in sets T that are the graph of a function $f\colon R \to R$. Such sets can be written in the form

$$T = \{(x, f(x)) \mid x \in R\}.$$

Let x and y be two different elements from R We now have the following possibilities:

1) if $x - y \notin \mathrm{rad}\, R$ then $(x, f(x))$ and $(y, f(y))$ determine the point (a), where

$$(a) = (f(x) - f(y))(x - y)^{-1}.$$

2) if $x - y \in \mathrm{rad}\, R \setminus \{0\}$, and $f(x) - f(y) \notin \mathrm{rad}\, R$ the points $(x, f(x))$ and $(y, f(y))$ determine the point (∞_i) if

$$(x - y)(f(x) - f(y))^{-1} = \theta\gamma_i, \gamma_i \in \Gamma.$$

3) if $x - y \in \mathrm{rad}\, R \setminus \{0\}$, and $f(x) - f(y) \in \mathrm{rad}\, R$, say $x - y = \theta a$, $f(x) - f(y) = \theta b$, $a, b \in \Gamma$.

a) if $b \neq 0$, $(x, f(x))$ and $(y, f(y))$ determine all points (c) with $c \in a/b + \mathrm{rad}\, R$;

b) if $b = 0$, $(x, f(x))$ and $(y, f(y))$ determine the infinite points $(\infty_0), \ldots, (\infty_q)$.

Furthermore, for every set T of point of $\mathrm{AHG}(R_R^2)$ of size q^2 determining at most $q^2 + q - 1$ directions, we can always choose the coordinate system so that T is the graph of a function from R to R.

3 Examples

Let R be a chain ring with $|R| = q^2$, $R/\mathrm{rad}\, R \cong \mathbb{F}_q$ that contains a proper subring isomorphic to its residue field $\mathbb{F}_q$ Then $R = \mathbb{F}_q[\theta; \sigma]$ for some $\sigma \in \mathrm{Aut}\, \mathbb{F}_q$.

It has been noted in [5] that $\mathrm{PHG}(R_R^3)$ contains a subgeometry isomorphic to $\mathrm{PG}(2, q)$ which is an irreducible blocking set with two intersection numbers. As noted at the end of the previous section, this blocking set is of Rédei type.

Below we give an explicit construction of this blocking set as a graph of a function from R to R.

Define

$$f : \left\{ \begin{array}{ccc} R & \to & R \\ a+\theta b & \to & b+\theta a \end{array} \right. . \tag{1}$$

We shall check that the set of points $T = \{(x, f(x)) \mid x \in R\}$ determines $q+1$ infinite points. Consider the points $P = (a+\theta b, b+\theta a)$ and $Q = (c+\theta d, d+\theta c)$, $a, b, c, d \in \Gamma = \mathbb{F}_q$.

1) Let $a = c$. Then $b \neq d$ since otherwise P and Q would coincide. We have

$$x - y = \theta(b-d), \ \ f(x) - f(y) = b - d.$$

Hence P and Q determine the infinite point (∞_1).

2) Let $a \neq c$. We have

$$\begin{array}{rl} (f(x) & - \ f(y))(x-y)^{-1} = \\ = & ((b-d)+\theta(a-c))((a-c)+\theta(b-d))^{-1} \\ = & ((b-d)+\theta(a-c))((a-c)^{-1} - \theta(((a-c)^{\sigma})^{-1}(b-d)(a-c)^{-1} \\ = & (b-d)(a-c)^{-1} - \theta(b-d)^{\sigma}((a-c)^{\sigma})^{-1}(b-d)(a-c)^{-1}. \end{array}$$

Assume that $P' = (a' + \theta b', b' + \theta a')$ and $Q' = (c' + \theta d', d' + \theta c')$ are two point that determine an infinite point which is a neighbour to the infinite point determined by P and Q. Then $(b-d)(a-c)^{-1} = (b'-d')(a'-c')^{-1}$ which implies that

$$\begin{array}{l} (b-d)(a-c)^{-1} - \theta(b-d)^{\sigma}((a-c)^{\sigma})^{-1}(b-d)(a-c)^{-1} = \\ \qquad (b'-d')(a'-c')^{-1} - \theta(b'-d')^{\sigma}((a'-c')^{\sigma})^{-1}(b'-d')(a'-c')^{-1}. \end{array} \tag{2}$$

Hence if P, Q on one side and P', Q' on the other determine infinite points that are neighbours, then they determine the same infinite point. Therefore, the points of T determine at most one point in each neighbour class of infinite points. On the other hand, $(b-a)(c-d)^{-1}$ runs all elements of $\mathbb{F}_q$ (take, for instance $a = 1, c = d = 0$, b free). Therefore exactly one infinite point is determined in each neighbour class. Hence the points of T determine exactly $q+1$ directions.

It is known that the projective Hjelmslev plane $\mathrm{PHG}(R_R^3)$, where $R = \mathrm{GR}(q^2, p^2)$, does not contain a subplane isomorphic to $\mathrm{PG}(2, q)$. It is interesting to know what are the parameters of the Rédei-type blocking sets given by (1). Let us note that (1) depends on the choice of Γ. Let $R = \mathbb{Z}_{p^2}/(f(X))$, where F is a monic polynomial of degree $n \geq 1$, that is irreducible over $\mathbb{Z}_p$. Then $|R| = p^{2n}$ and rad $R = (p)$. If

$$\Gamma = \{\gamma_0 \in \text{rad } R, \gamma_1 \in 1 + \text{rad } R, \gamma_2, \ldots, \gamma_{q-1}\},$$

where $\gamma_i - \gamma_j \notin$ rad R, for $0 \leq i < j \leq q-1$, it can be shown that the set T determines exactly $q^2 - q + 2$ directions and the size of the corresponding Rédei-type blocking set is $2q^2 - q + 2$.

Let $P = (a + bp, b + ap)$ and $Q = (c + dp, d + cp)$, where $a, b, c, d \in \Gamma$. If $a = c$ and $b \neq d$, the points P and Q determine the infinite point (∞_1). If $a \neq c$, they determine the infinite point (α) with

$$\alpha = \frac{b-d}{a-c} + \left(1 - \frac{(b-d)^2}{(a-c)^2}\right) p. \tag{3}$$

The number of different directions determined by the points of T is equal to the different values taken by $\frac{b-d}{a-c}$, $a \neq c$. In the special case $b = d$, we have $(\alpha) = (p)$.

Now we are going to prove that if for every $\alpha \in R \setminus$ rad R there exist $a, b, c, d \in \Gamma$ such that $\alpha(a - b) = c - d$. Consider the elements $\alpha x + y$ where $x, y \in \Gamma$. If $\{\alpha x + y | x, y \in \Gamma\} = R$, there is nothing to prove. Otherwise, there exist $x_1, x_2, y_1, y_2 \in \Gamma$, $(x_1, y_1) \neq (x_2, y_2)$ such that

$$\alpha x_1 + y_1 = \alpha x_2 + y_2.$$

Hence $\alpha(x_1 - x_2) = y_2 - y_1$. Since $x_1 - x_2 \in$ rad R implies $x_1 = x_2$ and, similarly, $y_1 - y_2 \in$ rad R implies $y_1 = y_2$. If one of the differences $x_1 - x_2$, $y_2 - y_1$ is 0 then the other is also 0, which is a contradiction. Hence it is enough to set $a = x_1, b = x_2, c = y_1, d = y_2$.

References

[1] A. Cronheim, Dual numbers, Witt vectors, and Hjelmslev planes, *Geom. Dedic.* 7, 1978, 287-302.

[2] T. Honold, I. Landjev, Projective Hjelmslev geometries, *Proc. Second Intern. Workshop OCRT*, Sozopol, Bulgaria, 1998, 116-126.

[3] T. Honold, I. Landjev. On arcs in projective Hjelmslev planes, *Discr. Math.* 231, 2001, 265-278.

[4] T. Honold, I. Landjev, Arcs in projective Hjelmslev planes, *Discr. Math. Appl.* 11, 2001, 53-70.

[5] I. Landjev, On blocking sets in projective Hjelmslev planes, *Adv. Math. Commun.* 1, 2007, 65-82.

[6] R. Raghavendran, Finite associative rings, *Compos. Math.* 21, 1969, 195-229.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 171-176*

A note on a result by Hamada on minihypers

IVAN LANDJEV ivan@math.bas.bg
Institute of Mathematics and Informatics Bulgarian Academy of Sciences,
8 G. Bonchev str., 1113 Sofia, BULGARIA
New Bulgarian University, 21 Montevideo str., 1618 Sofia, BULGARIA
ASSIA ROUSSEVA assia@fmi.uni-sofia.bg
Faculty of Mathematics and Informatics, Sofia University,
5 James Bouchier blvd, 1164 Sofia, BULGARIA

Abstract. Hamada [Bull. Osaka Women's Univ. 24:1–47, 1985; Discrete Math. 116:229-268, 1993] characterized the non-weighted minihypers having parameters $(\sum_{i=1}^{h} v_{\lambda_i+1}, \sum_{i=1}^{h} v_{\lambda_i}; t, q)$ with $t > \lambda_1 > \lambda_2 > \cdots > \lambda_h \geq 0$. This result has been generalized in [Des. Codes Cryptogr. 45:123-138,2007] where it was proved that a weighted $(\sum_{i=1}^{h} v_{\lambda_i+1}, \sum_{i=1}^{h} v_{\lambda_i}; t, q)$-minihyper $\mathfrak{F}$, with $k-1 > \lambda_1 > \lambda_2 > \cdots > \lambda_h \geq 0$, is a sum of the characteristic functions of spaces of dimension $\lambda_1, \ldots, \lambda_h$. In this note, we prove that we can relax further the restrictions on the integers λ_i by allowing $r(q)-1$ equalities in the chain of strict inequalities $\lambda_2 > \ldots > \lambda_h$.

1 Introduction

Let $\mathrm{PG}(t, q)$ be the t-dimensional projective space over $\mathbb{F}_q$. Denote by $\mathcal{P}$ the set of points of the projective geometry $\mathrm{PG}(t, q)$ and let $v_{t+1} = (q^{t+1}-1)/(q-1)$ denote the cardinality of $\mathcal{P}$. A *multiset* in $\mathrm{PG}(t, q)$ is any mapping $\mathfrak{K}\colon \mathcal{P} \rightarrow \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of all nonnegative integers. This mapping is extended in a natural way to the subsets of $\mathcal{P}$ (the extension is also denoted by $\mathfrak{K}$) by $\mathfrak{K}(\mathcal{Q}) = \sum_{P \in \mathcal{Q}} \mathfrak{K}(P)$ where $\mathcal{Q} \subseteq \mathcal{P}$. The integer $\mathfrak{K}(\mathcal{Q})$ is called the *multiplicity* of $\mathcal{Q}$. The *cardinality of a multiset* is defined by $|\mathfrak{K}| = \mathfrak{K}(\mathcal{P})$. The *support* $\operatorname{supp}\mathfrak{K}$ of a multiset $\mathfrak{K}$ is defined as $\operatorname{supp}\mathfrak{K} = \{P \in \mathcal{P} \mid \mathfrak{K}(P) > 0\}$. A multiset with $\mathfrak{K}(P) \in \{0, 1\}$ for every $P \in \mathcal{P}$ is called a *non-weighted* or *projective* multiset. Projective multisets can be viewed as sets by identifying them with their support.

Let $\mathcal{Q}$ be a set of points in $\mathrm{PG}(t, q)$. We define the characteristic multiset $\chi_{\mathcal{Q}}$ by

$$\chi_{\mathcal{Q}}(P) = \begin{cases} 1 & \text{if } P \in \mathcal{Q}, \\ 0 & \text{if } P \notin \mathcal{Q}. \end{cases}$$

Definition 1. *A multiset* $\mathfrak{F}$ *in* $\mathrm{PG}(t, q)$, $t \geq 2$, *is called an* $(f, m; t, q)$-minihyper *or* (f, m)*-minihyper if* *(a)* $\mathfrak{F}(\mathcal{P}) = f$*;*

(b) $\mathfrak{F}(H) \geq m$ *for any hyperplane* H*;*

(c) there exists a hyperplane H_0 *with* $\mathfrak{F}(H_0) = m$*.*

This definition of a minihyper is equivalent to the original one given by Hamada and Tamari in [3]. In order to save space, we refer to [4] for all notions that are not defined here.

In [1] and [2], Hamada characterized the non-weighted minihypers with parameters $(\sum_{i=1}^{h} v_{\lambda_i+1}, \sum_{i=1}^{h} v_{\lambda_i}; t, q)$ with $t > \lambda_1 > \lambda_2 > \cdots > \lambda_h \geq 0$, as the union of a λ_1-dimensional space, λ_2-dimensional space, …, λ_h-dimensional space, which all are pairwise disjoint. In [4], this result was extended to weighted minihypers.

Theorem 1. *Let* $\mathfrak{F}$ *be a* $(\sum_{i=1}^{h} v_{\lambda_i+1}, \sum_{i=1}^{h} v_{\lambda_i}; t, q)$*-minihyper, with* $t \geq 2$*,* $q \geq 3$*, and*

$$t > \lambda_1 > \lambda_2 > \ldots > \lambda_h \geq 0.$$

Then

$$\mathfrak{F} = \sum_{i=1}^{h} \chi_{\pi_i},$$

where π_i *is a* λ_i*-dimensional subspace of* $\mathrm{PG}(t, q)$*,* $i = 1, \ldots, h$*.*

In this note, we show that we can relax the restrictions on the numbers λ_i by allowing some of them to be equal. We prove the following theorem.

Theorem 2. *Let* $t \geq 2$ *be an integer and let* $q \geq 3$ *be a prime power. Let* $\lambda_1, \ldots, \lambda_h$ *be a sequence of non-negative integers such that*

(1) $t > \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_h \geq 0$*, and*

(2) equalities in (1) occur in at most $r(q) - 1$ *places, where* $q + 1 + r(q)$ *is the size of the smallest nontrivial blocking set in* $\mathrm{PG}(2, q)$*.*

Then every minihyper $\mathfrak{F}$ *in* $\mathrm{PG}(t, q)$ *with parameters* $(\sum_{i=1}^{h} v_{\lambda_i+1}, \sum_{i=1}^{h} v_{\lambda_i})$ *can be represented as*

$$\mathfrak{F} = \sum_{i=1}^{h} \chi_{\pi_i},$$

where π_i *is a* λ_i*-dimensional subspace of* $\mathrm{PG}(t, q)$*,* $i = 1, \ldots, h$*.*

2 The proof of Theorem 2

For the proof of our characterization result, we use induction on t and h. The theorem is obviously true for $h = 1$ for every t and for $t = 2$ for all $h \leq t + r(q) - 1$. Note that for $t = 2$, our statement follows by the definition of $r(q)$, namely, every blocking set with less than $q + r(q) + 1$ points contains a line in its support.

Lemma 3. *Let the integers $t, q, \lambda_1, \ldots, \lambda_h$ satisfy the conditions of Theorem 2. Let further $\mathfrak{F}$ be a $(\sum_{i=1}^{h} v_{\lambda_i+1}, \sum_{i=1}^{h} v_{\lambda_i}; t, q)$-minihyper. Then* $\operatorname{supp}\mathfrak{F}$ *contains a λ_1-dimensional subspace of* $\mathrm{PG}(t, q)$.

The proof of this fact follows *mutatis mutandis* the proof of Lemma 17 from [4].

Now we assume that Theorem 2 is proved for all dimensions up to $t_0 - 1$ for all possible h, and for dimension t_0 for all $(\sum_{i=1}^{h'} v_{\lambda_i+1}, \sum_{i=1}^{h'} v_{\lambda_i})$-minihypers with $t > \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda'_h \geq 0$, where $h' < h_0$. We want to prove that theorem for minihypers for which the sums in the parameters contain h_0 summands.

An easy counting argument shows that for a subspace S of codimension s

$$\mathfrak{F}(S) \geq v_{\lambda_1 - s+1} + \ldots + v_{\lambda_{h_0} - s+1}.$$

Here, $v_\alpha = 0$ for $\alpha < 0$. Hence the minimal multiplicity of a subspace S of dimension $t_0 - 2$ (codimension 2) is $v_{\lambda_1 - 1} + \ldots + v_{\lambda_{h_0} - 1}$ and all hyperplanes through S are also of minimal multiplicity $v_{\lambda_1} + \ldots + v_{\lambda_{h_0}}$.

(1) Assume that $\lambda_1 < t_0 - 1$. Then each hyperplane contains a 0-point. Consider a projection φ from an arbitrary 0-point P onto a hyperplane Δ. The induced minihyper $\mathfrak{F}^\varphi$ in $\Delta \cong \mathrm{PG}(t_0 - 1, q)$ has the parameters of $\mathfrak{F}$: $(\sum_{i=1}^{h_0} v_{\lambda_{i+1}}, \sum_{i=1}^{h_0} v_{\lambda_i})$. By the induction hypothesis

$$\mathfrak{F}^\varphi = \chi_{\delta_1} + \chi_{\delta_2} + \ldots + \chi_{\delta_{h_0}}, \tag{1}$$

where δ_i is a subspace of Δ with $\dim \delta_i = \lambda_i$, $i = 1, \ldots, h_0$.

The support of $\mathfrak{F}$ contains a λ_1-dimensional subspace π_1 (cf. Lemma 3). Set $\mathfrak{F}' = \mathfrak{F} - \chi_{\pi_1}$. Let H be a hyperplane that contains π_1. Consider a projection from a 0-point P in H. Clearly, $\mathfrak{F}^\varphi$ has the form (1). The image of π_1 under φ is exactly δ_1. Thus

$$\mathfrak{F}'^\varphi = \chi_{\delta_2} + \ldots + \chi_{\delta_{h_0}}.$$

This implies that

$$\mathfrak{F}'(H) = \mathfrak{F}'^\varphi(H') = v_{\lambda_2} + \ldots + v_{\lambda_{h_0}}.$$

Let H be a hyperplane that does not contain π_1. Now H meets π_1 in a $(\lambda_1 - 1)$-dimensional subspace and again

$$\mathfrak{F}'(H) = \mathfrak{F}(H) - |\pi_1 \cap H| = \mathfrak{F}(H) - v_{\lambda_1} \geq v_{\lambda_2} + \ldots + v_{\lambda_{h_0}}.$$

Hence $\mathfrak{F}'$ is a $(\sum_{i=2}^{h_0} v_{\lambda_{i+1}}, \sum_{i=2}^{h_0} v_{\lambda_i})$-minihyper in $\mathrm{PG}(t_0, q)$. By the induction hypothesis, $\mathfrak{F}' = \chi_{\pi_2} + \ldots + \chi_{\pi_{h_0}}$, where dim π_i are λ_i-dimensional subspaces of $\mathrm{PG}(t_0, q)$, $i = 2, \ldots, h_0$.

(2) Let $\lambda_1 = t_0 - 1$. Again by Lemma 3, we have that $\operatorname{supp} \mathfrak{F}$ contains a λ_1-space, i.e. a hyperplane π_1. Define the multiset $\mathfrak{F}' = \mathfrak{F} - \chi_{\pi_1}$. For each hyperplane $H \neq \pi_1$ we have $\mathfrak{F}'(H) \geq v_{\lambda_2} + \ldots + v_{\lambda_h}$. Now it is enough to demonstrate that

$$\mathfrak{F}'(\pi_1) \geq v_{\lambda_2} + \ldots + v_{\lambda_h}.$$

This will imply that $\mathfrak{F}'$ is a $(\sum_{i=2}^{h} v_{\lambda_i+1}, \sum_{i=2}^{h} v_{\lambda_i})$-minihyper and the result will follow by induction.

Fix a $(t_0 - 2)$-dimensional subspace δ of $\mathrm{PG}(t_0, q)$ of minimal multiplicity. Denote by Π_i, $i = 0, \ldots, q$, the hyperplanes through δ. All these hyperplanes are also of minimal multiplicity: $\mathfrak{F}(\Pi_j) = \sum_{i=1}^{h} v_{\lambda_i}$. By the induction hypothesis, the restriction of $\mathfrak{F}$ to every Π_j is a sum of subspaces:

$$\mathfrak{F}|_{\Pi_j} = \chi_{\pi_1^{(j)}} + \chi_{\pi_2^{(j)}} + \ldots + \chi_{\pi_h^{(j)}},$$

where $\dim \pi_i^{(j)} = \lambda_i - 1$, $i = 1, \ldots, h_0$, $j = 0, \ldots, q$. Note that the indices $i \in \{1, \ldots, h_0\}$ can be chosen in such way that the subspaces $\pi_i^{(j)}$ meet δ in the same $(\lambda_i - 2)$-dimensional subspace. This follows by the fact that δ is of minimal multiplicity. In other words, we can arrange the subspaces $\pi_i^{(j)}$ in such way that

$$\pi_i^{(0)} \cap \delta = \pi_i^{(1)} \cap \delta = \ldots = \pi_i^{(q)} \cap \delta = \delta_i, \ \ i = 1, 2, \ldots, h.$$

Now we can write

$$\begin{aligned}
\mathfrak{F} &= \sum_{j=0}^{q} \mathfrak{F}|_{\Pi_j} - q\mathfrak{F}|_{\delta} \\
&= \sum_{j=0}^{q}\sum_{i=1}^{h} \chi_{\pi_i^{(j)}} - q\sum_{i=1}^{h} \chi_{\delta_i} \\
&= \sum_{i=1}^{h}\sum_{j=0}^{q} \chi_{\pi_i^{(j)}} - q\sum_{i=1}^{h} \chi_{\delta_i} \\
&= \sum_{i=1}^{h}\left(\sum_{j=0}^{q} \chi_{\pi_i^{(j)}} - q\chi_{\delta_j}\right) \\
&= \sum_{i=1}^{h} \mathfrak{G}_i,
\end{aligned}$$

where we have set $\mathfrak{G}_i = \sum_{j=0}^{q} \chi_{\pi_i^{(j)}} - q\chi_{\delta_j}$. It is known that $\mathfrak{G}_1 = \chi_{\pi_1}$ which implies that $\mathfrak{F}' = \sum_{i=2}^{h} \mathfrak{G}_i$.

Let us fix an integer $i \in \{2, \ldots, h\}$. First, we consider the case when δ_i is not contained δ_1. In this case,

$$\begin{aligned}
\mathfrak{G}_i(\pi_1) &= \sum_{j=0}^{q} |\pi_1 \cap \pi_i^{(j)}| - |\pi_1 \cap \delta_i| \\
&= (q+1)v_{\lambda_i - 1} - qv_{\lambda_i - 2} \\
&= \frac{q^{\lambda_i} - 1}{q - 1} = v_{\lambda_i}.
\end{aligned}$$

Now suppose that δ_i is a subspace of δ_1. Fix a $(t-3)$-dimensional subspace δ_0 of δ that has minimal multiplicity. We have $\delta_0 \neq \delta_1$ since δ_1 is not of minimal multiplicity. Denote by $\tau_1, \ldots, \tau_q$ the $(t_0 - 2)$-dimensional subspaces of Π_0 through δ_0 other than δ. Since at most $r(q) - 1$ of the subspaces τ_i are not minimal (this happens when $\lambda_{h_0 - r(q)+2} = \cdots = \lambda_{h_0} = 1$), we can assume with no loss of generality that τ_1 is minimal, i.e. $\mathfrak{F}(\tau_1) = \sum_{i=1}^{h} v_{\lambda_i}$. If $\tau_1 \cap \pi_1^{(j)}$, $i = 2, \ldots, h$, is not contained in π_1 for some j, we can repeat the above argument to show that $\mathfrak{G}_i(\pi_1) \geq v_{\lambda_i}$.

Now assume that $\tau_1 \cap \pi_i^{(0)}$ is in π_1. Clearly, $\tau_1 \cap \pi_i^{(0)}$ does not coincide with δ_i since otherwise δ would coincide with τ_1. Hence $\pi_1^{(0}$ contains other points (apart from δ_i) that are from π_1. This implies that $\pi_1^{(0)}$ is contained completely in π_1. Now we have that

$$\mathfrak{G}_i(\pi_1) = \sum_{j=0}^{q} |\pi_1 \cap \pi_i^{(j)}| - q|\pi_1 \cap \delta_i| \geq |\pi_1 \cap \pi_i^{(0)}| = |\pi_i^{(0)}| = v_{\lambda_i}.$$

Thus we have proved that in all cases $\mathfrak{G}_i(\pi_1) \geq v_{\lambda_i}$. Now we have

$$\mathfrak{F}'(\pi_1) = \sum_{i=2}^{h} \mathfrak{G}_i(\pi_1) \geq \sum_{i=2}^{h} v_{\lambda_i},$$

which finishes the proof.

References

[1] N. Hamada, Characterization of minihypers in a finite projective geometry and its applications to error-correcting codes, *Bull. Osaka Women's Univ* 24, 1985, 1-47.

[2] N. Hamada, A characterization of some $[n, k, d; q]$-codes meeting the Griesmer bound using a minihyper in a finite projective geometry, *Discr. Math.* 116, 1993, 229-268.

[3] N. Hamada, F. Tamari, On a geometrical method of construction of maximal t-linearly independent sets, *J. Combin. Theory*, Ser. A 25, 1978, 14-28.

[4] I. Landjev, L. Storme, A weighted version of a result by Hamada on minihypers and on linear codes meeting the Griesmer bound, *Des., Codes Crypt.* 45, 2007, 123-138.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 177-180*

Colored superimposed codes[1]

VLADIMIR LEBEDEV `lebed37@iitp.ru`
Institute for Information Transmission Problems, Moscow, RUSSIA

1 Introduction

One of the most important applications of (w, r) superimposed codes (see, [1] - [3]) is the following cryptography problem. There are T users and N secret keys. Each user has his own set of keys, and a group of users can communicate if there exists a common secret key for the whole group. It is required that, for any group of w users and any group of other r users, there should exist a key such that all users of the first group have this key and thus can communicate, while neither of the r users of the second group possess this key. Thus, users of the first group can exchange information "secretly" from users of the second group.

Now assume that all users have the same set of keys, but any key has several states. Let all keys have the same numbers of the states. A user can not change key's state and the user can communicate with users who have the key with this state. There are several groups of users (the number of the groups is not more then the number of the key's states). We want that there is a key such that for any group of users the key has the same state (and for different groups - different states) and so the users from any group can communicate secretly from users of other groups.

This situation can naturally be thought of as a q-ary $N \times T$ matrix $C = \|c_{ij}\|$, where $c_{ij} = k$ if the jth user possesses the ith secret key with the state k. Then the property described above means that, for any subsets $R_0, R_1, \ldots, R_{q-1} \subset [T]$ of cardinalities $|R_s| = r_s$, there exists a row i in C such that $c_{ij} = s$ for all j from R_s, where $s = 0, 1, \ldots, q-1$. We will refer to the matrix as $(r_0, r_1, \ldots, r_{q-1})$ superimposed code or colored superimposed code.

Of course, we would like to minimize the number of secret keys with a fixed number of users, or, equivalently, maximize the number of users with a fixed number of keys. Thus, the problem consists in finding a matrix C that obeys this property, with the number of columns as large as possible (rows are of length N). We will often refer to columns of C as codewords and refer to the

[1] Supported in part by the Russian Foundation for Basic Research, project no. 06-01-00226.

matrix C itself as a q-ary code. Furthermore, in what follows, we use the term "code of size $N \times T$" rather than the more commonly used "code of length N and cardinality T."

Denote by $N(T, r_0, r_1, \dots, r_{q-1})$ the minimum possible length of a $(r_0, r_1, \dots, r_{q-1})$ superimposed code of a given cardinality T. A colored code is optimal if $N = N(T, r_0, r_1, \dots, r_{q-1})$. The rate of a q-ary code of length N and cardinality T is, as usual, $R = (\log T)/N$. We are interested in the asymptotic behavior of the rate

$$R(r_0, r_1, \dots, r_{q-1}) = \limsup_{T \to \infty} \frac{\log_q T}{N(T, r_0, r_1, \dots, r_{q-1})}$$

of such (optimal) codes.

2 Some results

Let us start with the formal definition of $(r_0, r_1, \dots, r_{q-1})$ superimposed codes.

Definition 1. *A q-ary $N \times T$ matrix $C = \|c_{ij}\|$ is called a $(r_0, r_1, \dots, r_{q-1})$* superimposed code of size $N \times T$ *if, for any disjoint subsets $R_0, R_1, \dots, R_{q-1} \subset [T]$ of cardinalities $|R_s| = r_s$, there exists a coordinate $i \in [N]$ such that $c_{ij} = s$ for all $j \in R_s$, where $s = 0, 1, \dots, q-1$.*

Theorem 1. *For colored superimposed codes we have*

$$R(r_0, r_1, \dots, r_{q-1}) \geq 1/(S-1) \log_q \frac{S^S}{S^S - r_0^{r_0} r_1^{r_1} \cdots r_{q-1}^{r_{q-1}}},$$

where $S = (r_0 + r_1 + \dots + r_{q-1})$.

The next important parameter will be defined for an arbitrary q-ary code C of size $N \times T$. Consider positive integers $(x_0, x_1, \dots, x_{q-1})$. Fix a collection I consisting of $X = x_0 + x_1 + \dots + x_{q-1}$ codewords and denote by $C_X(I)$ the submatrix of C formed by these codewords. Thus, the matrix $C_X(I)$ is of size $N \times X$. By the "X-distance" for the collection I, we call the number of rows of $C_X(I)$ such that each row has x_s elements with value s for all s, where $s = 0, 1, \dots, q-1$. We denote this number by $d(C_X(I))$.

Definition 2. *The* minimum "X-distance" *for a q-ary code C is the value $d_X = \min_{|I|=X} d(C_X(I))$. Denote by $R^{(N)}(d_X)$ the rate of a q-ary code of length N with minimum "X-distance" d_X.*

Theorem 2. *For q-ary code C of length N with minimum "X-distance" d_X we have the following asymptotic bound:*

$$R^{(N)}(d_X) \leq (1 - \frac{X^X x_0! x_1! \dots x_{q-1}!}{x_0^{x_0} x_1^{x_1} \dots x_{q-1}^{x_{q-1}} X!} \frac{d_X}{N})(1 - \log_q(q-1)).$$

The following lemma explains the relation between the parameter d_X and $(r_0, r_1, \dots, r_{q-1})$ superimposed codes.

Lemma. *If a $(r_0, r_1, \dots, r_{q-1})$ superimposed code C of size $N \times T$ exists, then a $(r_0 - x_0, r_1 - x_1, \dots, r_{q-1} - x_{q-1})$ superimposed code of size*

$$\lfloor d_X x_0! x_1! \dots x_{q-1}!/X! \rfloor \times (T - X).$$

exists.

Corollary. *If there exists a $(r_0, r_1, \dots, r_{q-1})$ superimposed code C of cardinality T with minimum "X-distance" d_X, then, for positive integers x_s ($x_s < r_s$), we have*

$$N(T - X, r_0 - x_0, r_1 - x_1, \dots, r_{q-1} - x_{q-1}) \leq \frac{d_X x_0! x_1! \dots x_{q-1}!}{X!}.$$

Theorem 3. *For the rate of $(r_0, r_1, \dots, r_{q-1})$ superimposed codes, we have the asymptotic bound:*

$$R(r_0, r_1, \dots, r_{q-1}) \leq \\ \frac{R(r_0 - x_0, r_1 - x_1, \dots, r_{q-1} - x_{q-1})}{R(r_0 - x_0, \dots, r_{q-1} - x_{q-1})/(1 - \log_q(q-1)) + X^X/(x_0^{x_0} \dots x_{q-1}^{x_{q-1}})}.$$

Proof. Consider an optimal $(r_0, r_1, \dots, r_{q-1})$ superimposed code of cardinality T, length $N(T, r_0, r_1, \dots, r_{q-1})$, and rate $R_T(r_0, r_1, \dots, r_{q-1})$. Theorem 2 implies that, as $T \to \infty$

$$R_T(r_0, r_1, \dots, r_{q-1}) \leq \\ 1 - \frac{X^X d_X x_0! x_1! \dots x_{q-1}!}{x_0^{x_0} x_1^{x_1} \dots x_{q-1}^{x_{q-1}} X! N(T, r_0, r_1, \dots, r_{q-1})})(1 - \log_q(q-1)) + o(1)$$

Using the corollary of the lemma, we get

$$R_T(r_0, r_1, \ldots, r_{q-1}) \leq$$
$$(1 - \frac{X^X N(T-X, r_0 - x_0, r_1 - x_1, \ldots, r_{q-1} - x_{q-1})}{x_0^{x_0} x_1^{x_1} \ldots x_{q-1}^{x_{q-1}} N(T, r_0, r_1, \ldots, r_{q-1})})(1 - \log_q(q-1)) + o(1).$$

Let us apply Definition 2 of $R_{T-X}(r_0 - x_0, r_1 - x_1, \ldots, r_{q-1} - x_{q-1})$ and pass to the limit as $T \to \infty$ on both sides of the above inequality. As a result, we get a recurrence inequality for the rate $R(r_0, r_1, \ldots, r_{q-1})$, which can be written as

$$R(r_0, r_1, \ldots, r_{q-1})(1 + \frac{X^X(1 - \log_q(q-1))}{x_0^{x_0} x_1^{x_1} \ldots x_{q-1}^{x_{q-1}} R(r_0 - x_0, r_1 - x_1, \ldots, r_{q-1} - x_{q-1})})$$
$$\leq 1 - \log_q(q-1).$$

From this inequality, the statement of the theorem follows.

References

[1] C. J. Mitchell, F. C. Piper, Key storage in secure networks, *Discr. Appl. Math.* 21, 1988, 215-228.

[2] H. K. Kim, V. Lebedev, On optimal superimposed codes, *J. Combin. Des.* 12, 2004, 79-91.

[3] A. D'yachkov, A. Macula, V. Torney, P. Vilenkin, Families of finite sets in which no intersection of l sets is covered by the union of s others, *J. Combin. Theory* Ser. A. 99, 2002, 195-218.

[4] D. R. Stinson, R. Wei, L. Zhu, Some new bounds for cover-free families, *J. Combin. Theory* Ser. A. 90, 2000, 224-234.

[5] A. D'yachkov, P. Vilenkin, S. Yekhanin, Upper bounds on the rate of superimposed (s, l)-codes based on Engel's inequality, *Proc. Eighth Intern. Workshop ACCT*, Tsarskoe Selo, Russia, 2002, 95-99.

[6] V. S. Lebedev, Some tables for (w, r) superimposed codes, *Proc. Eighth Intern. Workshop ACCT*, Tsarskoe Selo, Russia, 2002, 185-189.

[7] V. S. Lebedev, New asymptotic upper bound on the rate of (w, r) cover free codes, *Probl. Inform. Transm.* 39, 2003, 317-323.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
16-22 June, 2008, Pamporovo, Bulgaria *pp. 181–185*

Jacket conference matrices and Paley transformation

Moon Ho Lee moonho@chonbuk.ac.kr
Institute of Information and Communication, Chonbuk National University,
Jeonju 561-756, KOREA
Veselin Vl. Vavrek veselin@math.bas.bg
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
8 G.Bonchev str., 1113, Sofia, BULGARIA

Abstract. Paley construction can be generalized into two ways. First we apply it to construct Jacket matrices, staring from new type of matrices and second generalization is to increase the dimensions to obtain (bigger) Jacket matrices. In this paper we describe a modified Paley construction to produce Jacket matrices that can be denoted as a new type of matrices called "Jacket Conference matrices".

1 Introduction

Jacket matrices [1], [2] are defined to be $n \times n$ matrices $J = ||j_{i,k}||$ over a field F with the property $JJ' = n.I_n$, where $J' = ||j'_{i,k}||$ is the transpose matrix of the element inverse of J, i.e., $j'_{i,k} = (j_{k,i})^{-1}$. These matrices are used in Digital Signal Processing and Coding theory. In [3] Paley constructed two classes of Hadamard matrices (known as Paley type 1 and Paley type 2). In this paper we describe a modified Paley construction to produce Jacket matrices that can be denoted as a new type of matrices called "Jacket Conference matrices".

In Section 2, we present the Paley method [3] in more general form, in order to apply it for constructing Hadamard matrices from symmetric Conference matrices.

2 Paley transformation

Recall that the Kronecker product, $A \otimes B$, of two matrices $[A]_n = ||a_{i,j}||$ and $[B]_m = ||b_{i,j}||$ is defined as

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix} \otimes B := \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n}B \\ \vdots & \vdots & & \vdots \\ a_{m,1}B & a_{m,2}B & \dots & a_{m,n}B \end{pmatrix}$$

In the above formula $a_{ij}B$ stands for a sub-matrix (not for an element).

Let C be an $m \times m$ symmetric Conference matrix. Paley statement says, that for

$$[A]_2 = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}_2, \; [B]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_2,$$

the matrix

$$[S]_{2m} := [A]_2 \otimes [C]_m + [B]_2 \otimes [I]_m$$

is Hadamard [3].

Proposition 1. *Let C be an $m \times m$ symmetric Conference matrix, and A and B be $n \times n$ Hadamard matrices. The matrix*

$$[S]_{nm} := [A]_n \otimes [C]_m + [B]_n \otimes [I]_m \tag{1}$$

is an Hadamard matrix if and only if

$$[A]_n[B]_n^t + [B]_n[A]_n^t = 0, \tag{2}$$

where $[X]^t$ is the transpose matrix of $[X]$.

Proof. To check if $[S]_{mn}$ is an Hadamard matrix we must show, that $[S]_{mn}[S]_{mn}^t = mn[I]_{mn}$. Since

$$[S]_{mn}^t = [A]_n^t \otimes [C]_m^t + [B]_n^t \otimes [I]_m,$$

we have

$$\begin{aligned}
[S]_{mn}[S]_{mn}^t &= ([A]_n \otimes [C]_m + [B]_n \otimes [I]_m)([A]_n^t \otimes [C]_m^t + [B]_n^t \otimes [I]_m) = \\
&= ([A]_n \otimes [C]_m)([A]_n^t \otimes [C]_m^t) + ([A]_n \otimes [C]_m)([B]_n^t \otimes [I]_m) + \\
&\quad + ([B]_n \otimes [I]_m)([A]_n^t \otimes [C]_m^t) + ([B]_n[B]_n^t) \otimes [I]_m = \\
&= ([A]_n \otimes [C]_m)([A]_n^t \otimes [C]_m^t) + ([B]_n[B]_n^t) \otimes [I]_m + ([A]_n[B]_n^t) \otimes [C]_m + ([B]_n[A]_n^t) \otimes [C]_m^t
\end{aligned}$$

$[C]_m^t = [C]_m$, and $[A]_n$ and $[C]_n$ are orthogonal matrices, thus $[A]_n \otimes [C]_m$ is also orthogonal matrix, and $[B]_n$ is Hadamard matrix. Hence

$$[S]_{mn}[S]_{mn}^t = (mn - n)[I]_{mn} + n[I]_{mn} + ([A]_n[B]_n^t + [B]_n[A]_n^t) \otimes [C]_m.$$

To get $[S]_{mn}$ to be Hadamard matrix (i.e. $[S]_{mn}[S]_{mn}^t = mn[I]_{mn}$) would be equivalent to

$$[A]_n[B]_n^t + [B]_n[A]_n^t = 0.$$

Note that in the case $n = 2$ Proposition 1 gives the Paley construction. Also, the above proposition motivate us to give the following definition.

Definition 1. *The pair $([A]_n, [B]_n)$ of Hadamard matrices of order n is called matched, iff the equation (2) holds.*

3 Jacket conference matrices

In this section instead of using transpose matrix of $[X]_n$, we can use transpose matrix of element inverse, and we shall denote it with $[X]'_n$ (where $x'_{ij} = (x_{ji})^{-1}$). We preserve matrices $[A]_n$ and $[B]_n$ to be Hadamard, but $[C]_m$ must be selected properly, and $[C]'_m$ must be defined in little bit different way.

Definition 2. *We shall call the $n \times n$ square matrix A is a Jacket Conference matrix, if the following conditions hods:*

1. $a_{i,i} = 0$, for $i = 1, 2, \ldots, n$.
2. $a_{i,j} \neq 0$, for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$.
3. $\sum_{s\in\{1,2,\ldots,n\}\setminus\{i,j\}} a_{i,s}.(a_{s,j})^{-1} = 0$, for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$.

We shall call such a matrix is reciprocal, if also

4. $a_{i,j} = (a_{j,i})^{-1}$, for all $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$.

If matrix $[C]_m$ is Jacket Conference matrix, then we define matrix $[C]'_m$ as follow

$$[C]'_m : c'_{ij} = \begin{cases} (c_{ji})^{-1}, & i \neq j; \\ 0, & i = j. \end{cases}$$

It is easy to check, that the calculations in previous section continue to be true if we use prime instead of transpose symbol t. It is enough to check $([A]_n \otimes [C]_m)([A]'_n \otimes [C]'_m) = (mn - n)[I]_{mn}$.

If we try to construct a reciprocal Jacket Conference matrix $[JC]$ of order 4:

$$[JC]_4 = \begin{bmatrix} 0 & a & b & c \\ 1/a & 0 & d & e \\ 1/b & 1/d & 0 & f \\ 1/c & 1/e & 1/f & 0 \end{bmatrix}_4 , \quad \text{we obtain} \quad \left| \begin{array}{l} c = -adf \\ b = \ iad \\ e = -idf \end{array} \right. .$$

Here, i is the imaginary unit. Selecting $a = d = f = 1$, and calculating the other coefficient by the formulae above, we can obtain a reciprocal Jacket Conference matrix

$$[JC]_4 = \begin{bmatrix} 0 & 1 & i & -1 \\ 1 & 0 & 1 & -i \\ -i & 1 & 0 & 1 \\ -1 & i & 1 & 0 \end{bmatrix}_4 . \qquad (3)$$

Thus, applying the Paley construction we can construct larger Jacket matrices.

4 Generalized Paley construction

We proved in {Section 2}/{Section 3} that if $([A]_n, [B]_n)$ are matched, and $[C]_n$ is {symmetric}/{reciprocal Jacket} Conference matrix of order m, then the matrix $[S]_{mn}$, defined in (1) is {Hadamard}/{Jacket}.

In this section we shall find several matched pairs of matrices.

Proposition 2. *If $([A]_n, [B]_n)$ is matched pair, and $[D]_r$ is Hadamard matrix, then $([A]_n, [B]_n) \otimes [D]_r := ([A]_n \otimes [D]_r, [B]_n \otimes [D]_r)$ is a matched pair, too.*

Proof. Since $[A]_n$, $[B]_n$ and $[D]_r$ are Hadamard, the matrices $[A]_n \otimes [D]_r$ and $[B]_n \otimes [D]_r$ are also Hadamard matrices. We can obtain the following calculation to prove $([A]_n \otimes [D]_r, [B]_n \otimes [D]_r)$ is matched pair

$$[A \otimes D]_{nr}[B \otimes D]'_{nr} + [B \otimes D]_{nr}[A \otimes D]'_{nr} =$$

$$= ([A]_n[B]'_n + [B]_n[A]'_n) \otimes ([D]_r[D]'_r) = 0.$$

■

Proposition 3. *If $[X]$ and $[Y]$ are $2n \times n$ matrices and the composite matrix $\begin{bmatrix} X \\ Y \end{bmatrix}_{2n}$ is Hadamard, then*

$$([U]_{2n}, [V]_{2n}) := (\begin{bmatrix} X \\ Y \end{bmatrix}_{2n}, \begin{bmatrix} Y \\ -X \end{bmatrix}_{2n})$$

is matched pair of matrices.

Proof. The first matrix is Hadamard by definition, and obviously the second is also Hadamard. We shall check the condition (2), and write it as

$$\langle a_v, b_u \rangle + \langle a_u, b_v \rangle = 0, \forall u, v \in \{1, 2, \ldots, m\} \tag{4}$$

The matrices $[U]_{2n}$ and $[V]_{2n}$ (up to the sign) have rows of one and the same Hadamard matrix, it is easy to show, that the inner products in (4) will be 0, excluding the case in which row corresponds to its identical row. By definition

$$u_i = -v_{i+n}, \text{ for } i = 1, 2, \ldots, n$$

and

$$u_i = v_{i-n}, \text{ for } i = n+1, n+2, \ldots, 2n.$$

Thus if the first inner product in (4) is nonzero, it must be $\pm 2n$, while the other one would be $\mp 2n$. Thus the sum is 0. ■

5 Example

Applying the modified Paley transformation using matrix (3) we obtain the following Jacket matrix

$$[J]_8 = \begin{bmatrix} 1 & -1 & -i & 1 & 1 & 1 & i & -1 \\ -1 & 1 & -1 & i & 1 & 1 & 1 & -i \\ i & -1 & 1 & -1 & -i & 1 & 1 & 1 \\ 1 & -i & -1 & 1 & -1 & i & 1 & 1 \\ 1 & 1 & i & -1 & -1 & 1 & i & -1 \\ 1 & 1 & 1 & -i & 1 & -1 & 1 & -i \\ -i & 1 & 1 & 1 & -i & 1 & -1 & 1 \\ -1 & i & 1 & 1 & -1 & i & 1 & -1 \end{bmatrix}_8 .$$

6 Acknowledgments

This research was supported by the Ministry of Knowledge Economy, Korea, under the ITFSIP (IT foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement), C1012-0801-0001, and KRF D-2007-521-D00330, Korea. The research was also partially supported by Bulgarian NSF (project IO-03-02/2006).

References

[1] M. H. Lee, A new reverse jacket transform and its fast algorithm, *IEEE Trans. Circ. Syst.* 47, 2000, 39-47.

[2] http://en.wikipedia.org/wiki/Category:Matrices,
http://en.wikipedia.org/wiki/Jacket_matrix.

[3] R. E. A. C. Paley, On orthogonal matrices, *J. Math. Phys.* 12, 1933, 311-320.

[4] F. J. MacWilliams, N. J. A. Sloane, *The theory of error correcting codes*, Amsterdam, The Netherlands: North-Holland, 1977.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 186-191*

Sum covers in steganography

PETR LISONĚK[1] plisonek@sfu.ca
Department of Mathematics, Simon Fraser University
Burnaby, BC, CANADA V5A 1S6

Abstract. We extend the study of steganography schemes with pooling to the case when two changes per cell are allowed. We show that such schemes are equivalent to a new, symmetric version of sum covers known in combinatorial design theory. We give a construction that is better in the information versus distortion metric than the schemes with one change per cell. A number of interesting questions concerning the underlying sum cover sets remain open.

1 Steganography background

Steganography is the science of information hiding. The sender starts with a *cover object,* such as for example a digital multimedia file, and (s)he embeds a hidden message into the cover object by slightly distorting it in a way that enables the intended recipient to retrieve the hidden message from the distorted cover object; at the same time the very existence of the hidden message should be impossible to detect by any third party.

Typically the cover object is a sequence of elements of D, where $D = \{0, \ldots, m-1\}$, $m = 2^e$. In current applications we usually have $e \in \{8, 12, 16\}$. For example, $e = 8$ for grayscale digital images and $e = 16$ for CD quality audio.

Let S denote the set of message symbols. A message to be communicated by the sender to the recipient is a string of elements of S. In most steganographic schemes, the sender and the recipient agree on a *symbol-assignment function*

$$s : D \to S. \tag{1}$$

To embed a given message symbol $z \in S$ in a given element $x \in D$, the sender modifies x to x' so that $s(x') = z$ and $|x - x'|$, the *amplitude of the embedding change,* is as small as possible.

One of the goals of Steganography is to design schemes with high embedding efficiency, which can be broadly defined as the ratio between the amount of the communicated information (information rate) and the amount of introduced distortion (distortion rate) [3, 4].

The embedding efficiency can be increased by *applying covering codes,* and we recommend [1] or [5] for an introduction to this topic. In order to achieve a desired information rate (or a desired distortion rate), one can use direct sums of covering codes.

It has been established in the steganography literature that the impact of embedding becomes statistically detectable rather quickly with the increasing amplitude of embedding changes. Thus, from now on we limit ourselves to so-called *±1 embedding changes* in which the sender modifies each element of D by at most one, which is the smallest possible modification. Hence we will be measuring the total amount of distortion simply by *counting the number of embedding changes.*

We note that a problem will arise in the rare case when the sender is required to apply the $+1$ change to the value $m-1 \in D$ or the -1 change to the value $0 \in D$. Then the sender can choose a different cover object (or the sender can perform a change of a magnitude greater than 1 to achieve the same effect). If we neglect these rare events, then we can assume that $D = \mathbb{Z}$, which makes our algebraic treatment easier.

Let $\mathbb{Z}_n = \mathbb{Z}/(n)$ denote the integers modulo n. A concrete example of a symbol-assignment function (1) that requires only ± 1 embedding changes is given by $S = \mathbb{Z}_3$ and $s(x) = x \bmod 3$. This function has a better embedding efficiency than the notorious "least significant bit embedding" defined by $s(x) = x \bmod 2$.

1.1 Schemes with pooling

In [3] we proved that the embedding efficiency can be increased by *pooling* the elements of D. We partition the cover object into disjoint segments, each of which consists of d elements of D. That is, we partition the cover object into elements of D^d, which we will call *cells*. The details of partitioning into cells are immaterial for our study. For example, the cells can be formed by adjacent elements along some pseudo-random path through the cover object. This pseudo-random path can be generated by the sender and by the recipient from a shared secret seed.

In contrast to (1), the symbol-assignment function will now be a mapping

$$s : D^d \to S. \tag{2}$$

The information rate achieved by s in (2) is $d^{-1} \log_2 |S|$ bits per element of the cover object. Therefore, given the cell dimension d and the maximum number c of changes allowed per cell, we wish to maximize $|S|$. The upper bound on $|S|$ is

$$U_{d,c} := \sum_{i=0}^{c} \binom{d}{i} 2^i$$

since s must be surjective.

One example of a function of the type (2) is given by taking $S = \mathbb{Z}_{2d+1}$ and

$$s(x_1, \ldots, x_d) := \left(\sum_{i=1}^{d} i x_i \right) \bmod (2d+1). \tag{3}$$

In order to embed any symbol $u' \in \mathbb{Z}_{2d+1}$ into any cell $x \in D^d$ using (3), at most one ± 1-change is required. This can be seen as follows: Let (e_i) denote the standard basis of $\mathbb{Z}^d$, and assume $s(x) = u$. Let $\delta = u' - u$. If $\delta = 0$, then no change is performed. Otherwise let $\delta = \varepsilon_k k$ with $\varepsilon_k \in \{-1, 1\}$ and $k \in \{1, 2, \ldots, d\}$. We modify x to $x' = x + \varepsilon_k e_k$; indeed $s(x') = u'$. Note that the embedding defined by (3) is optimal if at most one ± 1-change per cell is allowed, since $|\mathbb{Z}_{2d+1}| = 2d + 1 = U_{d,1}$.

We finish this introductory section by an informal sketch of the main result of [3]. Suppose that $2d + 1$ is a prime power. Let the embedding scheme Σ_1 be defined by the symbol-assignment function (3) and using $(2d+1)$-ary Hamming codes as covering codes (see the note about covering codes above). Let the embedding scheme Σ_2 be defined by the symbol-assignment function $x \mapsto x \bmod 3$ and using ternary Hamming codes as covering codes to achieve the same distortion rate as Σ_1. Then the information rate of Σ_1 is never worse than the information rate of Σ_2. The precise statement with proofs can be found in [3].

2 Schemes with two changes per cell

The present paper is concerned with the embedding schemes that allow *at most two ± 1-changes per cell.* We will continue to use the definitions and notation introduced in Section 1. We start by presenting the mathematical background.

Let R be a ring, $C \subseteq R$, $u \in R$. We define

$$C + C = \{x + y : x, y \in C,\ x \neq y\} \tag{4}$$

and further let $-C = \{-x : x \in C\}$ and $C - u = \{x - u : x \in C\}$. We say that $A, B \subset R$ are *shift equivalent* if there exists a $v \in R$ such that $A = B - v$.

Definition 1. *A subset $S \subseteq \mathbb{Z}_n$ is called a* strict sum cover *of $\mathbb{Z}_n$, abbreviated* SSC(n), *if $S + S = \mathbb{Z}_n$.*

The adjective *strict* emphasizes the condition $x \neq y$ in (4). Many papers (e.g. [6, 7]) consider sumsets both with and without this distinctness condition, hence we feel the need to emphasize the choice made in our definition.

Definition 2. *A subset $S \subseteq \mathbb{Z}_n$ is called* symmetric *if $0 \in S$ and $-S = S$. A subset $S \subseteq \mathbb{Z}_n$ is called a* symmetric strict sum cover of $\mathbb{Z}_n$, *abbreviated* SSSC(n), *if S is symmetric and S is an* SSC(n).

Lemma 3. *If $A = \{0, \pm a_1, \ldots, \pm a_d\}$ is an* SSSC(n), *then*

$$s(x_1, \ldots, x_d) = \left(\sum_{i=1}^{d} a_i x_i \right) \bmod n$$

is a symbol-assignment function that allows the sender to embed any symbol in $\mathbb{Z}_n$ into any cell in $\mathbb{Z}^d$ by at most two ±1-changes.

The proof is a straightforward extension of the argument for the case of one ±1-change that was given near the end of Section 1. Note the importance of the condition $x \neq y$ in (4); without imposing this condition it could happen that we require one change of amplitude 2. However, per the discussion in Section 1, two changes of amplitude 1 are preferable to one change of amplitude 2.

Lemma 3 makes our objective fairly obvious: Given d, we wish to maximize n such that an SSSC(n) with $2d + 1$ elements exists.

Definition 4. *For a positive integer k we denote by $n_\gamma(k)$ the largest n such that an* SSC(n) *of cardinality k exists. For an odd positive integer k we denote by $\hat{n}_\gamma(k)$ the largest n such that an* SSSC(n) *of cardinality k exists.*

The notation $n_\gamma(k)$ was introduced in the influential paper by Graham and Sloane [6]. To the best of our knowledge the SSSC(n) have not been studied in the literature; hence the notation $\hat{n}_\gamma(k)$ is new.

Clearly for all odd k we have $\hat{n}_\gamma(k) \leq n_\gamma(k)$.

Proposition 5. *For $3 \leq k \leq 13$, k odd, we have $\hat{n}_\gamma(k) = n_\gamma(k)$.*

Proof. The values $n_\gamma(k)$ for $k \leq 14$ are determined in [7]; they are tabulated in the last row of Table 1 therein. The corresponding SSC$(n_\gamma(k))$ are tabulated in Table 4 of that paper. We will now show that for odd $k \in [3, 13]$, each optimal strict sum cover given in [7] is shift equivalent to a symmetric set:
$k = 3$, $C = \{0, 1, 2\} \subset \mathbb{Z}_3$, $C = \{0, \pm 1\} \subset \mathbb{Z}_3$
$k = 5$, $C = \{0, 1, 2, 3, 6\} \subset \mathbb{Z}_9$, $C - 6 = \{0, \pm 3, \pm 4\} \subset \mathbb{Z}_9$
$k = 7$, $C = \{0, 1, 2, 3, 4, 8, 13\} \subset \mathbb{Z}_{17}$, $C - 2 = \{0, \pm 1, \pm 2, \pm 6\} \subset \mathbb{Z}_{17}$
$k = 9$, $C = \{0, 1, 2, 6, 9, 12, 16, 17, 18\} \subset \mathbb{Z}_{30}$, $C - 9 = \{0, \pm 3, \pm 7, \pm 8, \pm 9\} \subset \mathbb{Z}_{30}$
$k = 11$, $C = \{0, 1, 11, 12, 18, 22, 24, 27, 30, 32, 36\} \subset \mathbb{Z}_{42}$,
$C - 27 = \{0, \pm 3, \pm 5, \pm 9, \pm 15, \pm 16\} \subset \mathbb{Z}_{42}$
$k = 13$, $C = \{0, 1, 2, 3, 4, 7, 13, 21, 29, 36, 44, 52, 58\} \subset \mathbb{Z}_{61}$,
$C - 2 = \{0, \pm 1, \pm 2, \pm 5, \pm 11, \pm 19, \pm 27\} \subset \mathbb{Z}_{61}$ □

A lower bound on $n_\gamma(k)$ is given in [2] by a simple construction that uses the sets

$$T(r,k) = \{0,1,\ldots,r-1\} \cup \{2r-2, 3r-2, \ldots, kr-2\}.$$

We make the following observation:

Proposition 6. *If r is odd or k is even, then $T(r,k)$ is shift equivalent to a symmetric set.*

Proof. If r is odd, then the set

$$T(r,k) - \frac{r-1}{2}$$

is symmetric. If k is even, then the set

$$T(r,k) - \left(\frac{k+2}{2}r - 2\right)$$

is symmetric. The verification is straightforward; we omit its details. □

Proposition 7. *Let $k = 2d+1$. Then $\hat{n}_\gamma(k) \geq d^2 + 3d - 1$.*

Proof. Proposition 2.3 of [2] and some extra calculations show that, for each d, $T(d+1, d+1)$ is an $\mathrm{SSC}(d^2+3d-1)$ of cardinality $2d+1$. By Proposition 6, $T(d+1,d+1)$ is shift equivalent to a symmetric set for each d. □

Proposition 7 shows that the scheme which uses two changes per cell is superior to the scheme using one change per cell, assuming of course a fair comparison when both schemes have the same overall distortion rate $2/d$. Indeed, if d is even, then using $\mathbb{Z}^d \simeq \mathbb{Z}^{d/2} \times \mathbb{Z}^{d/2}$ and defining the symbol-assignment function by applying (3) to each of the factors produces a symbol set of cardinality $(2d/2+1)(2d/2+1) = d^2+2d+1$. Similarly, if d is odd then using $\mathbb{Z}^d \simeq \mathbb{Z}^{(d-1)/2} \times \mathbb{Z}^{(d+1)/2}$ and defining the symbol-assignment function by applying (3) to each of the factors produces a symbol set of cardinality $(2(d-1)/2+1)(2(d+1)/2+1) = d^2+2d$. In either case this is less than the d^2+3d-1 symbols guaranteed by Proposition 7 combined with Lemma 3.

By non-exhaustive computer search we have verified that the bound of Proposition 7 is not tight for odd k in the range $9 \leq k \leq 61$. For example, $\{0, \pm 3, \pm 12, \pm 13, \pm 21, \pm 26, \pm 48, \pm 52, \pm 54, \pm 65, \pm 84, \pm 91\}$ is an SSSC(195) of cardinality 23 while Proposition 7 only guarantees $\hat{n}_\gamma(23) \geq 153$. An interesting open problem is to give a systematic construction of examples that improve the bound of Proposition 7.

3 Conclusion

We have extended our previous work on steganography schemes with pooling to the case when two changes per cell are allowed. We have shown that such schemes can be obtained from a specialized, symmetric version of sum covers known in combinatorial design theory. We gave a construction that is better in the information versus distortion metric than the schemes with one change per cell.

A number of interesting questions about the symmetric strict sum covers remain open. We conjecture that the equality $\hat{n}_\gamma(k) = n_\gamma(k)$ holds for a larger set of values k than those established in Proposition 5. A construction of examples that improve the bound of Proposition 7 would have practical value. It appears that the optimal covers achieving the value $\hat{n}_\gamma(k)$ often possess a lot of symmetry; it would be interesting to study this phenomenon theoretically.

References

[1] J. Bierbrauer, Personal communication, Available from http://www.ws.binghamton.edu/fridrich/covcodes.pdf, 1998.

[2] M. Chateauneuf, A. C. H. Ling, D. R. Stinson, Slope packings and coverings, and generic algorithms for the discrete logarithm problem. *J. Combin. Des.* 11, 2003, 36-50.

[3] J. Fridrich, P. Lisoněk, Grid colorings in steganography. *IEEE Trans. Inform. Theory* 53, 2007, 1547-1549.

[4] J. Fridrich, P. Lisoněk, D. Soukal, On steganographic embedding efficiency. *Proc. 8th Inform. Hiding Conf.* (J. Camenisch et al., Eds.), Lect. Notes Comp. Sci. 4437, 2007, 282-296.

[5] F. Galand, G. Kabatiansky, Steganography via covering codes, *IEEE Intern. Symp. Inform. Theory*, Yokohama, Japan, 2003. Slides available at www-rocq.inria.fr/secret/Fabien.Galand/PAPERS/galand_isit03.pdf.

[6] R. L. Graham, N. J. A. Sloane, On additive bases and harmonious graphs, *SIAM J. Algebr. Discr. Math.* 1, 1980, 382-404.

[7] H. Haanpää, Minimum sum and difference covers of abelian groups, *J. Integer Seq.* 7, 2004, Article 04.2.6, 10 pp. Available from http://www.emis.de/journals/JIS/VOL7/Haanpaa/haanpaa.pdf.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 192-198*

Properties of codes in rank metric

Pierre Loidreau Pierre.Loidreau@m4x.org
CELAR and IRMAR, Université de Rennes

Abstract. We study properties of rank metric and codes in rank metric over finite fields. We show that perfect codes do not exist. We derive an equivalent of the Varshamov-Gilbert bound in Hamming metric. We study the asymptotic behavior of the minimum rank distance of codes that are on GV. We show that the packing density of maximum rank distance codes is lower bounded by a function depending on the error-correcting capability. We show that there are asymptotically perfect codes correcting errors of rank 1 over fields of characteristic 2.

1 Introduction

Apart from cryptographic applications and applications in tape recording, rank metric found recently many more applications in the field of random network coding and construction of optimal rate-diversity tradeoff space-time codes.

In this paper, we first recall properties of rank metric and existing bounds. We show that perfect codes cannot exist in rank metric. Then we exhibit an asymptotic relation between parameters of a code which is said to be on GV, that is, which satisfies the Varshamov-Gilbert bound in rank metric.

We also study codes which reach the Singleton bound. These codes are called MRD-codes for *Maximum Rank Distance* codes. After recalling the formula given by Gabidulin on the rank distribution of linear MRD-codes, we present some simulations showing that rank distribution of random codes and of MRD-codes is very similar. In addition, we prove that the density of *correctable* errors for MRD-codes corresponding to codes formed with square matrices is lower bounded by a function depending only on the error-correcting capability of the code. In the special case of fields of characteristic 2, we show that we can construct a family of codes over fields of characteristic 2 that is asymptotically perfect.

2 Properties of rank metric

Let q be a power of a prime and let $\mathbf{b} = (\beta_1, \ldots, \beta_n)$ be a basis of $GF(q^m)$ over $GF(q)$. The integer n denotes the length of the code. The rank norm over $GF(q)$ of an element of $GF(q^m)^n$ is defined by

Definition 1 ([1]) *Let* $\mathbf{x} = (x_1, \ldots, x_n) \in GF(q^m)^n$. *The rank of* $\mathbf{x}$ *on* $GF(q)$, *is the rank of matrix*

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix},$$

where $x_j = \sum_{i=1}^{n} x_{ij}\beta_i$. *It is denoted by* $Rk(\mathbf{x})$

Rank metric is the metric over $GF(q^m)^n$ induced by the rank norm. Spheres and balls in rank metric have the following expression:

- Sphere of radius $t \geq 0$: $\mathcal{S}_t \stackrel{def}{=} \{\mathbf{y} \in GF(q^m)^n \mid \mathrm{Rk}(\mathbf{y}) = t\}$
- Ball of radius $t \geq 0$: $\mathcal{B}_t \stackrel{def}{=} \cup_{i=0}^{t} \mathcal{S}_i$

We have the following bounds:

$$\left\{ \begin{array}{lcl} q^{(m+n-2)t-t^2} & \leq \mathcal{S}_t \leq & q^{(m+n+1)t-t^2} \\ q^{(m+n-2)t-t^2} & \leq \mathcal{B}_t \leq & q^{(m+n+1)t-t^2+1} \end{array} \right. \tag{2.1}$$

Let $\mathcal{C} \subset GF(q^m)^n$ for m and n non-zero integers. If M denotes the cardinality of $\mathcal{C}$ and $d \stackrel{def}{=} \min_{\mathbf{c}_1 \neq \mathbf{c}_2 \in \mathcal{C}}(\mathrm{Rk}(\mathbf{c}_1 - \mathbf{c}_2))$ we say that $\mathcal{C}$ is a $(n, M, d)_r$ code over $GF(q^m)$. The integer d is called the *minimum rank distance* of $\mathcal{C}$.

3 Upper bounds and perfect codes

In this section we recall a Singleton-like bound for rank metric codes and state an equivalent to the sphere-packing bound. We show that there are no perfect codes in rank metric.

Proposition 1 *Let* $\mathcal{C}$ *be a* $(n, M, d)_r$ *code over* $GF(q^m)$. *We have*

- Singleton-like bound: $M \leq q^{\min(m(n-d+1), n(m-d+1))}$.
- Sphere packing-like bound: *If* $t = \lfloor (d-1)/2 \rfloor$, *then*

$$M \times \mathcal{B}_t \leq q^{mn}, \tag{3.2}$$

For the proof of Singleton-like bound see [1, 6]. The proof of the *sphere-packing* bound comes from the fact that, for rank metric, two balls of radius $t = \lfloor (d-1)/2 \rfloor$ centered on codewords do not intersect. Thus, the full packing has size less than the whole space. The proof is similar to that of Hamming metric.

If we define perfect codes as usual, that is: an $(n, M, d)_r$-code over $GF(q^m)$ is perfect if and only if $M \times \mathcal{B}_t = q^{mn}$, we can investigate the existence of perfect codes. The following proposition answers the question

Proposition 2 *There are no perfect codes in rank metric.*

Proof. The proof can be derived from the bounds (2.1)

4 A Varshamov–Gilbert like bound

In rank metric the equivalent of Varshamov–Gilbert (GV) bound is given by the following result:

Proposition 3 *Let m, n, M, d be positive integers. If*

$$M \times \mathcal{B}_{d-1} < q^{mn}, \tag{4.3}$$

then there exists a $(n, M+1, d)_r$-code over $GF(q^m)$.

From this result we define the property for some code to be on GV:

Definition 2 *An $(n, M, d)_r$-code is said to be on GV if*

$$(M-1) \times \mathcal{B}_{d-1} < q^{mn} \leq M \times \mathcal{B}_{d-1}, \tag{4.4}$$

Now we proove the following result given the relations between the parameters of a $(n, M, d)_r$, which is on *GV* and whose cardinality is not *too small.*

Proposition 4 *Consider an $(n, M, d)_r$-code $\mathcal{C}$ over $GF(q^m)$ where $m = m(n) \geq n$. Then, if $\mathcal{C}$ is on GV we have*

$$\frac{d}{m+n} \overset{n \to +\infty}{\sim} \frac{1}{2} - \frac{\sqrt{\log_q M}}{m+n} \sqrt{1 + \frac{(m-n)^2}{4 \log_q M}},$$

provided $\log_q M = \lambda(n)m$, where $\lambda(n) = o(n)$ tends to $+\infty$ with n.

Proof. By taking the base q logarithm of the inequalities (2.1), we obtain from property (4.4) that

$$\begin{cases} mn \leq (m+n+1)(d-1) - (d-1)^2 + 1 + \log_q M, \\ \log_q(M-1) + (m+n-2)(d-1) - (d-1)^2 < mn. \end{cases}$$

Since $M \geq 2$ we have further that $\log_q(M-1) \geq \log_q(M) - \log_q(2) \geq \log_q(M) - 1$. Hence the minimum distance of the code must satisfy

$$\begin{cases} 0 \leq -d^2 + (m+n+3)d + \log_q M - mn - (m+n+1), \\ 0 \geq -d^2 + (m+n)d + \log_q M - mn - (m+n). \end{cases}$$

The inequalities are given by second order equations whose discriminant are respectively

$$\Delta_1 = (m-n)^2 + 4\log_q(M) + 2(m+n) + 5,$$
$$\Delta_2 = (m-n)^2 + 4\log_q(M) - 4(m+n).$$

Therefore the minimum distance of a code on GV satisfies the inequalities

$$\frac{1}{2} - \frac{-\sqrt{\Delta_1} + 3}{2(m+n)} \leq \frac{d}{m+n} \leq \frac{1}{2} - \frac{\sqrt{\Delta_2}}{2(m+n)}.$$

Under the conditions of the theorem ($\log_q M = \lambda(n)(m+n)$, where $\lambda(n) = o(n)$ and tends to infinity with n), it is not very difficult to complete the proof of the proposition. ■

Example 1 *A special case is when $m = n$ and for a family of constant rate codes $0 < R < 1$ that is*

$$\log_q M = n^2 R.$$

In that case we have

$$\frac{d}{n} \sim 1 - \sqrt{R}.$$

This result implies that the ratio of the minimum rank distance on the length of the code is asymptotically constant.

5 Maximum rank distance codes

Singleton inequality gives an upper bound on the cardinality of codes with given parameters. We call optimal codes or MRD (*Maximum Rank Distance*) codes, codes attaining the Singleton bound

Definition 3 (MRD-codes – [1]) *A $(n, M, d)_r$-code over $GF(q^m)$ is called MRD if*

- $M = q^{m(n-d+1)}$, *if $n \leq m$.*
- $M = q^{m(n-d+1)}$, *if $n > m$*

We study properties of MRD codes such as the distribution of the rank of codewords as well as bounds on their packing density.

5.1 Rank weight distribution of MRD-codes

In Hamming metric, the weight distribution of MDS-codes is well-known [5]. Gabidulin showed the rank distribution of codes in rank metric can be expressed by

Proposition 5 ([1]) *Let $A_s(n,d)$ be the number of rank s codewords of an MRD-code over $GF(q^m)$. Then*

$$A_{d+\ell}(n,d) = \begin{bmatrix} n \\ d+\ell \end{bmatrix}_q \sum_{t=0}^{\ell} (-1)^{t+\ell} \begin{bmatrix} d+\ell \\ \ell+t \end{bmatrix}_q q^{\binom{\ell-t}{2}} \left(q^{m(t+1)}-1\right), \qquad (5.5)$$

where $\begin{bmatrix} n \\ i \end{bmatrix}_q$ is the Gaussian binomial.

Our contribution to this section comes from the simulations we made to evaluate the *randomness degree* of MRD-codes. By using these simulations we obtained that the rank distribution of random $GF(q)$-linear codes in rank metric was almost identical to the weight distribution of linear MRD-codes. Results are presented in table 5.1. The table gives the base 2 logarithm of the proportion $A_i(n,d)/2^{mn}$ for $n=32$, $m \geq 32$. The left-most curve corresponds to $m=32$, the right-most to $m=40$. We made simulations for random $GF(q)$-linear codes as well as for MRD-codes *sufficiently* large with the same parameters. For ranks significantly greater than the minimum rank distance both curves coincide very accurately.

5.2 Packing density of MRD codes

In section 3 we proved that no perfect codes existed in rank metric. However a natural question can be: what is the *defect* of *perfectitude* of MRD-codes, that is, given an $(n,M,d)_r$ MRD-code what is the volume of the space covered by balls of radius $\lfloor (d-1)/2 \rfloor$ compared to the volume of the whole space. The *packing density* of the code is thus defined by

$$D = \frac{M\mathcal{B}_t}{q^{mn}},$$

where $t = \lfloor (d-1)/2 \rfloor$ is the rank error-correcting capability of the code. By using the bounds (2.1), we prove

Proposition 6 (Packing density of MRD-codes) *Let C be a MRD-code, $(n, q^{m(n-2t)}, 2t+1)_r$ over $GF(q^m)$. The packing density of C satisfies*

$$\frac{1}{q^{(m-n+2)t+t^2}} \leq D \leq \frac{1}{q^{(m-n-1)t+t^2}},$$

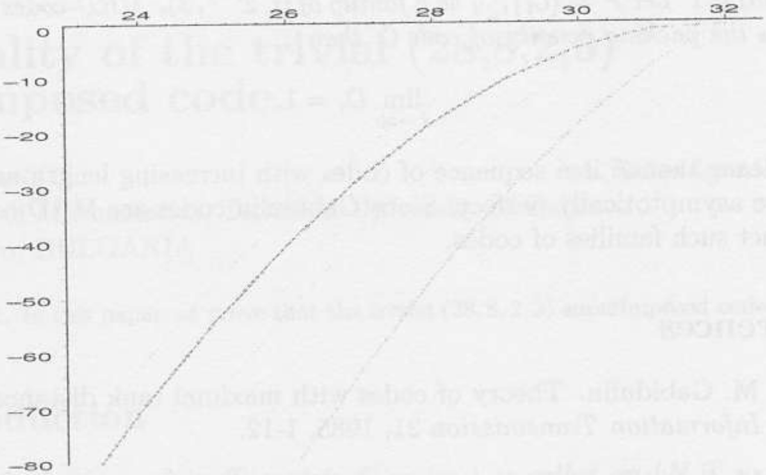

Table 1: Base 2 logarithm of proportion of words of given rank in an MRD-codes of length $n = 32$ over $GF(2^m)$, where $m = 32,\ 33,\ 35,\ 40$.

This proposition shows that, whenever the length of the code equals the extension degree, *i.e.* $n = m$, and if n tends to ∞, then the packing density is lower bounded by q^{-t^2-2t}, which depends only on the rank error-correcting capability of the code.

Particular case of rank 1 correcting MRD codes For rank 1 MRD codes where $m = n$, we can express the exact formulas and obtain

Proposition 7 *An* $(n, q^{n-2}, 3)_r$ *MRD-code over* $GF(q^n)$ *has a packing density equal to*

$$D = \frac{1 - 2q^{-n} + q^{-2n+1}}{q - 1}. \tag{5.6}$$

There is a special interest in the binary case. In section 3, we showed that there are no perfect codes in rank metric. However from previous proposition wd have

Corollary 1 *Let $\mathcal{F} = \{C_i\}_{i \geq 2}$ be a family of $(i, 2^{i-2}, 3)_r$ MRD-codes over $GF(2^i)$. If D_i is the packing density of code C_i then*

$$\lim_{i \to \infty} D_i = 1.$$

This means that $\mathcal{F}$ is a sequence of codes with increasing length and alphabet that are asymptotically perfect. Since Gabidulin codes are MRD codes we can construct such families of codes.

References

[1] E. M. Gabidulin. Theory of codes with maximal rank distance. *Problems of Information Transmission* 21, 1985, 1-12.

[2] E. M. Gabidulin, A fast matrix decoding algorithm for rank-error correcting codes, *Algebr. Coding, Lect. Notes Comp. Sci.* 573, G. Cohen, S. Litsyn, A. Lobstein, G. Zémor, editors, Springer-Verlag, 1991, 126-133.

[3] E .M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, Ideals over a non-commutative ring and their application in cryptology, *Adv. Cryptol. - EUROCRYPT'91, Lect. Notes Comp. Sci.* 547, D. W. Davies, editor, Springer-Verlag, 1991, 482-489.

[4] P. Loidreau, A Welch-Berlekamp like algorithm for decoding Gabidulin codes, *Proc. Fourth Intern. Workshop Cod. Crypt., Lect. Notes Comp. Sci.* 3969, Ø. Ytrehus, editor, 2006, 36-45.

[5] F. J. MacWilliams, N. J. A. Sloane, *The theory of error-correcting codes*, North Holland, 1977.

[6] A. V. Ourivski, E. M. Gabidulin, B. Honary, B. Ammar, Reducible rank codes and their applications to cryptography, *IEEE Trans. Inform. Theory* 49, 2003, 3289-3293.

[7] G. Richter, S. Plass, Fast decoding of rank-codes with rank errors and column erasures, *IEEE Intern. Symp. Inform. Theory*, 2004.

[8] R. M. Roth, Maximum-rank array codes and their application to crisscross error correction, *IEEE Trans. Inform. Theory* 37, 1991, 328-336.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 199-202*

Optimality of the trivial (28,8,2,3) superimposed code

MLADEN MANEV[1] ml.manev@gmail.com
Department of Mathematics, Technical University of Gabrovo
5300 Gabrovo, BULGARIA

Abstract. In this paper we prove that the trivial $(28, 8, 2, 3)$ superimposed code is optimal.

1 Introduction

Definition 1 *A binary $N \times T$ matrix $C = (c_{ij})$ is called an (N, T, w, r) superimposed code (SIC) if for any pair of subsets $W, R \subset \{1, 2, \ldots, T\}$ such that $|W| = w$, $|R| = r$ and $W \cap R = \varnothing$ there exists a row $i \in \{1, 2, \ldots, N\}$ such that $c_{ij} = 1$ for all $j \in W$ and $c_{ij} = 0$ for all $j \in R$. We say also that C is a (w, r) superimposed code of length N and size T.*

The trivial code is a simple example for an (N, T, w, r) superimposed code. The length N of the trivial code is $\binom{T}{w}$ and its rows are all possible binary vectors of weight w.

Let $N(T, w, r)$ is the minimum length of an (N, T, w, r) superimposed code for given values of T, w and r. The code is called optimal when $N = N(T, w, r)$. The exact values of $N(T, 2, 3)$ are known for $T \leq 7$.

T	5	6	7
$N(T, 2, 3)$	10	15	21

The trivial $(10, 5, 2, 3)$, $(15, 6, 2, 3)$ and $(21, 7, 2, 3)$ superimposed codes are optimal. Kim and Lebedev [2] have proved that $24 \leq N(8, 2, 3) \leq 28$ and $26 \leq N(9, 2, 3) \leq 30$. Therefore the trivial $(36, 9, 2, 3)$ superimposed code is not optimal. In this paper we prove the nonexistence of $(27, 8, 2, 3)$ superimposed code. Consequently the trivial $(28, 8, 2, 3)$ superimposed code is optimal.

2 Preliminaries

Definition 2 *Two (N, T, w, r) superimposed codes are equivalent if one of them can be transformed into the other by a permutation of the rows and a permutation of the columns.*

[1] Partially supported by the Technical University of Gabrovo under Grant C-801/2008.

Let C be a binary $N \times T$ matrix. Denote by $d(x,y)$ the Hamming distance between two columns x and y and by S_x and S_y – the characteristic sets of the columns x and y respectively. The following lemma is obvious.

Lemma 3 $d(x,y) = |S_x| + |S_y| - 2|S_x \cap S_y|$.

Let $d(x,y,z) = d(x,y) + d(x,z) + d(y,z)$. From Lemma 3 we obtain $d(x,y,z) = d(x,y) + d(x,z) + d(y,z) = 2(|S_x| + |S_y| + |S_z| - |S_x \cap S_y| - |S_x \cap S_z| - |S_y \cap S_z|)$. Consequently $d(x,y,z)$ is even number. Denote by $d_2 = \min\{d(x,y) \mid x,y \in C,\ x \neq y\}$ and by $d_3 = \min\{d(x,y,z) \mid x,y,z \in C,\ x \neq y, x \neq z, y \neq z\}$. It is clear that $3d_2 \leq d_3$. Let $d(C) = \sum_{x,y \in C,\ x \neq y} d(x,y)$.

Lemma 4 *(Plotkin bound) [3]* $\binom{T}{2} d_2 \leq d(C) \leq N \left\lfloor \frac{T}{2} \right\rfloor \left\lfloor \frac{T+1}{2} \right\rfloor$.

Corollary 5 $\binom{T}{3} d_3 \leq (T-2)d(C) \leq (T-2)N \left\lfloor \frac{T}{2} \right\rfloor \left\lfloor \frac{T+1}{2} \right\rfloor$.

Definition 6 *Let $x_1, x_2, ..., x_k$ be different columns of the superimposed code C. The residual code $Res(C, x_1 = v_1, x_2 = v_2, ..., x_k = v_k)$ of C is the code obtained by taking all the rows in which C has value v_i in the column x_i for $i = 1, 2, ..., k$ and deleting the columns $x_1, x_2, ..., x_k$ in the selected rows.*

Lemma 7 *Suppose C is an (N,T,w,r) superimposed code and x and y are two different columns of C. Then*
(a) $Res(C, x = 1)$ is a $(|S_x|, T-1, w-1, r)$ SIC;
(b) $Res(C, x = 0)$ is an $(N - |S_x|, T-1, w, r-1)$ SIC.

Lemma 8 *[2] $N(6,1,2) = 6$ and $N(7,2,2) = 14$.*

Lemma 9 *[1] Any $(6,6,1,2)$ superimposed code is equivalent to the trivial $(6,6,1,2)$ superimposed code*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Using computer programs for generation of $(1,2)$ and $(2,3)$ superimposed codes and for code equivalence we proved the following two lemmas:

Lemma 10 *Any $(7,6,1,2)$ superimposed code is equivalent to one of the codes*

$$C_{1,2,\ldots,7} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ * & * & * & * & * & * \end{pmatrix} \quad C_8 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

The last row of $C_{1,2,\ldots,7}$ *is* 0000000, 0000001, 0000011, 0000111, 0001111, 0011111, 0111111 *or* 1111111 *respectively.*

Lemma 11 *Any* $(21,7,2,3)$ *SIC is equivalent to the trivial* $(21,7,2,3)$ *SIC.*

3 The nonexistence of (27,8,2,3) SIC

Lemma 12 *Let* C *be a* $(27,8,2,3)$ *superimposed code. Then* $d_2 \geq 12$.

Proof. Let x and y be two different columns in C. Since $N(6,1,2) = 6$ (Lemma 8), $|S_x \cap \overline{S}_y| \geq 6$ and $|\overline{S}_x \cap S_y| \geq 6$. Therefore $d(x,y) \geq 12$ and $d_2 \geq 12$. □

Lemma 13 *Let* C *be a* $(27,8,2,3)$ *superimposed code and* x *and* y *are two different columns of* C. *Then* $Res(C, x=0, y=1)$ *contains at most 5 rows of weight* 0 *or* 1.

Proof. Suppose the matrix C contains at least 6 rows of weight 0 or 1. Let C' be the matrix obtained of C by deleting of the column y. C' is a $(27,7,2,3)$ superimposed code and contains 6 rows of weight 0 or 1. Consequently the remaining 21 rows of C' form a $(21,7,2,3)$ superimposed code. According to Lemma 11 this code is equivalent to the trivial $(21,7,2,3)$ superimposed code, hence all its rows are of weight 2. Therefore $d(C') \leq 21 \times 10 + 6 \times 6 = 246$. According to Lemma 12 the distance between any two columns of C' is at least 12. It follows from Lemma 4 that $d(C') \geq \binom{7}{2}.12 = 252$, which is a contradiction. Therefore $Res(C, x=0, y=1)$ contains at most 5 rows of weight 0 or 1. □

Lemma 14 *Let* C *be a* $(27,8,2,3)$ *superimposed code. Then* $d_2 = 14$.

Proof. Let x and y be two columns of C for which $d(x,y) = d_2$. It follows from Lemma 13 that the length of each of the codes $Res(C, x=0, y=1)$ and $Res(C, x=1, y=0)$ is at least 7, hence $d_2 \geq 14$. According to Lemma 4 $d_2 \leq 15$. Consequently one of this residual codes is of length 7 and is equivalent to the code C_8 of Lemma 10. Therefore $d(C) \leq 429$. It follows from Corollary 5 that $d_3 \leq 45\frac{27}{28}$. But d_3 is an even number, hence $d_3 \leq 44$. Consequently $d_2 = 14$. □

Theorem 15 *There is no* $(27, 8, 2, 3)$ *superimposed code.*

Proof. Let C be a (27,8,2,3) superimposed code. It follows from Lemma 14 that there exist two columns x and y such that $d(x, y) = 14$. Hence the residual codes $Res(C, x = 0, y = 1)$ and $Res(C, x = 1, y = 0)$ are equivalent to the code C_8 of Lemma 10. We can write C as follows:

$$\begin{array}{|c|c|}
\multicolumn{1}{c}{x\ y} & \multicolumn{1}{c}{} \\ \hline
0\ 1 & \\
\vdots\ \vdots & (7,6,1,2)\text{ SIC} \\
0\ 1 & \\ \hline
1\ 0 & \\
\vdots\ \vdots & (7,6,1,2)\text{ SIC} \\
1\ 0 & \\ \hline
0\ 0 & \\
\vdots\ \vdots & M\text{ rows} \\
0\ 0 & \\ \hline
1\ 1 & \\
\vdots\ \vdots & 13 - M\text{ rows} \\
1\ 1 & \\ \hline
\end{array}$$

Using a computer program we obtained that there are exactly 30 inequivalent possibilities for the first 14 rows of C. $Res(C, x = 0)$ is an $(M + 7, 7, 2, 2)$ SIC. According to Lemma 8 $M \geq 7$. C is a $(27, 8, 2, 3)$ SIC, hence $M \leq 12$.

Using a computer program we constructed the missing part column by column, checking at each step the condition of Lemma 14, the superimposed code property and the sorted last 13 rows property.

It turned out that the extension to a $(27, 8, 2, 3)$ superimposed code is impossible. Therefore there is no $(27, 8, 2, 3)$ superimposed code. □

Theorem 16 *The trivial* $(28, 8, 2, 3)$ *superimposed code is optimal.*

References

[1] S. Kapralov, M. Manev, The nonexistence of (19,10,2,2) superimposed codes, *Proc. Fourth Intern. Workshop OCRT*, Pamporovo, Bulgaria, 2005, 196-200.

[2] H. K. Kim, V. S. Lebedev, On optimal superimposed codes, *J. Combin. Des.* 12, 2004, 79-91.

[3] M. Plotkin, Binary codes with specified minimum distance, *IRE Trans. Inform. Theory* 6, 1960, 445-450.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria pp. 203-209

Extendability of linear codes over $\mathbb{F}_q$

TATSUYA MARUTA maruta@mi.s.osakafu-u.ac.jp
Department of Mathematics and Information Sciences,
Osaka Prefecture University, Sakai, Osaka 599-8531, JAPAN

Abstract. For an $[n,k,d]_q$ code $\mathcal{C}$, we define a mapping $w_{\mathcal{C}}$ from $\mathrm{PG}(k-1,q)$ to the set of weights of $\mathcal{C}$ via a generator matrix of $\mathcal{C}$. We give a geometric aspect derived from $w_{\mathcal{C}}$ to investigate the extendability of linear codes. We survey known extension theorems and some recent results.

1 Introduction

Let $\mathbb{F}_q^n$ denote the vector space of n-tuples over $\mathbb{F}_q$, the field of q elements. A linear code $\mathcal{C}$ of length n, dimension k and minimum (Hamming) distance d over $\mathbb{F}_q$ is referred to as an $[n,k,d]_q$ code. The *weight* of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$, denoted by $wt(\boldsymbol{x})$, is the number of nonzero coordinate positions in $\boldsymbol{x}$. The weight distribution of $\mathcal{C}$ is the list of numbers A_i which is the number of codewords of $\mathcal{C}$ with weight i. The weight distribution with $(A_0, A_d, ...) = (1, \alpha, ...)$ is also expressed as $0^1 d^\alpha \cdots$. We only consider *non-degenerate* codes having no coordinate which is identically zero.

For an $[n,k,d]_q$ code $\mathcal{C}$ with a generator matrix G, $\mathcal{C}$ is called (l,s)*-extendable* (to $\mathcal{C}'$) if there exist l vectors $h_1, \dots, h_l \in \mathbb{F}_q^k$ such that the extended matrix $[G, h_1^{\mathrm{T}}, \cdots, h_l^{\mathrm{T}}]$ generates an $[n+l, k, d+s]_q$ code $\mathcal{C}'$ ([7]). Then $\mathcal{C}'$ is called an (l,s)*-extension* of $\mathcal{C}$. A $(1,1)$-extendable code is simply called *extendable*. The following is well-known.

Theorem 1.1. [1] *Every $[n,k,d]_2$ code with d odd is extendable.*

As for the (l,s)-extendability, the next theorem is known as 'Construction X'.

Theorem 1.2. [1] *Let $\mathcal{C}$ and $\mathcal{C}_0$ be an $[n,k,d]_q$ code and an $[n,k_0,d_0]_q$ code, respectively, such that $\mathcal{C} \supset \mathcal{C}_0$ and $d < d_0$. If there exists an $[l, k-k_0, d']_q$ code $\mathcal{C}'$, then $\mathcal{C}$ is (l,s)-extendable, where $s = \min\{d', d_0 - d\}$.*

Proof. We give an elementary proof using generator matrices. Take a generator matrix G of $\mathcal{C}$ with two submatrices G_0 and G_1 so that G_0 consisting of the first k_0 rows of G is a generator matrix of $\mathcal{C}_0$ and that the remaining $k-k_0$ rows of G form G_1. Let G' be a generator matrix of $\mathcal{C}'$. Then, the matrix $\left[\begin{array}{c|c} G_0 & O \\ \hline G_1 & G' \end{array}\right]$ generates an (l,s)-extension of $\mathcal{C}$, where O is the zero matrix. □

For example, every $[n,k,d]_2$ code with odd d contains an $[n,k-1,d_0]_2$ code with $d_0 > d$ as a subcode. It might be possible to find a suitable subcode $\mathcal{C}_0$ of $\mathcal{C}$ when $\mathcal{C}$ is a BCH code, but It is not easy to find such a subcode for an arbitrary linear code $\mathcal{C}$ in general. We sometimes need to know the minimum l so that $\mathcal{C}$ is $(l,1)$-extendable.

Problem 1. Find easily checkable conditions to see whether a given $[n,k,d]_q$ code is $(l,1)$-extendable or not.

The aim of this paper is to give a geometric aspect to investigate the $(l,1)$-extendability of linear codes and survey known extension theorems with some applications mainly for $l=1$.

2 A geometric approach

We assume that $k \geq 3$, see [9] for $k=1,2$. Let $\mathcal{C}$ be an $[n,k,d]_q$ code with a generator matrix $G=[g_{ij}]=[g_1,\cdots,g_k]^{\mathrm{T}}$. Put $\Sigma =$PG$(k-1,q)$, the projective space of dimension $k-1$ over $\mathbb{F}_q$. We consider the mapping $w_{\mathcal{C}}$ from Σ to $\{i \mid A_i > 0\}$, the set of weights of $\mathcal{C}$. For $P=\mathbf{P}(p_1,\ldots,p_k)\in\Sigma$ we define *the weight of P with respect to $\mathcal{C}$*, denoted by $w_{\mathcal{C}}(P)$, as

$$w_{\mathcal{C}}(P) = |\{j \mid \sum_{i=1}^{k} g_{ij}p_i \neq 0\}| = wt(\sum_{i=1}^{k} p_i g_i).$$

Let $F_d=\{P\in\Sigma \mid w_{\mathcal{C}}(P)=d\}$. Recall that a hyperplane H of Σ is defined by a non-zero vector $h=(h_0,\ldots,h_{k-1})\in\mathbb{F}_q^k$ as $H=\{P=\mathbf{P}(p_0,\ldots,p_{k-1})\in\Sigma \mid h_0p_0+\cdots+h_{k-1}p_{k-1}=0\}$. h is called the *defining vector of H*.

Lemma 2.1. *$\mathcal{C}$ is extendable if and only if there exists a hyperplane H of Σ such that $F_d \cap H=\emptyset$. Moreover, the extended matrix of G by adding the defining vector of H as a column generates an extension of $\mathcal{C}$.*

Proof. For an $[n,k,d]_q$ code $\mathcal{C}$ with a generator matrix G, there exists a vector $h=(h_0,\ldots,h_{k-1})\in\mathbb{F}_q^k$ such that $[G,h^{\mathrm{T}}]$ generates an $[n+1,k,d+1]_q$ code if and only if $\sum_{i=0}^{k-1} h_ip_i \neq 0$ holds for all $P=\mathbf{P}(p_0,\ldots,p_{k-1})\in F_d$. Equivalently, there exists a hyperplane H with defining vector h such that $F_d\cap H=\emptyset$. □

The above lemma can be easily generalized to the $(l,1)$-extendability.

Theorem 2.2. *$\mathcal{C}$ is $(l,1)$-extendable if and only if there exist l hyperplanes $H_1,\ldots,H_l$ of Σ such that $F_d\cap H_1\cap\cdots\cap H_l=\emptyset$. Equivalently, there exists a $(k-1-l)$-flat Π with $F_d\cap\Pi=\emptyset$.*

Lemma 2.3. [3] *For two linearly independent vectors $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{F}_q^n$, it holds that*

$$\sum_{\lambda \in \mathbb{F}_q} wt(\boldsymbol{a}_1 + \lambda \boldsymbol{a}_2) + wt(\boldsymbol{a}_2) \equiv 0 \pmod{q}.$$

As a consequence of Lemma 2.3, we get the following.

Lemma 2.4. *For a line $L = \{P_0, P_1, \cdots, P_q\}$ in Σ, it holds that*

$$\sum_{i=0}^{q} w_{\mathcal{C}}(P_i) \equiv 0 \pmod{q}.$$

Now, let

$$\begin{aligned} F_0 &= \{P \in \Sigma \mid w_{\mathcal{C}}(P) \equiv 0 \pmod{q}\}, \\ \bar{F_d} &= \{P \in \Sigma \mid w_{\mathcal{C}}(P) \equiv d \pmod{q}\}, \quad F = \Sigma \setminus \bar{F_d}. \end{aligned}$$

The mapping $w_{\mathcal{C}}$ is *trivial* if $F = \emptyset$. For example, $w_{\mathcal{C}}$ is trivial if $\mathcal{C}$ attains the Griesmer bound and if $q|d$ when q is prime [17]. To avoid such cases we assume that $\gcd(d, q) = 1$. Then we have $F_0 \subset F$. If $\bar{F_d}$ contains a line L of Σ, then we have $d \equiv 0 \pmod{q}$ by Lemma 2.4, a contradiction. Hence we get the following.

Lemma 2.5. *F forms a blocking set with respect to lines in Σ if gcd$(d, q) = 1$.*

Most of the known extension theorems presented in the next section can be proved by showing that F contains a hyperplane of Σ.

3 Extension theorems and their applications

A q-ary linear code $\mathcal{C}$ is *w-weight (mod q)* if there exists a w-set $W = \{i_1, \ldots, i_w\} \subset \mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ such that $A_i > 0$ implies $i \equiv i_j \pmod{q}$ for some $i_j \in W$. The condition 'd is odd' in Theorem 1.1 would be replaced by 'gcd$(d, q) = 1$' for general q. But this is not enough for $q > 2$. In this section, we assume that $\mathcal{C}$ is an $[n, k, d]_q$ code with $k \geq 3$ and $\gcd(d, q) = 1$. As a solution of Problem 1, Hill & Lizak showed the following for 2-weight (mod q) codes.

Theorem 3.1. [3],[4] *Every $[n, k, d]_q$ code with gcd$(d, q) = 1$ whose weights (i's such that $A_i > 0$) are congruent to 0 or d (mod q) is extendable.*

Most of the cases one can apply Theorem 3.1 for $q > 3$ are when $d \equiv -1 \pmod{q}$.

Corollary 3.2. *Every $[n, k, d]_q$ code with $d \equiv -1 \pmod{q}$ whose weights are congruent to 0 or -1 (mod q) is extendable.*

The following is the first extension theorem for 3-weight (mod q) codes.

Theorem 3.3. [11] *Every* $[n,k,d]_q$ *code with odd* $q \geq 5$, $d \equiv -2 \pmod q$ *whose weights are congruent to* 0, -1 *or* $-2 \pmod q$ *is extendable.*

Throughout this section, we define the *diversity* of $\mathcal{C}$ as the pair (Φ_0, Φ_1) with

$$\Phi_0 = |F_0| = \frac{1}{q-1} \sum_{q|i, i>0} A_i, \quad \Phi_1 = |F \setminus F_0| = \frac{1}{q-1} \sum_{i \not\equiv 0, d \pmod q} A_i.$$

Theorem 3.4. [8] *Every* $[n,k,d]_q$ *code with* $gcd(d,q) = 1$ *is extendable if*

$$\Phi_1 \leq q^{k-3}(s(q) - q - 1)/(q-1)$$

where $s(q)$ *is the smallest size of a nontrivial blocking set in* $PG(2,q)$.

Theorem 3.5. [12] *Let* $\mathcal{C}$ *be an* $[n,k,d]_3$ *code with diversity* (Φ_0, Φ_1), $gcd(3,d) = 1$, $k \geq 3$. *Then* $\mathcal{C}$ *is extendable if one of the following conditions holds:*
(1) $\Phi_0 = \theta_{k-3}$, (2) $\Phi_1 = 0$, (3) $\Phi_0 + \Phi_1 < \theta_{k-2} + 3^{k-2}$,
(4) $\Phi_0 + \Phi_1 \geq \theta_{k-2} + 2 \cdot 3^{k-2}$, (5) $2\Phi_0 + \Phi_1 \leq 2\theta_{k-2}$,
where $\theta_j = (3^{j+1} - 1)/2$.

Theorem 3.6. [12] *Let* $\mathcal{C}$ *be an* $[n,k,d]_3$ *code with diversity* (Φ_0, Φ_1), $d \equiv 1 \pmod 3$, $k \geq 3$. *Then* $\mathcal{C}$ *is* $(2,2)$*-extendable if*

$$(\Phi_0, \Phi_1) \in \{(\theta_{k-2}, 0), (\theta_{k-3}, 2 \cdot 3^{k-2}), (\theta_{k-2} + 3^{k-2}, 3^{k-2})\}.$$

The condition (3) of Theorem 3.5 is generalized for other q as follows.

Theorem 3.7. [10] *Let* $\mathcal{C}$ *be an* $[n,k,d]_q$ *code with* $gcd(d,q) = 1$, $q = p^h$, p *prime. Then* $\mathcal{C}$ *is extendable if*

$$\sum_{i \not\equiv d \pmod p} A_i < q^{k-2}(2q-1)$$

and if one of the following conditions holds:
(1) $h = 1$ (*i.e.* q *is prime*),
(2) $q = 4$,
(3) $h = 2$ *with* $n \equiv 0 \pmod p$, $d \equiv -1 \pmod p$,
(4) $h = 2$ *with* $n \equiv d \equiv 1 \pmod p$ *and* $A_i = 0$ *for all* $i \equiv d \pmod p$ *with* $i \not\equiv n \pmod q$.

Theorem 3.7 for $q = 4$ was first found by Simonis [16]. When $h \geq 3$, the following result is known.

Theorem 3.8. [10] *Let $\mathcal{C}$ be an $[n,k,d]_q$ code with $gcd(d,q)=1$, $q=p^h$, p prime, $h \geq 3$. Then $\mathcal{C}$ is extendable if*

$$\sum_{i \not\equiv d (\text{mod } p^{h-1})} A_i < q^{k-2}(2q-1).$$

Theorem 3.5 (except for the condition (4)) can be generalized as follows.

Theorem 3.9. [14] *Let $\mathcal{C}$ be an $[n,k,d]_q$ code with diversity (Φ_0, Φ_1), $k \geq 3$, $d \equiv -1 \pmod q$, q odd, whose weights are congruent to 0 or $\pm 1 \pmod q$. Then $\mathcal{C}$ is extendable if one of the following conditions holds:*
(1) $\Phi_0 = \theta_{k-3}$, (2) $\Phi_1 = 0$,
(3) $\Phi_0 + \Phi_1 \geq \theta_{k-2} + \alpha q^{k-2}$, (4) $\alpha\Phi_0 + \Phi_1 \leq \alpha\theta_{k-2}$,

where $\theta_j = (q^{j+1}-1)/(q-1)$, $\alpha = \theta_1/2$.

When (Φ_0, Φ_1) is none of the types in Theorem 3.9(1), we need more information about $\mathcal{C}$.

Theorem 3.10. [14] *Let $\mathcal{C}$ be an $[n,k,d]_q$ code with diversity (Φ_0, Φ_1), $k \geq 3$, $d \equiv -1 \pmod q$, q odd, whose weights are congruent to 0 or $\pm 1 \pmod q$. Then $\mathcal{C}$ is not extendable if (Φ_0, Φ_1) satisfies none of the criteria of Theorem 3.9 and if*

$$\sum_{d<i\equiv d \pmod q} A_i < \frac{(q-1)^2 q^{k-3}}{2}. \tag{3.1}$$

As for even q, the following theorem can be proved.

Theorem 3.11. [14] *Let $\mathcal{C}$ be an $[n,k,d]_q$ code with q even, $d \equiv -1 \pmod q$, whose weights are congruent to 0 or $\pm 1 \pmod q$, $k \geq 3$. Then $\mathcal{C}$ is extendable.*

Extension theorems can be applied to find new codes from old ones or to prove the nonexistence of codes with certain parameters. For example, we demonstrate the nonexistence of $[245,5,183]_4$ codes. For a putative $[245,5,183]_4$ code $\mathcal{C}_1$, considering the residual codes (see Theorem 2.7.1 in [6]) yields that $A_i = 0$ for all $i \notin \{0,183,184,196,228,244,245\}$. Applying Theorem 3.11, $\mathcal{C}_1$ is extendable, which contradicts that a $[246,5,184]_4$ code does not exist. See also [15] for the extendability of quaternary linear codes.

Next, we give a typical example one can apply Theorems 3.10 and 3.11. Let $\mathcal{C}_2$ be a $[q+1,3,q-1]_q$ code, which is MDS (see [6]) and has the unique weight distribution

$$0^1 (q-1)^{(q+1)q(q-1)/2} q^{q^2-1} (q+1)^{q(q-1)^2/2}.$$

So, the weights of $\mathcal{C}_2$ are congruent to 0 or $\pm 1 \pmod q$ and its diversity $(\theta_1, q(q-1)/2)$ satisfies none of the conditions of Theorem 3.9. When q is odd, $\mathcal{C}_2$ is not extendable by Theorem 3.10 since the left hand side of (3.1) is 0. This fact is known as the completeness of $(q+1)$-arcs in PG(2,q) for q odd, see [5]. On the other hand, it is also known that $\mathcal{C}_2$ is extendable when q is even, as guaranteed by Theorem 3.11. The inequality (3.1) could be slightly improved according to diversities just as for the case when $q = 3$ ([12],[13]).

As for other types of 3-weight (mod q) codes, Cheon and Maruta recently proved the following.

Theorem 3.12. [2] *Let $\mathcal{C}$ be an $[n,k,d]_q$ code with even $q \geq 4$, $k \geq 3$, whose weights are congruent to $0, -1$ or $-2 \pmod q$ and $d \equiv -1 \pmod q$. Then $\mathcal{C}$ is extendable.*

Theorem 3.13. [2] *Let $\mathcal{C}$ be an $[n,k,d]_q$ code with odd $q \geq 5$, $k \geq 3$, whose weights are congruent to $0, -1$ or $-2 \pmod q$ and $d \equiv -1 \pmod q$. Then $\mathcal{C}$ is extendable if $(\Phi_0, \Phi_1) \neq (\binom{q}{2}q^{k-3} + \theta_{k-3}, \binom{q}{2}q^{k-3})$.*

Problem 2. Find a new extension theorem for 4-weight (mod q) codes.

References

[1] J. Bierbrauer, *Introduction to Coding Theory*, Chapman & Hall/CRC, 2005.

[2] E.J. Cheon, T. Maruta, A new extension theorem for 3-weight modulo q linear codes over $\mathbb{F}_q$, in preparation.

[3] R. Hill, An extension theorem for linear codes, *Des. Codes Crypt.* 17, 1999, 151-157.

[4] R. Hill, P. Lizak, Extensions of linear codes, *Proc. IEEE Intern. Symp. Inform. Theory*, Whistler, Canada, 1995.

[5] J. W. P. Hirschfeld, *Projective Geometries over Finite Fields*, 2nd ed., Clarendon Press, Oxford, 1998.

[6] W. C. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, 2003.

[7] A. Kohnert, (l,s)-extension of linear codes, *Discr. Math.*, to appear.

[8] I. Landjev, A. Rousseva, An extension theorem for arcs and linear codes, *Probl. Inform. Transm.* 42, 2006, 319-329.

[9] T. Maruta, On the extendability of linear codes, *Fin. Fields Their Appl.* 7, 2001, 350-354.

[10] T. Maruta, Extendability of linear codes over GF(q) with minimum distance d, $\gcd(d,q) = 1$, *Discr. Math.* 266, 2003, 377-385.

[11] T. Maruta, A new extension theorem for linear codes, *Fin. Fields Their Appl.* 10, 2004, 674-685.

[12] T. Maruta, Extendability of ternary linear codes, *Des. Codes Crypt.* 35, 2005, 175-190.

[13] T. Maruta, K. Okamoto, Some improvements to the extendability of ternary linear codes, *Fin. Fields Their Appl.* 13, 2007, 259-280.

[14] T. Maruta, K. Okamoto, Extendability of 3-weight (mod q) linear codes over $\mathbb{F}_q$, submitted.

[15] T. Maruta, M. Takeda, K. Kawakami, New sufficient conditions for the extendability of quaternary linear codes, *Fin. Fields Their Appl.*, to appear.

[16] J. Simonis, Adding a parity check bit, *IEEE Trans. Inform. Theory* 46, 2000, 1544-1545.

[17] H.N. Ward, Divisibility of codes meeting the Griesmer bound, —it J. Combin. Theory Ser. A 83, 1998, 79-93.

Doubles of Hadamard 2-(15,7,3) designs

ZLATKA MATEVA ziz@abv.bg
Department of Mathematics, Technical University, Varna, BULGARIA

Abstract. Nonisomorphic 2-(15,7,6) designs which are doubles of Hadamard 2-(15,7,3) designs are constructed. The automorphism groups of the Hadamard designs are considered to reduce the number of isomorphic ones among the constructed doubles. Canonical form of the incidence matrices is used to reject isomorphic designs and to establish the order of their automorphism groups. There are 5 non isomorphic 2-(15,7,3) designs H_1, H_2, H_3, H_4 and H_5. All doubles of H_1 and H_i are classified for $i = 1, 2, 3, 4$ and 5.

1 Introduction

Basic definitions. For the basic concepts and notations concerning combinatorial designs refer, for instance, to [1], [2], [14].

Let $\mathcal{P} = \{P_i\}_{i=1}^{v}$ be a finite set of *points*, and $\mathcal{B} = \{B_j\}_{j=1}^{b}$ a finite collection of k-element subsets of $\mathcal{P}$, called *blocks*. $D = (\mathcal{P}, \mathcal{B})$ is a *design* with parameters t-(v,k,λ) if any t-subset of $\mathcal{P}$ is contained in exactly λ blocks of $\mathcal{B}$. Any point $P_i \in \mathcal{P}$ occurs in the same number r of blocks of $\mathcal{B}$. If $v = b$ the design is *symmetric* and $r = k$ too. A symmetric 2-$(4m-1,2m-1,m-1)$ design is called a Hadamard 2-design.

Two designs D_1 and D_2 are *isomorphic* ($D_1 \sim D_2$) if there exists a one-to-one correspondence between the point and block sets of the first design and the point and block sets of the second design, and if this one-to-one correspondence does not change the incidence. Isomorphic designs are indistinguishable by algebraical means. In some cases, however, it is very important to distinguish isomorphic, but different designs. We then speak about *labelled* designs (see for instance [3]) and mean that the points are ordered in some way.

An *automorphism* is an isomorphism of the design to itself. The set of all automorphisms of a design forms a group called its *full group of automorphisms*. Each subgroup of this group is a group of automorphisms of the design.

Each 2-(v,k,λ) design determines the existence of 2-$(v,k,2\lambda)$ designs. These 2-$(v,k,2\lambda)$ designs are called *quasidoubles* of 2-(v,k,λ) designs. A quasidouble 2-$(v,k,2\lambda)$ design is *reducible* into two 2-(v,k,λ) designs if there is a partition of its blocks into two subcollections each of which forms a 2-(v,k,λ) design. A reducible quasidouble is called a *double*.

We denote the set $\{1, 2, ..., v\}$ by N_v, the symmetric group of all permutations of N_v by S_v, the full automorphism group of a design D by $Aut(D)$, and a double design which is reducible to the designs D_1 and D_2 by $[D_1 \parallel D_2]$.

Incidence matrix of a labeled 2-(v,k,λ) design is a $(0,1)$ matrix with v rows and b columns, where the element of the i-th row $(i \in N_v)$ and j-th column $(j \in N_b)$ is 1 if the i-th point of $\mathcal{P}$ occurs in the j-th block of $\mathcal{B}$ and 0 otherwise. The design is completely determined by its incidence matrix. The incidence matrices of two isomorphic designs are equivalent.

Let us denote the incidence matrix of a design D by $\mathcal{D}$. Define standard lexicographic order relations on the rows and columns of $\mathcal{D}$. We denote by $\mathcal{D}^{sort}$ a column-sorted matrix obtained from $\mathcal{D}$ by sorting the columns in decreasing order. Define a standard lexicographic order on the matrices considering each matrix as an ordered v-tuple of the v rows. Let $\mathcal{D}^{max} = max\{\varphi\mathcal{D}^{sort} : \varphi \in S_v\}$ (corresponds to the notation *romim* [13] about the incidence matrix of a graph). $\mathcal{D}^{max}$ is a canonical form of the incidence matrix $\mathcal{D}$.

Other notations. Let $D = (\mathcal{P}, \mathcal{B})$ be a $2-(v,k,\lambda)$ design, $G \subseteq S_v$ and the permutation $\sigma = (\sigma_1, \sigma_2, ..., \sigma_v) \in G$.

$\sigma P_i = P_{\sigma_i}$,

$\sigma D = (\sigma\mathcal{P}, \sigma\mathcal{B}) : B = \{P_{i_j}\}_{j=1}^{k} \in \mathcal{B} \Leftrightarrow \sigma B = \{P_{\sigma_{i_j}}\}_{j=1}^{k} \in \sigma\mathcal{B}$.

$\sigma D_1 = D_2 \Leftrightarrow D_1 \sim D_2$.

$P^G = \{\sigma P : \sigma \in G\}$ (*orbit* of P with respect to G).

$G_P = \{\sigma : \sigma \in G, \sigma P = P\}$ (*stabiliser* of the point P with respect to G).

$G_{\{i_1,i_2,...,i_m\}} = \{\sigma \in G : \forall j \in N_m, \sigma P_{i_j} = P_{i_j}\} = \bigcap G_{P_{i_j}}$ for $m \in N_v$ is the *stabiliser of the point set* $\{P_{i_1}, P_{i_2}, ..., P_{i_m}\}$ with respect to G.

Hadamard 2-(15,7,3) designs. There are five nonisomorphic 2-(15,7,3) designs. We denote them by $H_1, H_2, ..., H_5$ such that $\forall i \in N_4 : \ \mathcal{H}_i^{max} > \mathcal{H}_{i+1}^{max}$.

The full automorphism groups of H_1, H_2, H_3, H_4 and H_5 are of orders 20160, 576, 96, 168 and 168 respectively. We use automorphisms and point orbits of these groups to decrease the number of constructed isomorphic designs. The number of isomorphic but distinguished 2-(15,7,3) designs is $15! \sum_{n=1}^{5} \frac{1}{Aut(D_n)} = 31524292800$.

The present work. Subject of the present work are 2-(15,7,6) designs, which are reducible into two Hadamard 2-(15,7,3) designs H_1 and $H_i, i = 1, 2, ..., 5$. Their block collection is obtained as a union of the block collections of H_1 and φH_i, $\forall\varphi \in S_v$. The action of $Aut(H_1)$ and $Aut(H_i)$ is considered and doubles are not constructed for part of the permutations of S_v, because it is shown that they lead to isomorphic doubles.

Transformation of matrices in some canonical form is used by many authors for the rejection of equivalent solutions (see for instance [4], [6] and [7]). In the present work classification of the obtained designs is made by the help of $\mathcal{D}^{max}$.

There exist at least 57810 nonisomorphic 2-(15,7,6) designs [12]. This lower bound is improved in [10] and [11], where all 2-(15,7,6) designs with automorphisms of prime odd orders were constructed, their number was determined to be 92 323 and 12 786 of them were found to be reducible. Here a classification of all 2-(15,7,6) designs reducible into H_1 and H_i, $i = 1, 2, 3, 4, 5$ is presented. The results coincide with those in [10] and [11] and improve the lower bound to 1566454. A further classification of all reducible 2-(15,7,6) designs is also of interest for setting higher lower bounds on the number of Hadamard designs of greater parameters [8], [9].

2 Doubles of 2-(15,7,3) designs

Preliminaries. Consider a 2-(15,7,6) design $D = [D' \parallel D'']$. Without loss of generality we can assume that the first 15 columns of the incidence matrix $\mathcal{D}$ form a sub-matrix equal to $\mathcal{D}'$ and the next 15 columns form $\mathcal{D}''$. In this case we will write $D = D' \parallel D''$ instead of $[D' \| D'']$.

The number of doubles $H_1 \parallel \varphi H_i, i = 1, ..., 5$, is greater than $4, 7.10^{12}$. Our purpose is to construct exactly one representative of each isomorphism class. That is why it is very important to show which permutations applied to H_i lead to isomorphic designs and skip them.

The construction algorithm is based on the next simple proposition.

Proposition 1 *Let D' and D'' be two $2-(v, k, \lambda)$ designs and let α' and α'' be automorphisms of D' and D'' respectively. Then for all permutations $\varphi \in S_v$ the double designs $[D' \parallel \varphi D'']$, $[D' \parallel \varphi\alpha'' D'']$ and , $[D' \parallel \alpha'\varphi D'']$ are isomorphic.*

Proof. $\forall \alpha' \in Aut(D') \Rightarrow [D' \parallel \alpha'\varphi D''] \sim \alpha'^{-1}[D' \parallel \alpha'\varphi D''] = [D' \parallel \varphi D'']$ and $\forall \alpha'' \in Aut(D'') \Rightarrow [D' \parallel \varphi\alpha'' D''] = [D' \parallel \varphi D'']$.

Corolary 1 *If the double design $[D' \parallel \varphi D'']$ is already constructed, then all permutations in the set $Aut(D')\varphi \bigcup \varphi Aut(D'') \setminus \{\varphi\}$ can be omitted.*

We implement that with a back-track search algorithm.

Let the last considered permutation be $\varphi = (\varphi_1, \varphi_2, ..., \varphi_v)$. The next lexicographically greater than it permutation $\psi = (\psi_1, \psi_2, ..., \psi_v)$ is formed in the following way:

We look for the greatest $m \in N_{v-1} \bigcup \{0\}$, such that

- if $i \in N_m$ then $\varphi_i = \psi_i$ and $\varphi_{m+1} < \psi_{m+1}$, $\psi_{m+1} \in N_v \backslash \{\varphi_1, \varphi_2, ..., \varphi_m\}$.
- The number ψ_{m+1} is taken from the set N''_m that contains a unique representative of each of the orbits of the permutation group $Aut(D'')_{\{\varphi_1, \varphi_2, ..., \varphi_m\}}$.
- If $j \in N_m$ and $\psi_j > \psi_{m+1}$ then points P'_j and P'_{m+1} should not be in one orbit with respect to the stabilizer $Aut(D')_{\{1,2,...,j-1\}}$.

The isomorphism test is applied when a new double design D is constructed by the help of the canonical D^{max} form of its incidence matrix. The algorithm finding D^{max} gives as additional effect the full automorphism group of D.

3 Classification results

The number of nonisomorphic reducible 2-(15,7,6) designs from the five cases $H_1 \parallel H_i$, $i \in N_5$ is 1566454. Their classification with respect to the order of the automorphism groups is presented in Table 1.

A double design can have automorphisms of order 2 and automorphisms which preserve the two constituent designs (see for instance [5]). That is why among the constructed designs are all reducible 2-(15,7,6) designs with automorphisms of order 5. Their number is 6 and is the same as in [10].

Table 1: Order of the automorphism group of $H_1 \parallel H_i$, $i = 1, 2, 3, 4, 5$.

| Aut. gr. |	1	2	3	4	6	7	8	9
Designs	1 559 007	5 012	990	173	119	15	860	1
| Aut. gr. |	10	12	14	16	18	21	24	32
Designs	4	32	4	61	1	5	48	6
| Aut. gr. |	36	42	48	56	64	96	120	168
Designs	1	2	14	3	6	3	1	2
| Aut. gr. |	192	288	336	384	576	2048	2688	20160
Designs	4	1	1	4	1	1	1	1

References

[1] E. F. Jr. Assmus, J. D. Key, *Designs and their Codes*, Cambridge University Press, 1992, Cambridge Tracts in Mathematics, 103.

[2] Th. Beth, D. Jungnickel, H. Lenz, *Design Theory*, Cambridge University Press, 1993.

[3] A. Betten, M. Klin, R. Laue, C. Pech, A computer approach to the enumeration of block designs which are invariant with respect to a prescribed permutation group, Univ. Dresden, preprint MATH-AL-13-1997, 1997.

[4] I. Bouyukliev, Algorithmic approaches to the investigation of linear codes, Dr sci. dissert. Inst. Math., Sofia, 2008 (in Bulgarian).

[5] V. Fack, S. Topalova, J. Winne, R. Zlatarski, Enumeration of the doubles of the projective plane of order 4, *Discr. Math.* 306, 2006, 2141-2151.

[6] P. Kaski, P. Ostergard, *Classification Algorithms for codes and Designs*, Springer, 2006.

[7] S. Kapralov, Algorithms for generation of orbit matrices, *Sci. Conf. dedic. 100 ann. of L. Chakalov*, Bulgaria, 1986, 70-78.

[8] C. Lam, S. Lam, V. D. Tonchev, Bounds on the number of affine, symmetric and Hadamard designs and matrices, *J. Combin. Theory* Ser. A 92, 2000, 186-196.

[9] C. Lam, S. Lam, V. D. Tonchev, Bounds on the number of Hadamard designs of even order, *J. Combin. Des.* 9, 2001, 363-378.

[10] Z. Mateva, S. Topalova, Enumeration of 2-(15,7,6) designs with automorphisms of order 7 or 5,*Math. Educ. Math.* 2006, 270-274.

[11] Z. Mateva, S. Topalova, Quasidoubles of Hadamard 2-(15,7,3) designs with automorphisms of order 3,*Math. Educ. Math.*, 2007, 180-185.

[12] R. Mathon, A. Rosa, 2-(v, k, λ) designs of small order, *The CRC Handbook of Combinatorial Designs*, Boca Raton, FL., 2007, 25-57.

[13] A. Proskurovski, The maximal incidence matrix of a graph, *Technical report* 70, 1973, Royal Institute of Technology, Stokholm.

[14] V. D. Tonchev, *Combinatorial configurations*, Longman Scientific and Technical, New York, 1988.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 215-220*

Constructive algorithm of self-dual error-correcting codes

KIYOSHI NAGATA nagata@ic.daito.ac.jp
Faculty of Business Management, Daito Bunka University,
1-9-1 Takashimadaira, Itabasi-ku,Tokyo 175-8571, JAPAN

FIDEL NEMENZO fidel@math.upd.edu.ph
Department of Mathematics, University of the Philippines,
Dliman, Quezon City 1101, PHILIPPINES

HIDEO WADA wada@mm.sophia.ac.jp
Department of Science and Technology, Sophia University,
7-1 Kioi-cho, Chiyoda-ku, Tokyo, 102-0094, JAPAN

Abstract. In this paper, we consider self-dual codes over the finite ring $\mathbf{Z}_{p^s}$ of integer modulo p^s for any prime p and for an integer $s \geq 4$. We start with any self-dual code in lower modulo and give an necessary and sufficient condition for the self-duality of induced codes. Then we can give an inductive algorithm for construction of all self-dual codes and the mass formula in case of odd prime p.

1 Introduction

Since the discovery [4] of a relationship between non-linear binary codes and linear quaternary codes, there has been enormous interest in codes over the ring $\mathbf{Z}_m$ of integers modulo m and finite rings in general. We continue the ongoing investigations on the family of self-dual codes, from which many of the best known codes come from. By applying the Chinese Remainder Theorem [2] to self-dual codes over $\mathbf{Z}_m$, it suffices to classify codes over integers modulo prime powers.

We begin by giving the necessary definitions and notions. A *code* of length n over a finite ring R is a R-submodule of R^n. Elements of codes are called *codewords.* Two codewords $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_n)$ are *orthogonal* if their Euclidean inner product $\vec{x} \cdot \vec{y} = \sum_i x_i y_i$ is zero. Associated to a code $\mathcal{C}$ is a generator matrix, whose rows span $\mathcal{C}$ and the number of generators is minimal.

The *dual* $\mathcal{C}^\perp$ of a code $\mathcal{C}$ over a ring R consists of all elements of R^n which are orthogonal to every codeword in $\mathcal{C}$. A code $\mathcal{C}$ is said to be *self-dual* (resp. *self-orthogonal*) if $\mathcal{C} = \mathcal{C}^\perp$ (resp. $\mathcal{C} \subseteq \mathcal{C}^\perp$).

2 Condition for self-duality of codes over $\mathbf{Z}_{p^s}$

Every code $\mathcal{C}$ of length n over $\mathbf{Z}_{p^s}$ has a generator matrix which, after a suitable permutation of coordinates, can be written as

$$\mathcal{C} = \begin{bmatrix} T_1 \\ pT_2 \\ p^2T_3 \\ \vdots \\ p^{s-1}T_s \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11} & A_{12} & \dots & A_{1s-1} & A_{1s} \\ 0 & pI_{k_2} & pA_{22} & \dots & pA_{2s-1} & pA_{2s} \\ 0 & 0 & p^2I_{k_3} & \dots & p^2A_{3s-1} & p^2A_{3s} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & p^{s-1}I_{k_s} & p^{s-1}A_{ss} \end{bmatrix},$$

where I_{k_i} is the $k_i \times k_i$ identity matrix, and the other matrices A_{ij}'s $(1 \leq i \leq j \leq s)$ are considered modulo p^{j-i+1}. When we denote the inverse matrix of $[T_i]_{i=1,\dots,s+1}$ with an additional $T_{s+1} = (0\ 0\ \dots\ 0\ I_{k_{s+1}})$ by $[T_i^*]_{i=1,\dots,s+1}^t$, we have $\mathcal{C}^\perp = \left[p^{i-1}T_{s+2-i}^*\right]_{i=1,\dots,s}^t$. Thus we see that a necessary and sufficient condition for the self-duality of $\mathcal{C}$ is $k_1 = k_{s+1}, \dots, k_i = k_{s-i+2}, \dots$ and $\mathcal{C}$ is self-orthogonal. And we have following proposition and lemma.

Proposition 1 *Let $\mathcal{C} = \left[p^{i-1}T_i\right]_{i=1,\dots,s}$ be a code over $\mathbf{Z}_{p^s}$ with $T_i = (0 \dots 0\ I_{k_i}\ A_{ii} \dots A_{is})$. Then $\mathcal{C}$ is a self-dual code if and only if $k_i = k_{s-i+2}$ for $i = 1, \dots, s+1$ and the following holds:*

$$T_iT_j^t \equiv 0 \qquad \mod p^{s-i-j+2}, \tag{1}$$

for any integers i and j such that $1 \leq i \leq j \leq s$ and $i + j \leq s+1$.

Lemma 1 *When the condition (1) in Proposition 2.1 holds, the rank of $k_i \times (k_1 + \dots + k_i)$ matrix $(A_{is+1-i}A_{is+2-i} \cdots A_{is})$ $(1 \leq i < \frac{s+1}{2})$ is equal to k_i. Especially when $i = 1$, we have that A_{1s} is invertible.*

Proof. We rewrite the condition (1) using A's, and we have two modulo p conditions $I_{k_i} \equiv -\sum_{l=i}^{s} A_{il}A_{il}^t$ and $A_{ij-1} \equiv -\sum_{l=j}^{s} A_{il}A_{jl}^t$ for $i < \frac{s+1}{2}$. By recursive substitution, we have $A_{ij-1} \equiv \sum_{l=s+1-i}^{s} A_{il}A_{jl}'^t$ $(\exists A_{jl}')$, and $I_{k_i} \equiv (A_{is+1-i} \cdots A_{is})C_i^t$ $(\exists C_i)$. This completes the proof. □

3 Codes over $\mathbf{Z}_{p^s}$ from a code over $\mathbf{Z}_{p^{s-2}}$

Now we consider the code $\mathcal{C}'$ of length $n = k_1 + k_2 + \cdots + k_{s+1}$ over $\mathbf{Z}_{p^{s-2}}$ reduced from a self-dual code $\mathcal{C}$ as

$$\mathcal{C}' = \begin{bmatrix} T_1' \\ T_2' \\ pT_3' \\ \vdots \\ p^{s-3}T_{s-1}' \end{bmatrix} = \begin{bmatrix} I_{k_1} & A_{11} & A_{12}' & \cdots & A_{1s-2}' & A_{1s-1}' & A_{1s}' \\ 0 & I_{k_2} & A_{22} & \cdots & A_{2s-2} & A_{2s-1} & A_{2s}' \\ 0 & 0 & pI_{k_3} & \cdots & pA_{3s-2} & pA_{3s-1} & pA_{3s}' \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p^{s-3}I_{k_{s-1}} & p^{s-3}A_{s-1s-1} & p^{s-3}A_{s-1s}' \end{bmatrix},$$

To see the self-duality condition, we substitute $s-2$ for s, $i-2$ and $j-2$ for $(3 \leq)i$ and j respectively, and $\begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ for T_1 in (1). Then we have $T_1T_1^t \equiv 0 \mod p^{s-2}$, $T_1T_j^t \equiv 0 \mod p^{s-j}$, and $T_iT_j^t \equiv 0 \mod p^{s-i-j+2}$ $(2 \leq i \leq j \leq s-1)$. The third conditions are the completely same as that for $\mathcal{C}$, and the first and the second conditions hold as the corresponding equations for $\mathcal{C}$ are $T_1T_j^t \equiv 0 \pmod{p^{s-j+1}}$ for any $j \leq s$. Since we also have $k_1 + k_2 = k_{s+1} + k_s$, $k_3 = k_{s-1}, \ldots$, we see that $\mathcal{C}'$ is again self-dual.

Conversely we start with a self-dual code $\mathcal{C}'$ of length $n = k_1' + k_3 + \ldots + k_{s-1} + k_{s+1}'$ over $\mathbf{Z}_{p^{s-2}}$. At first, we divide the part of generation vectors modulo p of dimension k_1' into two parts of dimension k_1 and k_2, and we also divide last $k_{s+1}'(= k_s + k_{s+1} = k_1 + k_2)$ columns into two parts, like as described above. We should notice that different matrix at (1,2)-entry might induce a different code over $\mathbf{Z}_{p^s}$.

Before starting the construction of $\mathcal{C}$ over $\mathbf{Z}_{p^s}$, we need an important permutation operation. From Lemma 1, we see that $(k_s + k_{s+1})$-size square matrix $\begin{pmatrix} A_{1s-1}' & A_{1s}' \\ A_{2s-1} & A_{2s}' \end{pmatrix}$ (mod p) is invertible. So by some column permutation, we can suppose that A_{1s}' (mod p) is invertible. Moreover we need to make some kind of modification by adding $k_1 \times k_i$ matrix times T_i' to T_1' since A_{1i}s' are to be considered in $\mod p^i$ not in $\mod p^{i-1}$ in $\mathcal{C}$.

Now we denote the resulted matrices by A_{1i} $(i = 2, \ldots, s-2)$, and for such a given self-dual code $\mathcal{C}'$ over $\mathbf{Z}_{p^{s-2}}$ in the above form, we will construct the code $\mathcal{C}$ by multiplying p to $p^{i-2}T_i$ $(i = 2, \ldots, s-1)$ and adding a new $p^{s-1}T_s$ in the bottom. All A_{ij}'s except for A_{is} $(i = 1, \ldots, s-1)$ and A_{1s-1} are considered in the same modulo as in $\mathcal{C}'$. For any i, A_{is} is defined modulo p^{s-i+1} and A_{1s-1} is modulo p^{s-1}. Since A_{is}' is defined modulo p^{s-i} for $i \geq 2$, A_{1s}' is modulo p^{s-2}, and A_{1s-1}' is modulo p^{s-2} in $\mathcal{C}'$, we need following extension

$$T_1 = T_1' + p^{s-2}U_1 + p^{s-1}V \quad \text{and} \quad T_i = T_i' + p^{s-i}U_i \; (2 \leq i \leq s-1) \qquad (2)$$

where $U_1 = (0 \ \ldots \ A_{1s-1}^{(1)} \ A_{1s}^{(1)})$, $V = (0 \ \ldots \ 0 \ A_{1s}^{(2)})$, $U_i = (0 \ \ldots \ 0 \ A_{is}^{(1)})$ $(i = 2, \ldots, s-1)$ for some modulo p matrices $A^{(1)}$'s and $A^{(2)}$.

We have remaining two types of conditions in (1) for the self-duality of $\mathcal{C}'$. One is $T_1T_i^t \equiv 0 \mod p^{s-i+1}$ under the codition $T_1'T_i'^t \equiv 0 \mod p^{s-i}$ for $2 \leq i$, which becomes $T_1'T_i'^t + p^{s-2}U_1T_i'^t + p^{s-i}T_1'U_i^t \equiv 0 \mod p^{s-i+1}$ by substituting the right-hand sides of (2) and taking the assumption that $4 \leq s$ and $2s-i-2 = s-i+1+(s-3)$ in mind. If $3 \leq i$, then $s-i+1 \leq s-2$ and the equation is $T_1'T_i'^t + p^{s-i}A_{1s}A_{is}^{(1)t} \equiv 0 \mod p^{s-i+1}$, and we have $A_{is}^{(1)t}$ $(3 \leq i)$ is uniquely determined as $A_{is}^{(1)t} \equiv -A_{1s}^{-1}\left(\frac{1}{p^{s-i}}T_1'T_i'^t\right) \mod p$. If $i = s$, then $A_{ss}^t \equiv -A_{1s}^{-1}A_{1s-1} \mod p$. If $i=2$, then $T_1'T_2'^t + p^{s-2}(A_{1s-1}^{(1)}A_{2s-1}^t + A_{1s}^{(1)}A_{2s}^t + A_{1s}A_{2s}^{(1)t}) \equiv 0 \mod p^{s-1}$. Thus we have that $A_{2s}^{(1)t}$ is also uniquely determined as $A_{2s}^{(1)t} \equiv -A_{1s}^{-1}\left(\frac{1}{p^{s-2}}T_1'T_2'^t + A_{1s-1}^{(1)}A_{2s-1}^t + A_{1s}^{(1)}A_{2s}^t\right) \mod p$ for any $A_{1s-1}^{(1)}$ and $A_{1s}^{(1)}$. The other condition can be rewritten as $0 \equiv T_1'T_1'^t + p^{s-2}\widetilde{T_1'U_1^t} + p^{s-1}\widetilde{T_1'V^t}$ mod p^s, with $\widetilde{X} = X + X^t$. This includes the condition for $A_{1s-1}^{(1)}$ and $A_{1s}^{(1)}$, and using them we have following essential condition

$$T_1'T_1'^t + p^{s-2}(\widetilde{A_{1s-1}A_{1s-1}^{(1)t}} + \widetilde{A_{1s}A_{1s}^{(1)t}}) + p^{s-1}\widetilde{A_{1s}A_{1s}^{(2)t}} \equiv 0 \qquad \mod p^s. \qquad (3)$$

From now on, we consider the equation above only in odd p case. $\widetilde{A_{1s}A_{1s}^{(1)t}} \equiv -\left(\frac{1}{p^{s-2}}T_1'T_1'^t + \widetilde{A_{1s-1}A_{1s-1}^{(1)t}}\right) \mod p$ is given by reducing (3) modulo p^{s-1}. We put $(x_{ij}) = A_{1s}A_{1s}^{(1)t}$ and put (d_{ij}) the right-hand side of the equation for any $k_1 \times k_2$ matrix $A_{1s-1}^{(1)}$. Then the necessary and sufficient condition for x_{ij} are $x_{ji} = d_{ij} - x_{ij} \mod p$ $(i < j)$, and $x_{ii} = \frac{1}{2}d_{ii}$. For any $p^{\frac{1}{2}k_1(k_1-1)}$ number of (x_{ij}) satisfying above, $A_{1s}^{(1)t}$ is uniquely detemined by $A_{1s}^{-1}(x_{ij})$ (mod p). Once $A_{1s}^{(1)t}$ is determined, the condition (3) is just equivalent to $\widetilde{A_{1s}A_{1s}^{(2)t}} \equiv -\frac{1}{p}\left(\frac{1}{p^{s-2}}T_1'T_1'^t + \widetilde{A_{1s-1}A_{1s-1}^{(1)t}} + \widetilde{A_{1s}A_{1s}^{(1)t}}\right) \mod p$. We also put $(y_{ij}) = A_{1s}A_{1s}^{(2)t}$ and put (f_{ij}) the right-hand side of the equation. Then the necessary and sufficient condition for y_{ij} are $y_{ji} = f_{ij} - y_{ij} \mod p$ $(i < j)$, and $y_{ii} = \frac{1}{2}f_{ii}$. For any $p^{\frac{1}{2}k_1(k_1-1)}$ number of (y_{ij}) satisfying this, $A_{1s}^{(2)t}$ is uniquely detemined by $A_{1s}^{-1}(y_{ij})$ (mod p). Thus we have self-dual codes over $\mathbf{Z}_{p^s}$ and the following lemma.

Lemma 2 *The number of self-dual codes over $\mathbf{Z}_{p^s}$ of type $(k_1, k_2, \ldots, k_{s+1})$ induced from a self-dual code over $\mathbf{Z}_{p^{s-2}}$ of type $(k_1+k_2, k_3, \ldots, k_s+k_{s+1})$ is*

$$\rho(k_1+k_2, k_1) \times p^{k_1\sum_{i=3}^{s-1}k_i} \times p^{k_1k_2+k_1(k_1-1)} = \rho(k_1+k_2, k_1)p^{k_1(n-k_1-k_2-1)},$$

where $\rho(n,k) = \prod_{j=1}^{k}(p^n - p^{j-1})/\prod_{j=1}^{k}(p^k - p^{j-1})$, the number of subspace of dimension k of a vector space over $\mathbf{F}_p = \mathbf{Z}_p$ of dimension n.

Proof. The number of possible partitions $\left[\begin{array}{c} T_1' \\ T_2' \end{array} \right]$ in $\mathcal{C}'$ is given by considering the map $\mathcal{C}' \xrightarrow{p^{s-1}} p^{s-1}\mathcal{C}' \to 0$. The kernel is $< pT_1', pT_2', pT_3', ..., p^{s-3}T_{s-1}' >$ and noticing that the submodule is to be considered in $< T_1', pT_2', p^2T_3', ..., p^{s-1}T_s' >$, we should count the multiple of the number of partitions in the vector space $p^{s-1}\mathcal{C}'$ and a kind of modifications of T_1' by $T_3',..., T_{s-1}'$. The number of partitions in the vector space is just $\rho(k_1 + k_2, k_1)$ from the lemma3.2 in [5]. As the modifications are done by adding any $k_1 \times k_i$ matrix times T_i' to T_1', the number of such modification is just equal to $p^{k_1 \times k_3} \times p^{k_1 \times k_4} \times \cdots \times p^{k_1 \times k_{s-1}} = p^{k_1 \sum_{i=3}^{s-1} k_i} = p^{k_1(n-2(k_1+k_2))}$. □

When we calculate the product of $\rho(n_i, k_i)$, we have following lemma.

Lemma 3

$$\prod_{i=1}^{m} \rho(n_i, k_i) = \frac{\prod_{i=1}^{n_m}(p^i - 1)}{\prod_{i=1}^{m}\prod_{j=1}^{k_i}(p^j - 1)}, \quad \textit{with } n_i = k_1 + \cdots + k_i \quad (i = 1, ..., m).$$

Now we have the following formulae.

Theorem 1 *Let $N_{p^s}(n; k_1, ..., k_{s+1})$ be the number of self-dual codes over $\mathbf{Z}_{p^s}$ of type $(k_1, ..., k_{s+1})$ for an odd prime p and for an integer s $(1 < s)$. And put $n_i = k_1 + \cdots + k_i$ for $i = 1, ..., [\frac{s+1}{2}]$, and put $m_u = \sum_{i=1}^{u} n_i(n - n_{i+1} - 1)$.*

1. *If $s(= 2u)$ even, then*

$$N_{p^s}(n; k_1, ..., k_{s+1}) = D_{n,n_u} \frac{\prod_{i=1}^{n_u-1}(p^{n-2i-\delta} - 1)}{\prod_{i=1}^{u}\prod_{j=1}^{k_i}(p^j - 1)} \cdot p^{m_u - \frac{1}{2}n_u(n_u-1)},$$

   where $D_{n,n_u} = \left(p^{\frac{n}{2}-n_u} + \left(\frac{-1}{p}\right)^{\frac{n}{2}}\right)\left(p^{\frac{n}{2}} - \left(\frac{-1}{p}\right)^{\frac{n}{2}}\right)$ and $\delta = 0$ if n is even, and $D_{n,n_u} = \delta = 1$ if n is odd.

2. *If $s(= 2u + 1)$ odd, then n must be even and*

$$N_{p^s}(n; k_1, ..., k_{s+1}) = \left(1 + \left(\frac{-1}{p}\right)^{\frac{n}{2}}\right) \prod_{i=1}^{\frac{n}{2}-1}(p^i + 1) \frac{\prod_{i=0}^{n_u-1}(p^{n-i} - 1)}{\prod_{i=1}^{u}\prod_{j=1}^{k_i}(p^j - 1)} \cdot p^{m_u}.$$

Proof. ¿From the lemma 2 and the lemma 3,

$$\begin{aligned} & N_{p^s}(n; k_1, ..., k_{s+1}) = N_{p^{s-2}}(n; n_2, k_3, ..., k_{s-1}, n_2)\rho(n_2, k_2)p^{n_1(n-n_2-1)} \\ = & \left\{ \begin{array}{ll} N_{p^2}(n; n_u, k_{u+1}, n_u) \dfrac{\prod_{i=1}^{n_u}(p^i - 1)}{\prod_{i=1}^{u}\prod_{j=1}^{k_i}(p^j - 1)} \cdot p^{m_{u-1}} & \text{(if s is even)} \\ N_{p^3}(n; n_u, k_{u+1}, k_{u+1}, n_u) \dfrac{\prod_{i=1}^{n_u}(p^i - 1)}{\prod_{i=1}^{u}\prod_{j=1}^{k_i}(p^j - 1)} \cdot p^{m_{u-1}} & \text{(if s is odd)} \end{array} \right. . \end{aligned}$$

If s is even, then from Theorem 3.5 in [1]

$$N_{p^2}(n; n_u, k_{u+1}, n_u) = D_{n,n_u} \frac{\prod_{i=1}^{n_u-1}(p^{n-2i-\delta}-1)}{\prod_{i=1}^{n_u}(p^i-1)} p^{\frac{1}{2}n_u(n_u-1)},$$

and we have the resulted formula. If s is odd, then from Theorem 4.1 in [5]

$$\begin{aligned} & N_{p^3}(n; n_u, k_{u+1}, k_{u+1}, n_u) \\ = & \left(1+\left(\frac{-1}{p}\right)^{\frac{n}{2}}\right)\prod_{i=1}^{\frac{n}{2}-1}(p^{\frac{n}{2}-i}+1)\frac{\prod_{i=0}^{n_u-1}(p^{n-i}-1)}{\prod_{i=1}^{n_u}(p^i-1)}\cdot p^{n_u(n-n_{u+1}-1)}, \end{aligned}$$

and we have the resulted formula. □

4 Conclusions

We suceeded to give a formula for the number of self-dual codes of a given type for an odd prime p and for any integers $s \geq 4$. In order to obtain the mass formula for the self-dual codes of length n, we have only to add up the formulae in theorem 1. Since we already have the mass formula for each $\mathbf{Z}_p$, $\mathbf{Z}_{p^2}$, and $\mathbf{Z}_{p^3}$ [1, 3, 5], the mass formula problem for any odd prime is completely solved.

In case of $p = 2$, Gaborit [3] had the two types of mass formula for the doubly even binary code and for type II quarternary code. Our construction algorithm is similarly applied to this case, but somehow complicated because we need douby even property. We are now under investigating the mass formula for codes over $\mathbf{Z}_{2^s}$.

References

[1] Balmaceda, J., Betty, R., and Nemenzo, F. Mass formula for self-dual codes over $\mathbf{Z}_{p^2}$, *Discrete Mathematics* (to appear).

[2] Dougherty, S., Harada, H., and Solé, P. Self-dual codes over rings and the Chinese remainder theorem, *Hokkaido Math. J.* 28, 1999, 253-283.

[3] Gaborit, P. Mass formulas for self-dual codes over $\mathbf{Z}_4$ and $\mathbf{F}_q + u\mathbf{F}_q$ rings, *IEEE Trans. Inform. Theory* 42, 1996, 1222-1228.

[4] Hammons, A., Kumar, P., Calderbank, A., Sloane, N., and Solé, P. The $\mathbf{Z}_4$ linearity of Kerdock, Preparata, Goethals and related codes, *IEEE Trans. Inform. Theory* 40, 1994, 301-319.

[5] Nagata, K., Nemenzo, F., and Wada, H. The number of self-dual codes over $\mathbf{Z}_{p^3}$, (under review)

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 221-228*

Procedures of extending the alphabet for the PPM algorithm

Radu Radescu rradescu@gmail.com
George Liculescu
Polytechnic University of Bucharest, ROMANIA

Abstract. In this paper it is presented the lossless PPM (Prediction by Partial string Matching) algorithm and it is studied the way the alphabet can be extended for the PPM encoding so it will allow the use of symbols which are not present in the alphabet at the beginning of the encoding phase. The extended alphabet can contain symbols with the size larger than a byte. The paper presents the manner to extend the alphabet and the changes that need to be made to the PPM algorithm in order to use the extended alphabet. Three ways to extend the alphabet are proposed: manual, through a run over the text (executed before the encoding phase), and specialized (adapted during the evolution of the algorithm).

1 Introduction

Let us presume that a file contains a string of bytes (characters), which appears many times in the file. PPM must encode independently every byte from the string with a probability (which is preferable to have large value). Every time the character was not found in the past (the string preceding the current context), an escape symbol is send to decrease the level, leading to increment the information from the compressed data flow. The alphabet used by the PPM algorithm has 256 characters (all the characters that can be formed using 8 bits). If the regular alphabet is extended adding a new symbol (the string mentioned above) the algorithm could perform a good compression. An extended alphabet is an enriched known alphabet with a series of symbols that will not be presented in the alphabet offered to the decoder. The symbols that extend the alphabet need to be obtained in the decoding phase through different methods, so while decoding the alphabet will be enriched with new symbols. In the coding phase, the symbols that will extend the alphabet are known, but at the decoding these will be deduced gradually. Three solutions to extend the alphabet are considered:

1. manual adding of the words by the user;
2. search of the words that get repeated using a certain criterion (length, number of appearances, etc.) in a first step by running through the entire text and then adding these words to the alphabet;
3. adaptive search of the text words during the algorithm evolution.

In this paper, we consider that internal words are present at the decoding phase because they are internally generated, and they can be reproduced at the decoding.

2 The manual adding of text words

From the three options, this one is the simplest because it is based on the user experience and has no need for additional processing. For adding of a text word from the manual point of view, it is necessary to define all the parameters involved: length, number of appearances, presence at the decoding, and gain. The length of the word to be added is known. The number of appearances and the presence at the decoding must be set up manually. The set up number of appearances and the length will determine the gain of the text word. In this way, we can set manually the significance of the word. Implicitly, the manual added words would be marked as external.

3 The search of words by running through the text

The suggested solution in this paper is based on the suffix vector. This contains all the suffixes from a string, lexicographically ordered. For example, if we consider the abracadabra string, then the suffixes of this string will be (see Figure 1):

Suffix	Position
abracadabra	0
bracadabra	1
racadabra	2
acadabra	3
cadabra	4
adabra	5
dabra	6
abra	7
bra	8
ra	9
a	10

sorting ⟹

Suffix	Position
a	10
abra	7
abracadabra	0
acadabra	3
adabra	5
bra	8
bracadabra	1
cadabra	4
dabra	6
ra	9
racadabra	2

Fig. 1. The suffix vector.

To obtain a suffix vector we need to extract the suffixes and sort them. It can be noticed that in the suffix vector the side elements can have identical characters. We aim only the strings that start with the first character from the left of the suffix and are continued through the right. Based on this we can determine strings that are repeated in the text. These characters can make up words, which can be used to extend the alphabet used at the encoding In the suffix vector, we can find many words that are repeated but only few of them will be of some interest. To find out which words are significant we will need to induce some restrictions. To do this we will attach to every word a gain with which we will determine its significance. The restrictions will be related to the minimum length, the maximum length, and the minimum gain.

If a match does not have at least the minimum length then it is treated like it did not exist. In order to calculate the gain we need to know the length and the number of appearances. We know the length of the word but finding the number of appearances is a problem. To solve this problem we need to have a view only of words that do not overlap. The words that overlap cannot be compared because an exact delimitation does not appear so we cannot say that both of them exist at the same time in the compression string, because only one word can be coded. Knowing the text from where the suffixes are extracted, it is enough to keep a position vector from where the suffixes begin and a length vector, which will indicate the accepted lengths for every suffix, which begins at that position. When we want to sort the suffixes, we compare the suffixes that start at the specified positions in the position vector. The sorting of the suffixes will result in an arrangement of the positions in the position vector and the of length in the length vector (see Figure 2). The length vector, as it will be shown next, will help to solve the overlapping of words extracted from the text. The length of a suffix can be at most the maximum set up length. To keep track of the found words we will create a list, where every record will be made up of the positions where the word is found and the length of the word. The number of positions on which the word is situated will give the number of appearances.

Position	Length
0	l_0
1	l_1
2	l_2
3	l_3
4	l_4
5	l_5
6	l_6
7	l_7
8	l_8
9	l_9
10	l_{10}

sorting $\Longrightarrow$

Position	Length
10	l_{10}
7	l_7
0	l_0
3	l_3
5	l_5
8	l_8
1	l_1
4	l_4
6	l_6
9	l_9
2	l_2

Fig. 2. The sorting of suffixes.

A record from the list will look like this (see Figure 3):

Position 1	Position 2		Position n	Length

.

Fig. 3. A recording from the word list.

Example. We will run through the suffix vector and we will compare, in this order, pairs of suffixes: a with *abra*, *abra* with *abracadabra*, $\cdots$, *dabra* with ra, and ra with *racadabra*. For every pair of suffixes, we will find identical characters or not. Every time we find a word, this will be added in the list by creating a new record (in the case the word does not exist in the list) or by adding the position where it was found. We can see that *a* and *abra* have only the a character in common, and *abra* and *abracadabra* have the *abra* string in common. We can say that *a* is found in the *a* & *abra* suffixes, and in *abracadabra.* So the a recording will have three positions. *abra* is found in *abra* and in *abracadabra* so the *abra* recording will have two positions. We can see that the positions from *abra* can be found in *a*, because *a* is part of *abra*. This means the shorter strings from the word list that are part of other longer strings must have at least the positions of the longer strings.

Remark. If two suffixes do not have a character in common, then from that point there will be no suffixes having characters in common with the suffixes before that point. This means that portions of the suffix vector can be treated separately. For example, the suffixes from below do not have any character in common with the rest of the suffixes (see Figure 4).

Suffix	Position
a	10
abra	7
abracadabra	0
acadabra	3
adabra	5

Fig. 4. Suffixes that have at least one character in common.

Because we mentioned above that shorter strings that are part of longer strings must have at least the positions of the longer strings, we need to find a practical way to accomplish this. We can see that if between two suffixes on the p1 and p2 positions there is a n length match, then we need to update all the recordings smaller or equal to n and that are involved in that match. In order to reduce the number of steps, every time a match is found, and is smaller than the last record from the list, all the element from the list smaller or equal to n will be moved, and we will store the pmoved position in the list in which the element has been moved. This means that pmoved will be the end of the list before the move. From the pmoved position to the end of the list, the p2 position will be added in every record. If the n length match does not exist in the list, it will be added.

Example. In Figure 4, a and abra have in common a so a will be added in the list together with positions 10 and 7 because it is a new word and the list is empty. abra and abracadabra have abra in common so it will be added the 0 position to the previous recordings, after that abra is added in the list with 7 and 0 positions because it is a new word. This means that until now the list contains a and abra. abracadabra and acadabra have in common only a. Because a has length 1, it is smaller than the last recording (abra) so any word with length smaller or equal to 1 will be moved to the end of the list. As a result, the list that contained the recordings (in this order) a, abra now will have the recordings abra, a. This means that the last record will be added to the position where the new a was found, and this is 3 (see Figure 5).

a	10, 7

⟹

a	10, 7, 0
abra	7,0

⟹

abra	7, 0
a	10, 7, 0, 3

Fig. 5. Suffixes that have at least one character in common.

Because we need all the words from the list that have at least one character in common with analyzed suffixes, we cannot solve the overlapping between the words until there will be no suffixes that would have a common character with the ones in the list. This way, we will solve the overlapping each time there are no characters in common between the two currently analyzed words. The solving of the overlapping will be handled only inside of a recording from the list, because it is very demanding the check the positions of every recording through the comparison with the other recordings. In addition, keeping of some positions must be settled according to the gain. If there is an overlapping of the recordings, we cannot calculate the gain. This means that the adopted way will be to search the word that has a maximum gain and its positions do not overlap one another but can overlap with other words. After finding this word, we will mark the positions from the input text where the word is found as being occupied. After this, we repeat the search algorithm for a new word. In this manner, the founded words will not overlap. This procedure will repeat until no word can be found under these conditions.'

4 Adaptive search of the words

The first two methods can be used along with any compression algorithms. Only the PPM algorithm uses the method described below. To form the words, we will use the tree, with the help of which the contexts are maintained. The tree is changed every time we wish to insert a word. The tree has all the past contexts, if it has not been emptied to save memory, or only a part of them (from a near past). In the case the inserted word has been preceded by the same context in the past, the number of appearances is incremented and added to the actualization list. In the case the PPM context has been spotted as being followed by the algorithm word 5 times, we can say that in the past PPM algorithm has been seen 5 times. This means a word discovery can be made similar to the run through method previously described. Therefore if a minimum length, a maximum length and a minimum gain are set, some words created with the tree can be considered.

When a word, which is added in the three, has been seen preceded by a certain context, the tree is run through, starting from that word preceded by the context to the root. This means we begin at the node that has the certain word and we move to his parent, then his grandparent, and so on until we reach the root. Every time a move is made, the word from the current node is inserted in a stack and a *total_length* variable is incremented that holds the length of the word added in the stack. If the value of *total_length* is at least the minimum set length then the word preceded by that context is considered for further checking. In the case the gain calculated based on the minimum of *total_length* and the maximum set length and based on the number of appearances of the word preceded by the context is at least the minimum set gain, then the word preceded by the context is considered for further checking. Next, the words are extracted one by one from the stack and are concatenated until the stack is empty or until the maximum length is reached. This way, a word is constructed that may of may not be added to the alphabet. If the word is not in the alphabet, then it is inserted. The added word will have a number of appearances equal to zero and will be marked as being intern so it will be present at decoding.

Normally we are tempted to be less restrictive with the limits imposed on making of a new word, for it to exist in the alphabet. A word that already exists in the alphabet has the chance to be used as soon as possible. If the limits are too restrictive, the process of using a word will be much delayed. The setting of less restrictive rules will produce the negative effect of congesting the alphabet because numerous words are in the set limits. Because of this, we must have a compromise. Because the less restrictive rules allow the words to be used soon, the problem of the large number of generated words must be solved. The solution is the periodical cleaning of the alphabet (after a number of bytes). This way, we will search words that have not been used and are marked as being intern. The words are unused if the real appearances number (set from inside) is zero. If a word has been used at least one time in the encoding stage then it remains in the alphabet, and can no longer be eliminated. This may lead to the growth of the alphabet. This is why a maximum admitted memory would be set for the storing of words. If this memory is exceeded, then the extern words are kept and the intern ones are sorted decreasingly by the gain. Another memory limit will be set for the reducing of the alphabet. The structure is presented in Figure 6.

Example. In Figure 6, we considered that the last added word, a, had the length 1. Therefore, the actualization list will be formed from the nodes with thick lines. We consider the restrictions: *minimum_length* = *3*, *maximum_length* = *6*, and *minimum_gain* = *6*, the gain being computed using the formula *length* × *length* × *appearances.* We can form 3 words at most because there are 3 thickened nodes besides the root, which cannot participate at the forming of the word because they do not contain a word. The 3 words will be:

1) The first node that is checked is the thickened one from the level 3. We insert in a stack all the words from the path that starts at the current analyzed node and ends at root. In order to do this, we use pointers that indicate the parent of every node. In the stack, the words will be placed in the following order: a, a, merge. The total length is 6. We can see that the *total_length*, which is 7, is larger than the minimum length, which is 3. *mergeaa* was seen 7 times.

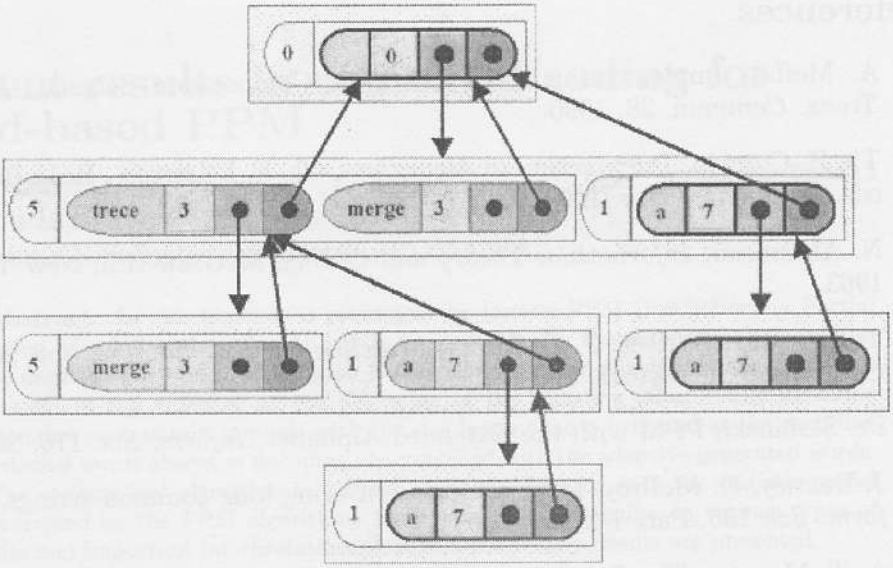

Fig. 6. The adding of the pointer that indicates the parent.

As a result, the gain will be *length* × *length* × *appearances= 6×6×7=252* , because the minimum between the total length and the maximum length is 6. We can observe that this gain is bigger than the minimum gain, which is 6. This means this word has all the characteristics to be used to extend the alphabet. Next, we extract the elements from the stack and we concatenate them until we reach the maximum length or the stack is left with no elements. The result string will be: *mergea* (was going in Romanian) and not *mergeaa*, because the maximum length is 6 (see Figure 7). The word is checked if it already exists in the alphabet. If it is not present, then it is added.

merge
a
a

⟹

merge	a	a

Fig. 7. The creation of a text word based on the information from the tree.

2) The second node that will be checked is the thickened one from level 2. We insert the words in the stack in the following order: a, a. The total length is 2 and it is smaller than the minimum set length, which is 3. This means we cannot form a word because it does not match the set limits.

3) The third node that will be checked is the thickened one from level 1. This time the only word inserted in the stack is a. Because the total length is 1, we will not be able to create a word with the use of this node, because the minimum length is 3. It is not possible to create a new word with only a node from the first level, even if this matches the set limits, because the word from a single node is certainly in the alphabet.

References

[1] A. Moffat, Implementing the PPM Data Compression Scheme, *IEEE Trans. Commun.* 38, 1990.

[2] Th. H. Cormen, *Introduction to Algorithms*, Ch. E. Leiserson , R. L. Rivest eds., The MIT Press. 1999.

[3] N. Abramson, *Information Theory and Coding*, McGraw-Hill, New York, 1963.

[4] T. Bell , I. H. Witten, J. G. Cleary, *Modeling for Text Compression*, ACM Computing Surveys 21, 1989.

[5] Pr. Skibinski, PPM with the Extended Alphabet, *Inform. Sci.* 176, 2006.

[6] J. Bentley, D. McIlroy, Data compression using long common strings, *Inform. Sci.* 135, Part 1-2, June 2001.

[7] A. T. Murgan, *The Principles of Information Theory in Information and Communication Engineering*, Romanian Academy Press, Bucharest, 1998.

[8] R. Radescu, *Lossless Compression - Methods and Applications*, Matrix Rom Press, Bucharest, 2003.

[9] M. Nelson, *The Data Compression Book*, 2nd Edition, Jean-Loup Gailly ed., M&T Books, 1995.

[10] *** *Data Compression - The Complete Reference*, 3rd Edition, David Salomon ed., Springer-Verlag, 2004.

[11] *** *Lossless Compression Handbook*, 1st Edition, Khalid Sayood ed., Academic Press, 2002.

[12] R. Radescu, C. Harbatovschi, Compression methods using prediction by partial matching, *Proc. 6th Intern. Conf. Commun.*, Bucharest, Romania, 2006, 65-68.

[13] R. Radescu, R. Popa, On the performances of symbol ranking text compression method, *Sci. Bull. "Politehnica" Univ. Timisoara, Romania, Trans. Electr. Commun., special issue dedic. Electr. Telecomm. Symp.* 49, ETC 2004, 25-27.

[14] The Calgary Corpus:
ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus

[15] www.winrar.com

Recent results in combined coding for word-based PPM

Radu Radescu rradescu@gmail.com
George Liculescu
Polytechnic University of Bucharest, ROMANIA

Abstract. In this paper it is presented the lossless PPM (Prediction by Partial string Matching) algorithm and it is studied the way the extended alphabet can be used for the PPM encoding so it will allow the use of symbols which are not present in the alphabet at the beginning of the encoding phase. The extended alphabet can contain symbols with the size larger than a byte and at the decoding external words absent at decoding are combined with the adaptive-generated words. The arithmetical algorithm is used to encoding of words with the statistic model generated by the PPM algorithm. Some experimental results on various types of files and important interpretations deducted from these results are presented.

1 Introduction

Let us presume that a file contains a string of bytes (characters), which appears many times in the file. PPM must encode independently every byte from the string with a probability (which is preferable to have large value). Every time the character was not found in the past (the string preceding the current context), an escape symbol is send to decrease the level, leading to increment the information from the compressed data stream. The alphabet used by the PPM algorithm has 256 characters (all the characters that can be formed on 8 bits). If the regular alphabet is extended adding a new symbol (the string mentioned above) the algorithm could perform a good compression.

An extended alphabet is an enriched known alphabet with a series of symbols that will not be presented in the alphabet offered to the decoder. The symbols that extend the alphabet need to be obtained in the decoding phase through different methods, so while decoding the alphabet will be enriched with new symbols. In the coding phase, the symbols that will extend the alphabet are known, but at the decoding these will be deduced gradually.

In this paper, we consider that internal words are present at the decoding phase because they are internally generated, and they can be reproduced at the decoding. The external words that could be present at the decoding are inserted externally at both coding and decoding stages. It is considered an optimization of the data tree, so it can be used on the purpose of word-based coding (strings of octets).

In order to minimize the searching time, an optimized algorithm must be used. The red-black tree is used for searching. The red-black tree is a binary tree, which keeps inside of every node an extra-information - the color of the node - that can be red or black. Through the constrain of the way the nodes can be colored with every line that starts at the base and ends in a leaf, the red-black tree ensures there is no other way which is longer than the other keeping the tree approximately balanced. The procedures that can be performed are that of a classic binary tree. The method used here is dedicated only to the PPM algorithm and it performs the adaptive search of the words.

2 PPM encoding with the extended alphabet

The extended alphabet encoding is similar to the basic alphabet one (made up of every 8 bits symbols). In order to determine which word is next coded, we need to check all the words, which can be made up based on the text bytes, starting from the current position from the considered coding stage. The length of words that can be formed with the text bytes must be smaller or equal to the maximum admitted length and smaller or equal to the maximum length present in the source alphabet. Every formed word is searched in the alphabet and if it is found, a gain is associated to it. In order to compute the gain, we must have information regarding the current state of every word from the alphabet (number of appearances, length, etc.).

The gain can be calculated in many ways, but here it will be calculated as a function of number of appearances and length. Usually, a formula for computing the gain should be used, and this would depend on the context where the word is situated.

In order to know the value of the word from the context point of view, a search in the tree must be performed. This search must be made for every word that has a chosen potential, this being a very big extra task. For this reason, in order to compute the gain, we will use the real number of appearances (imposed on inside basis), in the case of internal words, or the maximum between the real and false number of appearances (imposed on outside basis), for external words. The number obtained from the zero level node gives the real number of appearances. Therefore, every word will have to keep a reference to the correspondent zero-level node to find out the real number of appearances.

The external words, which are present at decoding, and the internal ones are coded using a regular PPM model. In this case, all the tasks that were executed on bytes must be executed on words. Thus in the tree a word and not a byte will be inserted, and the context will be one with words and not with bytes. The saving queue of the 2,048 symbols from the past for the actualization of the tree after the cleaning (if we want to use it) will contain also words and not bytes.

In order to reduce the time of adding and searching within the tree, all the words that are in other structures will be references to words from the used alphabet. In this way, all the comparisons between words could be made based on reference, but a comparison between elements will not be made. The only task that involves comparison between words at byte level is when a word must be searched inside the alphabet.

3 Combining the external words with the adaptive generated words

The encoder and decoder must always keep the same alphabet. When the -1 order is reached, we must encode a word with the probability (the alphabet length)-1. If the length of the alphabet is not identical at encoding and at decoding at that step, then the decoder will not be able to follow the coder's steps and the decoding will fail. In the alphabet, there can be external words present at decoding, external words that are not present at decoding, and internal words (automatically present at decoding). At the encoding step we know all the words of any type but at decoding we will not have at any moment all the words that were considered at encoding external and absent at decoding. For this reason, when we encode a symbol with a probability dependent on the alphabet length, we will consider only the words marked as being present at decoding. This is why it is important that every word which was external and absent at decoding to be marked as being extern and present at decoding only after this word has been encoded character by character and was followed by a special word and a counter type word. The disadvantage of combining external words, which are not present at decoding, with internal words is that the internal ones have priority, replacing the external ones absent at the decoding step. The external words are the result of other algorithms or of user's experience and many times this can be a useful information, which may improve the encoding. At the occurrence of an internal word, which replaces an external one, this useful information is ignored. The problem is that in the case of external words the lifetime is unlimited while the lifetime of internal words is limited if they have not been seen a few times in the past. (the internal words with zero appearance number are erased periodically). The advantage of this combination is that an absent word at decoding can be replaced with an adaptive generated one, which is seen many times until the end of the survival period. Because the adaptive generated word is encoded regularly, and the word absent at decoding is encoded character by character, the result is a gain.

4 Experimental results

The next two tables contain the best results obtained in two different experiments, using both plain and complex test files. The last line from every table represents the number of bits per character obtained on a compression with the standard RAR application.

We can remark that the adaptive mode is efficient when the text contains words that appear repeatedly in the text. For example, it is obvious that **aaa** looks all the same and **limit_comp.xmcd** is an XML format that contains elements of the same type in the tags. If the text has not a predefined structure, then the adaptive mode will generate words that initially can be good but later could be too long or too short. At first, a long word can be generated, but later we can find a piece of text that needs a shorter version of this word (it has a partial match with a text fragment). This is why it is possible to generate a shorter word if the restrictions are matched. We can first generate a short word and then a longer word, which includes the short one. The words that includes other words are efficient if they are used. Unfortunately, for the files that do not have a well-defined structure it is likely to be generated words that later will not be used as it should. If the word was not seen many times in the past, it will have a small probability.

Parameter / File	aaa	limit_comp.xmcd	concertBach	ByteEditor.exe
Normal	0.01201	5.77123	4.2048	9.32427
Time [sec]	6.783	12.844	10.542	13.846
Adaptive	0.00451	5.57401	4.19925	9.17483
Time [sec]	0.078	17.260	10.330	13.867
gainTypeadaptive	Equal	Length	Equal	Length
clearPeriod [octets]	1500	1500	1500	1500
Adaptive / max.word	0.00471	5.51759	4.44394	9.17169
Time [sec]	0.060	18.622	10.392	14.049
gainTypeadaptive	Equal	Length	Equal	Appearances
clearPeriod [octets]	1500	1500	1500	1500
Search	0.00431	5.43396	-	9.20337
Time [sec]	0.073	19.289	-	13.478
gainTypesearch	Equal	Length	-	Appearances
gainTypecoding	Length	Length	-	Length
Search / max.word	0.00431	5.4443	-	9.20337
Time [sec]	0.070	16.257	-	14.092
gainTypesearch	Equal	Equal	-	Appearances
Adaptive / Search	0.00411	5.42187	-	9.18202
Time [sec]	0.401	20.542	-	13.809
gainTypeadaptive	Length	Appearances	-	Length
gainTypesearch	Appearances	Length	-	Appearances
clearPeriod [octets]	1500	1500	-	1500
Adaptive / Search / max.word	0.00521	5.4284	-	9.17326
Time [sec]	0.065	20.566	-	14.050
gainTypeadaptive	Appearances	Length	-	Appearances
gainTypesearch	Appearances	Equal	-	Appearances
clearPeriod [octets]	1500	1500	-	1500
RAR ("best" mode)	0.01141	3.7816	2.53007	7.30202

Table 1. Comparison of the best results (first experiment)

From the performed experiments this negative effect was not noticed. For every file, the adaptive search produced better results. The adaptive search of words cannot see in the future and cannot view which is the best word to choose. This is why the adaptive search is recommended only for files with a specific structure. We can remark that it is best to use the word with the biggest length when the text has a defined structure, because it is very likely that this will show in the future.

The most efficient from the compression point of view is the search of words that appear repeatedly, before the encoding. For the files that contain redundant words which can be seen in a period of existence of a word (*clearPeriod*) so they can be added in the alphabet, the adaptive searched is combined with that performed in a separate stage from the encoding.

The **aaa** file is compressed the best by using the adaptive search together with that performed before the encoding, because the search of words in a separate stage is limited to 255 (bytes), while the adaptive search is unlimited. The encoder uses words found by search, in a separate phase, and it adaptively extends them based on the data tree. For the experiments, the adaptive search was limited to the length of 1,000.

Parameter / File	paper1	progc	obj1	trans
Normal	3.78187	3.84196	5.38653	2.52419
Time [sec]	7.059	5.468	4.801	10.452
Adaptive	3.78157	3.82035	5.11496	2.51019
Time [sec]	6.880	5.359	4.636	11.340
gainTypeadaptive	Appearances	Appearances	Length	Appearances
clearPeriod [octets]	1500	1500	1500	1500
Adaptive / max.word	3.89955	3.8975	5.07924	2.60624
Time [sec]	7.047	5.411	4.704	12.266
gainTypeadaptive	Equal	Equal	Length	Equal
clearPeriod [octets]	1500	1500	1500	1500
Search	3.77736	3.80298	5.18936	2.45068
Time [sec]	9.549	4.952	4.056	21.223
gainTypesearch	Length	Appearances	Appearances	Equal
gainTypecoding	Equal	Length	Length	Length
Search / max.word	3.78006	3.80298	5.20945	2.45068
Time [sec]	9.560	4.940	4.115	21.196
gainTypesearch	Length	Appearances	Appearances	Equal
Adaptive / Search	3.7915	3.82076	5.12128	2.46476
Time [sec]	9.874	5.977	4.656	22.842
gainTypeadaptive	Appearances	Appearances	Equal	Appearances
gainTypesearch	Length	Equal	Appearances	Equal
clearPeriod [octets]	1500	1500	1500	1500
Adaptive / Search /max.word	3.8699	3.89488	5.07961	2.57713
Time [sec]	9.399	7.315	5.402	20.992
gainTypeadaptive	Appearances	Equal	Length	Appearances
gainTypesearch	Length	Length	Appearances	Equal
clearPeriod [octets]	1500	1500	1500	1500
RAR ("best" mode)	2.20748	2.23312	3.65365	1.26385

Table 2. Comparison of the best results (second experiment)

The **ByteEditor.exe** file is an executable file, which is extended and not compressed. The problem comes from the PPM classic encoder and not from the extended alphabet. A smaller size is obtained with the help of adaptive search.

The **concertBach** file did not contain words that will match the rules imposed by the search in other stage than that of coding, so for this reason any experiment that had this type of search was not performed.

We can observe that the best performances of the **aaa, limit_comp.xmcd** and **progc.cs** compression is obtained by combining the adaptive search with that performed in a separate stage. This phenomenon is present because the coding of a missing word at the encoding means the coding of every of its character, while the coding of a present word does not have these disadvantages. If it is found a word with the adaptive search and this is present in the alphabet and is absent at decoding, then it is replaced with the one adaptively generated. After the insertion of the word in the text, we find (until the end of the word's existing period) a match with this word and therefore we have a gain because the word has not been coded character by character. In the case when the external word (absent at decoding) was replaced by an adaptive generated word that will not be found in the text, then we have a loss.

The encoding time usually increases compared to the PPM with regular alphabet because the words represented by strings of bytes, not only by bytes, must be checked. The file **aaa** is a special case where the coding time drops because there are few words, of very big length, which are coded.

The **concertBach** and ByteEditor.exe files are better coded by using the adaptive search because the restrictions imposed to the search performed in a separate stage of that of coding are too strong. These texts contain short words that appear repeatedly, and the adaptive search manages to find some of them because its minimum length is 5, while for the search before coding the minimum length is 20. For all the test files, the same encoding parameters were used. For this reason, we cannot say these are the best results that can be obtained. Still, an improvement is obtained.

Table 3 presents the experimental results obtained by using the adaptive search encoding for a larger set of original file types. One can analyze the compression efficiency obtained when all the words are accepted in the alphabet and the value of the existing period of a word (*clearPeriod*) is big. (e.g., 20,000 bytes).

File	Bits/character	Time [sec]
aaa	0.00510	0.030
limit_comp.xmcd	5.08220	78.550
concertBach	3.80084	34.441
ByteEditor.exe	9.18898	18.288
paper1	3.57781	23.326
progc	3.58870	16.224
obj1	4.92113	9.789
trans	2.40917	40.403

Table 3. Parameters for adaptive search without restrictions

One can see that for the proposed files the use of the adaptive search combined with the adding of words with no restrictions has better compression results but less quality time results from the gain and minimum length point of view. The time increases because there are many words in the alphabet and their search lasts a long time. The compression ratio is better because the words are early discovered and used. Although the alphabet has many words and the probability of a word at the -1 prediction level (the reverse value of the alphabet length) is small, the encoding is not strongly influenced by this because the -1 level situations are rare.

5 Conclusions

The most efficient is the search before the encoding together with the use of maximum length word at the encoding. The adaptive search can be performed in the case of files with many repeating words and has the advantage that it is performed at the coding phase. The combining of the two procedures of search can be used only for a certain types of files that contain words that get repeated nearby (so that the adaptive search can find them) and words situated much apart in the text so they won't be included in the alphabet by the adaptive search. From the tests, it results that the gain function depends on the type of used files for most of the files. The difference between the extended alphabet PPM encoding and the WinRar compression is about 1.5 bits/character. The file **aaa** (plain text) is compressed better with the extended alphabet PPM. From the previous results, one can observe that the encoding with adaptive search without restrictions is the most efficient and most files are compressed better with extended alphabet PPM.

References

[1] A. Moffat, Implementing the PPM Data Compression Scheme, *IEEE Trans. Commun.* 38, 1990.

[2] Th. H. Cormen, *Introduction to Algorithms*, Ch. E. Leiserson , R. L. Rivest eds., The MIT Press. 1999.

[3] N. Abramson, *Information Theory and Coding*, McGraw-Hill, NY, 1963.

[4] T. Bell , I. H. Witten, J. G. Cleary, *Modeling for Text Compression*, ACM Computing Surveys 21, 1989.

[5] Pr. Skibinski, PPM with the Extended Alphabet, *Inform. Sci.* 176, 2006.

[6] J. Bentley, D. McIlroy, Data compression using long common strings, *Inform. Sci.* 135, Part 1-2, June 2001.

[7] A. T. Murgan, *The Principles of Information Theory in Information and Communication Engineering*, Romanian Academy Press, Bucharest, 1998.

[8] R. Radescu, *Lossless Compression - Methods and Applications*, Matrix Rom Press, Bucharest, 2003.

[9] M. Nelson, *The Data Compression Book*, 2nd Ed., J.-L. Gailly ed., M&T Books, 1995.

[10] *** *Data Compression - The Complete Reference*, 3rd Ed., D. Salomon ed., Springer-Verlag, 2004.

[11] *** *Lossless Compression Handbook*, 1st Ed., Kh. Sayood ed., Acad. Press, 2002.

[12] R. Radescu, C. Harbatovschi, Compression methods using prediction by partial matching, *Proc. 6th Intern. Conf. Commun.*, Bucharest, 2006, 65-68.

[13] R. Radescu, R. Popa, On the performances of symbol ranking text compression method, *Sci. Bull. "Politehnica" Univ. Timisoara, Romania, Trans. Electr. Commun., Electr. Telecomm. Symp.* 49, ETC 2004, 25-27.

[14] The Calgary Corpus:
ftp://ftp.cpsc.ucalgary.ca/pub/projects/text.compression.corpus

[15] www.winrar.com

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 236-242*

Bounds for the minimum distance in constacyclic codes

DIANA RADKOVA dradkova@fmi.uni-sofia.bg
Faculty of Mathematics and Informatics, Sofia University,
5 James Bouchier blvd, 1164 Sofia, BULGARIA

A. J. VAN ZANTEN A.J.vanZanten@twi.tudelft.nl
Delft University of Technology, Faculty of Information Technology and Systems
Department of Mathematics, P.O. Box 5031,
2600 GA Delft, THE NETHERLANDS

Abstract. In algebraic coding theory it is common practice to require that $(n, q) = 1$, where n is the word length and $F = \mathrm{GF}(q)$ is the alphabet. In this paper, which is about constacyclic codes, we shall stick to this practice too. Since linear codes have the structure of linear subspaces of F^n, an alternative description of constacyclic codes in terms of linear algebra appears to be another quite natural approach. Due to this description we derive lower bounds for the minimum distance of constacyclic codes that are generalizations of the well known BCH bound, the Hartmann-Tzeng bound and the Roos bound.

Definition 1. *Let a be a nonzero element of $F = \mathrm{GF}(q)$. A code C of length n over F is called constacyclic with respect to a, if whenever $\mathbf{x} = (c_1, c_2, \ldots, c_n)$ is in C, so is $\mathbf{y} = (ac_n, c_1, \ldots, c_{n-1})$.*

Let a be a nonzero element of F and let

$$\psi_a : \begin{cases} F^n \to F^n \\ (x_1, x_2, \ldots, x_n) \mapsto (ax_n, x_1, \ldots, x_{n-1}) \end{cases}.$$

Then $\psi_a \in \mathrm{Hom}\, F^n$ and it has the following matrix

$$B_n(a) = B_n = \begin{pmatrix} 0\,0\,0 \ldots a \\ 1\,0\,0 \ldots 0 \\ 0\,1\,0 \ldots 0 \\ \vdots\ \vdots\ \vdots\ \ddots\ \vdots \\ 0\,0\,0 \ldots 0 \end{pmatrix}$$

with respect to the standard basis $e = (\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_n)$. The characteristic polynomial of B_n is $f_{B_n}(x) = (-1)^n(x^n - a)$. We shall denote it by $f(x)$. We assume that $(n, q) = 1$. The polynomial $f(x)$ has no multiple roots and splits into distinct irreducible monic factors $f(x) = (-1)^n f_1(x) \ldots f_t(x)$. Let $U_i = \mathrm{Ker}\, f_i(\psi_a)$, $i = 1, \ldots, n$. For the proof of the following theorem we refer to [1].

Theorem 1. *Let C be a linear constacyclic code of length n over F. Then the following facts hold.*

1) C is a constacyclic code iff C is a ψ_a-invariant subspace of F^n;

2) $C = U_{i_1} \oplus \cdots \oplus U_{i_s}$ for some minimal ψ_a-invariant subspaces U_{i_r} of F^n and $k := \dim_F C = k_{i_1} + \cdots + k_{i_s}$, where k_{i_r} is the dimension of U_{i_r};

3) $f_{\psi_a|C}(x) = (-1)^k f_{i_1}(x) \dots f_{i_s}(x) = g(x)$;

4) $\mathbf{c} \in C$ iff $g(B_n)\mathbf{c} = \mathbf{0}$;

5) the polynomial $g(x)$ has the smallest degree with respect to property 4);

6) $\mathrm{r}\,(g(B_n)) = n - k$, where $\mathrm{r}\,(g(B_n)) = n - k$ is the rank of the matrix $g(B_n)$.

Let $K = \mathrm{GF}(q^m)$ be the splitting field of the polynomial $f(x) = (-1)^n(x^n - a)$ over F, where $0 \neq a \in F$. Let the eigenvalues of ψ_a be $\alpha_1, \dots, \alpha_n$, with $\alpha_i = \sqrt[n]{a}\alpha^i$, $i = 1, \dots, n$, where α is a primitive $n-$th root of unity and $\sqrt[n]{a}$ is a fixed, but otherwise arbitrary, zero of the polynomial $x^n - a$. Let $\mathbf{v}_i$ be the respective eigenvectors, $i = 1, \dots, n$. More in particular we have

$$B_n \mathbf{v}_i^t = \alpha_i \mathbf{v}_i^t, \;\; \mathbf{v}_i = ({\alpha_i}^{n-1}, {\alpha_i}^{n-2}, \dots, \alpha_i, 1), \;\; i = 1, \dots, n.$$

Let us consider the basis $v = (\mathbf{v_1}, \dots, \mathbf{v_n})$ of eigenvectors of ψ_a. We carry out the basis transformation $e \to v$, and obtain that

$$D = \begin{pmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_n \end{pmatrix} = T^{-1} B_n T,$$

with

$$T = \begin{pmatrix} {\alpha_1}^{n-1} & {\alpha_2}^{n-1} & \dots & {\alpha_n}^{n-1} \\ {\alpha_1}^{n-2} & {\alpha_2}^{n-2} & \dots & {\alpha_n}^{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ 1 & 1 & \dots & 1 \end{pmatrix}.$$

Let $\mathbf{u}_i = (\alpha_i, {\alpha_i}^2, \dots, {\alpha_i}^{n-1}, {\alpha_i}^n)$, $i = 1, \dots, n$. Then

$$\langle \mathbf{v}_i, \mathbf{u}_j \rangle = \sum_{k=1}^{n} \Big(\frac{\alpha_i}{\alpha_j}\Big)^k = \sum_{k=1}^{n} (\alpha^{i-j})^k = \begin{cases} n, \; for \; i = j \\ 0, \; otherwise \end{cases}.$$

From this it follows immediately that

$$T^{-1}=\frac{1}{n}\begin{pmatrix}\mathbf{u}_1\\ \mathbf{u}_2\\ \vdots\\ \mathbf{u}_n\end{pmatrix}=\frac{1}{n}\begin{pmatrix}\alpha_1 & {\alpha_1}^2 & \dots & {\alpha_1}^{n-1} & {\alpha_1}^n\\ \alpha_2 & {\alpha_2}^2 & \dots & {\alpha_2}^{n-1} & {\alpha_2}^n\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \alpha_n & \alpha_n^2 & \dots & \alpha_n^{n-1} & {\alpha_n}^n\end{pmatrix}.$$

Let $h(x)=\frac{f(x)}{g(x)}$. Let $\deg h(x)=n-k=r$, and let its r zeros be $\alpha_{i_1},\alpha_{i_2},\dots,\alpha_{i_r}$ and its k nonzeros $\alpha_{j_1},\alpha_{j_2},\dots,\alpha_{j_k}$. It is obvious that the zeros of $g(x)$ are the nonzeros of $h(x)$ and vice versa. Assume that $\mathbf{c}=(c_1,c_2,\dots,c_n)\in F^n$ and let $\mathbf{c}'=T^{-1}\mathbf{c}$. We know $\mathbf{c}\in C$ iff $g(B_n)\mathbf{c}=\mathbf{0}$. The latter condition is equivalent to $g(D)\mathbf{c}'=T^{-1}g(B_n)TT^{-1}\mathbf{c}=T^{-1}g(B_n)\mathbf{c}=\mathbf{0}$, which, in its turn, is equivalent to $c'_{i_1}=c'_{i_2}=\dots=c'_{i_r}=0$. Hence, we get the following necessary and sufficient condition for $\mathbf{c}$ to be a codeword in C

$$\mathbf{u}_{i_l}\mathbf{c}=0,\ l=1,\dots,r.$$

Theorem 2. *Let C be a linear constacyclic code of length n over F, $g(x)=f_{\psi_a|_C}(x)$ and $h(x)=\frac{f(x)}{g(x)}$. Let for some integers $b\geq 1$, $\delta\geq 1$ the following equalities*

$$h(\alpha_b)=h(\alpha_{b+1})=\dots=h(\alpha_{b+\delta-2})=0$$

hold, i.e., the polynomial $h(x)$ has a string of $\delta-1$ consecutive zeros. Then the minimum distance of the code C is at least δ.

Proof. If $\mathbf{c}=(c_1,c_2,\dots,c_n)$ is in C, then

$$\mathbf{u}_i\mathbf{c}=0,\ i=b,b+1,\dots,b+\delta-2,$$

so that

$$\begin{pmatrix}\alpha_b & \alpha_b^2 & \dots & \alpha_b^{n-1} & \alpha_b^n\\ \alpha_{b+1} & \alpha_{b+1}^2 & \dots & \alpha_{b+1}^{n-1} & \alpha_{b+1}^n\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \alpha_{b+\delta-2} & \alpha_{b+\delta-2}^2 & \dots & \alpha_{b+\delta-2}^{n-1} & \alpha_{b+\delta-2}^n\end{pmatrix}\begin{pmatrix}c_1\\ c_2\\ \vdots\\ c_n\end{pmatrix}=\begin{pmatrix}0\\ 0\\ \vdots\\ 0\end{pmatrix}.$$

Now let us suppose that $\mathbf{c}$ has weight $w\leq\delta-1$, i.e., $c_i\neq 0$ iff $i\in\{a_1,a_2,\dots,a_w\}$. Then the last equality implies

$$\begin{pmatrix}\alpha_b^{a_1} & \dots & \alpha_b^{a_w}\\ \alpha_{b+1}^{a_1} & \dots & \alpha_{b+1}^{a_w}\\ \vdots & \ddots & \vdots\\ \alpha_{b+w-1}^{a_1} & \dots & \alpha_{b+w-1}^{a_w}\end{pmatrix}\begin{pmatrix}c_{a_1}\\ c_{a_2}\\ \vdots\\ c_{a_w}\end{pmatrix}=\begin{pmatrix}0\\ 0\\ \vdots\\ 0\end{pmatrix}.$$

Hence, the determinant of the matrix on the left is zero. But this determinant is equal to

$$\begin{vmatrix} \alpha_b^{a_1} & \cdots & \alpha_b^{a_w} \\ \alpha_{b+1}^{a_1} & \cdots & \alpha_{b+1}^{a_w} \\ \vdots & \ddots & \vdots \\ \alpha_{b+w-1}^{a_1} & \cdots & \alpha_{b+w-1}^{a_w} \end{vmatrix} = \begin{vmatrix} \mu^{a_1}\alpha^{a_1 b} & \cdots & \mu^{a_w}\alpha^{a_w b} \\ \mu^{a_1}\alpha^{a_1(b+1)} & \cdots & \mu^{a_w}\alpha^{a_w(b+1)} \\ \vdots & \ddots & \vdots \\ \mu^{a_1}\alpha^{a_1(b+w-1)} & \cdots & \mu^{a_w}\alpha^{a_w(b+w-1)} \end{vmatrix} =$$

$$= \mu^{a_1+\cdots+a_w}\alpha^{(a_1+\cdots+a_w)b} \begin{vmatrix} 1 & \cdots & 1 \\ \alpha^{a_1} & \cdots & \alpha^{a_w} \\ \vdots & \ddots & \vdots \\ \alpha^{a_1(w-1)} & \cdots & \alpha^{a_w(w-1)} \end{vmatrix} \neq 0,$$

where $\mu = \sqrt[n]{a}$. The contradiction proves the statement. □

The next result follows easily from Theorem 2.

Corollary 1. *Let C be a linear constacyclic code of length n over F and let*

$$\alpha_b, \alpha_{b+s}, \ldots, \alpha_{b+(\delta-2)s}$$

are zeros of $h(x)$, where $(s, n) = 1$. Then the minimum distance of C is at least δ.

The following theorem generalizes the Hartmann-Tzeng bound for linear constacyclic codes. Its proof is close to Roos' proof for cyclic codes in [2].

Theorem 3. *Let C be a constacyclic code of length n over F, $g(x) = f_{\varphi|_C}(x)$, $h(x) = \frac{f(x)}{g(x)}$ and let α be a primitive n-th root of unity in $K = \mathrm{GF}(q^m)$. Assume that there exist integers s, b, c_1 and c_2 where $s \geq 0$, $b \geq 0$, $(n, c_1) = 1$ and $(n, c_2) < \delta$, such that*

$$h(\alpha_{b+i_1c_1+i_2c_2}) = 0, \quad 0 \leq i_1 \leq \delta - 2,\ 0 \leq i_2 \leq s.$$

Then the minimum distance d of C satisfies $d \geq \delta + s$.

Proof. We use induction on s. For $s = 0$ the assertion follows from Corollary 1, since $(n, c_1) = 1$. Take some $s > 0$ and assume that the theorem holds, i.e.,

$$h(\alpha_{b+i_1c_1+i_2c_2}) = 0, \quad 0 \leq i_1 \leq \delta - 2,\ 0 \leq i_2 \leq s$$

defines a constacyclic code C of minimum distance $d \geq \delta + s$. We have that $\mathbf{c} \in C$ iff $\mathbf{u}_k\mathbf{c} = 0$, $k = b + i_1c_1 + i_2c_2$, $0 \leq i_1 \leq \delta - 2$, $0 \leq i_2 \leq s$. So, we obtain that $U\mathbf{c} = \mathbf{0}$, where U is the following matrix

$$U = \begin{pmatrix} \alpha_b & \alpha_b^2 & \cdots & \alpha_b^{n-1} & \alpha_b^n \\ \alpha_{b+c_1} & \alpha_{b+c_1}^2 & \cdots & \alpha_{b+c_1}^{n-1} & \alpha_{b+c_1}^n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{b+(\delta-2)c_1} & \alpha_{b+(\delta-2)c_1}^2 & \cdots & \alpha_{b+(\delta-2)c_1}^{n-1} & \alpha_{b+(\delta-2)c_1}^n \\ \alpha_{b+c_2} & \alpha_{b+c_2}^2 & \cdots & \alpha_{b+c_2}^{n-1} & \alpha_{b+c_2}^n \\ \alpha_{b+c_1+c_2} & \alpha_{b+c_1+c_2}^2 & \cdots & \alpha_{b+c_1+c_2}^{n-1} & \alpha_{b+c_1+c_2}^n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{b+(\delta-2)c_1+c_2} & \alpha_{b+(\delta-2)c_1+c_2}^2 & \cdots & \alpha_{b+(\delta-2)c_1+c_2}^{n-1} & \alpha_{b+(\delta-2)c_1+c_2}^n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{b+sc_2} & \alpha_{b+sc_2}^2 & \cdots & \alpha_{b+sc_2}^{n-1} & \alpha_{b+sc_2}^n \\ \alpha_{b+c_1+sc_2} & \alpha_{b+c_1+sc_2}^2 & \cdots & \alpha_{b+c_1+sc_2}^{n-1} & \alpha_{b+c_1+sc_2}^n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{b+(\delta-2)c_1+sc_2} & \alpha_{b+(\delta-2)c_1+sc_2}^2 & \cdots & \alpha_{b+(\delta-2)c_1+sc_2}^{n-1} & \alpha_{b+(\delta-2)c_1+sc_2}^n \end{pmatrix}.$$

From the definition of α_i it follows that $\alpha_{b+lc_2}\alpha^{c_2} = \alpha_{b+(l+1)c_2}$, $0 \le l \le s$ and $\alpha_{b+i_1c_1+lc_2}\alpha^{c_2} = \alpha_{b+i_1c_1+(l+1)c_2}$, $0 \le i_1 \le \delta - 2$. Hence, every set of $\delta - 1$ consecutive zeros of $h(x)$ is obtained from the previous one by multiplying by $\beta = \alpha^{c_2}$. It follows that if we multiply the first column $\mathbf{b}_1$ of U by β, the second column $\mathbf{b}_2$ by β^2,..., the $n-$th column $\mathbf{b}_n$ by β^n, the resulting matrix U_0 contains all rows of U except the first $\delta - 1$ rows, whereas its last $\delta - 1$ rows are new and correspond to the zeros $\alpha_{b+(s+1)c_2}, \dots, \alpha_{b+(\delta-2)c_1+(s+1)c_2}$. Note that U need not be the full parity check matrix of C. However, we can interpret U as parity check matrix for a code C^* over K. If C^* has minimum distance d^*, then clearly $d \ge d^*$. We shall show that $d^* \ge \delta + s$. Since $d \ge d^*$ this implies the theorem. Since $(n, c_2) < \delta$, β has order $e = \frac{n}{(n,c_2)} > \frac{n}{\delta} \ge \frac{n}{d^*}$ and hence in the sequence $\beta, \beta^2, \dots, \beta^n$ each element occurs $\frac{n}{e} < d^*$ times. We now define the matrix

$$U' = \begin{bmatrix} U \\ U_o \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_n \\ \beta\mathbf{b}_1 & \beta^2\mathbf{b}_2 & \dots & \beta^n\mathbf{b}_n \end{bmatrix}.$$

We know that every $d^* - 1$ columns of U are linearly independent. We shall prove now that every d^* columns of U' are independent. In order to show this, let us suppose that U' contains d^* columns which are linearly dependent. Without loss of generality we may assume that these are the first d^* columns.

Then there will exist elements $\lambda_1, \lambda_2, \dots \lambda_{d^*} \in K$ (not all zero) such that

$$\sum_{i=1}^{d^*} \lambda_i \mathbf{b}_i = \sum_{i=1}^{d^*} \lambda_i \beta^i \mathbf{b}_i = \mathbf{0}, \textit{ and so } \sum_{i=1}^{d^*-1} \lambda_i (\beta^i - \beta^{d^*}) \mathbf{b}_i = \mathbf{0}.$$

Since any $d^* - 1$ columns of U are linearly independent, it follows that $\lambda_i(\beta^i - \beta^{d^*}) = 0$ for $1 \leq i \leq d^* - 1$. However, $\lambda_i \neq 0$ for $1 \leq i \leq d^*$, again because no $d^* - 1$ columns of U are linearly dependent. Hence, we obtain $\beta = \beta^2 = \dots = \beta^{d^*}$, which contradicts the fact that in the sequence $\beta, \beta^2, \dots, \beta^n$ each element occurs less than d^* times. It immediately follows that the constacyclic code C' with zeros $\alpha_{b+i_1c_1+i_2c_2}$, $0 \leq i_1 \leq \delta - 2$, $0 \leq i_2 \leq s + 1$ of $h'(x)$, where $h'(x) = \frac{f(x)}{f_{\varphi|_{C'}}(x)}$, has minimum distance at least $d^* + 1$. □

Next, we shall derive an even more general bound for the minimum distance of constacyclic codes, which is similar to the so-called Roos bound for cyclic codes in [3]. Our proof and notation are also very close to the proof in [3], and therefore we shall partly omit it.

Let K be any finite field and $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ any matrix over K with n columns $\mathbf{a}_i$, $1 \leq i \leq n$. Let C_A denote the linear code over K with A as parity check matrix. The minimum distance of C_A will be denoted as d_A.

For any $m \times n$ matrix $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ with nonzero columns $\mathbf{x}_i \in K^m$ for $1 \leq i \leq n$, we define the matrix $A(X)$ as

$$A(X) := \begin{pmatrix} x_{11}\mathbf{a}_1 & x_{12}\mathbf{a}_2 & \dots & x_{1n}\mathbf{a}_n \\ x_{21}\mathbf{a}_1 & x_{22}\mathbf{a}_2 & \dots & x_{2n}\mathbf{a}_n \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}\mathbf{a}_1 & x_{m2}\mathbf{a}_2 & \dots & x_{mn}\mathbf{a}_n \end{pmatrix}.$$

The following lemma describes how the parity check matrix A for a linear code can be extended with new rows in such a way that the minimum distance increases. A proof of this result is given by Roos (cf. [3]).

Lemma. If $d_A \geq 2$ and every $m \times (m + d_A - 2)$ submatrix of X has full rank, then $d_{A(X)} \geq d_A + m - 1$.

Definition 2. *A set $M = \{\alpha_{j_1}, \alpha_{j_2}, \dots, \alpha_{j_l}\}$ of zeros of the polynomial $x^n - a$ in $K = \mathrm{GF}(q^m)$ will be called a consecutive set of length l if a primitive $n-$th root of unity β and an exponent i exist such that $M = \{\beta_i, \beta_{i+1}, \dots, \beta_{i+l-1}\}$, with $\beta_s = \sqrt[n]{a}\beta^s$. More generally, one says that M is a consecutive set of $n-$th roots of unity if there is some primitive $n-$th root of unity β in K such that M consists of consecutive powers of β.*

Let $N = \{\alpha_{j_1}, \alpha_{j_2}, \dots, \alpha_{j_t}\}$ be a set of zeros of the polynomial $x^n - a$. The $t \times n$ matrix over K the j_s-th row of which equals $(\alpha_{j_s}, \alpha_{j_s}^2, \dots, \alpha_{j_s}^n)$ will be denoted by U_N. (If N is a set of $n-$th roots of unity, the analogous matrix over K will be denoted as H_N.) So, it is clear that U_N is a parity check matrix for the constacyclic code C having N as a set of zeros for $h(x)$. Let C_N be the constacyclic code over K with U_N as parity check matrix, and let this code have minimum distance d_N. So, the minimum distance of C is at least d_N, since C is a subfield code of C_N (cf. [3]).

Theorem 4. *If N is a nonempty consecutive set of zeros of the polynomial $x^n - a$ and if M is a set of $n-$th roots of unity such that $|\overline{M}| < |M| + |N|$ for some consecutive set $\overline{M}$ containing M, then $d_{MN} \geq |M| + |N|$.*

Proof. Let us define $A := U_N$ and $X := H_M$. Then one may easily verify that $A(X) = U_{MN}$, where MN is the set of all products mn, $m \in M$, $n \in N$. Since N is a nonempty consecutive set, $d_N = |N| + 1 \geq 2$. Hence, the assertion of the theorem follows from the lemma above if in the matrix H_M every $|M| \times (|M| + |N| - 1)$ submatrix has full rank. It is sufficient to show that this is the case if $|\overline{M}| < |M| + |N|$ for some consecutive set $\overline{M} \supseteq M$. Observe that H_M is a submatrix of $H_{\overline{M}}$, and that in $H_{\overline{M}}$ every $|\overline{M}| \times |\overline{M}|$ submatrix is nonsingular, since the determinant of such a matrix is of Vandermonde type. So, it immediately follows that every $|M| \times |\overline{M}|$ submatrix of H_M has full rank. Since $|\overline{M}| < |M| + |N|$, this implies that every $|M| \times (|M| + |N| - 1)$ submatrix of H_M has full rank, which proves the theorem. □

References

[1] D. Radkova, A. Bojilov, A. J. van Zanten, Cyclic codes and quasi-twisted codes: an algebraic approach, Rep. MICC 07-08 Univ. Maastricht, 2007

[2] C. Roos, A generalization of the BCH bound for cyclic codes, including the Hartmann-Tzeng bound, *J. Combin. Theory* Ser. A, 33, 1982, 229-232.

[3] C. Roos, A new lower bound for the minimum distance of a cyclic code, *IEEE Trans. Inform. Theory* 29, 1983, 330-332.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 243-247*

On binary linear completely regular and completely transitive codes with arbitrary covering radius[1]

Josep Rifà josep.rifa@uab.es
Dept. of Information and Communications Engineering,
Autonomous University of Barcelona, 08193-Bellaterra, SPAIN
Victor Zinoviev zinov@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences,
Bol'shoi Karetnyi per. 19, GSP-4, Moscow, 101447, RUSSIA

Abstract. An infinite class of binary linear completely regular and completely transitive codes is given. The covering radius of these codes is growing with the length of the code.

1 Introduction

Let E be a binary alphabet. A binary (n, N, d)-code C is a subset of E^n where n is the *length*, d is the *minimum distance* and $N = |C|$ is the *cardinality* of C. For the case when C is a k-dimensional linear subspace of $\mathbb{F}^n$, the code C is a *linear* code denoted $[n, k, d]$, where $N = 2^k$.

Given any vector $\mathbf{v} \in E^n$, its *distance to the code* C is

$$d(\mathbf{v}, C) = \min_{\mathbf{x} \in C}\{d(\mathbf{v}, \mathbf{x})\}$$

and the *covering radius* of the code C is

$$\rho = \max_{\mathbf{v} \in E^n}\{d(\mathbf{v}, C)\}.$$

We assume that a code C always contains the zero vector. Let $D = C + \mathbf{x}$ be a *translate* of C. The *weight* $\mathrm{wt}(D)$ of D is the minimum weight of the codewords of D. For an arbitrary translate D of weight $i = \mathrm{wt}(D)$ denote by $\mu(D) = (\mu_0(D), \mu_1(D), ..., \mu_n(D))$ its weight distribution, where $\mu_i(D)$ denotes the number of words of D of weight i. Denote by C_j (respectively, D_j) the subset of C (respectively, of D), formed by all words of the weight j. In this terminology $\mu_i(D) = |D_i|$.

[1] This work was partially supported by Catalan DURSI Grant 2004PIV1-3, and also was partly supported by Russian fund of fundamental researches (the number of project 06 - 01 - 00226)

Definition 1 *A binary code C with covering radius ρ is called* completely regular *if the weight distribution of any its translate D is uniquely defined by the minimum weight of D, i.e. by the number $i = wt(D)$.*

2 Definitions and preliminary results

For a given code C with covering radius $\rho = \rho(C)$ define

$$C(i) = \{\mathbf{x} \in E^n : d(\mathbf{x}, C) = i\}, \ \ i = 1, 2, ..., \rho.$$

For any vector $\mathbf{x} \in E^n$ denote by $S(\mathbf{x})$ the sphere of radius one near $\mathbf{x}$, i.e. $S(\mathbf{x}) = \{\mathbf{y} \in E^n : d(\mathbf{x}, \mathbf{y}) = 1\}$.

Definition 2 *Let C be a code of length n with covering radius ρ. We say that C is* uniformly packed *in the wide sense, i.e. in the sense of [1], if there exist rational numbers $\alpha_0, \ \ldots, \ \alpha_\rho$ such that for any $\mathbf{v} \in E^n$*

$$\sum_{k=0}^{\rho} \alpha_k \, f_k(\mathbf{v}) = 1 \,, \qquad (1)$$

where $f_k(\mathbf{v})$ is the number of codewords at distance k from $\mathbf{v}$.

For any vector $\mathbf{x} \in E^n$ denote by $W_i(\mathbf{x})$ the sphere of radius i near $\mathbf{x}$, i.e. $W_i(\mathbf{x}) = \{\mathbf{y} \in E^n : d(\mathbf{x}, \mathbf{y}) = i\}$. Denote $W_1(\mathbf{x}) = W(\mathbf{x})$.

We say that two vectors $\mathbf{x}$ and $\mathbf{y}$ are *neighbors* if $d(\mathbf{x}, \mathbf{y}) = 1$. We use also the definition of completely regularity given in [10].

Definition 3 *A code C is a completely regular code if, for all $l \geq 0$, every vector $x \in C(l)$ has the same number c_l of neighbors in $C(l-1)$ and the same number b_l of neighbors in $C(l+1)$. Also, define $a_l = (q-1)n - b_l - c_l$ and note that $c_0 = b_\rho = 0$. Define by $\{b_0, \ldots, b_{\rho-1}; c_1, \ldots, c_\rho\}$ the intersection array of C.*

The *support* of $\mathbf{v} \in E^n$, $\mathbf{v} = (v_1, \ldots, \ v_n)$ is *supp*$(\mathbf{v}) = \{ \ \ell \mid v_\ell \neq 0 \ \}$. Say that a vector $\mathbf{v}$ *covers* a vector $\mathbf{z}$ if the condition $z_i \neq 0$ implies $z_i = v_i$.

For a binary (n, N, d) code C with zero codeword let $(\eta_0, \ldots, \eta_n)$ be its distance distribution, i.e. η_i is the number of ordered pairs of codewords at a distance i apart, divided by N. Let $(\eta'_0, \ldots, \eta'_n)$ be the MacWilliams transform of $(\eta_0, \ldots, \eta_n)$ and assume this vector has $s = s(C)$ nonzero components η'_i for $1 \leq i \leq n$. We call s the *external distance* of C. If C is a linear code, then $s(C)$ is the number of different nonzero weights of codewords in the dual code $C^\perp$.

Lemma 1 [7] *For any code C with covering radius $\rho(C)$ and external distance $s(C)$*

$$\rho(C) \leq s(C).$$

The case of equality above implies existence of uniformly packed code in the wide sense.

Lemma 2 [2] *Let C be a code with minimum distance $d = 2e + 1$, covering radius ρ, and external distance s. Then the code C is uniformly packed in the wide sense, if and only if $\rho = s$.*

For a binary code C let $\mathrm{Perm}(C)$ be its permutation stabilizer group. For any $\theta \in \mathrm{Perm}(C)$ and any translate $D = C + \mathbf{x}$ of C define the action of θ on D as: $\theta(D) = C + \theta(\mathbf{x})$.

Definition 4 *[13] Let C be a binary linear code with covering radius ρ. The code C is called* completely transitive, *if the set $\{C + \mathbf{x} : \mathbf{x} \in \mathbb{F}^n\}$ of all different cosets of C is partitioned under action of* $\mathrm{Perm}(C)$ *into exactly $\rho + 1$ orbits.*

Since two cosets in the same orbit should have the same weight distribution, it is clear, that any completely transitive code is completely regular.

It has been conjectured for a long time that if C is a completely regular code and $|C| > 2$, then $e \leq 3$. For the special case of linear completely transitive codes, the problem of existence is solved in [3, 4] in the sense that for $e \geq 4$ such nontrivial codes do not exist.

3 Main results

For a given natural number m where $m \geq 3$ denote by E_2^m the set of all binary vectors of length m and weight 2.

Definition 5 *Let $H^{(m)}$ be the binary matrix of size $m \times m(m-1)/2$, whose columns are exactly all the vectors from E_2^m (i.e. each vector from E_2^m occurs once as a column of $H^{(m)}$). Now define the binary linear code $C^{(m)}$ whose parity check matrix is the matrix $H^{(m)}$.*

For a fixed natural number m and any $i \in \{1, 2, \ldots, m\}$ define $f_i(m)$ as the weight of the vector sum of any i rows of $H^{(m)}$. Note that $f_i(m)$ is well defined and it does not depend on the specific rows taken in the computation as be can see in the next lemma.

Lemma 3 *For any natural number $m \geq 3$ the value $f_i(m)$ does not depend on the choice of i rows of $H^{(m)}$ and $f_i(m) = i \cdot (m - i)$ for $i \in \{1, 2, \ldots, m\}$.*

Lemma 4 *For any natural number $m \geq 3$ the code $C^{(m)}$ has the external distance $s(m) = \lfloor m/2 \rfloor$ and the covering radius $\rho(m) = \lfloor m/2 \rfloor$.*

Thus, the code $C^{(m)}$ has the same external distance and covering radius: $s(m) = \rho(m)$. By Lemma 2 the code $C^{(m)}$ is uniformly packed in the wide sense. The following statements shows that $C^{(m)}$ is, in fact, a completely transitive code and, so, a completely regular code too.

Theorem 1 *For any natural number $3 \leq m$ the code $C^{(m)}$ is a completely transitive $[n, k, d]$-code with the following parameters:*

$$n = \binom{m}{2}, \quad k = n - m + 1, \quad d = 3, \quad \rho = \lfloor m/2 \rfloor .$$

Theorem 2 *For any natural number $3 \leq m$ the code $C^{(m)}$ is a completely regular $[n, k, d]$-code with intersection numbers, for $\ell = 0, \ldots, \rho$:*

$$\begin{aligned} a_\ell &= 2\,\ell \cdot (m - 2\ell), \\ b_\ell &= \binom{m-2\ell}{2}, \\ c_\ell &= \binom{2\,\ell}{2}. \end{aligned}$$

The interesting fact is that generalization of this idea (i.e. using as a parity check matrix all possible binary vectors of length m and weight ℓ) above works only in three following cases. For given natural number m where $m \geq 3$ define by E_ℓ^m the set of all binary vectors of length m and weight ℓ.

Definition 6 *Denote by $H^{(m,\ell)}$ the binary matrix of size $m \times \binom{m}{\ell}$, whose columns are exactly all vectors from E_ℓ^m (i.e. each vector from E_ℓ^m occurs once as a column of $H^{(m,\ell)}$). Define the binary linear code $C^{(m,\ell)}$, whose parity check matrix is the matrix $H^{(m,\ell)}$.*

Theorem 3 *Let $C^{(m,\ell)}$ be the code defined above. Let $\ell \geq 3$. Let $C^{(m,\ell)}$ be a completely regular code. Then we are in one of the following three cases:*
(1) $m = 5$ and $\ell = 3$. The code $C^{(5,3)}$ is the $[10, 5, 4]$-code with covering radius $\rho = 3$ and with intersection array $(10, 9, 4; 1, 6, 10)$.
(2) $m = 6$ and $\ell = 4$. The code $C^{(6,4)}$ is the $[15, 10, 3]$-code with covering radius $\rho = 3$ and with intersection array $(15, 8, 1; 1, 8, 15)$.
(3) $m = 7$ and $\ell = 4$. The code $C^{(7,4)}$ is the $[35, 29, 3]$-code with covering radius $\rho = 2$ and with intersection array $(35, 16; 1, 20)$.
Furthermore, all these three codes are completely transitive.

References

[1] L. A. Bassalygo, G. V. Zaitsev, V. A. Zinoviev, Uniformly packed codes, *Probl. Inform. Transm.* 10, 1974, 9-14.

[2] L. A. Bassalygo, V. A. Zinoviev, Remark on uniformly packed codes, *Probl. Inform. Transm.* 13, 1977, 22-25.

[3] J. Borges, J. Rifa, On the nonexistence of completely transitive codes, *IEEE Trans. Inform. Theory* 46, 2000, 279-280.

[4] J. Borges, J. Rifa, V. A. Zinoviev, Nonexistence of completely transitive codes with error-correcting capability $e > 3$, *IEEE Trans. Inform. Theory* 47, 2001, 1619-1621.

[5] J. Borges, J. Rifa, V. A. Zinoviev, On non-antipodal binary completely regular codes, *Discr. Math.*, 2008, to appear.

[6] A. E. Brouwer, A. M. Cohen, A. Neumaier, *Distance-Regular Graphs*, Springer, Berlin, 1989.

[7] P. Delsarte, An algebraic approach to the association schemes of coding theory, *Philips Res. Rep. Suppl.* 10, 1973.

[8] J. M. Goethals, H. C. A. Van Tilborg, Uniformly packed codes, *Philips Res.* 30, 1975, 9-36.

[9] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error Correcting Codes*, North-Holland, New York, 1977.

[10] A. Neumaier, Completely regular codes, *Discr. Math.* 106/107, 1992, 335-360.

[11] J. Rifa, V. A. Zinoviev, On new completely regular q-ary codes, *Probl. Inform. Transm.*, 43, 2007.

[12] N. V. Semakov, V. A. Zinoviev, G. V. Zaitsev, Uniformly packed codes, *Probl. Inform. Transm.* 7, 1971, 38-50.

[13] P. Solé, Completely regular codes and completely transitive codes, *Discr. Math.* 81, 1990, 193-201.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 248-254*

On solving sparse algebraic equations over finite fields II. Extended abstract.

Igor Semaev Igor.Semaev@ii.uib.no
Department of Informatics, University of Bergen, NORWAY

1 Introduction

Let F_q be a finite field with q elements and X is a set of variables from F_q of size n. By X_i, $1 \leq i \leq m$ we denote subsets of X of size $l_i \leq l$. Equations

$$f_1(X_1) = 0, \ldots, f_m(X_m) = 0 \tag{1}$$

are considered, where f_i are polynomials over F_q and they only depend on variables X_i(l-sparse). We look for all solutions in F_q to (1). So we only consider polynomials of degree at most $q-1$ in each variable. They define mappings from all l_i-tuples over F_q to F_q and any such mapping is represented by a polynomial of degree at most $q-1$ in each variable. The equation $f_i(X_i) = 0$ is determined by (X_i, V_i), where V_i is the set of F_q-vectors in variables X_i, also called X_i-vectors, where f_i is zero. We call (X_i, V_i) a symbol. For $q = 2$ the polynomial f_i is uniquely defined by V_i. Given f_i, the set V_i is computed with q^{l_i} trials.

Deterministic Agreeing-Gluing Algorithm [6] and its average behavior are studied. Assume equiprobable distribution on (1). Given natural numbers m and $l_1, \ldots, l_m \leq l$, equations in (1) are independent. Each $f_i(X_i) = 0$ is determined by the subset X_i of size l_i taken uniformly at random, that is with the probability $\binom{n}{l_i}^{-1}$, and the mapping f_i taken, independently of X_i, with the probability $q^{-q^{l_i}}$. The running time of the Agreeing-Gluing Algorithm is a random variable.

For fixed q, l and $c \geq 1$ let $\beta = \beta(\alpha)$, where $0 \leq \alpha \leq l$, be the only root to

$$q^{\beta - \frac{\alpha}{l}} = q e^{g(\alpha)} (1 - \sum_{t=0}^{l} \binom{l}{t} \beta^{l-t} (1-\beta)^t (1 - \frac{1}{q})^{q^t})^{c - \frac{\alpha}{l}},$$

or $\beta(\alpha) = 0$ if there is not any root for some α. Here $g(\alpha) = f(z_\alpha) - \alpha + \alpha \ln \alpha - \frac{\alpha \ln q}{l}$ and $f(z) = \ln(e^z + q^{-1} - 1) - \alpha \ln(z)$, where by z_α we denote the only positive root of the equation $\frac{\partial f}{\partial z}(z) = 0$. We prove

Theorem 1 *Let $\frac{l_1 + l_2 + \ldots + l_m}{ln}$ tend to a constant $c \geq 1$ as n tends to ∞ while $q \geq 2$ and $l \geq 3$ are fixed. Let $r(q, l, c)$ be the maximal of* $\max_{0 \leq \alpha \leq l} q^{\beta(\alpha) - \frac{\alpha}{l}}$

Table 1: Algorithms' running time.

l	3	4	5	6
the worst case	1.324^n	1.474^n	1.569^n	1.637^n
Gluing1, expectation	1.262^n	1.355^n	1.425^n	1.479^n
Gluing2, expectation	1.238^n	1.326^n	1.393^n	1.446^n
Agreeing-Gluing1, expectation	1.113^n	1.205^n	1.276^n	1.334^n

.

and 1. *Then the expected complexity of the Agreeing-Gluing Algorithm is* $O((r(q,l,c)+\varepsilon)^n)$ *bit operations for any positive real* ε.

For any triple $q, l, c \geq 1$ the Theorem enables estimating the expected running time of the Agreeing-Gluing Algorithm with some mathematical software like Maple. To this end we realize that the equation $\frac{\partial f}{\partial z}(z) = 0$ is equivalent to $\frac{ze^z}{e^z+q^{-1}-1} = \alpha$. So $\alpha = \alpha(z)$ and $\beta = \beta(z)$ are functions in z and $z_\alpha = z$.

For some of $2, l, 1$(e.g. n Boolean equations in n variables each equation depends on l variables) we show the data obtained in Table 1 with the expected complexities of the Gluing1 and Gluing2 Algorithms from our previous work [7]. Agreeing-Gluing1 Algorithm is a variant of the Agreeing-Gluing Algorithm with the same asymptotical running time and polynomial in n memory requirement. In case $q = 2$ each instance of (1) may be encoded with a CNF formula in the same set of variables and of clause length at most l [7]. So l-SAT solving algorithms provide with the worst case complexity estimates, see [2], in the first line. We remark an exciting difference in the worst case complexity and expected complexity of the Agreeing-Gluing Algorithm. It is quite obvious that average instances of the l-SAT problem and that of (1) are different. That gives insight into why the expected complexity is so low in comparison with the worst case. The Agreeing-Gluing family algorithms seem better on sparse equation systems (1) than Gröbner Basis related algorithms, see conjectured estimates in [9].

This article was motivated by applications in cryptanalysis. Mappings implemented by modern ciphers are compositions of functions in small number of variables. Intermediate variables are introduced to simplify equations, describing the cipher, and get a system of sparse equations. We are studying an approach which exploits the sparsity of equations and doesn't depend on their algebraic degree. This approach was independently discovered in [10] and [5], where the Agreeing procedure(called local reduction in [10]) was described for the first time. The term Agreeing itself comes from [6]. No asymptotical estimates for that type of algorithms were given in [10, 5, 6]. We recommend to look also through our previous work [7], where some necessary basic facts were proved.

This is the extended abstract of [8]. The author is grateful to H.Raddum for careful reading the work and numerous remarks.

2 Gluing procedure and Gluing Algorithm

For symbols (X_i, V_i) for $i = 1, 2$, one defines $Z = X_1 \cup X_2$ and $Y = X_1 \cap X_2$ and the set of Z-vectors $U = \{(a_1, b, a_2) : (a_1, b) \in V_1, (b, a_2) \in V_2\}$. Here a_i is an $(X_i \backslash Y)$-vector and b is a Y-vector. We denote $(a_1, b, a_2) = (a_1, b) \circ (b, a_2)$ and say that (a_1, b, a_2) is the gluing of (a_1, b) and (b, a_2). To glue (X_1, V_1) and (X_2, V_2) one can sort V_1 or V_2 by Y-subvectors and only glues vectors with the same Y-subvector. So the complexity of the gluing is $O(|U| + (|V_1| + |V_2|) \log(|V_i|))$ operations. We use a simpler bound $O(|V_1||V_2| + |V_1| + |V_2|)$ in what follows. Denote $(Z, U) = (X_1, V_1) \circ (X_2, V_2)$.

```
Gluing Algorithm
input: the system (1) represented by symbols $(X_i, V_i)$, where $1 \le i \le m$.
output: the set $U$ of all solutions to (1) in variables $X(m) = X_1 \cup \ldots \cup X_m$.
put $(Z, U) \leftarrow (X_1, V_1)$ and $k \leftarrow 2$,
while $k \le m$ do $(Z, U) \leftarrow (Z, U) \circ (X_k, V_k)$ and $k \leftarrow k + 1$,
return $(Z, U)$.
```

The set U is all solutions to (1) in variables $X(m)$. The Gluing Algorithm takes $O(\sum_{k=1}^{m-1} |U_k| + m)$ operations with F_q-vectors of length at most n, where q and l are fixed, and n or m may grow. The memory requirement is of the same magnitude. Here $(X(k), U_k) = (X_1, V_1) \circ \ldots \circ (X_k, V_k)$. The set U_k consists of all solutions to the first k equations in variables $X(k) = X_1 \cup \ldots \cup X_k$. The sequence of $|U_k|$ fully characterizes the running time of the algorithm. The asymptotical analysis of $|U_k|$ is done in [7] using Random Allocations Theory results found in [4, 3, 1]. Two technical statements from [7] are formulated here.

Lemma 1 *(Lemma 4 in [7]) Let the subsets of variables $X_1, \ldots, X_k$ be fixed while $f_1, \ldots, f_k$ are randomly chosen according to our model. Then the expected number of solutions to the first k equations in (1) is $E_{f_1,\ldots,f_k}|U_k| = q^{|X(k)|-k}$.*

Lemma 2 *(Lemma 5 in [7]) Let $L_k = l_1 + \ldots + l_k$ and $\alpha = L_k/n$, and $k \le n$. Let $0 < \delta < 1$ be fixed as n tends to ∞. Then $E|U_k|$, the expected number of solutions to the first k equations, is $< q^{n^\delta}$, if $L_k < n^\delta$, and $O((qe^{g(\alpha)} + \epsilon)^n)$ otherwise for any positive real number ϵ. Here $g(\alpha) = f(z_\alpha) - \alpha + \alpha \ln \alpha - \frac{\alpha \ln q}{l}$ and $f(z) = \ln(e^z + q^{-1} - 1) - \alpha \ln(z)$, where by z_α we denote the only positive root of the equation $\frac{\partial f}{\partial z}(z) = 0$.*

3 Agreeing procedure and Agreeing-Gluing Algorithm

For symbols (X_i, V_i) for $i = 1, 2$, one defines $Y = X_1 \cap X_2$. Let $V_{1,2}(V_{2,1})$ be the set of Y-subvectors of $V_1(V_2)$. We say the symbols (X_1, V_1) and (X_2, V_2) agree if $V_{1,2} = V_{2,1}$. Otherwise, we apply the procedure called agreeing. We delete from V_i all vectors whose Y-subvectors are not in $V_{2,1} \cap V_{1,2}$. So new symbols (X_i, V_i') are determined, where $V_i' \subseteq V_i$ consist of the vectors in V_i survived after agreeing. To agree (X_1, V_1) and (X_2, V_2) one sorts V_1 or V_2 by Y-subvectors and do agreeing by table look ups. So the complexity of the agreeing is at most$O((|V_1|+|V_2|)\log(|V_i|))$ operations. The following Agreeing-Gluing Algorithm combines the Agreeing and Gluing procedures to solve (1).

Agreeing-Gluing Algorithm

```
  input: the system (1) represented by symbols $(X_i, V_i)$, where $1 \le i \le m$.
  output: the set $U$ of all solutions to (1) in variables $X(m) = X_1 \cup \ldots \cup X_m$.
  put $(Z, U) \leftarrow (X_1, V_1)$ and $k \leftarrow 2$,
  while $k \le m$ do $s \leftarrow k$,
      while $s \le m$ agree $(Z, U)$ and $(X_s, V_s)$, put $s \leftarrow s + 1$,
  put $(Z, U) \leftarrow (Z, U) \circ (X_k, V_k)$ and $k \leftarrow k + 1$,
  return $(Z, U)$.
```

Assume $(X(0), U_0')$ trivial. For any $0 \le k < m$ let $(X(k+1), U_{k+1}')$ denote the symbol $(X(k), U_k') \circ (X_{k+1}, V_{k+1})$ after agreeing with $(m-k-1)$ symbols (X_i, V_i), where $k+1 < i \le m$. The Agreeing-Gluing Algorithm produces the sequence of $(X(k), U_k')$ and takes

$$O(m(\sum_{k=1}^{m-1} |U_k'| + 1)) \tag{2}$$

operations with F_q-vectors of length at most n, where q and l are fixed, and n or m may grow. (2) incorporates the cost of the gluing $(X(k), U_k') \circ (X_{k+1}, V_{k+1})$, which is $O(|U_k'|)$ operations, and the agreeing the resulting set of $X(k+1)$-vectors, of size at most $O(|U_k'|)$, with the rest $m-k-1$ symbols. In our setting $|U_k'|$ is a random variable. We estimate the expectation of $|U_k'|$ in Section 4, see Theorem 2. That will imply Theorem 1. From the definition of Gluing and Agreeing procedures we get:

Lemma 3 *$(X(k), U_k')$ is the symbol $(X(k), U_k) = (X_1, V_1) \circ \ldots \circ (X_k, V_k)$ after agreeing with $(m-k)$ symbols (X_i, V_i), where $k < i \le m$.*

The space requirement of the Algorithm is as its running time. The Agreeing-Gluing1 Algorithm, similar to the Gluing1 Algorithm of [7], requires polynomial memory with the same running time. We do not go into detail here.

4 Complexity analysis of the Agreeing-Gluing Algorithm

We prove Theorem 1. Let $Z, X_1, \dots, X_k$ be fixed subsets of variables and U be a fixed set of Z-vectors, so that (Z,U) is defined by an equation $f(Z)=0$. Let V_i be the set of X_i-vectors, solutions to independent equations $f_i(X_i)=0$ generated uniformly at random.

Lemma 4 *Let (Z,U') be produced from (Z,U) by agreeing with all (X_i,V_i). Then the expectation of $|U'|$ is given by $E_{f_1,\dots,f_k}|U'| = |U|\prod_{i=1}^k(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}})$, where $|X_i \setminus Z|$ stands for the number of variables X_i not occurring in Z.*

Proof. Assume $k=1$. Let $Y_1 = Z\cap X_1$ and $|U| = \sum_a |U_a|$, where U_a is the subset of U-vectors whose projection to variables Y_1 is a. Similarly, $V_{1,a}$ is the subset of V_1-vectors whose projection to variables Y_1 is a. Then $|U'| = \sum_a |U_a| I_a$, where $I_a = 1$ for $V_{1,a}\neq\emptyset$ and $I_a = 0$ for $V_{1,a}=\emptyset$. Let W_a be the subset of all vectors in variables X_1 whose projection to variables Y_1 is a. We see that $|W_a| = q^{|X_1\setminus Y_1|}$. One computes $Pr(V_{1,a}=\emptyset) = Pr(f_1\neq 0 \,\mathrm{on}\, W_a) = (1-\frac{1}{q})^{q^{|X_1\setminus Y_1|}}$. So $E_{f_1}(I_a) = 1-(1-\frac{1}{q})^{q^{|X_1\setminus Y_1|}} = 1-(1-\frac{1}{q})^{q^{|X_1\setminus Z|}}$. Then $E_{f_1}|U'| = \sum_a |U_a| E_{f_1}(I_a) = |U|(1-(1-\frac{1}{q})^{q^{|X_1\setminus Z|}})$. This proves the statement for $k=1$. The Lemma is now shown true by induction.

Corollary 1 *Let f be generated independently to f_i. Then $E_{f,f_1,\dots,f_k}|U'| = E_f|U|\prod_{i=1}^k(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}})$.*

We will use the Corollary in order to estimate the expectation of $|U'_k|$.

Lemma 5 *Let $0\le\beta\le 1$ be any number. Then*

$$E|U'_k| \le q^{\beta n-k} + \sum_{|Z|>\beta n} Pr(X(k)=Z)\, q^{|Z|-k}\prod_{i=k+1}^{m} E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}}), \quad (3)$$

where Z runs over all subsets of X of size $>\beta n$.

Proof. For fixed X_i and random f_i, and by Lemma 3 and Corollary 1 we have

$$E_{f_1,\dots,f_m}|U'_k| = q^{|X(k)|-k}\prod_{i=k+1}^{m}(1-(1-\frac{1}{q})^{q^{|X_i\setminus X(k)|}}), \quad (4)$$

as $E_{f_1,\dots,f_k}|U_k| = q^{|X(k)|-k}$ by Lemma 2. Let We study the expectation of $|U'_k|$ when X_i are random too. So

$$E|U'_k| = \sum_{Z\subseteq X} Pr(X(k)=Z)\, q^{|Z|-k}\prod_{i=k+1}^{m} E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i\setminus Z|}})$$

We partition the last sum for $|Z| \leq \beta n$ and $|Z| > \beta n$, and get the statement. In next three Lemmas(without proof here) we estimate the expectation

$$E_{X_i}(1-(1-\frac{1}{q})^{q^{|X_i \setminus Z|}}). \tag{5}$$

Lemma 6 *Let $Z \subseteq X$ be a fixed subset of variables. Then (5) only depends on the size of Z and doesn't depend on the set itself. The expectation is not decreasing as $|Z|$ is decreasing or $|X_i|$ is increasing.*

Lemma 7 *Let Z be a fixed u-subset of X and X_i be an l_i-subset of X taken uniformly at random. Then $Pr(|X_i \setminus Z| = t) \doteq \frac{\binom{u}{l_i-t}\binom{n-u}{t}}{\binom{n}{l_i}}$.*

Lemma 8 *1. Let $|Z| > \beta n$, where $0 \leq \beta \leq 1$ is fixed as n tends to ∞, then (5) is bounded by $F(\beta) + O(\frac{1}{n})$, where $O(\frac{1}{n})$ doesn't depend on i.*

2. The function $F(\beta) = 1 - \sum_{t=0}^{l} \binom{l}{t} \beta^{l-t}(1-\beta)^t(1-\frac{1}{q})^{q^t}$ is not increasing in $0 \leq \beta \leq 1$ and $\frac{1}{q} \leq F(\beta) \leq 1-(1-\frac{1}{q})^{q^l} < 1$.

The inequality (3) then implies

$$E|U_k'| \leq q^{\beta n - k} + E_{X_1,\ldots,X_k}(q^{|X(k)|-k})\,(F(\beta)+\varepsilon)^{m-k}. \tag{6}$$

for any positive real ε as n tends to ∞. For $0 \leq \alpha \leq l$ we define the function $0 \leq \beta(\alpha) \leq 1$ by the rule: $\beta = \beta(\alpha)$ is the solution of the equation

$$q^{\beta - \frac{\alpha}{l}} = qe^{g(\alpha)} F(\beta)^{c - \frac{\alpha}{l}} \tag{7}$$

if such a solution exists and $\beta(\alpha) = 0$ otherwise. We know that $c_n = \frac{l_1+l_2+\ldots+l_m}{ln}$ tends to a constant $c \geq 1$ as n tends to ∞ while q and l are fixed.

Theorem 2 *1. The equation (7) has at most one solution for any $0 \leq \alpha \leq l$.*

2. Let $L_k = l_1 + \ldots + l_k$ and $\alpha = L_k/n$, and $k \leq n$. Let $0 < \delta < 1$ be fixed as n tends to ∞. Then

$$E|U_k'| = \begin{cases} < q^{n^\delta}, & if \quad L_k < n^\delta; \\ O((q^{\beta(\alpha)-\frac{\alpha}{l}} + \varepsilon)^n), & if \quad ln > L_k \geq n^\delta; \\ < 1, & if \quad L_k \geq ln, \end{cases}$$

for any positive real ε.

Proof. We prove the second statement here. It is true for $L_k < n^\delta$ and $L_k \geq ln$. Let $ln > L_k \geq n^\delta$. Then by Lemma 2 we get from (6) that

$$E|U_k'| \leq (q^{\beta-\frac{\alpha}{l}})^n + O((qe^{g(\alpha)}+\varepsilon)^n(F(\beta)+\varepsilon)^{\frac{m-k}{n}n}),$$

as $\frac{\alpha}{l} \leq \frac{k}{n}$ and for any positive ε. We realize that $\frac{m-k}{n} \geq c_n - \frac{\alpha}{l}$, so

$$E|U_k'| \leq (q^{\beta-\frac{\alpha}{l}})^n + O((qe^{g(\alpha)}F(\beta)^{c-\frac{\alpha}{l}}+\varepsilon)^n) \tag{8}$$

for any real positive ε as n tends to ∞. If (7) has one solution, then the inequality $E|U_k'| = O((q^{\beta(\alpha)-\frac{\alpha}{l}}+\varepsilon)^n)$ follows from (8) and (7). When (7) has no solutions, the statement is easy. The Theorem is proved.

The main Theorem 1 now follows from Theorem 2 and formula (2).

References

[1] V. P. Chistyakov, Discrete limit distributions in the problem of shots with arbitrary probabilities of occupancy of boxes, *Mat. Zametki* 1, 1967, 9-16.

[2] K. Iwama, Worst-case upper bounds for kSAT, *The Bull. EATCS* 82, 2004, 61-71.

[3] V. Kolchin, The rate of convergence to limit distributions in the classical problem of shots, *Teoriya veroyatn. i yeye primenen.*, 11, 1966, 144-156.

[4] V. Kolchin, A. Sevast'yanov, V. Chistyakov, *Random allocations*, John Wiley & Sons, 1978.

[5] H. Raddum, Solving non-linear sparse equation systems over $GF(2)$ using graphs, Univ. Bergen, preprint, 2004.

[6] H. Raddum, I. Semaev, New technique for solving sparse equation systems, Cryptology ePrint Archive, 2006/475.

[7] I. Semaev, On solving sparse algebraic equations over finite fields, to appear in *Des., Codes Crypt.*, extended abstract in Proc. WCC'07, Versailles, France, INRIA, 361-370.

[8] I. Semaev, On solving sparse algebraic equations over finite fields II, Cryptology ePrint Archive: Report 2007/280.

[9] B.-Y. Yang, J-M. Chen, N. Courtois, On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis, ICICS 2004, *Lect. Notes Comp. Sci.* 3269, Springer-Verlag, 2004, 401-413.

[10] A. Zakrevskij, I. Vasilkova, Reducing large systems of Boolean equations, *4th Intern. Workshop Bool. Probl.*, Freiberg Univ., 2000.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 255-259*

Relation between two classes of binary quasi-cyclic Goppa codes

Sergey Bezzateev bsv@aanet.ru
Natalia Shekhunova sna@delfa.net
Saint Petersburg State University of Airspace Instrumentation
Saint-Petersburg, RUSSIA

Abstract. Two classes of binary quasi-cyclic Goppa codes is considered. True parameters and codeword structure of these codes is proposed.

1 Inroduction

Let us consider the relation between two classes of quasi-cyclic Goppa codes $\Gamma(L, G(x))$ and $\Gamma^*(L^*, G^*(x))$, where

$$G(x) = x^{t-1} + 1, \tag{1}$$

$$G^*(x) = x^{t+1} + 1, \tag{2}$$

$t = 2^l, L \subset GF(2^{2l}), L^* \subset GF(2^{2l})$.

In [1], [2] the true values of parameters for these codes have been obtained. The code $\Gamma(L, G(x))$ has the minimal distance

$$d = 2t - 1 \tag{3}$$

and the number of information symbols is

$$k = t^2 - t - 2l(t - \frac{3}{2}). \tag{4}$$

The code $\Gamma^*(L^*, G^*(x))$ has the minimal distance

$$d^* = 2t + 3 \tag{5}$$

and the number of information symbols is

$$k^* = t^2 - t - 2l(t - \frac{3}{2}) - 1. \tag{6}$$

In this paper we will examine the codeword structure of these classes of the codes and we will show how the codewords from one class $\Gamma(L, G(x))$ can be transformed into the codewords of another class $\Gamma^*(L^*, G^*(x))$.

2 Codeword structure of the binary $\Gamma(L, G(x))$code

It is easy to show that $\Gamma(L, G(x))$ code is the quasi-cyclic code with the length of cycloid $(t-1)$ and number of cycloids t. Moreover, the codewords of this code have one fixed position - $\{0\}$. Therefore the total length of the code is

$$n = t(t-1)+1 \tag{7}$$

The numerators of the codewords of the $\Gamma(L, G(x))$ code can be represented in the following form:

$$L = \{\beta^i, \beta^i\alpha^{t+1}, \beta^i\alpha^{(t+1)2}, ..., \beta^i\alpha^{(t+1)(t-2)}\}_{i=1,...,t} \bigcup \{0\}, \tag{8}$$

where $\beta = \alpha^{2^l-1} = \alpha^{t-1}$, α is the primitive element of $GF(2^{2l})$, and $\{\beta^i, \beta^i\alpha^{t+1}, \beta^i\alpha^{(t+1)2}, ..., \beta^i\alpha^{(t+1)(t-2)}\}$ are numerators of positions that form the correspondent cycloids.

By using the representation of the set L as (8) it is possible to write the parity check matrix H of the code in the following form:

$$H = \left[\begin{bmatrix} \frac{1}{\beta^{i(t-1)}+1} & \frac{1}{\beta^{i(t-1)}+1} & \cdots & \frac{1}{\beta^{i(t-1)}+1} \\ \frac{\beta^i}{\beta^{i(t-1)}+1} & \frac{\beta^i\alpha^{t+1}}{\beta^{i(t-1)}+1} & \cdots & \frac{\beta^i\alpha^{(t+1)(t-2)}}{\beta^{i(t-1)}+1} \\ . & . & \cdots & . \\ \frac{\beta^{i(t-2)}}{\beta^{i(t-1)}+1} & \frac{\beta^{i(t-2)}\alpha^{(t+1)(t-2)}}{\beta^{i(t-1)}+1} & & \frac{\beta^{i(t-2)}\alpha^{(t+1)(t-2)(t-2)}}{\beta^{i(t-1)}+1} \\ 1 & 1 & & 1 \end{bmatrix}_{i=1,...,t} \begin{bmatrix} 1 \\ 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \right] \tag{9}$$

It follows from representation (9) that in any code from the $\Gamma(L, G(x))$ code class only the codewords that have 1 on position $\{0\}$ will be the codewords with the minimal weight $d = 2t-1$. The codewords with 0 on this position have an even weight and it will be shown that the minimal weight of such codewords is equal to $2t+4$.

3 Transformation of the codewords from the class $\Gamma(L, G(x))$ into codewords of the class $\Gamma^*(L^*, G^*(x))$

Let us consider now $\Gamma_1(L_1, G(x))$ code obtained as truncated $\Gamma(L, G(x))$ code by information position $\{0\}$, i.e., we remove all codewords with 1 on position $\{0\}$ from $\Gamma(L, G(x))$ code. Then $L_1 = L \backslash \{0\}$ and $\Gamma_1(L_1, G(x))$ code is still quasi-cyclic code with parity check matrix

$$H_1 = \begin{bmatrix} \frac{1}{\beta^{i(t-1)}+1} & \frac{1}{\beta^{i(t-1)}+1} & \cdots & \frac{1}{\beta^{i(t-1)}+1} \\ \frac{\beta^i}{\beta^{i(t-1)}+1} & \frac{\beta^i\alpha^{t+1}}{\beta^{i(t-1)}+1} & \cdots & \frac{\beta^i\alpha^{(t+1)(t-2)}}{\beta^{i(t-1)}+1} \\ . & . & \cdots & . \\ \frac{\beta^{i(t-2)}}{\beta^{i(t-1)}+1} & \frac{\beta^{i(t-2)}\alpha^{(t+1)(t-2)}}{\beta^{i(t-1)}+1} & \cdots & \frac{\beta^{i(t-2)}\alpha^{(t+1)(t-2)(t-2)}}{\beta^{i(t-1)}+1} \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{i=1,\dots,t}$$

Lemma 1 *The rows* $\left[\frac{\beta^{i(t-1)}}{\beta^{i(t-1)}+1} \ \frac{\beta^{i(t-1)}}{\beta^{i(t-1)}+1} \ \cdots \ \frac{\beta^{i(t-1)}}{\beta^{i(t-1)}+1} \right]_{i=1,\dots,t}$ *and* $\left[\frac{1}{\beta^i(\beta^{i(t-1)}+1)} \ \frac{1}{\beta^i\alpha^{t+1}(\beta^{i(t-1)}+1)} \ \cdots \ \frac{1}{\beta^i\alpha^{(t+1)(t-2)}(\beta^{i(t-1)}+1)} \right]_{i=1,\dots,t}$ *can be represented as a linear combination of the correponding rows of the parity check matrix* H_1.

From Lemma 1 we obtain that the matrix H_1 can be rewritten in the following form:

$$H_1 = \begin{bmatrix} \frac{1}{\beta^i(\beta^{i(t-1)}+1)} & \frac{1}{\beta^i\alpha^{t+1}(\beta^{i(t-1)}+1)} & \cdots & \frac{1}{\beta^i\alpha^{(t+1)(t-2)}(\beta^{i(t-1)}+1)} \\ \frac{1}{\beta^{i(t-1)}+1} & \frac{1}{\beta^{i(t-1)}+1} & \cdots & \frac{1}{\beta^{i(t-1)}+1} \\ \frac{\beta^i}{\beta^{i(t-1)}+1} & \frac{\beta^i\alpha^{t+1}}{\beta^{i(t-1)}+1} & \cdots & \frac{\beta^i\alpha^{(t+1)(t-2)}}{\beta^{i(t-1)}+1} \\ . & . & \cdots & . \\ \frac{\beta^{i(t-2)}}{\beta^{i(t-1)}+1} & \frac{\beta^{i(t-2)}\alpha^{(t+1)(t-2)}}{\beta^{i(t-1)}+1} & \cdots & \frac{\beta^{i(t-2)}\alpha^{(t+1)(t-2)(t-2)}}{\beta^{i(t-1)}+1} \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{i=1,\dots,t}$$

Obviously that this matrix is parity check matrix for the code $\Gamma_2(L_2, G_2(x))$ where $G_2(x) = x^t + x$, $L_2 = L_1$. This code is still quasi-cyclic with length of cycloid $t-1$ and the number of cycloids is t, i.e., $n_2 = t(t-1)$.

Theorem 1 *The minimal distance of* $\Gamma_2(L_2, G_2(x))$ *code is* $d_2 = 2t+4$ *and number of information symbols is* $k_2 = k_1 - 1$.

Lemma 2 $L_2 = \{GF(2^{2l})\} \backslash \{\{\alpha^{(t+1)i}, i = 0, ..., t-2\} \bigcup \{0\}\}$.

Let us consider now the following substitution: $x \longrightarrow z + \gamma$, where $\gamma \in GF(2^{2l})$ and $\gamma^t + \gamma + 1 = 0$. Then $G_2(x) = x^t + x = z^t + \gamma^t + z + \gamma = z^t + z + 1 = G_3(x)$.

Now, to proceed from the class $\Gamma(L, G(x))$ in to the class $\Gamma^*(L^*, G^*(x))$ let us prove the following statement.

Lemma 3 *There exist* t *different elements* $\gamma \in GF(2^{2l})$ *such that* $\gamma^t + \gamma + 1 = 0$ *where* $t = 2^l$.

Proof. Let us choose some element $\varpi_j \in GF(2^{2l})$ and let $\varpi_j^t + \varpi_j + 1 = \tau \neq 0$, then obviously, that $\tau \in GF(2^l)$. Indeed $\tau^{2l} = \varpi_j^{t2^l} + \varpi_j^{2^l} + 1 = \varpi_j^t + \varpi_j + 1 = \tau$. Therefore $\tau^{2l} = \tau$ and $\tau \in GF(2^l)$. It is easy to show that for any nonzero element τ there exists t different values ϖ_j such that $\varpi_j^t + \varpi_j + 1 = \tau$. Then, as the number of nonzero elements τ from $GF(2^l)$ is $2^l - 1$, we will have $N = (2^l - 1)t$ elements $\varpi_j \in GF(2^{2l})$ such that $\varpi_j^t + \varpi_j + 1 \neq 0$. $N = (2^l - 1)2^l = 2^{2l} - 2^l$.

Therefore in the field $GF(2^{2l})$ $\theta = 2^l$ elements ϖ_j such that $\varpi_j^t + \varpi_j + 1 = 0$ can be found. □

If we will choose one of these ϖ_j as γ then $\gamma^t + \gamma + 1 = 0$. It is easy to show that L_3 can be represented as:

$$L_3 = \{\beta^i + \gamma, \beta^i \alpha^{t+1} + \gamma, \beta^i \alpha^{(t+1)2} + \gamma, ..., \beta^i \alpha^{(t+1)(t-2)} + \gamma\}_{i=1,...,t}$$

Moreover, as $\gamma : G_2(\gamma) = 1$, i.e., element γ is not a root of the $G_2(\gamma)$, then accoding to the Lemma 2 there exist i, j such that :

$$\beta^i \alpha^{(t+1)j} = \gamma,$$

This means that in the set L_3 we have one cycloid with element $\{0\}$. In the set L_3 it is also exist element $\{1\}$, as $G_3(1) \neq 0$.

Obviously, the code $\Gamma_3(L_3, G_3(x))$ has parameters

$$n_3 = t(t-1),$$
$$k_3 = k_2 = k_1 - 1 \text{ and}$$
$$d_3 = 2t + 4.$$

Let us consider now $\Gamma_3^*(L_3^*, G_3(x))-$code obtained from $\Gamma_3(L_3^*, G_3(x))$-code by trancation on position $\{0\}$, i.e., $L_3^* = L_3 \backslash \{0\}$.

The code $\Gamma_3^*(L_3^*, G_3(x))$ has parameters:

$$n_3^* = n_3 - 1, \quad k_3^* = k_3 = k_2 = k_1 - 1, \quad d_3^* = d_3 - 1 = d_2 - 1 = 2t + 3.$$

Now let us use the following substitution: $z \longrightarrow \frac{1}{y}$. Then

$$G_3(z) = z^t + z + 1 = y^{-t} + y^{-1} + 1 \longrightarrow G_4(x) = y^t + y^{t-1} + 1.$$

The set L_4^* can be defined as a set of elements of $GF(2^{2l})$ that are inverse by multiplication to the elements of set L_3^*.

$$L_4^* = \{(\beta^i + \gamma)^{-1}, (\beta^i \alpha^{t+1} + \gamma)^{-1}, (\beta^i \alpha^{(t+1)2} + \gamma)^{-1}, ..., (\beta^i \alpha^{(t+1)(t-2)} + \gamma)^{-1}\}_{i=1,...,t}.$$

Code $\Gamma_4^*(L_4^*, G_4(x))$ has parameters

$$n_4^* = n_3^* = n_3 - 1,$$
$$k_4^* = k_3^* \text{ and}$$
$$d_4^* = d_3 - 1.$$

Lemma 4 *Code* $\Gamma_4^*(L_4^*, G_4(x)) \equiv \Gamma_5^*(L_5^*, G_5(x))$, *where* $G_5(y) = yG_4(y) = y^{t+1} + y^t + y$ *and* $L_5^* = L_4^*$.

Let us use the following substitution: $y \longrightarrow u+1$, then

$$G_5(y) = y^{t+1} + y^t + y \longrightarrow (u+1)^{t+1} + (u+1)^t + u = u^{t+1} + 1 = G_6(y).$$

$$L_6 = \{(\beta^i+\gamma)^{-1}+1, (\beta^i\alpha^{t+1}+\gamma)^{-1}+1, (\beta^i\alpha^{(t+1)2}+\gamma)^{-1}+1, \ldots, (\beta^i\alpha^{(t+1)(t-2)}+\gamma)^{-1}+1\}_{i=1,\ldots,t}$$

From Lemma 2 and the above obtained result about the existence of the element $\{1\}$ in the set L_3 it is obvious that the element $\{0\}$ will appear in set L_6.

Theorem 2 *The class of binary* $\Gamma_6(L_6, G_6(x))$ *codes is the class of binary quasi-cyclic* $\Gamma^*(L^*, G^*(x))$ *codes with Goppa polynomial defined by formula (2) and locator set* $L^* = L_6$.

Any codeword of this code is formed by $(t-2)$ cycloids of the length $t+1$ and one fixed position $\{0\}$.

$\Gamma^*(L^*, G^*(x))$ codes have the following parameters:

$$\begin{aligned} n^* = n_6 = n_5 = n_4^* = n_3 - 1 = t(t-1) - 1, \\ k^* = k_6 = k_5 = k_4^* = k_3^* = k - 1, \\ d^* = d_6 = d_5 = d_3^* = d_3 - 1 = 2t + 3. \end{aligned} \tag{10}$$

Let us write for the sequence of the accomplished transformations: $x \rightarrow z+\gamma \rightarrow \frac{1}{y}+\gamma \rightarrow \frac{1}{u+1}+\gamma$. Therefore $u = \frac{1}{x+\gamma}+1 = (x+\gamma)^{-1}+1$.

4 Conclusion

As it was shown above the codewords from the class of the binary quasi- cyclic $\Gamma_1(L_1, G_1(x))$-codes with cycloid length $(t-1)$ and cycloid number t and the fixed position $\{0\}$ can transformed into the class of the binary quasi-cyclic $\Gamma^*(L^*, G^*(x))$-codes with the cycloid length $(t+1)$ and cycloid number $(t-1)$ and fixed position $\{0\}$ by the sequence of simple transformations. The true values for parameters of these codes are defined by formulas (3), (4), (7) and (5), (6), (10) respectively.

References

[1] S. Bezzateev, N. Shekhunova, Subclass of binary Goppa codes with minimal distance equal to the design distance, *IEEE Trans. Inform. Theory* 41, 1995, 554-555.

[2] P. Veron, True dimension of some binary quadratic trace Goppa codes, *Des., Codes Crypt.* 24, 2001, 81-97.

Single-trial adaptive decoding of concatenated codes

Vladimir Sidorenko[1] vladimir.sidorenko@uni-ulm.de
Christian Senger[2] christian.senger@uni-ulm.de
Martin Bossert martin.bossert@uni-ulm.de
TAIT, Ulm University, Ulm, Germany
Victor Zyablov zyablov@iitp.ru
IITP, Russian Academy of Sciences, Moscow, Russia

Abstract. In this paper decoding of a concatenated code is considered. We use a Bounded Minimum Distance (BMD) decoder for the inner code correcting up to $(d^{\mathrm{i}}-1)/2$ errors and a Bounded Distance (BD) decoder for the outer code, which corrects ε errors and τ erasures if $\lambda\varepsilon+\tau \le d^{\mathrm{o}}-1$, where a real number $1<\lambda\le 2$ is the tradeoff rate between errors and erasures for this outer decoder. Here d^{o} and d^{i} are the minimum distances of the outer and the inner code, respectively. We consider a single-trial outer decoder, which extends Kovalev's approach [1] for the whole given range of λ. The error correcting radius of the suggested concatenated decoder is $\frac{d^{\mathrm{i}}d^{\mathrm{o}}}{2}\left(1-\left(\frac{\lambda-1}{\lambda}\right)^2\right)$. When using an outer Reed–Solomon code over $_{q^{\ell m}}$ of length $n^{\mathrm{o}} \le q^m$ with the BD decoder suggested in [2], $\lambda=\frac{\ell+1}{\ell}$, and the error correcting radius $\frac{d^{\mathrm{i}}d^{\mathrm{o}}}{2}\left(1-\frac{1}{(1-\ell)^2}\right)$ of the concatenated decoder quickly approaches $d^{\mathrm{i}}d^{\mathrm{o}}/2$ with increasing ℓ.

1 Introduction

Concatenated codes were suggested and investigated by Forney in 1966 [3]. A simple concatenated coding scheme uses an outer block code $\mathcal{C}^{\mathrm{o}}(n^{\mathrm{o}},k^{\mathrm{o}},d^{\mathrm{o}})$ over the finite field $_{q^{k^{\mathrm{i}}}}$ and an inner block code $\mathcal{C}^{\mathrm{i}}(n^{\mathrm{i}},k^{\mathrm{i}},d^{\mathrm{i}})$ over $_q$, where the upper indices o and i stand for the outer and the inner code, respectively. The information sequence composed of k^{o} $q^{k^{\mathrm{i}}}$-ary symbols is first encoded using the outer code into a $q^{k^{\mathrm{i}}}$-ary codeword $\mathbf{c}^{\mathrm{o}}=(c_1^{\mathrm{o}},\dots,c_{n^{\mathrm{o}}}^{\mathrm{o}})$. The inner code is then used to encode each symbol c_j^{o}, $j=1,\dots,n^{\mathrm{o}}$, into a q-ary column vector $\mathbf{c}^{\mathrm{i},T}=(c_1^{\mathrm{i}},\dots,c_{n^{\mathrm{i}}}^{\mathrm{i}})^T$. This results in an $n^{\mathrm{i}}\times n^{\mathrm{o}}$ matrix C of q-ary symbols, which is a codeword of the concatenated code $\mathcal{C}$. The code matrix C is transmitted over a q-ary channel and may suffer from channel errors. Denote by R the received matrix and by e the number of errors in the channel.

[1] The work of Vladimir Sidorenko is supported by DFG project BO 867/17. He is on leave from Institute for Information Transmission Problems, RAS, Moscow, Russia.

[2] The work of Christian Senger is supported DFG project BO 867/15.

The decoder of the concatenated code $\mathcal{C}$ consists of an inner decoder and an outer decoder. The inner decoder decodes each column $\mathbf{r}_j^{\mathrm{i},T}$, $j = 1, \ldots, n^{\mathrm{o}}$, of the received matrix R using a BMD decoder for $\mathcal{C}^{\mathrm{i}}$ correcting up to $(d^{\mathrm{i}} - 1)/2$ errors and producing either a codeword $\tilde{\mathbf{c}}_j^{\mathrm{i},T}$ or indicating a decoding failure. In case of successful decoding, the correspondent $q^{k^{\mathrm{i}}}$-ary symbol $\tilde{c}_j^{\mathrm{o}}$ is given an unreliability $\Delta_j = \mathrm{d_H}(\tilde{\mathbf{c}}_j^{\mathrm{i}}, \mathbf{r}_j^{\mathrm{i}})$ and both $\tilde{c}_j^{\mathrm{o}}$ and Δ_j are delivered to the outer decoder. Here, $\mathrm{d_H}(\cdot,\cdot)$ denotes the Hamming distance. In case of an inner decoding failure, the symbol $\tilde{c}_j^{\mathrm{o}}$ is considered to be erased which implies the greatest possible unreliability $\Delta_j = d^{\mathrm{i}}/2$.

The inner decoder provides the $q^{k^{\mathrm{i}}}$-ary vector $\tilde{\mathbf{c}}^{\mathrm{o}} = (\tilde{c}_1^{\mathrm{o}}, \ldots, \tilde{c}_{n^{\mathrm{o}}}^{\mathrm{o}})$ to the outer decoder, where potentially some of the symbols are erased, i.e. replaced by a special erasure symbol. We denote $\tilde{\mathbf{c}}^{\mathrm{o}} \triangleq \mathbf{r}^{\mathrm{o}} = (r_1^{\mathrm{o}}, \ldots, r_{n^{\mathrm{o}}}^{\mathrm{o}})$ to indicate that this is the received vector from point of view of the the outer decoder. In addition to $\mathbf{r}^{\mathrm{o}}$ the outer decoder is provided by the vector $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_{n^{\mathrm{o}}})$ of unreliabilities. The outer decoder should decode the received vector $\mathbf{r}^{\mathrm{o}}$ using the unreliabilities $\boldsymbol{\Delta}$, i.e. it should reconstruct the transmitted codeword $\mathbf{c}^{\mathrm{o}}$ of the outer code and the corresponding information sequence. This decoding problem is also known as Generalized Minimum Distance (GMD) decoding. *Our aim is to optimize the outer decoder if it is restricted to use the decoder of the outer code only once.*

Let us first assume an outer BMD decoder. It corrects ε errors and τ erasures if $2\varepsilon + \tau \leq d^{\mathrm{o}} - 1$, where the factor 2 can be considered as the *tradeoff rate* between errors and erasures for a BMD decoder. If the BMD decoder simply decodes the vector $\mathbf{r}^{\mathrm{o}}$ without using $\boldsymbol{\Delta}$ we can guarantee correction up to $e < d^{\mathrm{o}}d^{\mathrm{i}}/4$ channel errors, where $d^{\mathrm{o}}d^{\mathrm{i}}$ is the designed distance of the concatenated code. This fact was shown by Forney [3]. Forney also suggested multi-trial outer decoding, where in each trial a number of least reliable symbols of $\mathbf{r}^{\mathrm{o}}$ are erased and the obtained vector $\tilde{\mathbf{r}}^{\mathrm{o}}$ is decoded by the outer BMD decoder. This multi-trial decoding allows to correct up to $e < d^{\mathrm{o}}d^{\mathrm{i}}/2$ channel errors, if the number of trials is sufficiently large. However, in this paper we consider single-trial outer decoders only.

In 1973 Zyablov [4] suggested the following single-trial decoding: First, erase all symbols in $\mathbf{r}^{\mathrm{o}}$ whose unreliabilities exceed the fixed threshold $T = d^{\mathrm{i}}/3$. Then, decode the obtained vector $\tilde{\mathbf{r}}^{\mathrm{o}}$ with a BMD decoder for the outer code. This method allows to correct up to $e < d^{\mathrm{o}}d^{\mathrm{i}}/3$ channel errors. In 1986, Kovalev [1] proposed a single-trial decoding method, where the threshold T is not fixed, but is selected adaptively as a function of $\boldsymbol{\Delta}$. His algorithm is able to correct up to $e < 3d^{\mathrm{o}}d^{\mathrm{i}}/8$ channel errors. Some refinements of Kovalev's and Zyablov's approaches were done by Weber and Abdel-Ghaffar in [5]. We should also mention papers by Sorger [7], and Kötter [8] who suggested interesting modifications of a BMD decoder in such a way that multi-trial decoding of the outer code can be made "in one step".

In this paper we assume a BD decoder for the outer code, which corrects up to $(d^{\mathrm{o}}-1)/\lambda$ errors in the received vector $\mathbf{r}^{\mathrm{o}}$. More precisely, we assume that the BD decoder corrects ε errors and τ erasures if

$$\lambda\varepsilon + \tau \le d^{\mathrm{o}} - 1, \tag{1}$$

where the real number $1 < \lambda \le 2$ is the tradeoff rate between errors and erasures for the BD decoder.

For example, we can use for outer encoding a Reed–Solomon (RS) code $\mathcal{C}^{\mathrm{o}}(n^{\mathrm{o}}, k^{\mathrm{o}}, d^{\mathrm{o}})$ over the field $\;_{q^{\ell m}}$ of length $n^{\mathrm{o}} < q^m$ with locators taken from the field $\;_{q^m}$, where $m, \ell \in \{1, 2, \ldots\}$, $lm = k^{\mathrm{i}}$. This code can also be regarded as an ℓ-interleaved RS code [6]. In [2] an efficient algorithm is presented, which allows decoding of ε errors and τ erasures if $\varepsilon \le (d^{\mathrm{o}} - \tau - 1)\ell/(\ell+1)$, i.e. $\mathbf{r}^{\mathrm{o}}$ is decoded correctly if (1) is satisfied, where $\lambda = (\ell+1)/\ell$. The decoder from [2] may fail with probability $P_f(\varepsilon, \tau) \le \gamma q^{-m[(\ell+1)(\varepsilon_{\max}(\tau)-\varepsilon)+1]}$, where $\gamma \approx 1$ and $\varepsilon_{\max}(\tau) \triangleq (d^{\mathrm{o}} - \tau - 1)\ell/(\ell+1)$. If $P_f(\varepsilon, \tau)$ is not small enough we can make it negligibly small by slightly decreasing the error correcting radius [6].

Kovalev proposed an adaptive algorithm for $\lambda = 2$. In Section 2 we extend his algorithm for arbitrary λ, $1 < \lambda \le 2$. In Section 3 we estimate the error correction radius of this extended algorithm and show that the radius quickly approaches $d^{\mathrm{o}}d^{\mathrm{i}}/2$ when $\lambda \to 1$.

2 Single-trial adaptive decoding

From the inner decoder we have a received word $\mathbf{r}^{\mathrm{o}} = (r_1^{\mathrm{o}}, \ldots, r_{n^{\mathrm{o}}}^{\mathrm{o}})$ together with a vector $\mathbf{\Delta} = (\Delta_1, \ldots, \Delta_{n^{\mathrm{o}}})$ of unreliabilities for the components of $\mathbf{r}^{\mathrm{o}}$, where $0 \le \Delta_j \le d^{\mathrm{i}}/2$. Here, we assume w.l.o.g. that the components of $\mathbf{r}^{\mathrm{o}}$ are ordered according to their unreliabilities and hence $\Delta_1 \ge \Delta_2 \ge \cdots \ge \Delta_{n^{\mathrm{o}}}$.

The decoder of the outer code fails for $\mathbf{r}^{\mathrm{o}}$ with τ erasures and ε errors in unerased positions if

$$\lambda\varepsilon + \tau > d^{\mathrm{o}} - 1, \tag{2}$$

otherwise outer decoding will be correct (see assumption (1)). Given the number of erasures τ, we denote by $\varepsilon(\tau)$ the minimum number of (unerased) erroneous symbols in the input vector that guarantee to cause a decoding failure. From (2) we get

$$\varepsilon(\tau) = \left\lfloor \frac{d^{\mathrm{o}} - \tau - 1}{\lambda} \right\rfloor + 1. \tag{3}$$

Let us erase the first and thus least reliable τ components of $\mathbf{r}^{\mathrm{o}}$ and decode the obtained input word $\tilde{\mathbf{r}}^{\mathrm{o}}$ by a decoder for the outer code. The decoder will fail if there were at least $\varepsilon(\tau)$ (unerased) erroneous symbols. What is the minimum number $e_\tau(\mathbf{\Delta})$ of errors *in the channel* given the vector $\mathbf{\Delta}$ of unreliabilities to create $\varepsilon(\tau)$ (unerased) erroneous symbols in $\tilde{\mathbf{r}}^{\mathrm{o}}$?

To have an integer unreliability Δ_j the channel should spend Δ_j errors if inner decoding of the component r_j^{o} was correct, and at least $d^{\mathrm{i}} - \Delta_j$ errors otherwise. Consequently, the channel requires the minimum number of errors if the erroneous components r_j^{o} have minimum possible $d^{i} - \Delta_j$. This takes place when the $\varepsilon(\tau)$ erroneous components r_j^{o} are situated immediately after the τ erased (first) positions. We obtain

$$\begin{aligned} e_\tau(\mathbf{\Delta}) &= \sum_{j=1}^{\tau} \Delta_j + \sum_{j=\tau+1}^{\tau+\varepsilon(\tau)} (d^{\mathrm{i}} - \Delta_j) + \sum_{j=\tau+\varepsilon(\tau)+1}^{n^{\mathrm{o}}} \Delta_j \\ &= \sum_{j=1}^{n^{\mathrm{o}}} \Delta_j + \sum_{j=\tau+1}^{\tau+\varepsilon(\tau)} (d^{\mathrm{i}} - 2\Delta_j). \end{aligned} \tag{4}$$

Remark 1 *This is true for even d^{i}, in this case Δ_j is always integer since $\Delta_j \in \{0, \ldots, d^{\mathrm{i}}/2\}$. In case of odd d^{i} we can have non-integer $\Delta_j = d^{\mathrm{i}}/2$ and $e_\tau(\mathbf{\Delta})$ assumes a larger value then (4). Later on, we consider $e_\tau(\Delta)$ given by (4) only, despite the results can be slightly improved by methods similar to [5].*

Given $\mathbf{\Delta}$, if the number e of errors in the channel satisfies $e < e_\tau(\mathbf{\Delta})$ the decoding of $\mathcal{C}^{\mathrm{o}}$ with τ erasures will be successful. Hence, $e_\tau(\mathbf{\Delta})$ is an error correcting radius for given $\mathbf{\Delta}$ and τ. We are free to select $\tau \in \mathcal{T}$,

$$\mathcal{T} = \{0, \ldots, d^{\mathrm{o}} - 1\}. \tag{5}$$

Let us select $\tau = \tau^*$ which maximizes the radius $e_\tau(\mathbf{\Delta})$:

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} e_\tau(\mathbf{\Delta}). \tag{6}$$

As a result we obtain the following algorithm:

Algorithm A. Single-trial adaptive outer decoder

Input. Received vector $\mathbf{r}^{\mathrm{o}}$ with unreliability vector $\mathbf{\Delta}$ from the inner decoder. Code distances d^{i}, d^{o} and parameter $1 < \lambda \leq 2$.

Step 1. Find $\tau^* = \arg\max_{\tau \in \mathcal{T}} \sum_{j=\tau+1}^{\tau+\varepsilon(\tau)} (d^{\mathrm{i}} - 2\Delta_j)$, where $\varepsilon(\tau)$ is defined in (3).

Step 2. Decode $\mathbf{r}^{\mathrm{o}}$ with erased first τ^* positions by the BD decoder for the outer code.

Output. Either a codeword of the outer code or a decoding failure.

From Algorithm A we see that the complexity of the proposed adaptive decoder comprises the complexity of the decoder for the outer code and additionally the complexity of Step 1, which is upper bounded by $\mathcal{O}(d^{\mathrm{o}} \log d^{\mathrm{o}})$.

3 Error correcting radius

The goal of this section is to estimate the error correcting radius of Algorithm A with parameter λ. This means we should find the maximum (real) number $\rho(\lambda)$ such that any number $e < \rho(\lambda)$ of errors in the channel are guaranteed to be corrected by Algorithm A. For a given vector $\mathbf{\Delta}$ of unreliabilities the error correcting radius $\rho(\lambda)$ of Algorithm A is $e_\tau(\Delta)$, where τ^* is defined by (6) (see also Remark 1). The radius $\rho(\lambda)$ of Algorithm A for all possible $\mathbf{\Delta}$ can be found as the minimum of $e_\tau(\mathbf{\Delta})$ over all possible $\mathbf{\Delta}$ as follows:

$$\rho(\lambda) = \min_{\mathbf{\Delta}} \max_{\tau \in \mathcal{T}} e_\tau(\mathbf{\Delta}). \tag{7}$$

To simplify notations let us replace the unreliabilities Δ_j, $j = 1, \ldots, n^{\mathrm{o}}$, by real-valued reliabilities h_j as follows: $h_j = (d^{\mathrm{i}} - 2\Delta_j)/d^{\mathrm{i}}$, where

$$0 \le h_1 \le h_2 \le \cdots \le h_{n^{\mathrm{o}}} \le 1. \tag{8}$$

The greater the reliability value h_j the more reliable is the symbol r_j^{o} at the input of the outer decoder.

Definition 1 *Denote by* $\mathbf{h} = (h_1, \ldots, h_{n^{\mathrm{o}}})$ *the vector of reliabilities and by* $\mathcal{H}$ *the set of all possible real-valued vectors* $\mathbf{h}$ *that satisfy restriction (8).*

With these notations we rewrite (4) for $e_\tau(\mathbf{\Delta})$ as

$$e_\tau(\mathbf{h}) = d^{\mathrm{i}} \left(\frac{1}{2} \sum_{j=1}^{n^{\mathrm{o}}} (1 - h_j) + \sum_{j=\tau+1}^{\tau+\varepsilon(\tau)} h_j \right), \tag{9}$$

and from (7) we have for the error correcting radius

$$\rho(\lambda) = \min_{\mathbf{h} \in \mathcal{H}} \max_{\tau \in \mathcal{T}} \ e_\tau(\mathbf{h}). \tag{10}$$

Let us further simplify the task of finding $\rho(\lambda)$ and state it as a separate problem. First, notice that in (9) and (10) the parameter τ is selected independently of h_j, $j = d^{\mathrm{o}} + 1, \ldots, n^{\mathrm{o}}$. The contribution of these specific h_j into (9) is the sum $\sum_{j=d^{\mathrm{o}}+1}^{n^{\mathrm{o}}} (1 - h_j)$. Hence, to minimize in (10) over $\mathbf{h}$ we should select these h_j to have the maximum possible values $h_j = 1$ and this sum will vanish. As a result the summation $\sum_{j=1}^{n^{\mathrm{o}}}$ in (9) can be replaced by $\sum_{j=1}^{d^{\mathrm{o}}}$. Further, $\sum_{j=1}^{d^{\mathrm{o}}} 1$ is replaced by d^{o}. Now we can formulate our problem as follows.

Problem 1 *For any* $1 < \lambda \le 2$ *find the error correcting radius* $\rho(\lambda)$

$$\rho(\lambda) = d^{\mathrm{i}} \min_{\mathbf{h} \in \mathcal{H}} \max_{\tau \in \mathcal{T}} \ f_\tau(\mathbf{h}), \tag{11}$$

where

$$f_{\tau}(\mathbf{h}) = \frac{1}{2}\sum_{j=1}^{d^{\mathrm{o}}}(1-h_j) + \sum_{j=\tau+1}^{\tau+\varepsilon(\tau)} h_j. \tag{12}$$

The set $\mathcal{H}$ is given by Definition 1, $\varepsilon(\tau)$ is defined in (3), the integers d^{i} and d^{o} are the minimum distances of the component codes, and $\mathcal{T}$ is specified by (5).

Problem 1 coincides with finding the decoding radius of a single-trial adaptive GMD decoder. For this decoder (with $\lambda = 2$) Kovalev obtained in [1] the following bounds for $\rho(2)$:

$$\frac{d^{\mathrm{i}}}{2}\left(d^{\mathrm{o}}+1-\left\lceil\frac{d^{\mathrm{o}}+1}{4}\right\rceil\right) \leq \rho(2) < \frac{d^{\mathrm{i}}}{2}\left(d^{\mathrm{o}}+2-\left\lceil\frac{d^{\mathrm{o}}+1}{4}\right\rceil\right), \tag{13}$$

from where we get an approximation $\rho(2) \approx 3d^{\mathrm{i}}d^{\mathrm{o}}/8$. Our goal is to estimate $\rho(\lambda)$ for arbitrary $1 < \lambda \leq 2$. The following theorem gives a lower bound for $\rho(\lambda)$.

Theorem 1 *The error correcting radius $\rho(\lambda)$ of the single-trial adaptive algorithm (solution of Problem 1) with parameter λ satisfies the lower bound*

$$\rho(\lambda) \;\geq\; \underline{\rho}(\lambda) \;\triangleq\; \frac{d^{\mathrm{i}}}{2}\left(\left\lfloor\frac{d^{\mathrm{o}}-1}{\lambda}\right\rfloor + \left\lfloor\frac{d^{\mathrm{o}}-\lfloor\frac{d^{\mathrm{o}}-1}{\lambda}\rfloor-2}{\lambda}\right\rfloor + 2\right), \tag{14}$$

where $d^{\mathrm{i}}, d^{\mathrm{o}}$ are the distances of the component codes.

For $\lambda = (\ell+1)/\ell$ the arguments of the floor operations in (14) are integers if d^{o} satisfies

$$d^{\mathrm{o}} = s(\ell+1)^2 + \ell + 2, \qquad s = 0, 1, \ldots \tag{15}$$

In this case we can simplify (14) by omitting the floor operations and get the following expressions for $\underline{\rho}(\lambda)$:

$$\underline{\rho}(\lambda) = \frac{d^{\mathrm{i}}d^{\mathrm{o}}}{2}\left(1-\left(\frac{\lambda-1}{\lambda}\right)^2 + \frac{2\lambda^2-3\lambda+1}{d^{\mathrm{o}}\lambda^2}\right) \gtrsim \frac{d^{\mathrm{i}}d^{\mathrm{o}}}{2}\left(1-\left(\frac{\lambda-1}{\lambda}\right)^2\right). \tag{16}$$

If d^{o} does not satisfy (15), these expressions give a good approximation for $\underline{\rho}(\lambda)$. We see that for $\lambda = 2$ our results coincide with Kovalev's $\rho(2) \approx 3d^{\mathrm{i}}d^{\mathrm{o}}/8$. In terms of ℓ we equivalently have

$$\underline{\rho}\left(\frac{\ell+1}{\ell}\right) = \frac{d^{\mathrm{i}}d^{\mathrm{o}}}{2}\left(1-\frac{1}{(\ell+1)^2}+\frac{\ell+2}{d^{\mathrm{o}}(\ell+1)^2}\right) \gtrsim \frac{d^{\mathrm{i}}d^{\mathrm{o}}}{2}\left(1-\frac{1}{(\ell+1)^2}\right).$$

Theorem 2 *The error correcting radius $\rho(\lambda)$ of the single-trial adaptive algorithm (the solution of Problem 1) with parameter λ satisfies the following upper bound*

$$\rho(\lambda) \leq \bar{\rho}(\lambda) \triangleq \frac{d^{\mathrm{i}}}{\lambda}\left(d^{\mathrm{o}} - 1 - \frac{1}{2}\left\lfloor \frac{d^{\mathrm{o}}-1}{\lambda} \right\rfloor\right), \tag{17}$$

where $\varepsilon(\tau)$ is defined in (3) and $d^{\mathrm{i}}, d^{\mathrm{o}}$ are the distances of the component codes.

The obtained upper and lower bounds (17) and (14) are nearly tight and the approximation (16) holds for both bounds. Now we additionally show that the bounds are exact if d^{o} satisfies (15).

Corollary 1 *If $\lambda = (\ell+1)/\ell$ and d^{o} satisfies (15), then the error correcting radius $\rho(\lambda)$ is $\rho(\lambda) = \underline{\rho}(\lambda) = \bar{\rho}(\lambda)$.*

References

[1] S. I. Kovalev, Two classes of minimum generalized distance decoding algorithms, *Probl. Pered. Inform.* 22, 1986, 35-42.

[2] G. Schmidt, V. R. Sidorenko, and M. Bossert, Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs, Preprint, available online at ArXiv, `arXiv:cs.IT/0610074`, 2006.

[3] G. D. Forney Jr., *Concatenated Codes*, Cambridge, MA: MIT Press, 1966.

[4] V. V. Zyablov, Optimization of concatenated decoding algorithms, *Probl. Pered. Inform.* 9, 1973, 26-32.

[5] J. H. Weber, K. A. S. Abdel-Ghaffar, Reduced GMD decoding, *IEEE Trans. Inform. Theory* 49, 2003, 1013-1027.

[6] V. R. Sidorenko, G. Schmidt, M. Bossert, Decoding punctured Reed-Solomon codes up to the Singleton bound, in *Proc. Intern. ITG Conf. Source Channel Coding*, Ulm, Germany, January 2008.

[7] U. K. Sorger, A new Reed-Solomon code decoding algorithm based on Newton's interpolation, *IEEE Trans. Inform. Theory* 39, 1993, 358-365.

[8] R. Kötter, Fast generalized minimum-distance decoding of Algebraic-Geometry and Reed-Solomon codes, *IEEE Trans. Inform. Theory* 42, 1993, 721-737.

Existence of transitive partitions into binary codes

Faina Solov'eva sol@math.nsc.ru
Sobolev Institute of Mathematics, Novosibirsk State University
pr. ac. Koptyuga 4, Novosibirsk 630090, RUSSIA

Abstract. Some methods to construct transitive partitions of the set F_2^n of all binary vectors of length n into binary codes are presented. It is established that for any $n = 2^k - 1, k \geq 3$, there exist transitive partitions of F_2^n into perfect binary transitive codes of length n and distance 3.

1 Introduction

In this paper we continue the investigation of transitive objects beginning in [1-4]. Applying some switching constructions of partitions of the set F_2^n of all binary vectors of length n into perfect binary codes given in [5] (using Vasil'ev construction [6]) and also using Mollard construction [7] we construct transitive partitions of F_2^n into transitive binary codes. The methods permit us to construct transitive partitions of F_2^n into perfect binary codes. Mollard construction allows to get transitive partitions of F_2^n into nonparallel Hamming codes, i.e. the codes, which can not be obtained from each other using a translation by a vector of F_2^n (the method is essentially different from the method to construct partitions of F_2^n into nonparallel Hamming codes, see [8]). Transitive objects play an important role in the coding theory. Transitive codes are close to linear codes by some of their properties. Transitive partitions can be useful to construct new transitive codes.

In [2] several methods to construct transitive binary codes are given, in particular, we got a class of perfect and extended perfect transitive codes for any admissible length $n \geq 31$. The number of nonequivalent perfect transitive codes of length $n = 2^k - 1$ and distance 3 is not less than $\lfloor k/2 \rfloor^2$. An analogous estimate is true for extended perfect transitive codes. Earlier it was known $\lfloor (k+1)/2 \rfloor$ such perfect codes of length $n = 2^k - 1$, see [9]; analogous for the extended case, see [10]. Transitive codes obtained in [2] have different ranks, for example, for $n = 16^l - 1, l > 0$ the ranks vary from $n - \log(n+1)$ (the rank of the Hamming code of length n) to $n - \frac{\log(n+1)}{4}$. In [11] Potapov found the exponential number of transitive extended perfect codes of small rank. Transitive perfect binary codes of length 15 are investigated in [12]. It is easy to see that an extension of any transitive code by the parity checking give us a transitive code. The converse is not true, in [13] Malyugin has shown that there exists the transitive perfect binary code of length 16 such that any its puncturing perfect code is not transitive. Therefore it is worthwhile to investigate independently the extended case. Many known classes of good codes are transitive, for example, all additive codes, all Z_4-linear codes. In [13] perfect transitive codes of length 15 which belong to the switching class of the Hamming code are enumerated.

Two constructions of partitions of F_2^n into perfect codes were given in [5]. For any admissible $n \geq 15$ one of these construction allowed to get not less than $2^{2^{(n-1)/2}}$ different partitions of F_2^n into perfect binary codes of length $n > 15$, see [14]. In [15] a switching construction of the partitions of F_2^n into pairwise nonequivalent perfect binary codes of length n is presented for any $n = 2^k - 1$, $k \geq 5$.

2 Necessary definitions and notions

Let F_2^n be the set of all binary vectors of length n. Any subset of F_2^n is called a *binary code* of length n. A code C is *perfect binary code correcting single error* (briefly a perfect code) if for any vector $x \in F_2^n$ there exists exactly one vector $y \in C$ such that $d(x, y) \leq 1$. It is well known that perfect binary codes with code distance 3 exist if and only if $n = 2^k - 1, k > 1$. It is known that every isometry of F_2^n is defined as

$$\mathrm{Aut}(F_2^n) = F_2^n \leftthreetimes S_n = \{(v, \pi) \mid v \in F_2^n, \pi \in S_n\},$$

where $\leftthreetimes$ denotes a semidirect product, S_n is a group of symmetry of order n. The *automorphism group* $\mathrm{Aut}(C)$ of any code C of length n consists of all the isometries of F_2^n that transform the code into itself:

$$\mathrm{Aut}(C) = \{(v, \pi) \mid v + \pi(C) = C\}.$$

A code C is said to be *transitive* if its automorphism group acts transitively on all codewords. *The automorphism group of any family of codes* $\mathcal{P} = \{C_0, C_1, \ldots, C_m\}$, $\mathcal{P} \subseteq F_2^n$, $m \leq n$, is a group of isometries of F_2^n that transform the set $\mathcal{P}$ into itself such that for any $i \in M = \{0, 1, \ldots, m\}$ there exists $j \in M$, $v \in F_2^n$, $\pi \in S_n$ satisfying $v + \pi(C_i) = C_j$. Every such isometry induces a permutation τ on the index set M that permutes the codes in the partition $\mathcal{P}$:

$$\tau(\{C_0, C_1, \ldots, C_m\}) = \{C_{\tau(0)}, C_{\tau(1)}, \ldots, C_{\tau(m)}\},$$

i. e. the automorphism group of the family $\mathcal{P}$ is isomorphic to some subgroup of the group S_{m+1}. A family of codes $\mathcal{P}$ is *transitive* if its automorphism group acts transitively on the elements (the codes) of the family. Two partitions we call *equivalent* if there exists an isometry of the space F_2^n that transforms one partition into another one.

3 Constructions of transitive partitions

In this section we give two constructions of transitive partitions. As the starting point for the case of perfect codes we will take transitive Phelps partitions given in [16], where he classified all partitions of F_2^7 into Hamming codes of length 7. Regardless of the fact that the Hamming code is unique (up to equivalence) there are 11 such nonequivalent partitions. In the list of these partitions we will use one special partition $\mathcal{P}^7 = \{H_0^7, H_1^7, \ldots, H_7^7\}$, here $\mathbf{0}^7 \in H_0^7$ and $\mathbf{0}^7 \in H_i^7 + e_i$ for every $i \in \{1, \ldots, 7\}$, where e_i is the vector of length 7 with only one ith unit coordinate. For the partition it is true $|(H_i^7 + e_i) \cap (H_j^7 + e_j)| = 4$ for any $i \neq j$, $i, j \in \{1, \ldots, 7\}$, i.e. the codes in the partition are pairwise nonparallel. It is true the following known fact

Proposition 1. *The partition $\mathcal{P}^7$ is a transitive partition of F_2^7 into pairwise nonparallel Hamming codes of length 7.*

Construction A. In this section we show how the iterative construction of the partitions from [5] based on Vasil'ev codes from [6] allows to get transitive partitions of F_2^n into classes of codes. As a particular case we get transitive partitions of F_2^n into perfect codes for any admissible length.

Theorem 1. *Let $\mathcal{P}^n = \{C_0^n, C_1^n, \ldots, C_m^n\}$ be a transitive family of binary codes of length n; let B^n be any binary linear code of length n with odd code distance such that for any automorphism $(y, \pi) \in Aut(\mathcal{P}^n)$ it holds $\pi \in Sym(B^n)$. Then the family of the codes*
$\mathcal{P}^{2n+1} = \{C_0^{2n+1}, C_1^{2n+1}, \ldots, C_{2m+1}^{2n+1}\}$:
$C_i^{2n+1} = \{(x, |x|, x + y) : x \in B^n, y \in C_i^n\}$, $C_{m+i+1}^{2n+1} = C_i^{2n+1} + e_{n+1}$,
where $i = 0, 1, \ldots, m$, *is transitive.*

Codes from Theorem (1) we call Vasil'ev codes.

Taking into account that a translation of any transitive code by any vector of the space is again a transitive code we get from the last theorem and Theorem 1 in [2] the folowing

Corollary 1. *If every code in the family $\mathcal{P}^n$ is transitive than every code of the family $\mathcal{P}^{2n+1}$ from Theorem (1) is transitive.*

It is also true

Corollary 2. *Let $\mathcal{P}^n = \{C_0^n, C_1^n, \ldots, C_n^n\}$ be a transitive partition of F_2^n into perfect binary codes of length n. Then the family of the codes from Theorem (1) is a transitive partition of the space F_2^{2n+1} into perfect binary codes of length $2n + 1$.*

Taking into account the construction (1), Proposition 1 and corollaries 1 and 2 we can iteratively construct transitive partitions of the space F_2^n into transitive perfect codes for any admissible length $n = 2^m - 1, m \geq 3$, i.e. it is true

Theorem 2. *There exist transitive partitions of F_2^n into transitive perfect codes of length n for any $n = 2^m - 1,\ m \geq 3$.*

Corollary 3. *There exist transitive partitions of full-even binary code into extended transitive perfect codes of length n for any $n = 2^m,\ m \geq 4$.*

Construction B. Here we give another method to construct transitive partitions. The method is based on Mollard construction [7] for binary codes. It is known that Mollard construction is a generalization of Vasil'ev construction for the codes correcting single errors. The construction B given below is also a generalization of the construction A. As contrasted with the construction B the construction A gives transitive partitions into codes with big code distances. In turn the construction B allows to get partitions of F_2^n into nonparallel Hamming codes.

Further we will use the following particular case of Mollard construction [7] for binary codes. Let P^t and C^m be any two binary codes of lengths t and m respectively with code distances not less than 3. Let

$$x = (x_{11}, x_{12}, \ldots, x_{1m}, x_{21}, \ldots, x_{2m}, \ldots, x_{t1}, \ldots, x_{tm}) \in F_2^{tm}.$$

The generalized parity-check functions $p_1(x)$ and $p_2(x)$ are defined by $p_1(x) = (\sigma_1, \sigma_2, \ldots, \sigma_t) \in F_2^t$, $p_2(x) = (\sigma'_1, \sigma'_2, \ldots, \sigma'_m) \in F_2^m$, where $\sigma_i = \sum_{j=1}^m x_{ij}$ and $\sigma'_j = \sum_{i=1}^t x_{ij}$. The set

$$C^n = \{(x, y + p_1(x), z + p_2(x)) \mid x \in F_2^{tm}, y \in P^t, z \in C^m\}$$

is a binary Mollard code of length $n = tm + t + m$ and code distance 3, see [7]. It is true the following

Theorem 3. *Let $\mathcal{P}^t = \{C_0^t, C_1^t, \ldots, C_t^t\}$ and $\mathcal{P}^m = \{D_0^m, D_1^m, \ldots, D_m^m\}$ be any transitive families of the codes of length t and m respectively correcting single errors. Then the family of the codes*

$$\mathcal{P}^n = \{C_{00}^n, C_{01}^n, \ldots, C_{tm}^n\}$$

is transitive class of codes of length $n = tm + t + m$, correcting single errors, where

$$C_{ij}^n = \{(x, y + p_1(x), z + p_2(x)) \mid x \in F_2^{tm}, y \in C_i^t, z \in D_j^m\} \tag{1}$$

is Mollard code, $i = 0, 1, \ldots, t;\ j = 0, 1, \ldots, m$.

From this theorem and Theorem 3 of the paper [2] we get

Corollary 4. *Let $\mathcal{P}^t$ and $\mathcal{P}^m$ be any transitive partitions of F_2^t and F_2^m into perfect transitive codes of length $t = 2^r - 1$, $r \geq 3$, and $m = 2^l - 1$, $l \geq 3$, respectively. Then the construction (1) gives a transitive partition of F_2^n into perfect binary transitive codes of length $n = tm + t + m$.*

Remark. It should be noted that Theorem 3 is true to get transitive partitions into nontransitive codes. For $t = 1$ Corollary 2 can be obtain from Corollary 4 as a particular case.

Theorem 3 and Proposition 1 allow us to construct by induction transitive partitions of F_2^n into pairwise nonparallel Hamming codes.

Theorem 4. *Let $\mathcal{P}^t = \{H_0^t, H_1^t, \ldots, H_t^t\}$ and $\mathcal{P}^m = \{H_0^m, H_1^m, \ldots, H_m^m\}$ be any transitive partitions into pairwise nonparallel Hamming codes, $t = 2^r - 1$, $r \geq 3$, and $m = 2^l - 1$, $l \geq 3$. Then the family of the codes*

$$H_{ij}^n = \{(x, y + p_1(x), z + p_2(x)) \mid x \in F_2^{tm}, y \in H_i^t, z \in H_j^m\}, \qquad (2)$$

$i = 0, 1, \ldots, t$, $j = 0, 1, \ldots, m$, is a transitive partition of F_2^n into pairwise nonparallel Hamming codes of length $n = tm + t + m$.

Denote by $\bar{H}^n$ the code containing all-zero vector obtained from the code H^n of length n by a switch on some vector from F_2^n.

Remark. It holds from Theorem 4 that if we know the size of the sets $\bar{H}_i^t \cap \bar{H}_k^t$ and $\bar{H}_j^m \cap \bar{H}_s^m$ for any $i, k \in \{0, 1, \ldots, t,\}$ and $j, s \in \{0, 1, \ldots, m\}$ we can easily calculate the size of the intersection codes $\bar{H}_{ij}^n$ and $\bar{H}_{ks}^n$.

References

[1] F. I. Solov'eva, On transitive codes, it Proc. Intern. Workshop Discr. Anal. Oper. Res., Novosibirsk, Russia. 2004, 99.

[2] F. I. Solov'eva, On construction of transitive codes, *Probl. Inform. Transm.* 41, 2005, 204-211.

[3] F. I. Solov'eva, On $\mathbb{Z}_4$-linear codes with parameters of Reed-Muller codes, *Probl. Inform. Transm.* 43, 2007, 26-32.

[4] J. Pujol, J. Rifa, F. I. Solov'eva, Construction of Z_4-linear Reed-Muller codes, *IEEE Trans. Inform. Theory* submitted.

[5] F. I. Solov'eva, On binary nongroup codes, *Methody Discr. Analiza* 37, 1981, 65-76 (in Russian).

[6] Yu. L. Vasil'ev, On nongroup close-packed codes, *Probl. der Kybern.* 8, 1962, 92-95.

[7] M. Mollard, A generalized parity function and its use in the construction of perfect codes, *SIAM J. Alg. Discr. Math.* 7, 1986, 113-115.

[8] O. Heden, F. I. Solov'eva, On partitions of n-cube into nonparallel Hamming codes, submitted.

[9] J. Borges, J. K. Rifa, A characterization of 1-perfect additive codes, *IEEE Trans. Inform. Theory* 45, 1999, 1688-1697.

[10] D. S. Krotov, Z_4-linear perfect codes, *Discr. Anal. Oper. Res.* Ser. 1., 7, 2000, 78-90 (in Russian).

[11] V. N. Potapov, On lower bound on the number of transitive perfect codes, *Discr. Anal. Oper. Res.* Ser. 1, 13, 2000, 49-59 (in Russian).

[12] S. A. Malyugin, Transitive perfect codes of length 15, *Proc. Intern. Workshop Discr. Anal. Oper. Res.*, Novosibirsk, Russia, 2004, 96.

[13] S. A. Malyugin, On equivalent classes of perfect binary codes of length 15, Preprint 138. Novosibirsk: Inst. of Mathematics of SB RAS, 2004, 34.

[14] F. I. Solov'eva, On Perfect Codes and Related Topics, Com^{2}Mac Lecture Note Series 13, Pohang 2004.

[15] S. V. Avgustinovich, O. Heden, F. I. Solov'eva, On partitions of n-cube into nonequivalent perfect codes, *Probl. Inform. Transm.* 43, 2007, 45-50.

[16] K. T. Phelps, An enumeration of 1-perfect binary codes of length 15, *Australas. J. Comb.* 21, 2000, 287-298.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 273-279*

Burst-error correcting codes and lattice paths

Ulrich Tamm tamm@ieee.org
German Language Department of Business Informatics,
Marmara University, Istanbul, TURKEY and
Department of Mathematics, University of Bielefeld, GERMANY

1 Introduction

At the 3rd Waterloo Conference on Combinatorics [16, pp. 341-342], Berlekamp presented the following combinatorial problem. The problem will be illustrated with the following example also due to Berlekamp in [16].

$$\begin{array}{c|cccccc}
8 & & & & & & 1 \\
7 & & & & & 1 & 1 \\
6 & & & & & 1 & 2 \\
5 & & & & 1 & 1 & 3 \\
4 & & & 1 & 1 & 2 & 7 \\
3 & & & 1 & 2 & 5 & 19 \\
2 & & 1 & 1 & 3 & 9 & 37 \\
1 & 1 & 1 & 2 & 7 & 23 & 99 \\
0 & 1 & 2 & 5 & 19 & 66 & 293 \\
\hline
 & 0 & 1 & 2 & 3 & 4 & 5
\end{array} \tag{1}$$

Berlekamp defines an array to be unitary if any square submatrix whose upper left corner falls on the boundary of the array has a determinant equal to 1. For instance, in the array above

$$\det \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 5 \\ 1 & 3 & 9 \end{pmatrix} = 1$$

The problem then he states as follows: "... A periodic quasilinear boundary represents the best staircase approximation to a straight line of rational slope. ... Exact formulas are known for the values of the numbers in the unitary arrays generated by periodic quasilinear boundaries of slopes $1/n$ or n, but no such formulas are known (to me) for the values in the arrays with boundaries of slopes m/n where $1 < m < n$. The simplest such case is slope $2/3$" – this is shown above in (1).

This problem arose already in Berlekamp's paper [4], where the numbers in the array above reduced modulo 2 were suggested as a convolution code As Berlekamp pointed out [6], the density of ones in these codes is very low, which was regarded as a disadvantage that time, "...But in the past decades we have seen great popularity of parity-check codes and of turbo codes, both of which are of low density (and hard to design in any highly structured way, compared with, say, RS block codes). The key is to find a criteria other than constraint length. Anyway, I think the topic of structured generator sequences for convolutional codes merits attention again, although my '63 paper seems to be perhaps the only one that attempts to initiate work in that area...".

The case slope 2/3 and 3/2, which yield codes of rates 2/5 and 3/5, respectively, was studied in further detail later on by Berlekamp in [5], where he derived some formulas for special parameters and then stated: "The patterns are clear but I know no explanation. Why does the formula apply to an individual entry, then to sums of pairs of entries from different rows, and then to the *negative* of an entry?". This question had been answered in our papers [13] (without being aware of the reference [5] at that time) and [15]. In Section IV we shall give the generating function for the entries in the array (1). The methods using lattice path enumeration are presented in Section II and III. They apply to further lattice path models, which is the topic of actual research, as briefly mentioned in Section V.

2 Lattice path enumeration

Carlitz, Roselle, and Scoville [8] later presented a fast algorithm for the computation of the number of such lattice paths by getting rid of the determinant calculation. They showed that the entries in this array enumerate the lattice paths from the beginning of the row to the top of the column which determine the respective entry, where these paths are not allowed to cross the boundary given by the 1's. For instance, in the array (1) above the positions of the 1's are below the boundary determined by $u_0 = 2$, $u_1 = 3$, $u_2 = 5$, $u_3 = 6$, $u_4 = 8$, $u_5 = 9$, etc.

A path here is a sequence of pairs (s_i, t_i), $i = 0, 1, \dots$ of nonnegative integers where (s_i, t_i) is either $(s_{i-1} + 1, t_{i-1})$ or $(s_{i-1}, t_{i-1} + 1)$. So, a particle following such a path can move either one step to the right, i. e. $s_i = s_{i-1} + 1$, or one step upwards, i. e. $t_i = t_{i-1} + 1$ in each time unit i. We shall assume that a path starts in the origin $(0, 0)$. There is a one-to-one correspondence between a $\{0, 1\}$- sequence x^m and a path with m steps: a 0 in the sequence x^m corresponds to a step upwards, a 1 to a step to the right in the corresponding path. The (infinite) path determined by this boundary hence corresponds to the periodic, binary sequence

$$0010100101001010 01...$$

Observe that the positions of the 1's in this sequence are at $v_{i+1} = u_i + i$ for all $i = 0, 1, 2, \ldots$. This holds, because there is exactly one step to the right after each u_i steps upwards in the boundary lattice path.

The rows in the array (1) above behave periodically in the sense that every third row has the same entries, which are only shifted according to the boundary. Because of this fact only two further sequences have to be considered in order to analyze Berlekamp's problem for slope $\frac{2}{3}$, namely the sequences

$$01010010100101001\ldots \quad \text{and} \quad 01001010010100101\ldots,$$

since the paths corresponding to these sequences characterize all possible boundaries arising in the array (1).

In terms of these sequences, Berlekamp's problem was analyzed in [15] by studying the size of the downsets $N(y^m)$ of the initial segments y^m of these three sequences in the so called pushing order (cf. [3] and [11]), which played a central role in Ahlswede's and Khachatrian's solution of the Erdös–Ko–Rado Problem [1, 2].

3 Gessels's probabilistic method

We shall consider paths in an integer lattice from the origin $(0, 0)$ to the point (n, u_n), which never touch any of the points (i, u_i), $i = 0, 1, \ldots, n-1$. In [9] Gessel introduced a general probabilistic method to determine the number of such paths, denoted by f_n , which he studied for the case that the subsequence $(u_i)_{i=1,2\ldots}$ is periodic. For period length 2 the elements of the sequence $(u_i)_{m=0,1,2,\ldots}$ are on the 2 lines (for $i = 0, 1, 2, \ldots$)

$$u_{2i} = s + ci \text{ and } u_{2i+1} = s + \mu + ci, \tag{2}$$

Gessel's probabilistic method is as follows. A particle starts at the origin $(0, 0)$ and successively moves with probability p one unit to the right and with probability $q = 1 - p$ one unit up. The particle stops if it touches one of the points (i, u_i). The probability that the particle stops at (n, u_n) is $p^n q^{u_n} \cdot f_n$. Setting

$$f(t) = \sum_{n=0}^{\infty} f_n t^n = \sum_{n=0}^{\infty} f_{2n} t^{2n} + \sum_{n=0}^{\infty} f_{2n+1} t^{2n+1} = g(t^2) + t \cdot h(t^2)$$

the probability that the particle eventually stops is

$$q^{u_0} g(p^2 q^c) + p q^{u_1} h(p^2 q^c)$$

If p is sufficiently small, the particle will touch the boundary $(i, u_i)_{i=0,1,\cdots}$ with probability 1. So for small p and with $t = pq^{c/2}$ we have

$$q(t)^{u_0} g(t^2) + p(t) q(t)^{u_1} h(t^2) = 1$$

For p sufficiently small one may invert $t = p(1-p)^{c/2}$ to express p as a power series in t, namely $p = p(t)$. Then changing t to $-t$ and denoting $p(-t)$ by $\overline{p}(t)$ and similarly $q(-t)$ by $\overline{q}(t)$ yields the system of equations

$$q^s \cdot g(t^2) + p \cdot q^{s+\mu} \cdot h(t^2) = 1,$$

$$\overline{q}^s \cdot g(t^2) + \overline{p} \cdot \overline{q}^{s+\mu} \cdot h(t^2) = 1 \tag{3}$$

which for $g(t^2)$ and $h(t^2)$ yield the solutions

$$g(t^2) = \frac{p^{-1} q^{-s-\mu} - \overline{p}^{-1} \overline{q}^{-s-\mu}}{p^{-1} q^{-\mu} - \overline{p}^{-1} \overline{q}^{-\mu}} = \frac{q^{c/2-\mu-s} + \overline{q}^{c/2-\mu-s}}{q^{c/2-\mu} + \overline{q}^{c/2-\mu}} \tag{4}$$

and

$$h(t^2) = \frac{q^{-s} - \overline{q}^{-s}}{t \cdot (q^{\mu-c/2} + \overline{q}^{\mu-c/2})} \tag{5}$$

By Lagrange inversion (cf. e.g. [12]) for any α we have

$$q^{-\alpha} = \sum_{n=0}^{\infty} \frac{\alpha}{(c/2+1)n+\alpha} \binom{(c/2+1)n+\alpha}{n} \cdot t^n \tag{6}$$

The following identities were derived in [9] and [13]. Since we are going to look at several random walks in parallel, we shall write the parameters determining the restrictions as superscripts. So, $g^{(s,c,\mu)}$ and $h^{(s,c,\mu)}$ are the generating functions (4) and (5) for even and odd n, respectively, for the random walk of a particle starting at the origin and first touching the boundary $(i, u_i)_{i=0,1,\ldots}$ determined by the parameters s, c, and μ as defined under (2) in the lattice point (n, u_n).

Theorem [9, 13] a) Let c be an odd positive integer, $s = 1$ and $\mu = \frac{c-1}{2}$. Then

$$h^{(1,c,\frac{c-1}{2})}(t^2) = \frac{q^{-1/2} - \overline{q}^{-1/2}}{t} = \sum_{n=0}^{\infty} \frac{1}{(c+2)n+\mu+2} \binom{(c+2)n+\mu+2}{2n+1} t^{2n}.$$

b) For $0 \leq \mu < \frac{c}{2}$ it is

$$g^{(s,c,\mu)}(t^2) + g^{(s,c,c-\mu)}(t^2) = q^{-s} + \overline{q}^{-s} = \sum_{n=0}^{\infty} \frac{2s}{(c+2)n+s} \binom{(c+2)n+s}{2n} t^{2n}$$

and

$$g^{(s,c,c-\mu)}(t^2) - g^{(s,c,\mu)}(t^2) = t^2 \cdot h^{(s,c,\mu)}(t^2) \cdot h^{(c-2\mu,c,\mu)}(t^2).$$

c) Let $s+\mu=c$ with $s\geq\mu$, then

$$h^{(s,c,c-s)}(t^2)+h^{(c-s,c,s)}(t^2)=\frac{1}{t^2}\cdot(p+\overline{p})=\sum_{n=1}^{\infty}\frac{2}{(c+2)n-1}\binom{(c+2)n-1}{2n}\cdot t^{2(n-1)}$$

In the special case c odd, $s=\frac{c+1}{2}$ and $\mu=\frac{c-1}{2}$ we have

$$h^{(\frac{c+1}{2},c,\frac{c-1}{2})}(t^2)-h^{(\frac{c-1}{2},c,\frac{c+1}{2})}(t^2)=\left(g^{(\frac{c+1}{2},c,\frac{c-1}{2})}(t^2)\right)^2,$$

where

$$g^{(\frac{c+1}{2},c,\frac{c-1}{2})}(t^2)=\frac{1}{t}\cdot(\overline{q}^{\frac{1}{2}}-q^{\frac{1}{2}})=\sum_{n=0}^{\infty}\frac{1}{(c+2)n+\frac{c+1}{2}}\binom{(c+2)n+\frac{c+1}{2}}{2n+1}\cdot t^{2n}.$$

d)

$$\left(g^{(s,c,\mu)}(t^2)+g^{(s,c,c-\mu)}(t^2)\right)\cdot h^{(s,c,\mu)}(t^2)=h^{(2s,c,\mu)}(t^2).$$

e)

$$g^{(c-2\mu,c,\mu)}(t^2)\cdot g^{(\mu,c,c-\mu)}(t^2)=g^{(c-\mu,c,\mu)}(t^2).$$

f) For $s_1+\mu_1+\mu_2=c$ we have

$$g^{(s_1,c,\mu_1)}(t^2)\cdot h^{(s_2,c,\mu_2)}(t^2)=h^{(s_2,c,s_1+\mu_2)}(t^2).$$

Especially, for odd c

$$g^{(1,c,\frac{c-1}{2})}(t^2)\cdot h^{(1,c,\frac{c-1}{2})}(t^2)=h^{(1,c,\frac{c+1}{2})}(t^2).$$

4 Solution of Berlekamp's problem

Now we are able to explain the entries from Berlekamp's example array for slope $\frac{2}{3}$. We have to inspect the parameter choices $(s=1,\mu=1)$, $(s=1,\mu=2)$, and $(s=2,\mu=1)$. By application of the previous theorem, the generating functions for these parameters (after mapping $t^2\to x$) look as follows.

Corollary [15]:

$$g^{(1,3,1)}(x)=\sum_{n=0}^{\infty}\frac{1}{5n+1}\binom{5n+1}{2n}x^n-\frac{x}{2}\cdot[h^{(1,3,1)}(x)]^2=1+2x+23x^2+377x^3+\ldots,$$

$$g^{(1,3,2)}(x)=\sum_{n=0}^{\infty}\frac{1}{5n+1}\binom{5n+1}{2n}x^n+\frac{x}{2}\cdot[h^{(1,3,1)}(x)]^2=1+3x+37x^2+624x^3+\ldots,$$

$$g^{(2,3,1)}(x) = \sum_{n=0}^{\infty} \frac{1}{5n+2} \binom{5n+2}{2n+1} x^n = 1 + 5x + 66x^2 + 1156x^3 + \dots,$$

$$h^{(1,3,1)}(x) = \sum_{n=0}^{\infty} \frac{1}{5n+3} \binom{5n+3}{2n+1} x^n = 1 + 7x + 99x^2 + 1768x^3 + \dots,$$

$$h^{(1,3,2)}(x) = \sum_{n=1}^{\infty} \frac{1}{5n-1} \binom{5n-1}{2n} x^{n-1} - \frac{1}{2}[g^{(2,3,1)}(x)]^2 = 1 + 9x + 136x^2 + \dots,$$

$$h^{(2,3,1)}(x) = \sum_{n=1}^{\infty} \frac{1}{5n-1} \binom{5n-1}{2n} x^{n-1} + \frac{1}{2} \cdot [g^{(2,3,1)}(x)]^2 = 2 + 19x + 293x^2 + 5332x^3 + \dots.$$

Using the results in the above theorem, in [15] we also derived the generating functions for the array in Berlekamp's problem with slope $\frac{2}{5}$.

5 Concluding remarks

1) There is a one–to–one correspondence between s–ary regular trees and ballot–type $\{0,1\}$–sequences $x^{sn} = (x_1, \dots, x_{sn})$ of weight (= number of 1's) $wt(x^{sn}) = n$ fulfilling the condition $wt(x_1, \dots, x_i) \geq \frac{i}{s}$ for all $i = 1, \dots, sn-1$. This correspondence can be exploited to store regular trees, by assigning to them as codewords the ballot - type sequence. The codes thus obtained form a prefix code, cf. [10].

2) Probably most interesting, and indeed the topic of actual research, is the fact that the formulae from the above theorem also arise in the enumeration of a different type of lattice paths. Here, the boundary not allowed to be crossed is obtained by repeatedly moving s steps upwards, and t steps to the right. This model was seemingly first studied in [7]. Again we analyzed the case $s = 2, t = 3$ and $s = 3, t = 2$. Here the formulae from the above theorem enumerate the number of paths to any point on the boundary. Interestingly, all the six formulae have a natural interpretation, whereas in the analysis of Berlekamp's array only four of them really come into play. However, for further periodic slopes, the analysis is more difficult.

References

[1] R. Ahlswede, L. H. Khachatrian, The complete nontrivial–intersection theorem for systems of finite sets, *J. Combin. Theory* Ser. A 76, 1996, 121-138.

[2] R. Ahlswede, L. H. Khachatrian, The complete intersection theorem for systems of finite sets, *Europ. J. Combin.* 18, 1997, 125-136.

[3] R. Ahlswede, Z. Zhang, On maximal shadows of members in left – compressed sets, *Discr. Appl. Math.* 95, 1999, 3-9.

[4] E. R. Berlekamp, A class of convolutional codes, *Inform. Control* 6, 1963, 1-13.

[5] E. R. Berlekamp, Unimodular arrays, *Comp. Math. Appl.* 39, 2000, 77-88.

[6] E. R. Berlekamp, private communication.

[7] R. J. Chapman, T. Y. Chow, A. Khetan, D. P. Moulton, R. J. Waters, Simple formulas for lattice paths avoiding certain periodic staircase boundaries, Preprint, 2007.

[8] L. Carlitz, D. P. Roselle, R. A. Scoville, Some remarks on ballot – type sequences, *J. Combin. Theory* 11, 1971, 258-271.

[9] I. Gessel, A probabilistic method for lattice path enumeration, *J. Stat. Plann. Infer.* 14, 1986, 49-58.

[10] K. Kobayashi, H. Morita, M. Hoshi, Enumerative coding for k–ary trees, *Proc. 19th Symp. Inform. Theory Appl. (SITA96)*, Hakone, Japan, 1996, 377-379.

[11] R. P. Stanley, Weyl groups, the hard Lefschetz theorem, and the Sperner property, *SIAM J. Algebr. Discr. Meth.*, 1980, 168-184.

[12] R.P. Stanley, *Enumerative Combinatorics 2*, Cambridge, 1999.

[13] U. Tamm, Lattice paths not touching a given boundary, *J. Stat. Plann. Infer.* 105, 2002, 433-448.

[14] U. Tamm, On a problem of Berlekamp, *Proc. Intern. Symp. Inform. Theory*, Yokohama, Japan, 2003, 41.

[15] U. Tamm, Size of downsets in the pushing order and a problem of Berlekamp, *Discr. Appl. Math.*, to appear.

[16] W. Tutte (ed.), *Recent Progress in Combinatorics*, Proc. 3rd Waterloo Conf. Combin., 1968.

Sets of mutually orthogonal resolutions of BIBDs[1]

SVETLANA TOPALOVA svetlana@math.bas.bg
STELA ZHELEZOVA stela@math.bas.bg
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
P.O. Box 323, 5000 Veliko Tarnovo, BULGARIA

Abstract. The nonisomorphic sets of m mutually orthogonal resolutions (m-MORs) of doubly resolvable $2-(v,k,\lambda)$ designs with small parameters are constructed and lower bounds on the number of m-MORs of multiple designs are obtained.

1 Introduction

For the basic concepts and notations concerning combinatorial designs and their resolutions refer, for instance, to [2], [3], [7].

Let $V = \{P_i\}_{i=1}^{v}$ be a finite set of *points*, and $\mathcal{B} = \{B_j\}_{j=1}^{b}$ – a finite collection of k-element subsets of V, called *blocks*. If any 2-subset of V is contained in exactly λ blocks of $\mathcal{B}$, then $D = (V, \mathcal{B})$ is a 2-(v,k,λ) *design*, or *balanced incomplete block design (BIBD)*. We shall call two blocks B_1 and B_2 equal ($B_1 = B_2$) if they are incident with the same set of points.

Two designs are *isomorphic* if there exists a one-to-one correspondence between the point and block sets of the first design and respectively, the point and block sets of the second design, and if this one-to-one correspondence does not change the incidence. An *automorphism* is an isomorphism of the design to itself, i.e. a permutation of the points that transforms the blocks into blocks.

A 2-(v,k,$m\lambda$) design is called an *m-fold multiple* of 2-(v,k,λ) designs if there is a partition of its blocks into m subcollections $\mathcal{B}_1$, $\mathcal{B}_2$, ... $\mathcal{B}_m$, which form 2-(v,k,λ) designs $D_1, D_2, ..., D_m$. If $D_1 = D_2 = ... = D_m$ we call the design *true m-fold multiple* of D_1.

A *resolution* of the design is a partition of the collection of blocks into *parallel classes*, such that each point is in exactly one block of each parallel class. We shall call two parallel classes of the resolution $\mathcal{R}$, R_1 and R_2 equal ($R_1 = R_2$) if each block of R_1 is equal to a block of R_2. The design is *resolvable* if it has at least one resolution. Two resolutions are isomorphic if there exists

[1] This work was partially supported by the Bulgarian National Science Fund under Contract No MM 1405.

an automorphism of the design transforming each parallel class of the first resolution into a parallel class of the second one.

There is a one-to-one correspondence [5] between the resolutions of $2-(qk,k,\lambda)$ designs and the $(r,qk,r-\lambda)_q$ equidistant codes, where $r=\lambda(qk-1)/(k-1)$ and $q>1$.

Two resolutions $\mathcal{R}$ and $\mathcal{T}$ of one and the same design are *orthogonal* if the number of blocks in $R_i \cap T_j$ is either 0 or 1 for all $1 \le i,j \le r$. Orthogonal resolutions may or may not be isomorphic to each other. A *doubly resolvable design (DRD)* is a design which has at least two orthogonal resolutions. We denote by *ROR* a resolution which is orthogonal to at least one other resolution, by *m-MOR* a set of m mutually orthogonal resolutions, and by *m-MORs* sets of m mutually orthogonal resolutions. Two m-MORs are isomorphic if there is an automorphism of the design transforming them into each other. The m-MOR is maximal if no more resolutions can be added to it.

The newest results and an extended bibliography and summary of previous works on the existence of DRDs can be found in [1] and a method for construction and classification of RORs and DRDs in [6].

The aim of the present work is the classification up to isomorphism of m-MORs of 2-(v,k,λ) DRDs with small parameters and the establishment of some lower bounds on their number for multiple designs.

2 m-MORs construction and classification

We start with a DRD and construct its resolutions block by block. For each resolution $\mathcal{R}$ we check if it is isomorphic to a lexicographically smaller one, and if not, we try to construct another resolution $\mathcal{R}_1$, which is lexicographically greater than $\mathcal{R}$ and orthogonal to it. We next repeat the same for $\mathcal{R}_1$, $\mathcal{R}_2$, etc, constructing at each step a resolution $\mathcal{R}_m$ orthogonal to all the resolutions $\mathcal{R}, \mathcal{R}_1, ..., \mathcal{R}_{m-1}$, and checking if this m-MOR is isomorphic to a lexicographically smaller one. We output a new m-MOR if it is maximal.

The results are summarized in Table 1, where the last column shows the number of the design in the tables [4] and a/b means that the number of nonisomorphic MORs is b, a of them maximal.

3 m-MORs of multiple designs

We first recall definitions and notations concerning sets of orthogonal Latin squares (see for instance [3]).

Table 1: Classification of inequivalent m-MORs

q	v	k	λ	b	r	DRDs	RORs	2-MORs	3-MORs	4-MORs	No
2	6	3	8	40	20	1	1	1/1	-	-	236
2	6	3	12	60	30	1	1	0/1	1/1	-	596
2	6	3	16	80	40	1	1	$0/\geq485$	$0/\geq485$	$\geq485/\geq485$	1078
2	8	4	6	28	14	1	1	1/1	-	-	101
2	8	4	9	42	21	1	1	0/1	1/1	-	278
2	8	4	12	56	28	4	4	7/17	0/60	60/60	524
2	10	5	16	72	36	5	5	5/5	-	-	891
2	10	5	24	108	54	6	6	2/7	5/5	-	-
2	12	6	10	44	22	1	1	1/1	-	-	319
2	12	6	15	66	33	1	1	0/1	1/1	-	743
2	12	6	20	88	44	546	546	$691/\geq718$	$0/\geq27$	$\geq27/\geq27$	-
2	16	8	14	60	30	5	5	5/5	-	-	618
2	16	8	21	90	45	5	5	0/5	5/5	-	-
2	20	10	18	76	38	3	3	3/3	-	-	1007
3	9	3	3	36	12	3	5	2/7	5/5	-	66
3	9	3	4	48	16	38	83	388/495	333/334	1/1	145
4	12	3	2	44	11	20	70	319/321	1/2	1/1	55
4	16	4	2	40	10	1	1	0/1	1/1	-	44

A *Latin square of side (order)* n is an $n \times n$ array in which each cell contains a single symbol from an n-set S, such that each symbol occurs exactly once in each row and exactly once in each column. A Latin square exists for any integer side n. An $m \times n$ *Latin rectangle* is an $m \times n$ array in which each cell contains a single symbol from an n-set S, such that each symbol occurs exactly once in each row and at most once in each column. An $m \times n$ Latin rectangle can always be completed to a Latin square of side n.

Let L be a Latin square of side n on symbol set E_3 with rows indexed by the elements of the n-set E_1 and columns indexed by the elements of the n-set E_2. Let $\tau = \{(x_1, x_2, x_3) : L(x_1, x_2) = x_3\}$. Let $\{a, b, c\} = \{1, 2, 3\}$. The (a, b, c)-conjugate of L, $L_{(a,b,c)}$ has rows indexed by E_a, columns by E_b, and symbols by E_c, and is defined by $L_{(a,b,c)}(x_a, x_b) = x_c$ for each $(x_1, x_2, x_3) \in \tau$.

Two Latin squares L_1 and L_2 are *equivalent (isotopic)* if there are three bijections from the rows, columns and symbols of L_1 to the rows, columns and symbols, respectively of L_2 that map L_1 to L_2. L_1 and L_2 are *main class equivalent* if L_1 is equivalent to any conjugate of L_2.

Two Latin squares of side n $L_1 = (a_{ij})$ on symbol set S_1 and $L_2 = (b_{ij})$ on symbol set S_2 are *orthogonal* if every element in $S_1 \times S_2$ occurs exactly once among the n^2 pairs $(a_{ij}, b_{ij}), i, j = 1, 2, ..., n$. A set of Latin squares of side n, $L_1, L_2, ..., L_m$ is *mutually orthogonal(a set of MOLS)*, if L_i and L_j are orthogonal for $i, j = 1, 2, ..., n, i \neq j$. A set of MOLS of side n can have at most $n - 1$ elements.

Different types of equivalence of MOLS can be defined. In this paper we use the following definitions of conjugates and equivalence of MOLS:

Let $\mathcal{M}$ be a set of m MOLS $L_1, L_2, ..., L_m$ of side n on symbol sets respectively $E_3, E_4, ..., E_{m+2}$ and with rows and columns indexed by the elements of the n-sets E_1 and E_2 respectively. Let $\tau = \{(x_1, x_2, ..., x_{m+2}) : L_i(x_1, x_2) = x_{i+2}, i = 1, 2, ..., m\}$. Let $\{a_1, a_2, ..., a_{m+2}\} = \{1, 2, ..., m+2\}$. The $(a_1, a_2, ..., a_{m+2})$ conjugate of $\mathcal{M}$, $\mathcal{M}_{(a_1,a_2,...,a_{m+2})}$ contains the Latin squares $L_i : L_i(a_1, a_2) = a_{i+2}$, $i = 1, 2, ..., m$ for each $(x_1, x_2, ..., x_{m+2}) \in \tau$.

Two sets of MOLS $\mathcal{M}_1$ and $\mathcal{M}_2$ are *equivalent (isotopic)* if there are three bijections from the rows, columns and symbols of $\mathcal{M}_1$ to the rows, columns and symbols, respectively of $\mathcal{M}_2$ that map $\mathcal{M}_1$ to $\mathcal{M}_2$. $\mathcal{M}_1$ and $\mathcal{M}_2$ are *main class equivalent* if $\mathcal{M}_1$ is equivalent to any conjugate of $\mathcal{M}_2$.

Proposition 3.1 *Let D be a 2-(v,k,λ) design and $v = 2k$.*

1) D is doubly resolvable iff it is resolvable and each set of k points is either incident with no block, or with at least two blocks of the design.

2) If D is doubly resolvable and at least one set of k points is in m blocks, and the rest in 0 or more than m blocks, then D has at least one maximal m-MOR, no i-MORs for $i > m$ and no maximal i-MORs for $i < m$.

The proof is based on:

1) If one block of a parallel class is known, the point set of the second one is known too. Suppose D has m-MOR $\mathcal{R}_1, \mathcal{R}_2, ...\mathcal{R}_m$. Consider a block with exactly $p - 1$ equal blocks. Denote by $1, 2, ..., p$ the parallel classes of $\mathcal{R}_1$, in which these blocks are, the blocks themselves by $1_1, 2_1, ..., p_1$ and the second blocks in the classes by $1_2, 2_2, ..., p_2$. Since block i_1 should be with block j_2 $(i, j = 1, 2, ..., p)$ at most once in a parallel class of the m-MOR, the class numbers of the second blocks form an $m \times p$ Latin rectangle. An example for $p = 4$ and $m = 3$ is presented in Fig.1.

2) A $2 \times m$ Latin rectangle can be completed to a Latin square of order m.

Proposition 3.2 *Let $l_{q-1,m}$ be the number of main class inequivalent sets of $q - 1$ MOLS of side m. Let $q = v/k$ and $m \geq q$. Let the 2-(v,k,$m\lambda$) design D be a true m-fold multiple of a resolvable 2-(v,k,λ) design d. If $l_{q-1,m} > 0$, then*

Figure 1: 4 equal parallel classes of 3 mutually orthogonal resolutions, $v = 2k$

$$\begin{array}{ccccc} & 1 & 2 & 3 & 4 \\ \mathcal{R}_1 & 1_1 1_2 & 2_1 2_2 & 3_1 3_2 & 4_1 4_2 \\ \mathcal{R}_2 & 1_1 2_2 & 2_1 1_2 & 3_1 4_2 & 4_1 3_2 \\ \mathcal{R}_3 & 1_1 3_2 & 2_1 4_2 & 3_1 1_2 & 4_1 2_2 \end{array} \Longrightarrow \begin{array}{cccc} \multicolumn{4}{c}{\textit{Latin rectangle}} \\ 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \end{array}$$

D is doubly resolvable and has at least $\binom{\frac{r}{m} - 1 + l_{q-1,m}}{\frac{r}{m}}$ *m-MORs.*

The proof is based on:

Consider a resolution $\mathcal{R}_1$ of D, such that each parallel class of $\mathcal{R}_1$ is equal to a parallel class of the resolution $\mathcal{T}$ of d. We can partition the collection of parallel classes of $\mathcal{R}_1$ into subcollections $P_1, P_2, ..., P_{r/m}$ of size m, such that the classes in a subcollection are equal. m-MOR containing $\mathcal{R}_1$ can be constructed as follows: the first block of each class equals the first block of the corresponding class of $\mathcal{R}_1$ and the other blocks of P_i form a set $\mathcal{M}_i$ of $q-1$ MOLS of side m. An example for $m = 4$ and $q = 3$ is presented in Fig. 2a.

Figure 2: 4 equal parallel classes of 4 mutually orthogonal resolutions, $v = 3k$

a)relation to a set $\mathcal{M}$ of two MOLS of side 4

$$\begin{array}{ccccc} & 1 & 2 & 3 & 4 \\ \mathcal{R}_1 & 1_1 1_2 1_3 & 2_1 2_2 2_3 & 3_1 3_2 3_3 & 4_1 4_2 4_3 \\ \mathcal{R}_2 & 1_1 2_2 3_3 & 2_1 1_2 4_3 & 3_1 4_2 1_3 & 4_1 3_2 2_3 \\ \mathcal{R}_3 & 1_1 3_2 4_3 & 2_1 4_2 3_3 & 3_1 1_2 2_3 & 4_1 2_2 1_3 \\ \mathcal{R}_4 & 1_1 4_2 2_3 & 2_1 3_2 1_3 & 3_1 2_2 4_3 & 4_1 1_2 3_3 \end{array} \Longrightarrow \begin{array}{ccccccccc} \multicolumn{9}{c}{\mathcal{M} = \mathcal{M}_{(1,2,3,4)}} \\ 1 & 2 & 3 & 4 & & 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 & & 3 & 4 & 1 & 2 \\ 3 & 4 & 1 & 2 & & 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 1 & & 2 & 1 & 4 & 3 \end{array}$$

b)automorphism α transforming first blocks into second blocks

$$\begin{array}{ccccc} & 1 & 2 & 3 & 4 \\ \mathcal{R}_1 & 1_2 1_1 1_3 & 2_2 2_1 2_3 & 3_2 3_1 3_3 & 4_2 4_1 4_3 \\ \mathcal{R}_2 & 1_2 2_1 3_3 & 2_2 1_1 4_3 & 3_2 4_1 1_3 & 4_2 3_1 2_3 \\ \mathcal{R}_3 & 1_2 3_1 4_3 & 2_2 4_1 3_3 & 3_2 1_1 2_3 & 4_2 2_1 1_3 \\ \mathcal{R}_4 & 1_2 4_1 2_3 & 2_2 3_1 1_3 & 3_2 2_1 4_3 & 4_2 1_1 3_3 \end{array}$$

c)relation to $\mathcal{M}_{(1,3,2,4)}$ - the $(1,3,2,4)$ conjugate of S

$$\begin{array}{ccccc} & 1 & 2 & 3 & 4 \\ \mathcal{R}_1 & 1_1 1_2 1_3 & 2_1 2_2 2_3 & 3_1 3_2 3_3 & 4_1 4_2 4_3 \\ \mathcal{R}_2 & 1_1 2_2 4_3 & 2_1 1_2 3_3 & 3_1 4_2 2_3 & 4_1 3_2 1_3 \\ \mathcal{R}_3 & 1_1 3_2 2_3 & 2_1 4_2 1_3 & 3_1 1_2 4_3 & 4_1 2_2 3_3 \\ \mathcal{R}_4 & 1_1 4_2 3_3 & 2_1 3_2 4_3 & 3_1 2_2 1_3 & 4_1 1_2 2_3 \end{array} \Longrightarrow \begin{array}{ccccccccc} \multicolumn{9}{c}{\mathcal{M}_{(1,3,2,4)}} \\ 1 & 2 & 3 & 4 & & 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 & & 4 & 3 & 2 & 1 \\ 3 & 4 & 1 & 2 & & 2 & 1 & 4 & 3 \\ 4 & 3 & 2 & 1 & & 3 & 4 & 1 & 2 \end{array}$$

Permutation of design classes, numbers of equal classes, or resolutions of the m-MOR invokes respectively permutation of columns, symbols and rows of all Latin squares in $\mathcal{M}_i$. A nontrivial point automorphism α can invoke a transformation of $\mathcal{M}_i$ into one of its conjugates (an example is presented in Fig. 2b,c.) or into a conjugate of $\mathcal{M}_j, i, j = 1, 2, ..., r/m, i \neq j$. Thus there are at least $l_{q-1,m}$ inequivalent ways to fix $\mathcal{M}_i$.

The number of different ways to choose u integers $i_1, i_2, ...i_u$, such that $i_1 + i_2 + ... + i_u = w$ is $Q(u, w) = \binom{u+w-1}{w}$.

Corollary 3.3 *Let l_m be the number of main class inequivalent Latin squares of side m. Let $v/k = 2$ and $m \geq 2$. Let the 2-$(v,k,m\lambda)$ design D be a true m-fold multiple of a 2-(v,k,λ) design d, and let d be resolvable, but not doubly resolvable. Then D is doubly resolvable and has at least $\binom{\frac{r}{m}-1+l_m}{\frac{r}{m}}$ m-MORs, no maximal i-MORs for $i < m$ and no i-MORs for $i > m$.*

References

[1] R. J. R. Abel, E. R. Lamken, J. Wang, A few more Kirkman squares and doubly near resolvable BIBDS with block size 3, *Discr. Math.* 308, 2008, 1102-1123.

[2] Th. Beth, D. Jungnickel, H. Lenz, *Design Theory*, Cambridge University Press, 1993.

[3] C. J. Colbourn, J. H. Dinitz (Eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton, FL., 2007.

[4] R. Mathon, A. Rosa, 2-(v, k, λ) designs of small order, *The CRC Handbook of Combinatorial Designs*, Boca Raton, FL., 2007, 3-41.

[5] N. V. Semakov, V. A. Zinoviev, Equidistant q-ary codes with maximal distance and resolvable balanced incomplete block designs, *Probl. Inform. Transm.* 4, 1968, 1-7.

[6] S. Topalova, S. Zhelezova, On the classification of doubly resolvable designs, *Proc. Fourth Intern. Workshop OCRT*, Pamporovo, Bulgaria, 2005, 265-268.

[7] V. D. Tonchev, *Combinatorial Configurations*, Longman Scientific and Technical, New York, 1988.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 286-291*

New results on s-extremal additive codes over $\mathbb{F}_4$

Zlatko Varbanov vtgold@yahoo.com
Department of Mathematics and Informatics,
Veliko Tarnovo University, 5000 Veliko Tarnovo, BULGARIA

Abstract. The purpose of this paper is to study s-extremal additive codes over F_4. The concept of s-extremality was introduced in [2] and the s-extremal additive codes with minimum distance up to 4 were classified. In this paper, our goal is to construct (or to classify if possible) new s-extremal codes with minimum distance $d = 5$ or 6. For $d = 5$ we classify the codes of length 13, and we construct 1075 new codes of length 14. For $d = 6$ we obtain that there is a unique code of length 14.

1 Introduction

The shadow of a binary self-dual code was introduced by Conway and Sloane [5] in order to get additional constraints in the weight enumerator of a singly-even binary self-dual code. Let C be a singly-even binary self-dual code. The shadow S of C is

$$S = \{w \in \mathbb{F}_2^n | (v, w) \equiv \tfrac{1}{2} wt(v) \ (mod\ 2) \text{ for all } v \in C\},$$

where (v, w) is an Euclidean inner product in $\mathbb{F}_2^n$.

Let d be the minimum distance of C and s be the minimum weight of S. It is known [1] that $2d + s \leq n/2 + 4$ except in the case $n \equiv 22 \pmod{24}$ and $d = 4[n/24] + 6$ where $2d + s = n/2 + 8$. Binary codes attaining these bounds are called s-extremal.

After the introduction of s-extremal binary self-dual codes, it is natural to ask whether there exists a concept of s-extremal additive F_4 codes. If so, can we classify them? This concept was introduced by Bautista, Gaborit, Kim, and Walker [2]. They gave a bound on the possible lengths of such codes related to their distances for even d and classified them up to $d = 4$. Also, they gave possible lengths (only strongly conjectured for odd d) and (shadow) weight enumerators for which there exist s-extremal codes with $5 \leq d \leq 11$.

In this paper, we investigate a s-extremal codes with minimum distance 5 and 6. For $d = 5$ we give a full classification of the codes of length 13 and we construct 1075 new codes of length 14. For $d = 6$ we obtain that there is a unique code (up to equivalence) of length 14.

2 Preliminaries

Let $F_4 = \{0, 1, \omega, \bar{\omega}\}$ with convention that $\bar{\omega} = \omega^2 = 1 + \omega$. We recall some definitions on additive codes over F_4 from [4], [6].

Definition 2.1 *An* ***additive code*** C ***over*** F_4 ***of length*** n *is an additive subgroup of* F_4^n. *As* C *is a free* F_2*-module, it has size* 2^k *for some* $0 \le k \le 2n$. *We call* C *an* $(n, 2^k)$ *code. It has a basis, as a* F_2*-module, consisting of* k *basis vectors; a* ***generator matrix*** *of* C *is any* $k \times n$ *matrix with entries in* F_4 *whose rows are a basis of* C.

Definition 2.2 *The* ***weight*** *of a codeword* $c \in C$ *(denoted by* $wt(c)$*) is the number of nonzero components of c and the* ***minimum weight*** *(or* ***minimum distance****)* d *of* C *is the smallest weight among all nonzero codewords in* C. *We call* C *an* $(n, 2^k, d)$ *code.*

There is an inner product arising from the trace map. The trace map $Tr : F_4 \to F_2$ is given by $Tr(x) = x + x^2$. The *conjugate* of $x \in F_4$, denoted $\bar{x}$, is the following image: $\bar{0} = 0, \bar{1} = 1$, and $\bar{\bar{\omega}} = \omega$.

Definition 2.3 ***Trace inner product*** *of two vectors* $x = (x_1, x_2, \ldots, x_n)$, $y = (y_1, y_2, \ldots, y_n)$ *in* F_4^n *is*

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

Definition 2.4 *If* C *is an additive code, its* ***dual****, denoted* $C^{\perp}$, *is the additive code* $\{x \in F_4^n | x \star c = 0 \text{ for all } c \in C\}$. *If* C *is an* $(n, 2^k)$ *code, then* $C^{\perp}$ *is an* $(n, 2^{2n-k})$ *code. As usual,* C *is* ***self-orthogonal*** *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. 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 [6]. If C is self-dual but some codeword has odd weight (in which case the code cannot be F_4-linear), the code is *Type I*. There is a bound on the minimum weight of an additive self-dual code ([10], 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

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

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

A code that meets the appropriate bound is called *extremal*.

Definition 2.5 *Two additive codes* C_1 *and* C_2 *are* ***equivalent*** *if 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* F_4*, followed by conjugation of some of the coordinates.*

Definition 2.6 *Let C be an additive F_4-code of length n which is self-dual with respect to the trace inner product. The* **shadow** *$S = S(C)$ of C is given by*

$$S = \{w \in \mathbb{F}_4^n | v \star w \equiv wt(v) \ (mod\ 2) \text{ for all } v \in C\}.$$

If C is $Type\ II$ $S(C) = C$, while if C is $Type\ I$ $S(C)$ is a coset of C.

The next theorem is the F_4-analog of Theorem 1 in [1].

Theorem 2.7 [2] *Let C be a Type I additive code over F_4 of length n, let $d = d_{min}(C)$ be the minimum distance of C, let $S = S(C)$ be the shadow of C, and let $s = wt_{min}(S)$ be the minimum weight of S. Then $2d + s \le n + 2$ unless $n = 6m + 5$ and $d = 2m + 3$, in which case $2d + s = n + 4$.*

This theorem motivates the following definition.

Definition 2.8 *Let C be a Type I additive code over F_4 of length n, let $d = d_{min}(C)$ be the minimum distance of C, let $S = S(C)$ be the shadow of C, and let $s = wt_{min}(S)$ be the minimum weight of S. We call C* ***s-extremal*** *if the bound of Theorem 7 is met, i.e., if $2d + s = n + 2$ except $n = 6m + 5$ and $d = 2m + 3$, in which case $2d + s = n + 4$.*

There are some known bounds for the length of s-extremal codes.

Theorem 2.9 [9] *Let C be an $(n, 2^n, d)$ s-extremal code. Then $n \ge 3d - 4$. If d is even, then $3d - 4 \le n \le 3d - 2$.*

For odd $d > 3$, there are the following bounds [9]:

$d = 5 \ : \ 11 \le n \le 15$	$d = 7 \ : \ 17 \le n \le 21$
$d = 9 \ : \ 23 \le n \le 27$	$d = 11 \ : \ 29 \le n \le 33$

3 Lengthening of graph codes

We recall the lengthening of graph codes from [12]. A *graph* is a pair $G = (V, E)$, where $V = \{v_0, v_1, \ldots, v_n\}$ is a set of n *vertices* (or *nodes*), and E is a set of distinct pairs of elements from V, i.e., $E \subseteq V \times V$. A pair $\{v_i, v_j\} \in E$ is called *edge*. We will only consider *undirected* graphs, which are graphs where E is a set of distinct unordered pairs of elements from V, and no self-loops ($\{v_i, v_i\} \notin E$). A graph may be represented by an *adjacency matrix* Γ. This is a $|V| \times |V|$ matrix where $\Gamma_{i,j} = 1$ if $\{v_i, v_j\} \in E$ and $\Gamma_{i,j} = 0$ otherwise. The adjacency matrix of an undirected graph will be symmetric, i.e., $\Gamma_{i,j} = \Gamma_{j,i}$, and $\Gamma_{i,i} = 0$ (because no self-loops).

A *graph code* is an additive self-dual code over F_4 with generator matrix $C = \Gamma + \omega I$ where I is the identity matrix and Γ is the adjacency matrix of a undirected graph, which must be symmetric with 0's along the diagonal.

Example: $\Gamma = \begin{pmatrix} 0\,1\,1 \\ 1\,0\,1 \\ 1\,1\,0 \end{pmatrix}, \quad C = \Gamma + \omega I = \begin{pmatrix} \omega\,1\,1 \\ 1\,\omega\,1 \\ 1\,1\,\omega \end{pmatrix}.$

Schlingemann [11] first proved the following theorem in terms of *quantum stabilizer states*.

Theorem 3.1 ([11], [7]) *For any additive self-dual code, there is an equivalent graph code. This means that there is a one-to-one correspondence between the set of undirected graphs and the set of additive self-dual codes over* F_4.

We have seen that every graph represents an additive self-dual code over F_4, and that every additive self-dual code over F_4 can be represented by a graph. It follows from Theorem 3.1 that, without loss of generality, we can restrict our study of additive self-dual codes over F_4 to those with generator matrices of the form $\Gamma + \omega I$ (graph form).

The lengthening of graph codes is based on the following theorem.

Theorem 3.2 [12] *If* G *is a generator matrix of a graph code* C *of length* n, *and* x *is a binary vector, then*

$$G' = \left(\begin{array}{c|c} G & x^t \\ \hline x & \omega \end{array} \right)$$

is a generator matrix of a graph code of length $n+1$.

Using this construction we obtain new results described in the following section.

4 Results

To obtain new s-extremal additive codes we use some preliminary results.

Theorem 4.1 (Theorem 4.1 [12]) *There are* 85845 *nonequivalent additive self-dual* $(13, 2^{13}, 5)$ *codes,* 2 *nonequivalent* $(14, 2^{14}, 6)$ *Type I codes, and* 1020 *nonequivalent* $(14, 2^{14}, 6)$ *Type II codes.*

These codes were obtained by lengthening of graph codes. Then, their generator matrices are given in graph form and we can use the same method to get new results.

It is known that the weight enumerator of the s-extremal codes of length 13 is $C(z) = 1 + 39z^5 + 156z^6 + 468z^7 + 1053z^8 + 1690z^9 + 2028z^{10} + 1716z^{11} + 858z^{12} + 183z^{13}$ and the number of these codes is ≥ 9 [2]. Using the results in Theorem 4.1 by computer check we obtain the following classification.

Theorem 4.2 *There are exactly* 33428 *nonequivalent* s*-extremal codes of length* 13.

In Table 1 we give full classification (by group order) of the s-extremal codes of length 13.

Table 1 Number of s-extremal codes of length 13 with group order α

α	1	2	3	4	6	8	12	52	156
Number	32134	1228	5	49	7	1	2	1	1

In our work we use the program package $Q-Extension$ [3] to obtain the number of nonequivalent codes and their group orders.

We use the generator matrices of the codes of length 13 to obtain new codes of length 14 with $d=5$. By lengthening of graph codes we construct 1075 new codes with these parameters (one code is already known [8]). Therefore

Theorem 4.3 *There are at least* 1076 *nonequivalent s-extremal codes of length* 14 *with $d=5$.*

In Table 2 we give a group order of the constructed s-extremal codes of length 14. It is known [2] that these codes have weight enumerator $C(z) = 1+42z^5+119z^6+408z^7+1281z^8+2492z^9+3486z^{10}+3864z^{11}+3038z^{12}+1386z^{13}+267z^{14}$, and shadow enumerator $S(z) = 308z^6+2352z^8+7224z^{10}+5936z^{12}+564z^{14}$.

Table 2 Number of s-extremal codes of length 14 with group order α

α	1	2	3	4	6	8	24	28
Number	≥ 915	≥ 125	≥ 8	≥ 16	≥ 5	≥ 5	≥ 1	≥ 1

By results in Theorem 4.1 we obtain that there is a unique s-extremal code of length 14. This code has weight enumerator $C(z) = 1+161z^6+576z^7+1113z^8+2240z^9+3738z^{10}+4032z^{11}+2870z^{12}+1344z^{13}+309z^{14}$, shadow enumerator $S(z) = 21z^4+203z^6+2562z^8+7014z^{10}+6041z^{12}+543z^{14}$, and group order 48. The generator matrix G_{14} of this code is

$$G_{14} = \begin{pmatrix}
\omega & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & \omega & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & \omega & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & \omega & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & \omega & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & \omega & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & \omega & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & \omega & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \omega & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & \omega & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & \omega & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & \omega & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \omega & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & \omega
\end{pmatrix}$$

References

[1] C. Bachoc, P. Gaborit, Designs and self-dual codes with long shadows, *J. Combin. Theory* Ser.A 105, 2004, 15-34.

[2] E. Bautista, P. Gaborit, J.-L. Kim, J. Walker, *s*-extremal additive $\mathbb{F}_4$ codes, *Adv. Math. Commun.* 1, 2007, 111-130.

[3] I. Bouyukliev, What is Q-extension?, *Serdica J. Comput.* 1, 2007, 115-130.

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

[5] J. H. Conway, N. J. A. Sloane, A new upper bound on the minimal distance of selfdual codes, *IEEE Trans. Inform. Theory* 36, 1990, 1319-1333.

[6] P. Gaborit, W. C. Huffman, J.-L. Kim, V. Pless, On additive GF(4)-codes, *DIMACS Workshop Codes Assoc. Schemes*, DIMACS Series Discr. Math. Theoret. Comp. Sci., AMS 56, 2001, 135-149.

[7] D. G. Glynn, On self-dual quantum codes and graphs, submitted to Electr. J. Combin. http://homepage.mac.com/dglynn/.cv/dglynn/Public/SD-G3.pdf-link.pdf

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

[9] S. Han, J.-L. Kim, Upper bounds for the length of *s*-extremal codes over $\mathbb{F}_2, \mathbb{F}_4$, and $\mathbb{F}_2 + u\mathbb{F}_2$, submitted, 2007.

[10] E. M. Rains, N. J. A. Sloane, Selfdual codes, in Handbook of Coding Theory, ed. V. S. Pless and W. C. Huffman, Amsterdam: Elsevier (1998), 177-294.

[11] D. Schlingemann, Stabilizer codes can be realized as graph codes, *Quantum Inf. Comput.* 2, 2002, 307-323, arXiv:quant-ph/0111080.

[12] Z. Varbanov, Some new results for additive self-dual codes over $GF(4)$, *Serdica J. Comput.* 1, 2007, 213-227.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 292-299*

On mobile sets in the binary hypercube

Yu. L. Vasil'ev, S. V. Avgustinovich*, D. S. Krotov krotov@math.nsc.ru
Department of Mechanics and Mathematics, Novosibirsk State University
Pirogova Str. 2, Novosibirsk, 630090, RUSSIA

Abstract. If two distance-3 codes have the same neighborhood, then each of them is called a mobile set. In the $(4k+3)$-dimensional binary hypercube, there exists a mobile set of cardinality $2 \cdot 6^k$ that cannot be split into mobile sets of smaller cardinalities or represented as a natural extension of a mobile set in a hypercube of smaller dimension.

1 Introduction

By E^n we denote the metric space of all length-n binary words with the Hamming metric. The space E^n is called the binary, or unary, or Boolean hypercube. The basis vector with one in the ith coordinate and zeros in the other is denoted by e_i. A subset M of E^n is called a 1-code if the radius-1 balls with centers in M are disjoint. The union of the radius-1 balls with the centers in M is called the *neighborhood* of M and denoted by $\Omega(M)$, i.e.,

$$\Omega(M) = \{x \in E^n : d(x, M) \leq 1\}.$$

If a 1-code M satisfies $\Omega(M) = E^n$, then it is called perfect, or a *1-perfect code*. 1-Perfect codes exist only when the dimension has the form $n = 2^k - 1$. For $n = 7$, such a code is unique (up to isometries of the space), the linear Hamming code. For $n = 15$, the problem of characterization and enumeration of the 1-perfect codes is not solved yet, in spite of the increasing computation abilities (considerable results are obtained in [10, 2]). In this context, it is topical to study objects that generalize, in different senses, the concept of 1-perfect code and exist in intermediate dimensions, not only of type $n = 2^k - 1$. Examples of such objects are the perfect colorings (in particular, with two colors [1]), the centered functions [8], and the mobile sets, discussed in this paper.

A set $M \subseteq E^n$ is called *mobile* (*m.s.*) iff:
1) M is a 1-code;
2) there exists a 1-code M' disjoint with M and with the same neighborhood, i.e., $M \cap M' = \emptyset$ and $\Omega(M) = \Omega(M')$;

such a set M' will be called the *alternative* of M.

In other words, a 1-code is a m.s. iff it has an alternative.

For every odd $n = 2m+1$, we can construct a linear (i.e., closed with respect to coordinatewise modulo-2 addition) m.s. in E^n:

$$M = \{(x, x, |x|) : x \in E^m\}. \qquad (1)$$

*This author is partially supported by the RFBR grant 07-01-00248.

(Here and below $|x|$ denotes the modulo-2 sum of the coordinates of x.) Respectively,

$$M' = \{(x, x, |x| \oplus 1) : x \in E^m\}.$$

It is not difficult to check the conditions 1 and 2 for these M and M'.

Our main goal is to prove the following:

Theorem. *For all $n \geq 7$ congruent to 3 modulo 4, there exists an irreducible unsplittable mobile set in E^n.*

A nonempty m.s. M is called *splittable* (*unsplittable*), iff if can (respectively, cannot) be represented as the union of two nonempty m.s. The concept of reducibility, which will be defined in Section 4, reflects a natural reducibility of mobile sets to mobile sets in the hypercube of the two-less dimension.

A simple way to construct a m.s. in a hypercube of a code dimension $n = 2^k - 1$ is the following. Let C and C' are 1-perfect codes in E^n. Then $M = C \setminus C'$ is a m.s. Indeed, we can take $C' \setminus C$ as M'. The cardinality of this m.s. is $C - |C \cap C'|$. We study the existence of m.s. that cannot be reduced to code dimensions.

In Section 2 we define extended mobile sets; that concept is convenient for the description of our construction. In Section 3 we describe a connection between the mobile sets and the i-components, which were studied earlier. In Section 4 we describe a construction of increasing dimension for mobile sets; that construction leads to the natural concept of a reducible m.s. In Section 5 we give the main construction and prove Theorem. In the final section, we formulate several problems.

2 Extended mobile sets

Like as with 1-perfect codes, it is sometimes convenient to work with mobile sets extending them by the all-parity check to the next dimension. In some cases we get more symmetrical objects, which simplifies proofs and formulations of statements. And. Some statements become more simple and intuitive while being formulated for the extended case, although geometrical interpretations of extended objects can seem to be not so elegant and natural as for the original.

Recall that the *extension* of the set $M \subseteq E^n$ is the set $\overline{M} \subseteq E^{n+1}$ obtained by the addition of the all-parity-check bit to all the words of M:

$$\overline{M} = \{(x, |x|) : x \in M)\} \quad \text{or} \quad \overline{M} = \{(x, |x| \oplus 1) : x \in M)\}.$$

Puncturing the ith coordinate for some set of words in E^n means removing the ith symbol from all the words of the set (the result is in E^{n-1}). Obviously, the extension and puncturing the last coordinate lead to the original set; so, these operations are opposite to each other, in some sence.

A set $\overline{M} \subseteq E^n$ is called *extended mobile* (an *e.m.s.*) iff it can be obtained as the extension of some m.s.

We will use the following lemma, which gives alternative definitions of an e.m.s. As a usual m.s., an e.m.s. M can be defined together with some other e.m.s. M', which can also be referred as an *alternative* of M (usually, it is clear from the context what we are talking about, mobile sets or extended mobile sets). For the formulation of the lemma and further using, it is convenient to define the concept of the *spherical neighborhood*

$$\dot\Omega(M) = \Omega(M) \setminus M,$$

which, for the extended mobile sets, plays the role similar to the role of the usual ("ball") neighborhood for the m.s. In particular, part (c) of Lemma 1 defines an e.m.s. and an alternative similarly to the case of a m.s.

Lemma 1 (alternative definitions of an e.m.s.). *Let M and M' be disjoint 1-codes in E^n, and let their vectors have the same parity (either all vectors are even, or odd). Let $i \in \{1,\ldots,n\}$. The following conditions are equivalent and imply that M (as like as M') is an e.m.s.*
(a) *The sets M_i and M'_i obtained from M and M' by puncturing ith coordinate are mobile and, moreover, are alternatives of each other.*
(b) *The (bipartite) distance-2 graph $G(M \cup M')$ of the union $M \cup M'$ has the degree $n/2$.*
(c) $\dot\Omega(M) = \dot\Omega(M')$. △

Taking into account (b) and the existence of a linear m.s., we have the following:

Corollary 1. *Nonempty m.s. (e.m.s.) exist in E^n if and only if n is odd (resp., even).*

3. i-Components

A m.s. M is called an *i-component* iff $\Omega(M) = \Omega(M \oplus e_i)$. Consider the set M_i obtained from M by puncturing the ith coordinate. Let us construct the so-called minimal-distance graph $G(M_i)$ with the vertex set M_i, connecting vertices at the distance 2 from each other. The proof of the following lemma is similar to Lemma 1, and we omit it.

Lemma 2. A 1-code M is an i-component if and only if the graph $G(M_i)$ is regular of degree $(n-1)/2$ and bipartite.

So, Lemmas 1 and 2 establish a correspondence between pairs of alternative m.s. in E^{n-1} and i-components in E^{n+1} (for fixed i, say, $i = n+1$). This correspondence is evident as both objects correspond to a set in E^n whose distance-2 graph is bipartite and has the degree $n/2$. In the first case, all the vertices of this set have the same parity. In the second case, this is not not necessary, but the subsets of different parity will correspond to a partition of the i-component into independent i-components, "i-even" and "i-odd". Formally, we can formulate the following.

Corollary 2. *Sets $M, M' \subseteq E^{n-1}$ are a m.s. and an alternative if and only if the set*

$$\{(x, |x|, 0) : x \in M\} \cup \{(x, |x|, 1) : x \in M'\}$$

is an i-component with $i = n + 1$.

Corollary 3. *A set $M \subseteq E^{n+1}$ is an i-component with $i = n + 1$ if and only if the sets*

$$M_a^b = \{x : (x, |x| \oplus a, b) \in M\}, \quad a, b \in \{0, 1\}$$

are m.s., where M_a^0 and M_a^1 are alternatives to each other (the sets M_0^0 and M_0^1 correspond to the "i-even" part of the i-component; M_1^0 and M_1^1, to the "i-odd"; each of these parts can be empty; and if both are nonempty, then the i-component is splittable).

An example of i-component is the linear m.s. (1), $i = n$. Formerly [4, 5] many examples of nonlinear i-components were constructed. Each of them is embeddable to a 1-perfect codes and has the cardinality, divisible by the cardinality of the linear component. Moreover, it was only proved that these i-components cannot be split into smaller i-components. Their splittability onto mobile sets are still questionable. So, in spite of the fact that the researches are devoted to common problems and a common approach, the lines are slightly different and the results do not overlap but complement each other: we give the embeddability to 1-perfect codes up (which is a weakening) but deal with a stronger splittability and a wider specter of dimensions.

4 Reducibility

Lemma 3 (on the linear extension of a m.s.). *Let $M \subseteq E^n$ be an e.m.s. and let $M' \subseteq E^n$ be an alternative of M. Then the set*

$$R = \{(x, 0, 0) : x \in M\} \cup \{(x, 1, 1) : x \in M'\} \tag{2}$$

is an e.m.s. with an alternative

$$R' = \{(x, 1, 1) : x \in M\} \cup \{(x, 0, 0) : x \in M'\}.$$

Proof. Condition (b) of Lemma 1 for M and M' implies the validity of this condition for R and R'. $\triangle$.

An e.m.s. $R \in E^n$ is called *reducible* iff it can be obtained by the construction (2) and applying some isometry of the space (i.e., a coordinate permutation and the inversion in some coordinates). A m.s. is called *reducible* iff the corresponding e.m.s. is reducible.

So, the existence of reducible m.s. is reduced to the existence of m.s. in smaller dimensions. From this point of view, the formulation of the main theorem is natural.

Remark. As we can see from Corollary 3, any i-component is either reducible m.s. or can be split into two i-components ("i-even" and "i-odd"), which are reducible m.s. In particular, the linear m.s. (1) is reducible. Moreover, the linear e.m.s., up to a coordinate permutation, can be obtained from the trivial e.m.s. $\{00\}$ in E^2 by sequential applying the construction from Lemma 3.

5 Proof of Theorem

Let us fix n divisible by 4: $n = 4k$. Partition the coordinate numbers into k groups with 4 numbers in each group; rename the corresponding orts as follows: $e_0^1, e_1^1, e_2^1, e_3^1, e_0^2, \ldots, e_3^k$. In each quadruple of type $\{e_0^i, e_1^i, e_2^i, e_3^i\}$ we chose arbitrarily (there exist 6 possibilities) a pair of different orts e_j^i and e_t^i; by the index of the pair we shell mean the number $p\{j,t\}$ where

$$p\{0,1\} = p\{2,3\} = 0, \qquad p\{0,2\} = p\{1,3\} = 1, \qquad p\{0,3\} = p\{1,2\} = 2.$$

Summarizing the chosen pairs for all $i = 1, 2, \ldots, k$, we get a vector of weight $2k$, which will be called *standard*. Totally, there exist 6^k standard vectors. By the index $I(v)$ of a standard vector v we shell mean the modulo-3 sum of the indexes of all the pairs of orts that constitute v.

Let us partition the set of standard vectors into disjoint subsets S_0, S_1, and S_2 in compliance with the indexes of the vectors.

Claim 1. *Let $i \neq j$, $i, j \in \{0, 1, 2\}$. Then the distance-two graph $G(S_i \cup S_j)$ induced by the set $S_i \cup S_j$ is bipartite and regular of degree $2k$.*

We first note that the graphs $G(S_i)$ and $G(S_j)$ are empty. Indeed, consider two vectors $v, u \in S_i$. Either v and u differ in exactly one quadruple of coordinates, and thus $d(v, u) = 4$, because $I(u) = I(v)$; or v and u differ in more than one quadruples, and thus $d(v, u) \geq 4$, because the distance between standard vectors is even in every quadruple. So, $G(S_i \cup S_j)$ is bipartite.

Further, it is easy to see that every vector of index i has exactly two distance-2 neighbors in S_j. This means that the graph degree is $2k$. Claim 1 is proved.

So, S_0 (for example) is an e.m.s. of cardinality $2 \cdot 6^{k-1}$.

Claim 2. *The e.m.s. S_0 is unsplittable.* Assume that $P \subseteq S_0$ and $Q = S_0 \backslash P$ are nonempty e.m.s. Then P and Q have alternative, say P' and Q'.

We will first show that

(*) *P' (similarly, Q') consists of only standard vectors*, i.e., such vectors that contains exactly two ones in every quadruple. Indeed, otherwise P' contains a vector with non-standard quadruple; consequently, $\Omega(P')$ contains a vector with two non-standard quadruples. But $\Omega(P)$ consists of vectors with exactly one non-standard quadruple and, thus, cannot coincide with $\dot{\Omega}(P')$, which contradicts to Lemma 1. (*) is proved.

The following is another simple statement we will use:
(**) *The distance-two graph* $G(S_i \cup S_j)$ *is connected* $(i, j \in \{0, 1, 2\},\ i \neq j)$. Let us show this by induction on k. For $k = 2$ the the statement can be checked directly. Let $k > 2$. It is sufficient to show that arbitrary u and v from $S_i \cup S_j$ belong to the same connected component. If u and v coincide in some coordinate quadruple, then this fact follows from the inductive assumption (fixing this quadruple, we get a subgraph isomorphic to a graph considered in the previous inductive step). Otherwise, there exists a word w in $S_i \cup S_j$ that coincide with u in the first quadruple and with v in the second quadruple (the values in the other quadruples are chosen to make the index of w being i or j). Similarly to the considered case, u, w and v belong to the same connected component. (**) is proved.

Since P' and Q' consist of standard vectors, they are included in $S_1 \cup S_2$. Denote

$$P_1 = P' \cap S_1, \qquad P_2 = P' \cap S_2, \qquad Q_1 = Q' \cap S_1, \qquad Q_2 = Q' \cap S_2.$$

If $P_1 = Q_1 = \emptyset$, then, as follows from Lemma 1(b), $P \cup P_2$ and $Q \cup Q_2$ correspond to connected components of $G(S_0 \cup S_2)$, which contradicts to (**). Similarly, $P_2 = Q_2 = \emptyset$ is impossible.

We have:

$$\dot{\Omega}(P_1 \cup P_2) \cup \dot{\Omega}(Q_1 \cup Q_2) = \dot{\Omega}(P) \cup \dot{\Omega}(Q) = \dot{\Omega}(S_0).$$

Further, $\dot{\Omega}(S_1) = \dot{\Omega}(S_0)$; thus,

$$\dot{\Omega}(S_1 \setminus (P_1 \cup Q_1)) = \dot{\Omega}(P_2 \cup Q_2).$$

Similarly,

$$\dot{\Omega}(S_2 \setminus (P_2 \cup Q_2)) = \dot{\Omega}(P_1 \cup Q_1).$$

So, S_1 is partitioned into two nonempty sets with alternatives in S_2. It follows from Lemma 1(b) that the graph $G(S_1 \cup S_2)$ is not connected, which contradicts to (**) and proves Claim 2.

Claim 3. *The e.m.s.* S_0 *is irreducible.*

Note that in the construction (2) the sum of the last two coordinates is 0 for every word in R. Taking into account coordinate permutations and symbol inversions, we can claim that for any reducible e.m.s. there exist two coordinates whose sum is either 0 or 1 simultaneously for all words of the e.m.s. It is easy to see that S_0 does not satisfy this condition: in every two coordinates there occur all four combinations of 0 and 1. Claim 3 is proved. The theorem is proved.

6. Conclusion

We have constructed an infinite class of unsplittable irreducible m.s. Our construction generalizes the example mentioned in [7]. In conclusion, we formulate several problems, which are naturally connected with the study of mobile sets and with the problem of characterization of their variety.

For constructing m.s., one can apply the generalized concatenation principle which works for 1-perfect codes [9]. In particular, the construction from Section 5 can be treated in such terms. Unsplittable m.s. constructed in such the way will have non-full rank, i.e., for all the words of the set the coordinates will satisfy some linear equation.

Problem 1. Construct an infinite family of full-rank unsplittable m.s.

Example. Consider the four words

$$\begin{pmatrix}100\\110\\010\end{pmatrix},\quad\begin{pmatrix}011\\110\\000\end{pmatrix},\quad\begin{pmatrix}101\\001\\011\end{pmatrix},\quad\begin{pmatrix}001\\100\\111\end{pmatrix}$$

in E^9, listed, for convenience, as 3×3 arrays, and all the words obtained from them by cyclic permutations of rows and/or columns of the array. We get full-rank unsplittable m.s. of cardinality 36. An alternative can be obtained by the inversion of all the words.

Problem 2. Construct a rich class of transitive unsplittable m.s., e.m.s. A set $M \subseteq E^n$ is called transitive iff the stabilizer $\mathrm{Stab}_I(M)$ of M in the group I of isometries of the hypercube acts transitively on the elements of M; i.e., for every x, y from M there exists an isometry $\sigma \in \mathrm{Stab}_I(M)$ such that $\sigma(x) = y$. For example, it is not difficult to see that the m.s. constructed in the current paper are transitive. There are several constructions of transitive 1-perfect and extended 1-perfect codes, see [6, 3] for the last results.

Problem 3. Study the embeddability of m.s. into 1-perfect codes: the existence of nonembeddable m.s. in the code dimensions $n = 2^k - 1$; the existence of m.s. that cannot be embedded with help of the linear extension (Lemma 3) into a 1-perfect code in a larger dimension. In particular, for m.s. constructed in Section 5, the embedding questions are open provided $n \geq 11$.

Problem 4. Estimate the maximal cardinality of an unsplittable m.s.

Problem 5. Estimate the minimal cardinality of a nonlinear m.s. (the construction of Section 5 together with Lemma 3 give the upper bound $1{,}5L(n)$, where $L(n) = 2^{(n-1)/2}$ is the cardinality of the linear m.s.), of an irreducible unsplittable m.s. (the construction gives the upper bound $1{,}5^{(n-3)/4}L(n)$), unsplittable m.s. of full rank.

Problem 6. Study mobile sets in other spaces, in particular, in q-ary Hamming spaces where $q > 2$ is an arbitrary integer, not necessarily a prime power.

References

[1] D. G. Fon-Der-Flaass, Perfect 2-colorings of a hypercube, *Sib. Math. J.*, 48, 2007, 740-745. dx.doi.org/10.1007/s11202-007-0075-4, DOI:10.1007/s11202-007-0075-4 transl. from Sib. Mat. Zh. 48, 2007, 923-930.

[2] S. A. Malyugin, On enumeration of nonequivalent perfect binary codes of length 15 and rank 15, *J. Appl. Ind. Math.* 1, 2007, 77-89, dx.doi.org/10.1134/S1990478907010085, DOI: 10.1134/S1990478907010085 transl. from Diskr. Anal. Issled. Oper., Ser. 1 13, 2006, 77-98.

[3] V. N. Potapov, A lower bound for the number of transitive perfect codes. *J. Appl. Ind. Math.* 1, 2007, 373-379, dx.doi.org/10.1134/S199047890703012X, DOI: 10.1134/S199047890703012X transl. from Diskr. Anal. Issled. Oper. Ser. 1, 13, 2006, 49-59.

[4] F. I. Solov'eva, On the factorization of code-generating d.n.f. *Met. Diskr. Anal. Issled. Funkts. Sistem* 47, Inst. Math. SB AS USSR, Novosibirsk, 1988, 66-88 (in Russian).

[5] F. I. Solov'eva, Structure of i-components of perfect binary codes, *Discr. Appl. Math.* 111, 2001, 189-197. dx.doi.org/10.1016/S0166-218X(00)00352-8, DOI: 10.1016/S0166-218X(00)00352-8.

[6] F. I. Solov'eva, On the construction of transitive codes, *Probl. Inform. Transm.* 41, 2005, 204-211. dx.doi.org/10.1007/s11122-005-0025-3, DOI: 10.1007/s11122-005-0025-3, transl. from Probl. Peredachi Inf. 41, 2005, 23-31.

[7] Yu. L. Vasil'ev, F. I. Solov'eva, Codegenerating factorization on n-dimensional unite cube and perfect binary codes, *Probl. Inform. Transm.* 33, 1997, 64-74, transl. from Probl. Peredachi Inf. 33, 1997, 64-74.

[8] A. Yu. Vasil'eva, A representation of perfect binary codes, *Proc. Seventh Intern. Workshop ACCT*, Bansko, Bulgaria, 2000, 311-315.

[9] V. A. Zinoviev, A. Lobstein, On generalized concatenated constructions of perfect binary nonlinear codes, *Probl. Inform. Transm.* 36, 2000, 336-348.

[10] V. A. Zinoviev, D. V. Zinoviev, Binary extended perfect codes of length 16 and rank 14, *Probl. Inform. Transm.* 42, 2006, 123-138, dx.doi.org/10.1134/S0032946006020062, DOI:10.1134/S0032946006020062 transl. from Probl. Peredachi Inf. 42, 2006, 63-80.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 300-304*

An upper bound on the covering radius of a class of cyclic codes[1]

EVGENIYA VELIKOVA velikova@fmi.uni-sofia.bg
ASEN BOJILOV bojilov@fmi.uni-sofia.bg
Faculty of Mathematics and Informatics, Sofia University,
5 James Baucher blvd, Sofia, BULGARIA

Abstract. In this paper we consider a class of cyclic $[p^m-1, p^m-2m-1]$-codes over $\mathbb{Z}_p$, where $p \neq 2$ is a prime number, and we show that these codes have covering radius at most 3.

1 On the number of solutions of some equations

Let F be the Galois field $\mathrm{GF}(q)$ where $q = p^m$ and $p = \mathrm{char}\, F$ is prime. We assume that $p \neq 2$ and that β is a generator of the multiplicative group F^* of the field F. Let us define the following sets

$$Q = \langle \beta^2 \rangle \cup \{0\} = \{a \in F \mid \exists b \in F \colon a = b^2\}$$

of the perfect squares in F and

$$N = \beta \langle \beta^2 \rangle = F \backslash Q = \{a \in F \mid \exists b \in F \colon a = \beta b^2\}$$

of nonsquares in F.

We shall prove the next lemma following [5].

Lemma 1.1 *Let M be the set of the solutions (x, y) of the equation $Ax^2 + By^2 = C$ in the finite field F with q elements and let $D = AB \neq 0$. Then the following fact holds*

$$|M| = \begin{cases} q - \left(\dfrac{-D}{q}\right), & \text{if } C \neq 0, \\ q + \left(\dfrac{-D}{q}\right)(q-1), & \text{if } C = 0, \end{cases}$$

[1] This work was partially supported by the SF of Sofia University under Contract 171/05.2008.

Proof. Let us denote

$$M_{x_0} = \{y \in F \mid Ax_0^2 + By^2 = C\} =$$
$$\{y \in F \mid y^2 = -D\left(x_0^2 - \frac{C}{A}\right) \in Q\}.$$

Therefore,

$$|M_{x_0}| = \begin{cases} 0, \text{ if } \ -D\left(x_0^2 - \frac{C}{A}\right) \in N, \\ 1, \text{ if } \ \left(x_0^2 - \frac{C}{A}\right) = 0, \\ 2, \text{ if } \ {}_D\left(x_0^2 - \frac{C}{A}\right) \neq 0 \text{ and } \left(x_0^2 - \frac{C}{A}\right) \in Q \end{cases}$$

and

$$|M_{x_0}| = \left(\frac{-D(x_0^2 - \frac{C}{A})}{q}\right) + 1 = \left(\frac{-D}{q}\right)\left(\frac{x_0^2 - \frac{C}{A}}{q}\right) + 1,$$

where

$$\left(\frac{a}{q}\right) = \begin{cases} 0, & \text{if } \ a = 0, \\ 1, & \text{if } \ a \in Q, \, a \neq 0, \\ -1, & \text{if } \ a \in N \end{cases}$$

is the generalized symbol of Legendre in the finite field F with q elements.

Therefore,

$$|M| = \sum_{x\in F} |M_x| = \sum_{x \in F}\left(\left(\frac{-D}{q}\right)\left(\frac{x^2 - \frac{C}{A}}{q}\right) + 1\right) = q + \left(\frac{-D}{q}\right)\sum_{x\in F}\left(\frac{x^2 - \frac{C}{A}}{q}\right).$$

First, let us consider the case $A = 1$ and $B = -1$. It is clear that

$$|M| = \begin{cases} q-1, & \text{if } \ C \neq 0, \\ 2q-1, & \text{if } \ C = 0 \end{cases}.$$

In this case $D = -1$

$$\sum_{x\in F}\left(\frac{x^2 - \frac{C}{A}}{q}\right) = \begin{cases} -1, & \text{if } \ C \neq 0, \\ q-1, & \text{if } \ C = 0. \end{cases}$$

Now in the general case we have that

$$|M| = q + \left(\frac{-D}{q}\right)\sum_{x\in F}\left(\frac{x^2 - \frac{C}{A}}{q}\right) = \begin{cases} q + \left(\frac{-D}{q}\right)(-1) = q - \left(\frac{-D}{q}\right), & \text{if } \ C \neq 0, \\ q + \left(\frac{-D}{q}\right)(q-1), & \text{if } \ C = 0. \end{cases}$$

□

Lemma 1.2 *Let* $f(x) = Ax^2 + Bx + C \in F[x]$, $A \neq 0$, $B \neq 0$, *and let*

$$M = \{x^2 \mid x \in F,\ f(x^2) = f(\gamma x^2) \text{ for some } \gamma \in N\}.$$

Then $|M| = \dfrac{q+1}{2}$.

Proof. Let x be a solution of the equation $f(x^2) = f(\gamma x^2)$ for some $\gamma \in N$. Obviously $x = 0$ is a solution of that equation. For the next considerations we shall assume that $x \neq 0$. Then

$$\begin{aligned} Ax^4 + Bx^2 + C &= A\gamma^2 x^4 + B\gamma x^2 + C \\ Ax^2 + B &= A\gamma^2 x^2 + B\gamma \\ A(1-\gamma^2)x^2 &= B(\gamma - 1) \end{aligned}$$

and

$$-A(1+\gamma)x^2 = B,$$

since $\gamma \neq 1$ $(1 \in Q)$.

Note that $\gamma \in N$ iff there exists $u \in F$, $b \neq 0$ such that $\gamma = \beta u^2$.

We are looking for γ in such form and $u \neq 0$.

It is clear that $\gamma \neq -1$ $(B \neq 0)$. Then

$$x^2 = -\frac{B}{A} \cdot \frac{1}{1+\gamma}.$$

If $AB \in N$ then $1 + \gamma \in N$ and we must find $v \in F$ such that $1 + \beta u^2 = \beta v^2$. From Lemma 1.1 we know that there exist $q - 1$ pairs (u, v) which are the solutions of the last equation. Note that $u = 0$ is not a solution and therefore we have $\dfrac{q-1}{2}$ different elements γ such that $1 + \gamma \in N$ and $|M| = \dfrac{q-1}{2} + 1 = \dfrac{q+1}{2}$.

Analogously, the case $AB \in Q$ give us again that $|M| = \dfrac{q+1}{2}$. Indeed, $1 + \gamma \in Q$ and we must find $v \in F$ such that $1 + \beta u^2 = v^2$. By Lemma 1.1 it follows that there exist $q + 1$ pairs (u, v) which are the solutions of the last equation. Note that $u = 0$ is a solution and therefore we have $\dfrac{q-1}{2}$ $(\gamma \neq 0)$ different elements γ such that $1 + \gamma \in N$ and $|M| = \dfrac{q-1}{2} + 1 = \dfrac{q+1}{2}$. □

2 On covering radius of some cyclic codes

Let us denote by $f_a(x) \in \mathbb{Z}_p[x]$ the minimal polynomial of $a \in F$, $|F| = q = p^m$. Clearly, f_a is an irreducible polynomial and $\deg f_\beta = \deg f_{\beta^{-1}} = m$. We consider the cyclic code C of length $q-1$ over the field F generated by $g(x) = f_\beta(x).f_{\beta^{-1}}(x)$. Hence, C is a $[q-1, q-1-2m]$-code.

Following the techniques of [1], [3] and [4], we obtain the next theorem.

Theorem 2.1 *The $[p^m-1, p^m-1-2m]$-code C defined above has covering radius at most 3 for $p \neq 2$ and $q > 36$.*

Proof.

Let

$$H = \begin{pmatrix} 1 & \beta & \beta^2 & \dots & \beta^{q-1} \\ 1 & \gamma & \gamma^2 & \dots & \gamma^{q-1} \end{pmatrix}$$

be a parity check matrix of the code C.

Let $s = (a, b) \in F^2$, $(a, b) \neq (0, 0)$. We shall prove that there exists a vector $e \in F^{q-1}$ with syndrome s. For that purpose we must prove that the system

$$\left| \begin{array}{l} a_1x_1 + a_2x_2 + \dots + a_lx_l = a \\ a_1\dfrac{1}{x_1} + a_2\dfrac{1}{x_2} + \dots + a_l\dfrac{1}{x_l} = b \end{array} \right. \tag{1}$$

has a solution with $a_1, a_2, \dots, a_l \in \mathbb{Z}_p$ and $x_1, x_2, \dots, x_l \in F$ for some natural number $l \leq 3$.

For $l = 1$ it is clear that the system (1) has a solution iff ab is a nonzero perfect square in $\mathbb{Z}_p$.

Let us consider the following system

$$\left| \begin{array}{l} x_1 + x_2 + x_3 = a \\ \dfrac{1}{x_1} + \dfrac{1}{x_2} + \dfrac{1}{x_3} = b \end{array} \right., \tag{2}$$

where $(a, b) \neq (0, 0)$ and $ab \neq 1$.

Set $y_i = \dfrac{1}{x_i}$. Then we obtain an analogous system as (2) in which a and b are changed. Hence, we may assume that $b \neq 0$.

Let us consider the function $D_1(y) = 4by^2 + (-a^2b^2 + 6ab + 3)y + 4a$.

In the case $-a^2b^2 + 6ab + 3 \neq 0$ by Lemma1.2 it follows that there are $c \in F$ and $\gamma \in N$ such that $D_1(c^2) = D_1(\gamma c^2)$ and c^2 takes $\dfrac{q+1}{2}$ different values. We choose $y = c^2$ or $y = \gamma c^2$ in such a way that $D = -yD_1(y)$ is a perfect square.

If $q > 35$, it is clear that there exists y such that $y \neq 0$, $y \neq \frac{-1}{b}$, $y \neq -a$ and the system (2) has a solution

$$x_1 = \frac{a+y}{1+yb}, \quad x_2 = \frac{(ab-1)y+\sqrt{D}}{2b(1+yb)}, \quad x_3 = \frac{(ab-1)y-\sqrt{D}}{2b(1+yb)}.$$

In the case $-a^2b^2 + 6ab + 3 = 0$ we consider the system

$$\left| \begin{array}{l} x_1 + x_2 + x_3 = \frac{a}{2} \\ \frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} = \frac{b}{2} \end{array} \right. .$$

It is clear that $-\frac{a^2b^2}{16} + 3\frac{ab}{2} + 3 \neq 0$ and this system has a solution x_1, x_2, x_3 which is a solution of the system (1) with $a_1 = a_2 = a_3 = 2$.

Therefore, the covering radius of code C is at most 3. □

References

[1] T. Helleseth, On the covering radius of cyclic linear codes and arithmetic codes, *Discr. Appl. Math.* 11, 1985, 157-173.

[2] J. A. van der Horst, T. Berger, Complete decoding of triple-error-correcting binary BCH Codes, *IEEE Trans. Inform. Theory* 22, 1976, 138-147.

[3] D. Danev, S. Dodunekov, A family of ternary quasi-perfect BCH codes, to appear in *Des., Codes Crypt.*, 2008.

[4] O. Moreno, F. N. Castro, On the covering radius of certain cyclic codes, Springer-Verlag, Berlin-Heidelberg, 2003, 129-138.

[5] S. A. Stepanov, *Arithmetic of algebraic curves*, Moskva Nauka, 1991.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 305-311*

On the $(2, 1)$-extendability of ternary linear codes

Yuri Yoshida
Tatsuya Maruta maruta@mi.s.osakafu-u.ac.jp
Department of Mathematics and Information Sciences,
Osaka Prefecture University, Sakai, Osaka 599-8531, JAPAN

Abstract. We show that every $[n, k, d]_3$ code with diversity (Φ_0, Φ_1), $3 \le k \le 5$, $gcd(d, 3) = 1$, is $(2, 1)$-extendable except for the case $(\Phi_0, \Phi_1) = (40, 36)$ for $k = 5$, and that an $[n, 5, d]_3$ code with diversity $(40, 36)$, $gcd(d, 3) = 1$, is $(2, 1)$-extendable if $A_d \le 50$. Geometric conditions for the $(2, 1)$-extendability of not necessarily extendable $[n, k, d]_3$ codes for $k = 5, 6$ are also given.

1 Introduction

Let $\mathbb{F}_q^n$ denote the vector space of n-tuples over $\mathbb{F}_q$, the field of q elements. A linear code $\mathcal{C}$ of length n, dimension k and minimum (Hamming) distance d over $\mathbb{F}_q$ is referred to as an $[n, k, d]_q$ code. The *weight* of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$, denoted by $wt(\boldsymbol{x})$, is the number of nonzero coordinate positions in $\boldsymbol{x}$. The weight distribution of $\mathcal{C}$ is the list of numbers A_i which is the number of codewords of $\mathcal{C}$ with weight i. The weight distribution with $(A_0, A_d, ...) = (1, \alpha, ...)$ is also expressed as $0^1 d^\alpha \cdots$. We only consider *non-degenerate* codes having no coordinate which is identically zero.

For an $[n, k, d]_q$ code $\mathcal{C}$ with a generator matrix G, $\mathcal{C}$ is called (l, s)*-extendable* (to $\mathcal{C}'$) if there exist l vectors $h_1, \dots, h_l \in \mathbb{F}_q^k$ so that the extended matrix $[G, h_1^{\mathrm{T}}, \cdots, h_l^{\mathrm{T}}]$ generates an $[n + l, k, d + s]_q$ code $\mathcal{C}'$ ([1]). Then $\mathcal{C}'$ is called an (l, s)*-extension* of $\mathcal{C}$. '(1,1)-extendable' is simply called *extendable*. In this paper we are concerned with (2,1)-extendability of ternary linear codes with dimension $k \le 6$.

Let $\mathcal{C}$ be an $[n, k, d]_3$ code with $k \ge 3$, $\gcd(3, d) = 1$. The *diversity* (Φ_0, Φ_1) of $\mathcal{C}$ is given as the pair of integers:

$$\Phi_0 = \frac{1}{2} \sum_{3|i, i \ne 0} A_i, \quad \Phi_1 = \frac{1}{2} \sum_{i \not\equiv 0, d \pmod 3} A_i,$$

where the notation $x|y$ means that x is a divisor of y.

Let $\mathcal{D}_k$ be the set of all possible diversities of $\mathcal{C}$. $\mathcal{D}_k$ has been determined in [3] for $k \le 6$ and in [5] for $k \ge 7$. For $k \ge 3$, let $\mathcal{D}_k^*$ and $\mathcal{D}_k^+$ be as follows:

$$\mathcal{D}_k^* = \{(\theta_{k-2}, 0), (\theta_{k-3}, 2 \cdot 3^{k-2}), (\theta_{k-2}, 2 \cdot 3^{k-2}), (\theta_{k-2} + 3^{k-2}, 3^{k-2})\}, \;\; \mathcal{D}_k^+ = \mathcal{D}_k \setminus \mathcal{D}_k^*,$$

where $\theta_j = (3^{j+1} - 1)/2$. It is known that $\mathcal{D}_k^*$ is included in $\mathcal{D}_k$ and that $\mathcal{C}$ is extendable if $(\Phi_0, \Phi_1) \in \mathcal{D}_k^*$ ([3]). The necessary and sufficient conditions for the extendability of $\mathcal{C}$ with $(\Phi_0, \Phi_1) \in \mathcal{D}_k^+$ are given in [3-7] for $k \le 6$.

We denote by $\mathrm{PG}(r, q)$ the projective geometry of dimension r over $\mathbb{F}_q$. A *j-flat* is a projective subspace of dimension j in $\mathrm{PG}(r, q)$. 0-flats, 1-flats, 2-flats, 3-flats, $(r-2)$-flats and $(r-1)$-flats are called *points, lines, planes, solids, secundums* and *hyperplanes*, respectively.

Let $\mathcal{C}$ be an $[n, k, d]_q$ code with a generator matrix $G = [g_1, \cdots, g_k]^{\mathrm{T}}$. For $P = \mathbf{P}(p_1, \cdots, p_k) \in \Sigma$, *the weight of P with respect to $\mathcal{C}$* is defined in [4] as

$$w_{\mathcal{C}}(P) = wt(\sum_{i=1}^{k} p_i g_i).$$

¿From now on, let $\mathcal{C}$ be an $[n, k, d]_3$ code with $\gcd(d, 3) = 1$, $k \ge 3$. Let

$$\begin{aligned} F_0 &= \{P \in \Sigma \mid w_{\mathcal{C}}(P) \equiv 0 \pmod 3\}, \\ F_2 &= \{P \in \Sigma \mid w_{\mathcal{C}}(P) \equiv d \pmod 3\}, \\ F_d &= \{P \in \Sigma \mid w_{\mathcal{C}}(P) = d\}, \;\; F_e = F_2 \setminus F_d \\ F_1 &= \Sigma \setminus (F_0 \cup F_2), \;\; F = F_0 \cup F_1. \end{aligned}$$

Lemma 1.1([4]). *$\mathcal{C}$ is (2,1)-extendable iff there exist two hyperplanes H_1, H_2 of Σ such that $F_d \cap H_1 \cap H_2 = \emptyset$. Equivalently, $F \cup F_e$ contains a secundum of Σ.*

We give the necessary and sufficient conditions for the (2,1)-extendability of $[n, k, d]_3$ codes with diversity (Φ_0, Φ_1), $3 \le k \le 6$, $d \equiv 1$ or $2 \pmod 3$ from this geometrical point of view.

A t-flat Π of Σ with $|\Pi \cap F_0| = i$, $|\Pi \cap F_1| = j$ is called an *$(i, j)_t$ flat.* An $(i, j)_1$ flat is called an *(i, j)-line*. An *(i, j)-plane* and an *(i, j)-solid* are defined similarly. We denote by $\mathcal{F}_j$ the set of j-flats of Σ. Let Λ_t be the set of all possible (i, j) for which an $(i, j)_t$ flat exists in Σ. Then we have
$\Lambda_1 = \{(1,0), (0,2), (2,1), (1,3), (4,0)\}$,
$\Lambda_2 = \{(4,0), (1,6), (4,3), (4,6), (7,3), (4,9), (13,0)\}$,
$\Lambda_3 = \{(13,0), (4,18), (13,9), (10,15), (16,12), (13,18), (22,9), (13,27), (40,0)\}$,
$\Lambda_4 = \{(40,0), (13,54), (40,27), (31,45), (40,36), (40,45), (49,36), (40,54), (67,27),$

$$(40,81),(121,0)\},$$
$$\Lambda_5 = \{(121,0),(40,162),(121,81),(94,135),(121,108),(112,126),(130,117),$$
$$(121,135),(148,108),(121,162),(202,81),(121,243),(364,0)\},$$

see [3]. Let $\Pi_t \in \mathcal{F}_t$. Denote by $c_{i,j}^{(t)}$ the number of $(i,j)_{t-1}$ flats in Π_t and let ${\varphi_s}^{(t)} = |\Pi_t \cap F_s|$, $s = 0,1$. $({\varphi_0}^{(t)}, {\varphi_1}^{(t)})$ is called the *diversity of* Π_t and the list of $c_{i,j}^{(t)}$'s is called its *spectrum*. Thus Λ_t is the set of all possible diversities of Π_t.

According to the diversity of $\mathcal{C}$ we give the necessary and sufficient conditions for the (2,1)-extendability of $\mathcal{C}$ using some of the following six conditions:

For $k \geq 4$, let (C_k-0), (C_k-1) and (C_k-2) be the following conditions:
($\mathbf{C}_k$-0) there exists a $(\theta_{k-4},0)_{k-3}$ flat δ_1 in Σ satisfying $\delta_1 \setminus F_0 \subset F_e$;
($\mathbf{C}_k$-1) (C_k-0) holds and there exists a $(\theta_{k-4},3^{k-3})_{k-3}$ flat δ_2 in Σ such that $\delta_1 \cap \delta_2$ is a $(\theta_{k-4},0)_{k-4}$ flat;
($\mathbf{C}_k$-2) there exist two $(\theta_{k-4},0)_{k-3}$ flats δ_1, δ_2 in Σ such that $\delta_1 \cap \delta_2$ is a $(\theta_{k-4},0)_{k-4}$ flat with $(\delta_1 \cup \delta_2) \setminus (\delta_1 \cap \delta_2) \subset F_e$;

For $k \geq 4$, let (C_k-3) and (C_k-4) be the following conditions:
($\mathbf{C}_4$-3) there are three non-collinear points $Q_1, Q_2, Q_3 \in F_e$ such that the three lines $\langle Q_1,Q_2\rangle$, $\langle Q_2,Q_3\rangle$, $\langle Q_3,Q_1\rangle$ are $(0,2)$-lines;
($\mathbf{C}_k$-3) there exist three $(\theta_{k-5},0)_{k-4}$ flats δ_1, δ_2, δ_3 through a fixed $(\theta_{k-5},0)_{k-5}$ flat L such that $\langle\delta_1,\delta_2\rangle$, $\langle\delta_2,\delta_3\rangle$, $\langle\delta_3,\delta_1\rangle$ form distinct $(\theta_{k-5}, 2\cdot 3^{k-4})_{k-3}$ flats and that $(\delta_1 \cup \delta_2 \cup \delta_3) \setminus L \subset F_e$ holds;
($\mathbf{C}_4$-4) there are three non-collinear points $P_1, P_2, P_3 \in F_1$ such that the three lines $\langle P_1,P_2\rangle$, $\langle P_2,P_3\rangle$, $\langle P_3,P_1\rangle$ are $(0,2)$-lines each of which contains two points of F_e;
($\mathbf{C}_k$-4) there exist a $(\theta_{k-5},0)_{k-5}$ flat L, three $(\theta_{k-5},3^{k-4})_{k-4}$ flats δ_1', δ_2', δ_3' through L, and six $(\theta_{k-5},0)_{k-4}$ flats $\delta_1,\cdots,\delta_6$ through L such that $\langle\delta_i',\delta_j'\rangle$ forms a $(\theta_{k-5},2\cdot 3^{k-4})_{k-3}$ flat containing two of $\delta_1,\cdots,\delta_6$ for $1 \leq i < j \leq 3$ and that $(\cup_{i=1}^{6}\delta_i) \setminus L \subset F_e$ holds.

For $k = 5$, let (C_k-5) and (C_k-6) be the following conditions:
($\mathbf{C}_5$-5) there exist a (4,0)-line l and four skew (1,0)-lines l_1, l_2, l_3, l_4 such that each of $l_1, ..., l_4$ meets l and that $\langle l_1,l_2,l_3,l_4\rangle \in \mathcal{F}_3$ and $(\cup_{i=1}^{4} l_i) \setminus l \subset F_e$ hold;
($\mathbf{C}_5$-6) there exists a $(4,3)$-plane δ in Σ and a point $R \in F_e$ such that $l_i = \langle R,P_i\rangle$ is a $(1,0)$-line for $i = 1,2,3,4$ and $(\delta \cup l_1 \cup l_2 \cup l_3 \cup l_4) \cap F_d = \emptyset$, where $F_0 \cap \delta = \{P_1,\ldots,P_4\}$.

Theorem 1.2. *Let $\mathcal{C}$ be an $[n,k,d]_3$ code with diversity (Φ_0,Φ_1), $k = 3$ or 4, $gcd(3,d) = 1$. Then $\mathcal{C}$ is $(2,1)$-extendable.*

Theorem 1.3. *Let $\mathcal{C}$ be an $[n,5,d]_3$ code with diversity $(\Phi_0,\Phi_1) \neq (40,36)$, $gcd(3,d) = 1$. Then $\mathcal{C}$ is $(2,1)$-extendable.*

Theorem 1.4. *Let $\mathcal{C}$ be an $[n,5,d]_3$ code with diversity $(40,36)$, $gcd(3,d)=1$. Then $\mathcal{C}$ is $(2,1)$-extendable iff one of the conditions* (C$_4$-0), (C$_4$-3) *holds.*

Theorem 1.5. *Let $\mathcal{C}$ be an $[n,5,d]_3$ code with diversity $(40,36)$, $gcd(3,d)=1$. Then $\mathcal{C}$ is $(2,1)$-extendable if $A_d \le 50$.*

Theorem 1.6. *Let $\mathcal{C}$ be an $[n,6,d]_3$ code with diversity $(\Phi_0,\Phi_1) \notin \{(121,108),$ $(112,126),(130,117)\}$, $gcd(3,d)=1$. Then $\mathcal{C}$ is $(2,1)$-extendable.*

Theorem 1.7. *Let $\mathcal{C}$ be an $[n,6,d]_3$ code with diversity $(121,108)$, $gcd(3,d)=1$. Then $\mathcal{C}$ is $(2,1)$-extendable iff one of the conditions* (C$_5$-1), (C$_5$-3), (C$_5$-4), (C$_5$-5), (C$_5$-6) *holds.*

Theorem 1.8. *Let $\mathcal{C}$ be an $[n,6,d]_3$ code with diversity $(112,126)$, $gcd(3,d)=1$. Then $\mathcal{C}$ is $(2,1)$-extendable iff one of the conditions* (C$_5$-2), (C$_5$-3), (C$_5$-4), (C$_5$-5), (C$_5$-6) *holds.*

Theorem 1.9. *Let $\mathcal{C}$ be an $[n,6,d]_3$ code with diversity $(130,117)$, $gcd(3,d)=1$. Then $\mathcal{C}$ is $(2,1)$-extendable iff one of the conditions* (C$_5$-0), (C$_5$-3), (C$_5$-4), (C$_5$-5), (C$_5$-6) *holds.*

Example. Let $\mathcal{C}$ be a $[15,5,8]_3$ code with a generator matrix

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

whose weight distribution is $0^1 8^{60} 9^{40} 10^{62} 11^{20} 12^{40} 13^{10} 14^{10}$ (diversity (40,36)). Then we can take $Q_1=(0,0,1,1,1)$, $Q_2=(1,2,1,2,2)$, $Q_3=(1,1,0,0,1)$ so that the condition (C$_4$-3) of Theorem 1.4 holds. Since $V(x_0+2x_1+2x_2+x_3) \cap V(x_0+x_2+2x_4) = \langle Q_1,Q_2,Q_3 \rangle$, by adding the column $(1,2,2,1,0)^{\mathrm{T}}$ and $(1,0,1,0,2)^{\mathrm{T}}$ to G, we get a (2,1)-extension of $\mathcal{C}$ whose weight distribution is $0^1 9^{38} 10^{56} 11^{46} 12^{34} 13^{30} 14^{26} 15^8 16^4$.

2. Proof of Theorems 1.2–1.4, 1.6–1.9.

Proof of Theorem 1.2. When $k=3$, there is a point $P \notin F_d$ iff $\mathcal{C}$ is (2,1)-extendable. Obviously, any plane have such a point P in F. When $k=4$, there is an (i,j)-line l with $l \cap F_d = \emptyset$ iff $\mathcal{C}$ is (2,1)-extendable. ¿From Table 1 and Table 2 in [3], it can be checked that any solid has an (i,j)-line with $(i,j)=(1,3)$ or $(4,0)$. Hence $\mathcal{C}$ is (2,1)-extendable by Lemma 1.1. □

Proof of Theorem 1.3. There is an (i,j)-plane π satisfying $\pi \cap F_d = \emptyset$ iff $\mathcal{C}$ is (2,1)-extendable. From Table 2 and Table 3 in [3], any $(\Phi_0, \Phi_1)_4$ flat with $(\Phi_0, \Phi_1) \neq (40, 36)$ contains an (i,j)-plane with $(i,j) = (4,9)$ or $(13,0)$. Hence $\mathcal{C}$ is (2,1)-extendable by Lemma 1.1. □

Proof of Theorem 1.4. ("only if" part:) Assume that $\mathcal{C}$ is (2,1)-extendable. Then there is an (i,j)-plane π satisfying $\pi \cap F_d = \emptyset$. From Table 2 and Table 3 in [3], an (i,j)-plane in the $(40,36)_4$ flat Σ satisfies $(i,j) \in \{(4,0),(1,6),(4,3),(4,6),(7,3)\}$. The condition ($C_4$-0) holds if $(i,j) \in \{(4,0),(1,6),(4,3),(7,3)\}$ and the condition (C_4-3) holds if $(i,j) = (4,6)$.
("if" part:) Assume that the condition (C_4-0) holds. Let l be a (1,0)-line satisfying $l \setminus F \subset F_e$. Then, it can be proved that there is a (7,3)-plane through l in the $(40,36)_4$ flat Σ. Hence, $\mathcal{C}$ is (2,1)-extendable by Lemma 1.1.
Assume that The condition (C_4-3) holds. Then the plane $\delta = \langle Q_1, Q_2, Q_3 \rangle$ forms a (4,6)-plane satisfying $\delta \cap F_d = \emptyset$. □

Proof of Theorem 1.6. There is an (i,j)-solid π satisfying $\pi \cap F_d = \emptyset$ iff $\mathcal{C}$ is (2,1)-extendable. From Table 3 and Table 4 in [3], any $(\Phi_0, \Phi_1)_5$ flat with $(\Phi_0, \Phi_1) \notin \{(121,108),(112,126),(130,117)\}$ contains an (i,j)-solid with $(i,j) = (13,27)$ or (40,0). Hence $\mathcal{C}$ is (2,1)-extendable by Lemma 1.1. □

Proof of Theorem 1.7. ("only if" part:) Assume that $\mathcal{C}$ is (2,1)-extendable. Then there is an (i,j)-solid π satisfying $\pi \setminus F \subset F_e$. From Table 3 and Table 4 in [3], an (i,j)-solid in the $(121,108)_5$ flat Σ satisfies $(i,j) \in \{$(13,0), (4,18), (13,9), (10,15), (16,12), (13,18), (22,9)$\}$. The condition (C_5-1) holds if $(i,j) = (4,18)$ or (22,9). The conditions (C_5-3), (C_5-4), (C_5-5), (C_5-6) hold if $(i,j) = (13,18)$, (13,9), (16,12), (10,15), respectively.
("if" part:) Assume that the condition (C_5-1) holds. Then there exist a (4,0)-plane δ_1 and a (4,9)-plane δ_2 such that $l = \delta_1 \cap \delta_2$ is a (4,0)-line and that $\delta_1 \setminus l \subset F_e$. Since $\Delta = \langle \delta_1, \delta_2 \rangle$ is necessarily a (22,9)-solid in the $(121,108)_5$ flat Σ, we have $\Delta \cap F_d = \emptyset$. Hence $\mathcal{C}$ is (2,1)-extendable. Similarly, the conditions (C_5-3), (C_5-4), (C_5-5), (C_5-6) imply the existence of an (i,j)-solid Δ with $(i,j) = (13,18)$, (13,9), (16,12), (10,15), respectively, satisfying $\Delta \cap F_d = \emptyset$. □

Theorems 1.8 and 1.9 can be proved similarly to Theorem 1.7.

3. Proof of Theorem 1.5.

Assume that $\mathcal{C}$ is not $(2,1)$-extendable. Then no three points of F_e are collinear by Theorem 1.4. Thus, F_e forms a *cap* and hence $|F_e| \leq 20$ since the largest size of a cap in $\mathrm{PG}(4,3)$ is 20. Every 20-cap in $\mathrm{PG}(4,3)$ is either a Γ-*cap* or Δ-*cap* ([2]).

Theorem 3.1([2]). *Let E be a 10-cap in a solid H of* $\mathrm{PG}(4,3)$. *Let $P_1, \cdots, P_{10}$ be the points of E and let V be a point of* $\mathrm{PG}(4,3) \setminus H$. *Then the set consisting any two of the three points different from V on each of the 10 lines $\langle V, P_i \rangle$ forms a 20-cap. Such a cap is called a Γ-cap or a cap of type Γ.*

Theorem 3.2([2]). *Let C_1, C_2, C_3, C_4 be the points of a 4-arc in a plane π in* $\mathrm{PG}(4,3)$. *Let Z_1, Z_2 and Z_3 be the points $\langle C_1, C_2 \rangle \cap \langle C_3, C_4 \rangle, \langle C_2, C_3 \rangle \cap \langle C_4, C_1 \rangle$ and $\langle C_2, C_4 \rangle \cap \langle C_1, C_3 \rangle$ respectively. Let Z_{i1}, Z_{i2} be the points on the line $\langle Z_j, Z_k \rangle (j, k \neq i)$ other than Z_j and Z_k. Let $L = \{V_1, V_2, V_3, V_4\}$ be a line skew to π. Then the set of points C_1, C_2, C_3, C_4 together with the points on the following lines $\langle A, B \rangle$ other than A and B:*

$$\langle V_i, Z_j \rangle \ (i = 1, 2;\ j = 2, 3),\ \langle V_3, Z_{3h} \rangle \ (h = 1, 2),\ \langle V_4, Z_{2h} \rangle \ (h = 1, 2)$$

forms a 20-cap. Such a cap is called a Δ-cap or a cap of type of Δ.

For $i = 1, 2$, a point $P \in F_i$ is called a *focal point* of a hyperplane H if the following three conditions hold:

(f-1) $\langle P, Q \rangle$ is a $(0, 2)$-line for $Q \in F_i \cap H$,
(f-2) $\langle P, Q \rangle$ is a $(2, 1)$-line for $Q \in F_{3-i} \cap H$,
(f-3) $\langle P, Q \rangle$ is a $(1, 6 - 3i)$-line for $Q \in F_0 \cap H$.

Such a hyperplane H is called a *focal hyperplane* of P. We also employ the following theorem.

Theorem 3.3([8]). *In a $(40, 36)_4$-flat, every point of F_1 (resp. F_2) has the unique focal* $(10, 15)$*-solid (resp.* $(16, 12)$*-solid), and vice versa.*

We show that F_e cannot form a 20-cap. Then $|F_e| = |F_2| - |F_d| = 45 - A_d/2 < 20$, giving Theorem 1.5. First, suppose that F_e is a 20-cap of type Γ in Theorem 3.1.

If $V \in F_0$, then the line $\langle V, P_i \rangle$ is necessarily a $(1, 0)$-line for $i = 1, \cdots, 10$. This contradicts that there are exactly six $(1, 0)$-lines through a fixed point of F_0 in the $(40, 36)_4$-flat Σ.

If $V \in F_1$, then $\langle V, P_i \rangle$ is a $(0, 2)$-line for $i = 1, \cdots, 10$. Let H' be the focal solid of V. Then H' is a $(10, 15)$-solid by Theorem 3.3. Let E' be the projection of E from V onto H' if $H \neq H'$, otherwise let $E' = E$. Then $E' \subset F_1$. Since E is an elliptic quadric, so is E'. Hence, there are exactly 10 planes (resp. 30 planes) in H' meeting E' in one point (resp. four points) in H'. On the other hand, the spectrum of a $(10, 15)$-solid is $(c^{(3)}_{1,6}, c^{(3)}_{4,3}, c^{(3)}_{4,6}) =$ (10,15,15). Hence there are at most $c^{(3)}_{1,6} + c^{(3)}_{4,6} = 25$ planes in H meeting E in four points, a contradiction.

If $V \in F_2$, then $\langle V, P_i \rangle$ is a $(1, 0)$-line for $i = 1, \cdots, 10$. Let H' be a $(16, 12)$-solid which is the focal solid of V, and let E' be the projection of E from V

onto H' as the previous case. Then $E' \subset F_0$. Since $F_0 \cap H'$ is a hyperbolic quadric in H', $F_0 \cap H'$ cannot contain the 10-cap E', a contradiction. Thus, F_e cannot form a Γ-cap.

Next, suppose that F_e is a 20-cap of type Δ in Theorem 3.2. Since $C_1, \dots, C_4$ are points of F_e in the case, π is a $(4,0)$-plane or a $(1,6)$-plane or a $(4,3)$-plane.

Assume π is a $(4,0)$-plane. Then, one of the lines $\langle Z_2, Z_3\rangle$, $\langle Z_1, Z_2\rangle$, $\langle Z_1, Z_3\rangle$ must be a $(4,0)$-line. If $\langle Z_2, Z_3\rangle$ is a $(4,0)$-line, then $\langle V_1, Z_3\rangle$, $\langle V_2, Z_3\rangle$, $\langle V_3, Z_{31}\rangle$, and $\langle V_4, Z_{21}\rangle$ are $(1,0)$-lines, and L is a $(2,0)$-line, a contradiction. One can get a contradiction similarly for other cases.

Assume π is a $(1,6)$-plane. Since $C_1, \dots, C_4$ form a 4-arc contained in two (1,0)-lines of π, one of Z_1, Z_2, Z_3 must be the point $\pi \cap F_0$. Suppose $Z_3 \in F_0$. Then $\langle Z_3, C_i\rangle$ is a (1,0)-line for $i = 1, 2$ and $\langle Z_3, Z_j\rangle$ is a (1,3)-line for $j = 1, 2$. Since $\langle V_1, Z_3\rangle$ is a $(1,0)$-line, we have $V_1 \in F_2$, so $\langle V_1, Z_2\rangle$ is a $(0,1)$-line, a contradiction. One can get a contradiction similarly if $Z_1 \in F_0$ or $Z_2 \in F_0$.

Assume π is a $(4,3)$-plane. If Z_3 is a point of F_0, then $\langle Z_3, C_1\rangle$ and $\langle Z_3, C_2\rangle$ are $(1,0)$-lines, which contradicts that there is only one (1,0)-line through a fixed point of F_0 in a $(4,3)$-plane. One can get a contradiction similarly if $Z_3 \in F_1$ or $Z_3 \in F_2$. Thus, F_e cannot form a Δ-cap as well. □

References

[1] A. Kohnert, (l, s)-extension of linear codes, *Discr. Math.*, to appear.

[2] R. Hill, On Pellegrino's 20-caps in $S_{4,3}$, *Combin. '81* 18 of *Ann. Discr. Math.*, North-Holland, Amsterdam, 1983, 433-448.

[3] T. Maruta, Extendability of ternary linear codes, *Des. Codes Cryptogr.* 35, 2005, 175-190.

[4] T. Maruta, Extendability of linear codes over $\mathbb{F}_q$, preprint.

[5] T. Maruta, K. Okamoto, Some improvements on the extendability of ternary linear codes, *Finite Fields Appl.* 13, 2007, 259-280.

[6] T. Maruta, K. Okamoto, Geometric conditions for the extendability of ternary linear codes, Ø. Ytrehus (Ed.), Cod. Crypt., *Lect. Notes Comp. Sci.* 3969, Springer-Verlag, 2006, 85-99.

[7] K. Okamoto, T. Maruta, Extendability of ternary linear codes of dimension five, *Proc. Ninthth Intern. Workshop ACCT*, Kranevo, Bulgaria, 2004, 312-318.

[8] Y. Yoshida, T. Maruta, Ternary linear codes and quadrics, in preparation.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 312-317*

Partitions and constant-value codes

A. J. VAN ZANTEN A.J.vanZanten@twi.tudelft.nl
Delft University of Technology, Faculty of Information Technology and Systems
Dept. of Mathematics, P.O. Box 5031, 2600 GA Delft, THE NETHERLANDS
VESELIN VAVREK veselin@chonbuk.ac.kr
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
P.O. Box 323, 5000 V. Tarnovo, BULGARIA

Abstract. We study the relationship between partitions of some integer a in $GF(p)$ in unequal parts of size at most $(p-1)/2$, and binary vectors with so-called value a. In particular we investigate a group of transformations acting on the family $A = \{A, A, ..., A\}$, where A stands for the set of all vectors of value i.

1 Preliminaries

Let p be some odd prime. We shall study the partitions of positive integers consisting of unequal parts the size of which is at most $(p-1)/2$. It will be obvious that we can represent such partitions by binary vectors $c = (c_1, c_2, \cdots, c_{(p-1)/2})$ of length $(p-1)/2$. Here, $c_i = 1$ if and only if the partition contains a part of size i . We interpret all vectors as row vectors. The number of ones in such a vector c is called the weight of the partition and is denoted by $|c|$. It stands for the number of parts in the partition. Let c be some partition. We define

$$a = \sum_{j=1}^{(p-1)/2} jc_j \mod p \tag{1}$$

and call a the value of c or $val(c)$, with $a \in \{0, 1, \cdots, p-1\}$. For a fixed value a, we collect all vectors having this value in a set A_a consisting of $|A_a|$ binary vectors of length $(p-1)/2$. So, this set contains all "conventional" partitions of the integers $a, a+p, a_2p, \cdots$ into unequal parts. We shall call such a set a *constant-value code*. We also introduce integers n_e and n_o, being the number of vectors in A_a with an even number of ones and an odd number, respectively. (We suppress the a-dependency of these integers in our notation). The complement of a partition c is defined as the partition corresponding to the vector $c = c + 1$, where 1 is the all-one vector of length $(p-2)/2$. Since the value of 1 is equal to

$$L := (p^2 - 1)/8 \mod p \tag{2}$$

all vectors of a set A_a have a complement of the same value $L-a$. Hence, we can write $A_a^c = A_{L-a}$ and we call A_a^c the complement of A_a. We also need the "value of the first halve of 1", defined by

$$K = 1 + 2 + \cdots + [(p-1)/4] = (p^2 \mp 2p - 3)/32 \mod p, \tag{3}$$

for $p = \mp 1 \mod 4$. Consequently we have

$$L - 4K = (1 \pm p)/4 \mod p \tag{4}$$

Finally, we introduce the number $k \in GF(p)$, defined by

$$2k = L = (p^2-1)/8 \tag{5}$$

as equality in $GF(p)$. In order to deal with the sets A_a, $a \in \{0, 1, \cdots, p-1\}$, we also introduce

$$N(p) = \begin{cases} \dfrac{2^{(p-1)/2}+1}{p}, & p = \pm 3 \mod p; \\ \dfrac{2^{(p-1)/2}-1}{p}, & p = \pm 1 \mod p. \end{cases} \tag{6}$$

2 A group of transformations

Let $I = \{1, 2, \cdots, (p-1)/2\}$ and let m be some integer with $1 \leq m \leq p-1$. We introduce index sets

$$I_1 = \{i : i \in I,\ mi \mod p \in I\}, \ \ I_2 := I \backslash I_1 \tag{7}$$

and a permutation matrix P with elements

$$p_{ij} = 1, \ j = mi \mod p, \ i \in I_1, or\ j = -mi \mod p, \ i \in I_2 \tag{8}$$

while $p_{i,j} = 0$ otherwise.

Theorem 1. *Let l be the order of $m \mod p$. Then the matrix P defined by (8) represents a permutation on I consisting of $(p-1)/l$ cycles of length $l/2$, for l is even, and of $(p-1)/2l$ cycles of length l, for l is odd.*

Proof. Consider the mapping $\mathcal{P} : GF(p) \to GF(p), \mathcal{P} = ma$. This mapping gives rise to a permutation of the elements of I in the following way. First, $\mathcal{P}$ permutes the nonzero elements of $GF(p)$ according to $(p-1)/l$ cycles of length l. Next, we change all elements a in these cycles which are not in I into $a' := a - p$, and then omit the minus sign of a'. If -1 is in the same cycle as 1, which is the case for l is even, this cycle of length l is transformed into a cycle of length $l/2$ followed by the same cycle of length $l/2$. while all elements now are in I. The same holds for all other cycles. If -1 and 1 are in different cycles of length l, which is the case for l is odd, then both cycles become identical after changing the minus signs. So, when omitting repeated cycles, we end up with a permutation of the elements of I as described in the theorem. For the matrix P the same holds. More precisely, this matrix represents the mapping $\mathcal{P}^{-1}$, modified by the above procedure. □

Next, we define a translation vector $t = (t_1, t_2, \cdots, t_{(p-1)/2})$, with $t_j = 1$ for $j = mi \mod p$, for $i \in I_1$, and $t_j = 0$ otherwise. Furthermore, we consider the transformation $T_m := GF(p)^{(p-1)/2} \to GF(p)^{(p-1)/2}$ defined by

$$T_m(c) = cP + t \tag{9}$$

Theorem 2. *For each m, $1 \leq m \leq p-1$, T_m induces a permutation τ_m on the set $A = A_0, A_1, \cdots, A_{p-1}$ such that $\tau_m(A_a) = A_b$, with $b = m(S_m - a)$ and $S_m = \sum_{i\in I_1} i$.*

Proof. We shall determine the value w' of the vector $b = T_m(a)$, with $val(a) = w$. The components $i \in I$ contribute $\sum_{i\in I_1} mi(1-a_i)$ to w' and those in I_2 yield $\sum_{i\in I_2}(p - mia_i)$. Hence, both contributions together and taken mod p, give $w' = \sum_{i\in I_1} mi - \sum_{i\in I} mia_i = mS - mw$. □

Special cases

$m = 2$ $\quad I_1 = 1, 2, \cdots, [(p-1)/4]$, $I_2 = I\backslash I_1$,
$\quad t = (0, 1, 0, 1\cdots)$, $w' = 2(S_2 - w) = 2(K - w)$;

$m = (p-1)/2$ $\quad I_1 = 1, 3, 5, \ldots$, $I_2 = 2, 4, 6, \ldots$, $t = (1, 0, 1, 0\ldots)$,
$\quad w' = (p-1)/2 \cdot (S_{(p-1)/2} - w) = (p-1)/2 \cdot (L - K - w)$;

$m = p - 1$ $\quad I_1 = \emptyset$, $I_2 = I$, $t = 0, P = E$.

Let $w_{i,n}$ be the value of the set $\tau_m^n(A_i)$. The integers $w_{i,n}$ satisfy in $GF(p)$ the recurrence relation

$$w_{i,n} = m(S_m - w_{i,n-1}),\ w_{i,0} = i, \tag{10}$$

which has as solution

$$w_{i,n} = \frac{m}{m+1} S_m(1 - (-m)^n) + i(-m)^n. \tag{11}$$

The permutations $\tau_m, 1 \leq m \leq p-1$, generate a permutation group G_A on A.

Theorem 3.

(i) G_A can be generated by a permutation $\tau_{-\alpha}$, where α is a generator of $GF(p)^*$.

(ii) G_A has one orbit A_k of size 1, whereas all other A_i, $i \neq k$, are in one orbit of size $p-1$.

Proof. Since α generates the multiplicative group of $GF(p)$, we can write $m = \alpha^e$ for any $m \in 1, 2, \ldots, p-1$. The permutation τ_m generates a subgroup of G_A. Equality (10) implies that $\frac{m}{m+1} S_m$ has the same value for all m. Since $S_1 = L$, it follows that $\frac{m}{m+1} S_m = \frac{L}{2}$. Next, from (10) and (11) we have that $w_{i,n} = i$ is equivalent to

$$(L/2 - i)(1 - (-m)^n) = 0 \tag{12}$$

The only i-value which satisfies this equation is $i = L/2 = k$. So, A_k is invariant with respect to all transformations of G_A. Furthermore, it will be clear from (11), that the length of the orbit to which $A_i, i \neq k$, belongs under the action of τ_m, is equal to the order of $-m \mod p$. So, if we take $m = -\alpha$, the orbit has length $p - 1$. □

Example. For $p = 11$ we have the following data: $L = 4, k = 3, K = 3$. The family A of constant-value codes consists of the sets:

$$A_0 = (0,0,0,0,0), (0,1,0,1,1), (1,1,1,0,1) \quad A_1 = (1,0,0,0,0), (0,0,1,1,1), (1,1,0,1,1)$$
$$A_2 = (0,1,0,0,0), (1,0,1,1,1)$$
$$A_3 = (0,0,1,0,0), (1,1,0,0,0), (0,1,1,1,1) \quad A_4 = (0,0,0,1,0), (1,0,1,0,0), (1,1,1,1,1)$$
$$A_5 = (0,0,0,0,1), (1,0,0,1,0), (0,1,1,0,0) \quad A_6 = (1,0,0,0,1), (0,1,0,1,0), (1,1,1,0,0)$$
$$A_7 = (0,1,0,0,1), (0,0,1,1,0), (1,1,0,1,0) \quad A_8 = (0,0,1,0,1), (1,0,1,1,0), (1,1,0,0,1)$$
$$A_9 = (0,0,0,1,1), (0,1,1,1,0), (1,0,1,0,1) \quad A_{10} = (1,0,0,1,1), (0,1,1,0,1), (1,1,1,1,0)$$

In this case, 2 generates the multiplicative group of the relevant field, i.e. $GF(11)^*$. So, according to Theorem 3 the transformation $\tau_{-2} = \tau_9$ is a generator of G_A, and it acts transitively on the family $A_i | i \neq k$. In order to apply Theorem 2, we obtain $I_1 = 3, 4, 5$, and hence $S_9 = 3+4+5 = 1 \mod 11$. Indeed, the relations $\tau_9(A_a) = A_b$ and $b = 9(1-a)$ provide us with the transformations:

$$A_2 \rightarrow A_2, \ A_0 \rightarrow A_9 \rightarrow A_5 \rightarrow A_8 \rightarrow A_3 \rightarrow A_4 \rightarrow A_6 \rightarrow A_{10} \rightarrow A_7 \rightarrow A_1 \rightarrow A_0$$

3 Constructing A_{i+1} from A_i

Next, we shall discuss a method to transform a vector $a \in A_i$ into a vector $b \in A_{i+1}$. For the sake of convenience we assume that 2 is a generator of $GF(p)^*$. So, the matrix P in (8) corresponds to a $(p-1)/2$-cycle which we denote by

$$d := (d_1(=1), d_2, \ldots, d_{(p-1)/2}), \quad d_i \in I \tag{13}$$

Corresponding to (13) we define a binary vector p of length $(p-1)/2$, such that its i-th component is equal to the parity of the number of $d_j, j < i$, which are in I_2.

Now, let a be a binary vector representing some partition, and let $val(a) = i$. We define a translation vector t as follows. If $a_{d_j} \neq p_i$, $1 < j < k$, and $a_{d_k} = p_k$

for some k, $1 \le k \le (p-1)/2$, we put $t_{d_j} = 1$, whereas all other components are zero. Formally, we can obtain t by

$$t = (1, \dots, 1, 0, \dots, 0)Q \tag{14}$$

where the vector at the rhs contains k ones followed by $(p-1)/2 - k$ zeros, while the transformation matrix Q has elements $q_{i,j} = 1$ if $j = d_i$ and $q_{i,j} = 0$ otherwise.

Theorem 4.

(i) If $a \in A_i$, *then* $b = a + t \in A_{i+1}$, *unless* $a = a_0 := p^c Q$;

(ii) For $p = \pm 3 \mod 8$, *the translation in (i) gives one-to-one mappings* $A_i \to \cdot A_{i+1}, \forall i \in GF(p) \backslash \{k-1, k\}$, $A_{k-1} \backslash \{a_0\} \to A_k$ *and* $A_k \to A_{k+1} \backslash \{a_0^c\}$;

(iii) For $p = \pm 1 \mod 8$, *the translation in (i) gives one-to-one mappings* $A_i \to A_{i+1}, \forall i \in GF(p) \backslash \{k-1, k\}$, $A_{k-1} \to A_k \backslash \{a_0^c\}$ *and* $A_k \backslash \{a_0\} \to A_{k+1}$.

Proof. We only have to take into account the change in the contribution to $val(a)$ due to the components $a_{d_1}, \dots, a_{d_k}$. These contribute an amount of

$$\sum_{i=1}^{k} (-1)^{p_i} a_{d_i} 2^{i-1} \mod p,$$

where the signs are determined by the components of p. Because of the definition of k, we only have $(-1)^{p_i} = -1$ for those positions where $a_{d_i} = 0$, for $1 \le i < k$. But these are precisely the positions where b has ones. Hence, we find

$$val(b) - val(a) = -\sum_{i=1}^{k} 2^{i-1} + (-1)^{p_k}(b_k - a_k)2^{k-1}. \tag{15}$$

If $a_k = p_k = 1$, then $b_k = 0$, and if $a_k = p_k = 0$, then $b_k = 1$, so the second term in the rhs always equals 2^{k-1}. We conclude that $val(b) - val(a) = -(2^{k-1} - 1) + 2^{k-1} = 1$. The only exception occurs when $a_{d_j} = p_j$ for all $j,$‘ $\le j \le (p-1)/2$. In that case k is not defined. So,we proved parts (i) and (ii) under the assumption that 2 generates $GF(p)$, which is true if and only if $p = \pm 3 \mod 8$, or equivalently, when $\chi(2) = -1$. Similar results can be obtained in the case $p = \pm 1 \mod 8$. □

We may conclude from Theorems 3 and 4, applying eq. (6), that for all p the following result holds.

Corollary *For all* $i \ne k$ *one has* $|A_i| = N(p)$, *whereas* $|A_i| = N(p) + 1$ *for* $p = \pm 1 \mod 8$, *and* $|A_i| = N(p) - 1$ *for* $p = \pm 3 \mod 8$.

Example In our example $p = 11$, we now take $m = 2$. For this m-value, $I_1 = \{1, 2\}$ and $I_2 = \{3, 4, 5\}$. The 5-cycle (13) equals $d = (1\ 2\ 4\ 3\ 5)$, and hence $p = (0, 0, 0, 1, 0)$.

For $a = (1, 1, 1, 0, 1) \in A_0$, we find $k = 3$ and $t = (1, 1, 1, 0, 0)Q = (1, 1, 0, 1, 1)$. So, $b = a + t = (0, 0, 1, 1, 1)$, which indeed is a vector in A_1. If we take $a = (1, 1, 0, 1, 1) \in A_2$, then k is not defined, illustrating Theorem 4(i), since $a_p = (1, 1, 1, 0, 1)Q = (1, 1, 0, 1, 1)$. Taking for a the vectors $((0, 1, 0, 0, 0)$ and $(1, 0, 1, 1, 1)$, both from A_2, yields $(1, 1, 0, 0, 0)$ and $(0, 1, 1, 1, 1)$, respectively. The third vector $(0, 0, 1, 0, 0) \in A_3$ is the complement a_p, thus confirming Theorem 4(ii).

As an illustration of Theorem 4(iii), we consider the simple case of $p = 7$, where $k = 3$. A generator of $GF(7)^*$ is -2. The corresponding matrix P, as defined by (8), stands for the cycle $(1\ 2\ 3)$. Now, if we continue our construction with 2 (though 2 is not a generator), we have $I_1 = \{1\}$ and $I_2 = \{2, 3\}$, and therefore $p = (0, 0, 1)$. Applying this vector, yields the following translations:

$$a = (0, 1, 0) \in A_2 \rightarrow (1, 1, 0) \in A_3, \quad a = (0, 0, 1) \in A_3 \rightarrow (1, 0, 1) \in A_4$$

In both translations k is equal to 1, while k is not defined for the vector $p^c = (1, 1, 0)$.

4 Remarks

Research on this topic is still in progress. Our primary motive was to develop a new approach, i.e in the context of algebraic coding theory, to the old and famous problem of determining the sign of the Gauss sum $G(2)$ (cf. [1] for a probably exhausting list of papers on this issue). It turns out that this problem is equivalent to determining the sign of $n_e - n_0$ (see Section 1) in the codes A_i. It was this background of which forced us to require the size of the parts in a partition not to exceed $(p-1)/2$. Actually, this condition is not too restrictive, since partitions of a containing one part of size $(p-1)/2$, can be dealt with by considering the partitions of $a - (p-1)/2$ as defined in this paper. Theorems 1 and 2 have their origin in [2, Lemma 4.2.4.4].

References

[1] B. Bruce, C. Berndt, R. J. Evans, The determination of Gauss sums, *Bull. Amer. Math. Soc.* 5, 1981, 107-129.

[2] V. V. Vavrek, Linear Codes and Conference Matrices (diss.), Delft University Press, Delft, 2005.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 318-324*

On the switching construction of Steiner quadruple systems [1]

Victor Zinoviev zinov@iitp.ru
Dmitry Zinoviev
Institute for Information Transmission Problems, Russian Academy of Sciences
Bol'shoi Karetnyi, 19, GSP-4, Moscow, 101447, RUSSIA

Abstract. The structure of Steiner quadruple system $S(v, 4, 3)$ of full 2-rank $v - 1$ is considered. It is shown that there are two types (induced and singular) of such systems. It is shown that induced Steiner systems can be obtained from Steiner systems $S(v, 4, 3)$ of 2-rank $v - 2$ by switching construction which is introduced here. Moreover, all non-isomorphic induced Steiner systems $S(16, 4, 3)$ of full 2-rank 15 are enumerated. It is found that there are 305616 such non-isomorphic systems $S(v, 4, 3)$, which are obtained from all 708103 non-isomorphic such systems of rank 14 studied earlier.

1 Introduction

A Stener system $S(n, k, t)$ is a pair (J, B) where J is a v-set and B is a collection of k-subsets of J such that every t-subset of J is contained in exactly one member of B. The necessary condition for existence of an $\mathrm{SQS}(v)$ is that $v \equiv 2$ or $4 \mod 6$. Hanani [1] proved that the necessary condition for the existence of an $S(v, 4, 3)$ is also sufficient. A Steiner system $S(v, 4, 3)$ is called resolvable if it can be split into mutually non-overlapping sets so that every set is a Steiner system $S(v, 4, 1)$. More on the Steiner systems can be found in [2-4] and on $S(16, 4, 3)$ in [5-8].

In this work, we consider the structure of the Steiner systems $S(v, 4, 3)$ of full 2-rank, i.e. of rank $v - 1$ over $\mathbb{F}_2$. Any such system is one of two types, which we call *induced* and *singular*. The induced systems can be obtained by a switching operation from Steiner systems $S(v, 4, 3)$ of 2-rank $v - 2$. This operation allows to construct Steiner systems of rank $r + 1$ from systems of rank r. This operation, introduced later, is also interesting for the construction of resolvable Steiner systems. Namely, it keeps this property under certain conditions on the original resolvable systems.

The case $n = 16$ is considered in details. In particular, we found exactly 305616 non-isomorphic induced Steiner systems $S(16, 4, 3)$, which were constructed by the switching operation from all 708103 non-isomorphic systems $S(16, 4, 3)$ of rank 14. We described the structure of singular Steiner systems.

[1] The paper has been written under the partial financial support of the Russian fund for the fundamental research (the number of project 06 - 01 - 00226)

2 Preliminary results

Let $E = \{0, 1\}$. A binary code of length n is an arbitrary subset of E^n. Denote a binary code C with length n, with minimum distance d and cardinality N as a (n, d, N)-code. Denote by $\mathrm{wt}(\boldsymbol{x})$ the Hamming weight of vector $\boldsymbol{x}$ over E. For a (binary) code C denote by $\langle C \rangle$ the linear envelope of words of C over $\mathbb{F}_2$. The dimension of space $\langle C \rangle$ is called the *rank* of C over $\mathbb{F}_2$ and is denoted $\mathrm{rank}(C)$.

Denote by (n, w, d, N) a binary constant weight code C of length n, with weight of all codewords w, with minimum distance d and cardinality N. For vector $\boldsymbol{v} = (v_1, ..., v_n) \in E^n$ denote by $\mathrm{supp}(\boldsymbol{v})$ its support: $\mathrm{supp}(\boldsymbol{v}) = \{i : v_i \neq 0\}$.

The binary (n, d, N)-code A which is a linear k-dimensional space over $\mathbb{F}_2$ is denoted by $[n, k, d]$-code. For binary vector $\boldsymbol{x} = (x_1, \cdots, x_n)$ and $\boldsymbol{y} = (y_1, \cdots, y_n)$ denote by $(\boldsymbol{x} \cdot \boldsymbol{y}) = x_1 y_1 + \cdots + x_n y_n$ their inner product over $\mathbb{F}_2$. For any (n, d, N)-code (linear, nonlinear, or constant weight) denote by $C^\perp$ its dual code: $C^\perp = \{\boldsymbol{v} \in \mathbb{F}_2^n : (\boldsymbol{v} \cdot \boldsymbol{c}) = 0, \ \forall \ \boldsymbol{c} \in C\}$. Clearly $C^\perp$ is a linear $[n, n-k, d^\perp]$-code with some minimum distance $d^\perp$, where $k = \mathrm{rank}(C)$.

Denote by E_2^n the set of all binary vectors of length n of weight 2. Let $J_n = \{1, 2, \dots, n\}$ be the coordinate set of E^n and let $\mathcal{S}_n$ be the full group of permutations of n elements (thus $|\mathcal{S}_n| = n!$). A binary incidence matrix of a Steiner system $S(v, 4, 3)$ is a constant weight $(v, 4, 4, v(v-1)(v-2)/24)$-code C which is strongly optimal [8]. In our notation the connection between the system (X, B) and the code C is: $B = \{\mathrm{supp}(\boldsymbol{v}) \subset X : \boldsymbol{v} \in C\}$. In this note, the Steiner system $S(v, 4, 3)$ is identified with the constant weight $(v, 4, 4, v(v-1)(v-2)/24)$-code, which uniquely defines this system [8].

Definition 1 *Two Steiner systems (X, B) and (X', B') of order n are isomorphic, if their incidence matrices S and S' are equivalent as constant weight codes, i.e. if there exists some permutation $\tau \in \mathcal{S}_n$ such that S and $\tau S'$ coincide up to the permutation of rows.*

3 Switching constructions of SQS(v)

Let C be a Steiner system $S(v, 4, 3)$ of rank $r \leq v - 2$ over $\mathbb{F}_2$. Applying the appropriate permutation of coordinates, C can be presented in the form, when the $[v, v/2, 2]$-code $C^\perp$, orthogonal to $\langle C \rangle$, is of the following form:

$$C^\perp = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_1 + \boldsymbol{u}_2\}, \tag{1}$$

where $\boldsymbol{u}_0$ is the zero vector, $\boldsymbol{u}_1 = (11\dots1|00\dots0)$, and $\boldsymbol{u}_2 = (00\dots0|11\dots1)$. Thus we split n coordinates into two blocks of $v/2$ coordinates such that any $\boldsymbol{c} \in C$ consists of two vectors $\boldsymbol{c} = (\boldsymbol{c}_1 \mid \boldsymbol{c}_2)$ where each vector $\boldsymbol{c}_i$ satisfies to the overall parity checking: $\mathrm{wt}(\boldsymbol{c}_i) \equiv 0 \pmod 2$, $i = 1, 2$ (we call it a *parity rule*).

Definition 2 *Let C be a Steiner system $S(v,4,3)$ of rank less or equal to $v-2$ over $\mathbb{F}_2$ with orthogonal code* (1). *Define the subset $C_{(w_1|w_2)}$ of C where $w_1, w_2 \in \{0,2,4\}$ as follows:*

$$C_{(w_1|w_2)} = \{\boldsymbol{c} = (\boldsymbol{a}\,|\,\boldsymbol{b}) \in C : \ \mathrm{wt}(\boldsymbol{a}) = w_1, \ \mathrm{wt}(\boldsymbol{b}) = w_2\}.$$

Lemma 1 *Let $v \geq 16$ be an integer such that $v/2 \equiv 2$ or 4 (mod 6) and let C be a Steiner system $S(v,4,3)$ of rank less or equal to $v-2$ over $\mathbb{F}_2$ with dual code* (1). *Then C is a union of three subsets*

$$C = C_{(4|0)} \bigcup C_{(0|4)} \bigcup C_{(2|2)}$$

where $C_{(4|0)}$ (respectively $C_{(0|4)}$) is a Steiner system $S(v/2,4,3)$ and $C_{(2|2)}$ has cardinality $\binom{v/2}{2} \times (v/2-1)$.

Definition 3 *Define the following (constant weight) $(8,4,4,8)$-codes:*

$$C_P = \left\{ \begin{array}{ll} (1111|0000), & (0000|1111), \\ (1100|1100), & (0011|0011), \\ (1010|1010), & (0101|0101), \\ (1001|0110), & (0110|1001) \end{array} \right\}, \ C_N = \left\{ \begin{array}{ll} (1110|1000), & (1101|0100), \\ (1011|0010), & (0111|0001), \\ (1000|1110), & (0100|1101), \\ (0010|1011), & (0001|0111) \end{array} \right\}.$$

For a given permutation $\pi \in S_4$ denote by $C_{\pi(P)}$ (respectively, by $C_{\pi(N)}$) the code obtained from C_P (respectively, from C_N) by applying π to the last 4 columns of the code C_P (respectively C_N).

Note that the middle six columns of C_P define two Pasch configurations.

Theorem 1 (*switching construction*). *Let S be a Steiner system $S(v,4,3)$ and let C be the corresponding constant weight $(v,4,4,v(v-1)(v-2)/24)$-code with dual code* (1). *Assume that C contains as a subcode the code $C_{\pi(P)}$ for some $\pi \in \mathcal{S}$. Define the new code*

$$C^*(\pi(P)) = (C \setminus C_{\pi(P)}) \cup C_{\pi(N)}.$$

Then:
1). *The set $C^* = C^*(\pi(P))$ is a constant weight $(v,4,4,v(v-1)(v-2)/24)$-code, which defines a new Steiner system $S(v,4,3)$, denoted by $S^* = S^*(\pi(P))$.*
2). *The new system S^* is not isomorphic to the initial system S (since they have different number of Pasch configurations).*
3). *If the initial system S is resolvable and if the code $C_{\pi(P)}$ belongs to exactly four parallel classes of C, then the resulting system S^* is resolvable too.*

4 The structure of Steiner systems $S(v,4,3)$ with rank $v-1$ over $\mathbb{F}_2$

Let $S = S(v,4,3)$ be of rank $v-1$ over $\mathbb{F}_2$. Recall $J = \{1, 2, \ldots, v\}$ is the coordinate set of S. Divide J into two arbitrary equal halves: J_1 and J_2. Applying some permutation $\pi \in \mathcal{S}_n$, any vector $\boldsymbol{c} \in \pi(C)$ can be presented in the form $\boldsymbol{c} = (\boldsymbol{c}_1 \,|\, \boldsymbol{c}_2)$, where $\mathrm{supp}(\boldsymbol{c}_i) \in J_i$ for $i = 1, 2$. Hence without loss of generality assume that J_1 is the left half of J and J_2 is the right half of J_2.

Definition 4 *For any Steiner system $S(v,4,3)$ of rank $v-1$ over $\mathbb{F}_2$ define the left and right spectrum (x_i, y_i, z_i), $i = 1, 2$ as follows:*

$$\begin{aligned} x_i &= \left|\{\boldsymbol{c} = (\boldsymbol{c}_1 \,|\, \boldsymbol{c}_2) : \mathrm{wt}(\boldsymbol{c}_i) = 4\}\right|, \\ y_i &= \left|\{\boldsymbol{c} = (\boldsymbol{c}_1 \,|\, \boldsymbol{c}_2) : \mathrm{wt}(\boldsymbol{c}_i) = 3\}\right|, \\ z_i &= \left|\{\boldsymbol{c} = (\boldsymbol{c}_1 \,|\, \boldsymbol{c}_2) : \mathrm{wt}(\boldsymbol{c}_i) = 2\}\right|. \end{aligned}$$

Lemma 2 *Let C be an arbitrary Steiner system $(v,4,3)$ of rank $v-1$ over $\mathbb{F}_2$. Then $x = x_1 = x_2$, $y = y_1 = y_2$, $z = z_1 = z_2$. Furthermore*

$$y = \binom{v/2}{3} - 4x, \quad z = 6x + \binom{v/2}{2}. \tag{2}$$

Clearly for the same system the numbers x, y and z depend on the choice of subsets J_i.

Definition 5 *For a Steiner system $S = S(v,4,3)$ of rank $v-1$ over $\mathbb{F}_2$ define the spectrum (x, y, z), where x takes the maximal value for given S and y and z satisfies* (2).

Lemma 3 *For a Steiner system $S(v,4,3)$ of rank $v-1$ over $\mathbb{F}_2$ with spectrum (x, y, z), we have*

$$x \geq \left\lceil \frac{v(v-1)(v-2)}{24} \cdot \frac{\binom{v-4}{v/2}}{\binom{v}{v/2}} \right\rceil.$$

In particularly, $x \geq 6$ when $v = 16$.

Definition 6 *We say that* 4 *different binary vectors of length v and weight* 3 *form a 4-clique, if*

$$\left| \cup_{i=1}^{4} \mathrm{supp}(\boldsymbol{y}_i) \right| = 4.$$

Lemma 4 *Let X be a constant weight $(v,4,4,x)$ code with cardinality $x \leq v(v-1)(v-2)/24 - 2$. Denote by Y the constant weight $(v,3,2,y)$ code, formed by all vectors of weight* 3*, which are not covered by codewords of X, i.e. $y = \binom{v}{3} - 4x$. Then X can be imbedded into a Steiner system $S(v,4,3)$, if and only if all the codewords of Y can be partitioned into disjoint 4-cliques $C_1, \ldots, C_k$, $k = y/4$, such that $|\mathrm{supp}(C_i) \cap \mathrm{supp}(C_j)| \leq 2$ for any $i \neq j$.*

5 Induced Steiner systems $S(v, 4, 3)$

We say that a Steiner system $S = S(v, 4, 3)$ of full rank $r = v - 1$ is *induced*, if it is obtained by the switching construction from some Steiner system $S' = S(v, 4, 3)$ of rank $\leq v - 2$. In the contrary case, we call this system *singular.*

Theorem 2 *Let $S = S(v, 4, 3)$ be a Steiner system of rank $r = v - 1$ over $\mathbb{F}_2$ with spectrum (x, y, z) and let v is a multiple of 4. Let X_i and Y_j be the corresponding $(v/2, 4, 4, x)$- and $(v/2, 3, 2, y)$-codes, where y satisfies (2) and $i, j \in \{1, 2\}$. If X_1 and X_2 are any subcodes of a Steiner system $S' = S(v/2, 4, 3)$, then S is an induced system.*

It is known from [6,7] there are exactly 708103 non-isomorphic Steiner systems SQS(16) of rank 14 over $\mathbb{F}_2$. By computations it was found that all these 708103 systems give 295488 different Pasch configurations. For each system SQS(16) of rank 14, containing some Pasch configurations we have applied all possible switchings.

Theorem 3 (*Computational results*). *There are* 305616 *non-isomorphic induced Steiner systems* $S(16, 4, 3)$ *of rank* 15 *over* $\mathbb{F}_2$. *They are obtained from* 708103 *non-isomorphic Steiner systems SQS*(16) *of rank* 14 *over* $\mathbb{F}_2$ *by applying all possible switchings.*

Remark 1 *Taking into account the result of* [7] *we conclude that there are exactly* 27715 *non-isomorphic singular Steiner systems* $S(16, 4, 3)$ *of rank* 15.

6 Derived triple systems

For a system $S(v, 4, 3)$, given by the pair of sets (J, B), a derived triple system of (J, B) is a pair (J_a, B_a), where $J_a = J \setminus \{a\}$ and $B_a = \{b \setminus \{a\} : a \in b \in B\}$. It is obvious, that every derived triple system is a Steiner triple system $S(v-1, 3, 2)$. For $v = 16$ we obtain a system $S(15, 3, 2)$. It is known [9] that there are exactly 80 non-isomorphic systems $S(15, 3, 2)$. There is a standard numbering of these systems by the indices from 1 to 80, related to the number of Pasch configurations (see [4]).

Given a system $S = S(v, 4, 3)$, let $\beta = \beta(S)$ denote the number of its pairwise non-isomorphic $S(v - 1, 3, 2)$. Clearly $1 \leq \beta \leq v$ for any S. A system S is said to be *homogeneous* (respectively, *heterogeneous*), if $\beta = 1$ (respectively, $\beta = v$). Among all induced Steiner systems $S(16, 4, 3)$, the derived systems $S(15, 3, 2)$ that we found are those with indices $1, 2, \ldots, 77$ missing $35, 38, 43, 68, 69, 70, 73, 74$, i.e. all together 69 non-isomorphic $S(15, 3, 2)$ out of total 80 such systems. All Steiner triple systems with these numbers occur as derived in the homogeneous $S(16, 4, 3)$.

Denote by $N_{hom}(i)$ the number of non-isomorphic homogeneous systems $S(16, 4, 3)$ with rank 15, whose derived systems are $S(15, 3, 2)$ with number i, where $i \in \{1, 2, \ldots, 7\}$. Denote by $N(\beta)$ the number of such non-isomorphic systems $S(16, 4, 3)$ with rank 15 with given β. Denote by $N(\mu(i_1), \mu(i_2), \ldots, \mu(i_\beta))$ the number of non-isomorphic systems $S(16, 4, 3)$ with rank 15 which have $\mu(i_s) > 0$ derived systems with index i_s, where $i_s \in \{1, 2, \ldots, 7\}$ for $s = 1, \ldots, \beta$, i.e. in our notation $N_{hom}(i) = N(\mu(i) = 16)$.

Proposition 1 (*Computational results*). *Among the non-isomorphic induced* $S(16, 4, 3)$ *of rank* 15 *over* $\mathbb{F}_2$, *there are* 245 *homogeneous systems. Among these systems there are:*

$$\begin{array}{rcl@{\qquad}rcl}
N_{hom}(4) & = & 1, & N_{hom}(8) & = & 12,\\
N_{hom}(9) & = & 3, & N_{hom}(10) & = & 6,\\
N_{hom}(11) & = & 1, & N_{hom}(12) & = & 15,\\
N_{hom}(13) & = & 6, & N_{hom}(14) & = & 2,\\
N_{hom}(16) & = & 1, & N_{hom}(17) & = & 1,\\
N_{hom}(20) & = & 2, & N_{hom}(23) & = & 6,\\
N_{hom}(24) & = & 5, & N_{hom}(25) & = & 52,\\
N_{hom}(26) & = & 64, & N_{hom}(27) & = & 5,\\
N_{hom}(28) & = & 5, & N_{hom}(29) & = & 14,\\
N_{hom}(30) & = & 1, & N_{hom}(32) & = & 10,\\
N_{hom}(33) & = & 5, & N_{hom}(34) & = & 5,\\
N_{hom}(36) & = & 2, & N_{hom}(52) & = & 1,\\
N_{hom}(53) & = & 3, & N_{hom}(54) & = & 5,\\
N_{hom}(59) & = & 2, & N_{hom}(60) & = & 3,\\
N_{hom}(63) & = & 2, & N_{hom}(64) & = & 2,\\
N_{hom}(65) & = & 2, & N_{hom}(71) & = & 1,
\end{array}$$

Proposition 2 (*Computational results*). *For induced Steiner systems* $S(16, 4, 3)$ *of rank* 15 *over* $\mathbb{F}_2$, *the distribution of the value* $N(\beta)$ *is the following:*

$$\begin{array}{rcl@{\qquad}rcl}
N(1) & = & 245, & N(2) & = & 1412,\\
N(3) & = & 2732, & N(4) & = & 7553,\\
N(5) & = & 9674, & N(6) & = & 19187,\\
N(7) & = & 19187, & N(8) & = & 33896,\\
N(9) & = & 47645, & N(10) & = & 57794,\\
N(11) & = & 57794, & N(12) & = & 34250,\\
N(13) & = & 15607, & N(14) & = & 4758,\\
N(15) & = & 884, & N(16) & = & 77.
\end{array}$$

Taking into account the results of [7], among all Steiner systems $S(16, 4, 3)$ there are 77 heterogeneous induced systems of rank 15.

References

[1] H. Hanani, On quadruple systems, *Canad. J. Math.* 12, 1960, 145-157.

[2] C. C. Lindner, A. Rosa, Steiner quadruple systems - A survey, *Discr. Math.* 21, 1978, 147-181.

[3] A. Hartman, K. T. Phelps, Steiner Quadruple Systems, in Contemporary Design Theory: A Collection of Surveys. Dinitz J.H., Stinson D.R., Eds. John Wiley & Sons. 1992, Ch. 6, 205-240.

[4] C. J. Colbourn, J. H. Dinitz, *The CRC Handbook of Combinatorial Designs*, Boca Raton, FL: CRC Press, 1996.

[5] V. A. Zinoviev, D. V. Zinoviev, Classification of Steiner quadruple systems of order 16 and rank at most thirteen, *Probl. Inform. Transm.* 40, 2004, 48-67.

[6] V. A. Zinoviev, D. V. Zinoviev, Classification of Steiner quadruple systems of order 16 and rank 14, *Probl. Inform. Transm.* 42, 2006, 59-72.

[7] P. Kaski, P. R. J. Östergård, O. Pottonen, The Steiner quadruple systems of order 16, *J. Comb. Theory*, Ser. A 113, 2006, 1764-1770.

[8] N. V. Semakov, V. A. Zinoviev, Constant weight codes and tactical configurations, *Probl. Inform. Transm.* 5, 1969, 29-38.

[9] F. N. Cole, L. D. Cummings, H. S. White, The complete enumeration of trial systems in 15 elements, *Proc. Nat. Acad. Sci. USA* 3, 1917, 197-199.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 325-337*

On the error-correcting capabilities of low-complexity decoded LDPC codes with constituent Hamming codes

VICTOR ZYABLOV* zyablov@iitp.ru
MAJA LONČAR** maja@eit.lth.se
ROLF JOHANNESSON** rolf@eit.lth.se
PAVEL RYBIN* prybin@iitp.ru
* Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow 101447, RUSSIA
** Dept. of Electrical and Information Technology, Lund University, P. O. Box 118, SE-22100 Lund, SWEDEN

Abstract. Hamming code-based LDPC (H-LDPC) block codes are obtained by replacing the single parity-check constituent codes in Gallager's LDPC codes with Hamming codes. This paper investigates the asymptotic performance of ensembles of random H-LDPC codes, used over the binary symmetric channel and decoded with a low-complexity hard-decision iterative decoding algorithm. It is shown that there exist H-LDPC codes for which such iterative decoding corrects any error pattern with a number of errors that grows linearly with the code length. The number of required decoding iterations is a logarithmic function of the code length. The fraction of correctable errors is computed numerically for different code parameters.

1 Introduction

Concatenated code structures can yield powerful codes, which achieve good performance with low-complexity decoding, based on using simple constituent decoders as separate modules. A method for constructing long codes from short constituent codes, based on bipartite graphs, was introduced by Tanner in [1]. In his method, one of the two sets of nodes in a bipartite graph is associated with code symbols, while the other set is associated with constituent block codes whose length is equal to the node degree. These two sets of nodes are hereinafter referred to as variable nodes and constraint nodes, respectively. Tanner's general code construction unifies many known code families that can be obtained by choosing different underlying bipartite graphs and associating different constituent codes with their constraint nodes. For example, product codes [2], Gallager's Low-Density Parity-Check (LDPC) codes [3], expander codes [4], [5], and woven graph codes [6], [7] can all be described using a bipartite graph-based approach.

This work was supported in part by the Royal Swedish Academy of Sciences in cooperation with the Russian Academy of Sciences and in part by the Swedish Research Council under Grant 621-2007-6281.

For Gallager's LDPC codes [3], each constraint node in the corresponding bipartite graph represents a single parity-check (SPC) code over the variable nodes connected to it. In this case, the parity-check matrix of the code coincides with the adjacency matrix[1] of the corresponding bipartite graph. If the degree of each node is very small compared to the number of variable nodes (code length) the parity-check matrix is sparse. When the bipartite graph is regular, all variable nodes have degree j and all constraint nodes have degree k. Then the parity-check matrix contains j ones in each column and k ones in each row, and it specifies a (j,k)-regular LDPC code.

The error-correcting capabilities of LDPC codes for the binary symmetric channel (BSC) were studied in [8], where it was shown that there exist LDPC codes capable of correcting a portion of errors that grows linearly with the code length n, with decoding complexity $\mathcal{O}(n \log n)$. A similar result for expander codes was proven in [4], [5], [9].

The SPC codes associated with the constraint nodes in the Tanner graph of an LDPC code can be replaced with other constituent block codes (*e.g.*, Hamming codes [10], BCH codes [11], or Reed-Solomon codes [12]), which yields alternative constructions of LDPC codes, often referred to as generalized LDPC codes. The parity-check matrix of such an LDPC code is obtained by replacing every 1 in the graph's adjacency matrix with a column of the constituent code's parity-check matrix, and every 0 with the all-zero column.

Hamming code-based LDPC (H-LDPC) codes were first studied in [13]. Distance properties and iterative soft-decision decoding of the H-LDPC codes were further investigated in [10] and [11]. In [14], it was shown that an ensemble of H-LDPC codes contains codes with a minimum distance that asymptotically almost meets the Varshamov-Gilbert (VG) bound.

In this paper, we consider the asymptotic performance of random H-LDPC codes, when the code length n grows to infinity. We will prove that there exist H-LDPC codes which, when decoded with a simple iterative decoder of complexity $\mathcal{O}(n \log n)$, can correct any error pattern with a number of errors growing linearly with the code length. Our approach builds upon the work of [8] where such a result was proved for LDPC codes with constituent SPC codes which have minimum distance $d_0 = 2$. A similar result holds for the expander codes if the constituent codes have large enough minimum distance, cf. [4], [5], [9]. The work presented here, with constituent Hamming codes of minimum distance $d_0 = 3$, is a step towards 'closing the gap' between these two results.

2 Construction and properties of H-LDPC codes

An (n_0, k_0, d_0) Hamming code has length $n_0 = 2^m - 1$, dimension $k_0 = n_0 - m$, code rate $R_0 = 1 - m/n_0$, and minimum distance $d_0 = 3$, where $m \geq 2$ (for $m =$

[1] We define the adjacency matrix $\boldsymbol{A}$ of a bipartite graph with two vertex sets, $\mathcal{V}_1$ and $\mathcal{V}_2$, as a $|\mathcal{V}_1| \times |\mathcal{V}_2|$ binary matrix specifying connections among vertices, that is, $(\boldsymbol{A})_{ij} = 1$ iff nodes $v_i \in \mathcal{V}_1$ and $v_j \in \mathcal{V}_2$ are connected with a branch.

2 the code reduces to the length-3 repetition code). Hamming codes are perfect single-error correcting codes, that is, they correct all error patterns with one error, and no others, and their covering radius is equal to $\rho = \lfloor (d_{\min}-1)/2 \rfloor = 1$.

A parity-check matrix $\boldsymbol{H}_0$ of a Hamming code is an $m \times n_0$ matrix whose columns are all the nonzero binary m-tuples. We will consider H-LDPC codes with identical constituent Hamming codes. Let $\boldsymbol{H}_{\mathrm{b}}$ denote a block-diagonal matrix with the b constituent parity-check matrices $\boldsymbol{H}_0$ on the main diagonal, that is,

$$\boldsymbol{H}_{\mathrm{b}} = \begin{pmatrix} \boldsymbol{H}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{H}_0 \end{pmatrix} \tag{1}$$

where b is very large. The matrix $\boldsymbol{H}_{\mathrm{b}}$ is of size $bm \times bn_0$. Let $\pi(\boldsymbol{H}_{\mathrm{b}})$ denote a random column permutation of $\boldsymbol{H}_{\mathrm{b}}$. Then the matrix constructed using $\ell \geq 2$ such permutations as *layers*,

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_\ell \end{pmatrix} = \begin{pmatrix} \pi_1(\boldsymbol{H}_{\mathrm{b}}) \\ \pi_2(\boldsymbol{H}_{\mathrm{b}}) \\ \vdots \\ \pi_\ell(\boldsymbol{H}_{\mathrm{b}}) \end{pmatrix} \tag{2}$$

is a sparse $\ell bm \times bn_0$ parity-check matrix which characterizes the ensemble of Hamming code-based LDPC codes of length $n = bn_0$, where $n \gg n_0$. Let $\mathscr{C}(n_0, \ell, b)$ denote this ensemble. For a given constituent Hamming code with parity-check matrix $\boldsymbol{H}_0$, the elements of the ensemble $\mathscr{C}(n_0, \ell, b)$ are obtained by sampling independently the permutations π_l, $l = 1, 2, ..., \ell$, which are all equiprobable. The rate of a code $\mathcal{C} \in \mathscr{C}(n_0, \ell, b)$ is lower-bounded by [1]

$$R \geq 1 - \frac{\ell b(n_0 - k_0)}{n} = 1 - \ell(1 - R_0) \tag{3}$$

with equality iff the matrix $\boldsymbol{H}$ has full rank. This imposes a restriction on the rate of the constituent codes, namely,

$$R_0 > 1 - \frac{1}{\ell}$$

that is, the more layers there are, the higher the rate of the constituent codes must be.

The construction defined by (2) is a generalization of Gallager's construction [3] of the LDPC matrices, where the constituent codes are $(n_0, n_0 - 1)$ single parity-check (SPC) codes, for which $\boldsymbol{H}_0 = (1\,1\,...\,1)$. In that case, the matrix $\boldsymbol{H}$ has n_0 ones in each row, and ℓ ones in each column. Such a matrix has density $1/b$, and it specifies an (ℓ, n_0)-regular LDPC code. The parity-check matrix of an H-LDPC code, given by (2), is, in general, irregular.

The H-LDPC codes from the ensemble $\mathscr{C}(n_0, \ell, b)$ contain ℓb constituent Hamming codes; b in each layer. Such H-LDPC codes can be represented by a Tanner graph [1] with $n = bn_0$ variable nodes, and ℓb constraint nodes, as illustrated in Figure 1. Each constraint node comprises $n_0 - k_0$ parity-check constraints specified by the rows of the corresponding constituent parity-check matrix. If a codesymbol is checked by a constituent code (that is, by at least one row of its parity-check matrix), there is a branch connecting the corresponding variable node and the constraint node. Each codesymbol is checked by exactly one Hamming code in each layer. The graph is regular, with the variable-node degree equal to ℓ, and the constraint-node degree equal to n_0. Moreover, it is required that the ℓ constraint nodes adjacent to each variable node all belong

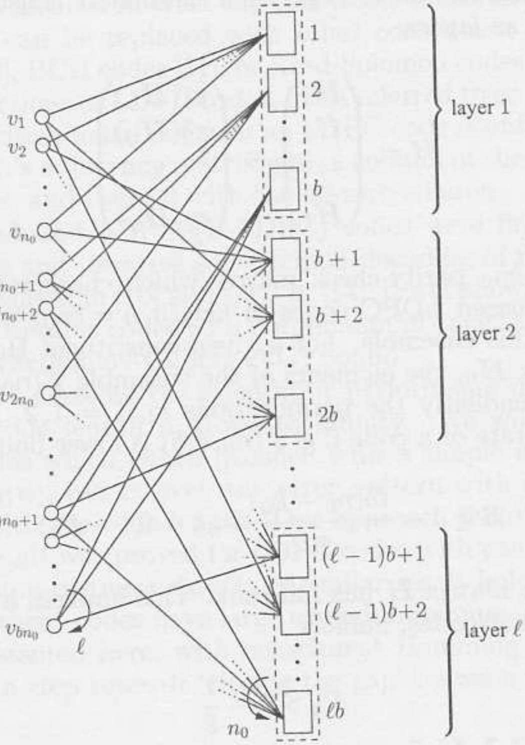

Figure 1: A Tanner graph of an H-LDPC code defined by the parity-check matrix $\boldsymbol{H}$ given in (2). The graph illustrates the case when the first layer of $\boldsymbol{H}$ is the matrix $\boldsymbol{H}_{\mathrm{b}}$ itself, *i.e.*, $\pi_1(\boldsymbol{H}_{\mathrm{b}}) = \boldsymbol{H}_{\mathrm{b}}$ (then the b constraint nodes in layer 1 are connected to the consecutive blocks of n_0 variable nodes). Other layers are obtained with arbitrary permutations.

to different layers.

Consider communication over a binary symmetric channel (BSC) using H-LDPC codes with hard-decision decoding. Let $\boldsymbol{v}$ be the transmitted codeword and $\boldsymbol{e}$ be the error pattern. Then the received sequence is given by $\boldsymbol{r} = \boldsymbol{v} + \boldsymbol{e}$. The weight of the error sequence is $W = |\boldsymbol{e}|$ and the fraction of erroneous symbols is $\omega = W/n$. For code length $n \to \infty$, the fraction of erroneous symbols ω converges in probability to the crossover probability of the BSC.

The syndrome vector computed at the receiver is given by

$$\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^{\mathrm{T}} = (\boldsymbol{e}\boldsymbol{H}_1^{\mathrm{T}}\ \boldsymbol{e}\boldsymbol{H}_2^{\mathrm{T}}\ ...\ \boldsymbol{e}\boldsymbol{H}_\ell^{\mathrm{T}}) = (\boldsymbol{s}_1\ \boldsymbol{s}_2\ ...\ \boldsymbol{s}_\ell) \tag{4}$$

where $\boldsymbol{s}_l = \boldsymbol{e}\boldsymbol{H}_l^{\mathrm{T}}$ is the syndrome of length $b(n_0 - k_0)$, corresponding to the lth layer of $\boldsymbol{H}$, which can be written as

$$\boldsymbol{s}_l = (\boldsymbol{s}_{1,l}\ \boldsymbol{s}_{2,l}\ ...\ \boldsymbol{s}_{b,l})$$

where $\boldsymbol{s}_{j,l}$ is the syndrome of the jth constituent Hamming code in layer l, $j = 1, 2, ..., b$, $l = 1, 2, ..., \ell$. If at least one of the $n_0 - k_0$ parity-checks of that constituent code is not satisfied, then $\boldsymbol{s}_{j,l} \neq \boldsymbol{0}$, indicating that the constituent code is affected by one or more errors. When the syndrome $\boldsymbol{s}_{j,l}$ is nonzero, a maximum-likelihood decoder of that Hamming code assumes that a single error occurred at the position pointed at by the syndrome value.

We further define a *generalized syndrome*

$$\boldsymbol{S} = (\boldsymbol{S}_1\, \boldsymbol{S}_2 \ldots \boldsymbol{S}_\ell) \tag{5}$$

where $\boldsymbol{S}_l = (S_{1,l}\ S_{2,l}\ ...\ S_{b,l})$, whose elements are indicators whether the constituent codes have detected an error or not, that is,

$$S_{j,l} = \begin{cases} 0, & \boldsymbol{s}_{j,l} = \boldsymbol{0} \\ 1, & \boldsymbol{s}_{j,l} \neq \boldsymbol{0} \end{cases} \qquad l = 1, 2, ..., \ell;\ j = 1, 2, ..., b.$$

The generalized syndrome is illustrated in Figure 2 for an H-LDPC code with constituent $(7, 4, 3)$ Hamming codes and $\ell = 3$ layers. Suppose that $W = 3$ codesymbols are received in error; the corresponding variable nodes are marked with black circles in Figure 2. They are connected to 5 constraint nodes, marked with black squares. The first four of these nodes are connected to less than $d_0 = 3$ variable nodes with erroneously received values and thus, these errors can be detected or corrected. Hence, the generalized syndrome for layer 1 is $\boldsymbol{S}_1 = (0\ 1\ 0\ ...\ 0\ 1)$, with weight $|\boldsymbol{S}_1| = 2$, and for layer 2, $\boldsymbol{S}_2 = (1\ 0\ 0\ ...\ 0\ 1)$, with $|\boldsymbol{S}_2| = 2$. In layer 3, the constituent code marked with a black square is affected by 3 errors. If this error pattern is a codeword of the Hamming code, then $\boldsymbol{S}_3 = \boldsymbol{0}$, otherwise, the pattern is correctable and $|\boldsymbol{S}_3| = 1$. Thus, in total, we either have $\boldsymbol{S} = (2\ 2\ 1)$, $|\boldsymbol{S}| = 5$, or $\boldsymbol{S} = (2\ 2\ 0)$, $|\boldsymbol{S}| = 4$.

Let a denote the weight of the generalized syndrome, $|\boldsymbol{S}| = a$. For an error pattern with W errors, it holds that $a \leq \ell W$. Furthermore, let a_1 denote the number of codes affected by exactly one error (that is, by a correctable error pattern). Clearly, $a_1 \leq a$. If the W errors all affect different constituent codes, then $a = a_1 = \ell W$. We can state the following lemma:

Lemma 1 *For an arbitrary error pattern with W errors, let the number of*

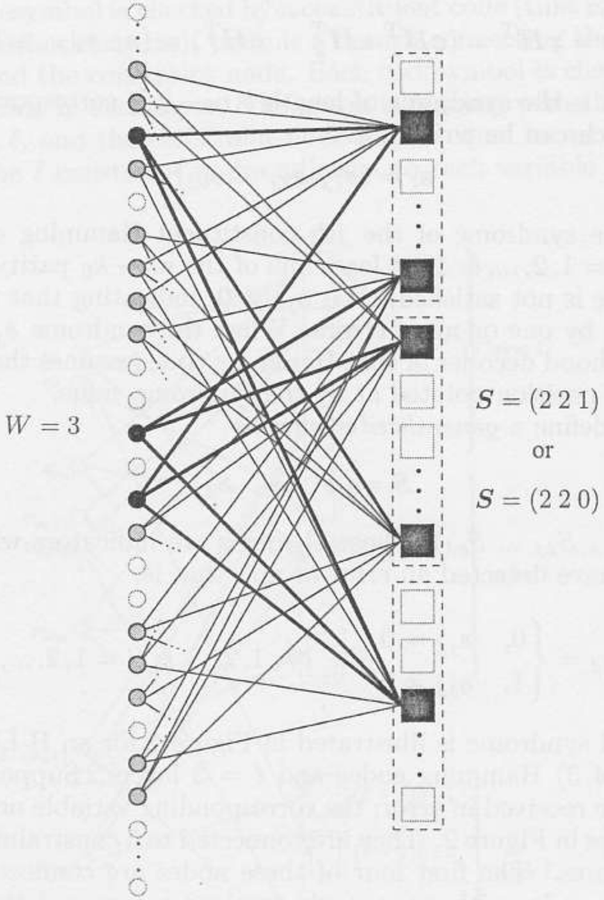

Figure 2: An illustration of the generalized syndrome value using the Tanner graph of an H-LDPC code with $\ell = 3$ layers and constituent code length $n_0 = 7$. $W = 3$ variable nodes with erroneously received values (black circles) are connected to 5 constraint nodes (black squares). The generalized syndrome vector is $\boldsymbol{S} = (2\ 2\ 1)$, or $\boldsymbol{S} = (2\ 2\ 0)$, depending on whether the three errors that affect the constituent code in the third layer form a codeword of a Hamming code or not.

constituent Hamming codes affected by exactly one error satisfy the condition

$$a_1 \geq \frac{\ell W}{2}. \tag{6}$$

Then, if the number of constituent codes with one error is a_1, the number of errors is bounded by the inequalities

$$a_1 \leq W \leq \frac{2a_1}{\ell}. \tag{7}$$

Proof. The lower bound in (7) follows from the fact that if there are W errors, the weight of the generalized syndrome a cannot be larger than ℓW, and it equals ℓW only if all the constituent codes whose generalized syndrome component is nonzero are affected by exactly one error. Then $a_1 = a = \ell W$. Consequently, the number of errors cannot be smaller than a_1/ℓ. The upper bound in (7) follows directly from condition (6). □

3 Decoding algorithm

Consider an iterative hard-decision decoding algorithm $\mathscr{A}$, whose decoding iteration i, $i = 1, 2, ..., i_{\max}$, consists of the following two steps:

(1) For the tentative sequence $\boldsymbol{r}^{(i)}$, where $\boldsymbol{r}^{(1)}$ is the received sequence $\boldsymbol{r}$, decode independently the ℓb constituent Hamming codes (that is, compute their syndromes $\boldsymbol{s}_{j,l}$, $j = 1, 2, ..., b$, $l = 1, 2, ..., \ell$, and if the value is nonzero, output the n_0-tuple where the position indicated by the syndrome is flipped). This yields ℓ independent decisions for each of the n symbols. If all the syndromes $\boldsymbol{s}_{j,l}$ are zero, output $\boldsymbol{r}^{(i)}$ as the decoding decision and stop. Otherwise, proceed with the next step.

(2) For every symbol $r_k^{(i)}$, $k = 1, 2, ..., n$, in the sequence $\boldsymbol{r}^{(i)}$, for which at least one of the ℓ decisions requires that the symbol is flipped, check if flipping the symbol reduces the weight of the generalized syndrome. If so, flip the symbol, otherwise, leave it unchanged. This yields the updated sequence $\boldsymbol{r}^{(i+1)}$. If $\boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)}$, declare the decoding failure and stop. Otherwise, return to step (1).

The following lemma provides a sufficient condition for reducing the weight of the generalized syndrome in one iteration of the algorithm $\mathscr{A}$.

Lemma 2 *For an arbitrary error pattern with W errors, if the number of constituent Hamming codes affected by a single error satisfies the condition*

$$a_1 > \frac{\ell W}{2}$$

then when decoding the constituent codes there exists a symbol such that flipping its value results in a reduction of the generalized syndrome weight.

Proof. Each received symbol is connected to exactly ℓ constituent codes. Some of these codes are affected by exactly one error, that is, they contribute to a_1. If there are more than $\ell/2$ such constituent codes, then flipping this symbol results in a reduction of the generalized syndrome weight. Assume first that there is no such symbol. Then each symbol is connected to at most $\mu\ell$ constituent codes which are affected by one error, where $\mu \leq 0.5$. The total number of codes affected by one error would then be at most

$$\mu\ell W \leq \frac{\ell W}{2}.$$

However, according to the condition of the lemma, the number of constituent codes affected by one error must satisfy

$$a_1 > \frac{\ell W}{2}.$$

Consequently, there must exist at least one symbol which is connected to more than $\ell/2$ constituent codes affected by one error. □

The number of errors that can be corrected by the algorithm $\mathscr{A}$ is stated by the following lemma.

Lemma 3 *Let W_{α_1} be the largest weight of the error pattern such that, for any $W \leq W_{\alpha_1}$, the number of constituent codes affected by a single error satisfies the condition*

$$a_1 > \frac{\ell W}{2}. \tag{8}$$

Then, if the number of errors is such that

$$W < \frac{W_{\alpha_1}}{2} \tag{9}$$

these errors will be corrected by algorithm $\mathscr{A}$. Furthermore, the maximum number of errors that may be introduced during the decoding process (until reaching the correct decision) is smaller than the initial number of errors.

Proof. In each decoding iteration, the weight of the generalized syndrome is reduced. Thus, if the algorithm does not declare a failure, it yields the all-zero syndrome, which corresponds a codeword. If $W \leq W_{\alpha_1}$, condition (8) holds, and, according to Lemma 2, there exists a symbol whose flipping reduces the generalized syndrome weight. Thus, the algorithm will yield the all-zero syndrome if the number of errors in each decoding iteration is not larger than W_{α_1}.

Let W_0 be the initial number of errors, and let W_+ be the number of errors that are added during decoding. According to condition (9), the initial weight of the generalized syndrome is not greater than

$$\ell W_0 < \frac{\ell W_{\alpha_1}}{2}.$$

Since in each decoding iteration the weight of the generalized syndrome is only decreasing, then due to Lemmas 1 and 2, the total number of errors during decoding cannot exceed W_{α_1}, and the decoder does not fail but corrects all the errors.

Let us now estimate the maximum number W_+ of the added errors. Assume that in each iteration in the beginning of the decoding we only add errors[2]. Then, W_+ errors are added during at most W_+ iterations. Thus, the weight of the generalized syndrome a is bounded from above by $\ell W_0 - W_+$, since in each iteration, the weight of the generalized syndrome has to decrease at least by 1. On the other hand, since $a \geq a_1$, then from (8) it follows that a must be larger than $\ell(W_0 + W_+)/2$. Thus, we obtain the inequality

$$\ell W_0 - W_+ > \frac{\ell(W_0 + W_+)}{2}$$

which yields

$$W_+ < W_0 \frac{\ell}{\ell + 2}.$$

Thus, we obtain from the conditions of the lemma that the number of errors introduced by decoding does not exceed the initial number of errors. □

Lemma 4 *For any H-LDPC code from the ensemble $\mathscr{C}(n_0, \ell, b)$, if an error pattern is such that in each decoding iteration of the algorithm $\mathscr{A}$ the number of corrected errors is larger than the number of introduced errors, then the algorithm $\mathscr{A}$ yields a correct decision after $\mathcal{O}(\log n)$ iterations, where $n = bn_0$ is the code length.*

Proof. Let $W = \omega n$ be the weight of the error pattern, and let ε denote a lower bound on the fraction of errors that are corrected in each iteration, $0 < \varepsilon < 1$. Then, after x iterations, the number of remaining errors is at most $\omega n(1-\varepsilon)^x$. The final decoding iteration $i_{\max}$ is reached when

$$\omega n(1-\varepsilon)^{i_{\max}} < 1$$

that is,

$$\log(\omega n) + i_{\max} \log(1-\varepsilon) < 0$$

which yields

$$i_{\max} < \frac{1}{\log\left(\frac{1}{1-\varepsilon}\right)} \log(\omega n). \tag{10}$$

Thus, the number of iterations is a logarithmic function of the code length. □

[2] Hamming codes are perfect single-error correcting codes with covering radius equal to 1. Thus, erroneous maximum-likelihood decoding of a Hamming code introduces at most one additional error.

The complexity of each decoding iteration of the algorithm $\mathscr{A}$ is proportional to the code length n. Thus, according to Lemma 4, the overall decoding complexity is $\mathcal{O}(n \log n)$, given that the number of introduced errors is smaller than the initial number of errors, which was shown to hold under the conditions of Lemma 3.

4 Asymptotic performance

As shown in the previous section, the iterative algorithm $\mathscr{A}$ corrects any error pattern with $W < W_{\alpha_1}/2$ errors, if the number of constituent codes affected by exactly one error is $a_1 > \ell W/2$. The question that arises, however, is whether such a code exists within the ensemble $\mathscr{C}(n_0, \ell, b)$. The following theorem allows us to receive the positive answer.

Theorem 1 *In the ensemble $\mathscr{C}(n_0, \ell, b)$ of H-LDPC codes, there exist codes (with probability p, where $\lim_{n\to\infty} p = 1$), which can correct any error pattern of weight up to $\omega_{\alpha_1} n/2$, with decoding complexity $\mathcal{O}(n \log n)$. The value ω_{α_1} is the largest root of the equation*

$$h(\omega) - \ell F(\alpha_1, \omega, n_0) = 0 \tag{11}$$

where $h(\omega) = -\omega \log_2 \omega - (1-\omega)\log_2(1-\omega)$ and the function $F(\alpha_1, \omega, n_0)$ is given by

$$F(\alpha_1, \omega, n_0) \triangleq h(\omega) - \frac{1}{n_0} h(\alpha_1 \omega n_0) + \max\Bigg\{\omega \log_2 s \\ -\frac{1}{n_0}\log_2\Big((1+s)^{n_0} - n_0 s\Big) + \alpha_1 \omega \log_2\left(\frac{(1+s)^{n_0}}{n_0 s} - 1\right)\Bigg\} \tag{12}$$

where $\alpha_1 > 1/2$ and the maximization is performed over all s such that

$$\frac{(1+s)^{n_0}}{n_0 s} \leq \frac{1}{\alpha_1 \omega n_0}.$$

Proof. For a *fixed* combination of $W = \omega n$ errors, the probability that the number of constituent Hamming codes of an H-LDPC code from the ensemble $\mathscr{C}(n_0, \ell, b)$ that are affected by a single error, will not exceed a certain value $\alpha_1 \ell W$ is upper-bounded by:

$$P(a_1 \leq \alpha_1 \ell W) \leq 2^{-n\ell F(\alpha_1, \omega, n_0)} \tag{13}$$

where the function $F(\alpha_1, \omega, n_0)$ is given by (12). The proof of this statement follows Appendix 1 in [8] and is omitted here for brevity.

Now consider the probability that the number of constituent codes with a single error is at most $\alpha_1 \ell W$ for *any* error pattern of a given weight W. If this

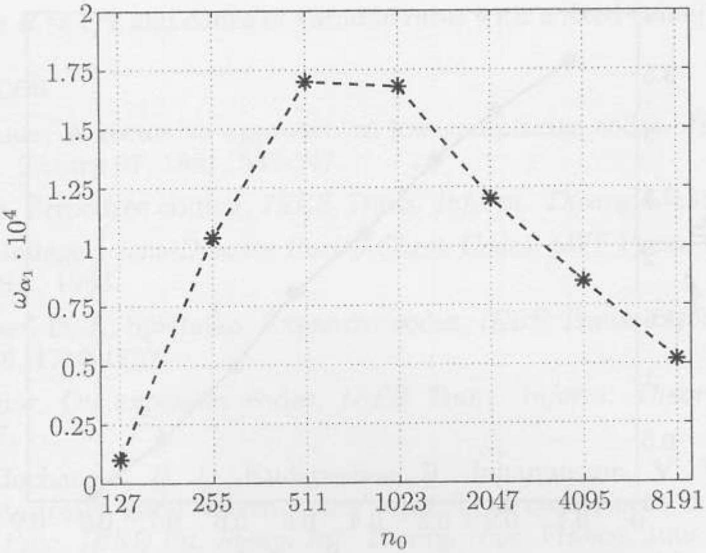

Figure 3: Values of ω_{α_1} computed according to Theorem 1 for seven code ensembles of rates approximately $R \approx 1/2$, with the number of layers $\ell \in \{9, 16, 28, 51, 93, 171, 315\}$. The maximum is achieved with the constituent code length $n_0 = 511$ and $\ell = 28$.

probability is strictly smaller than 1, then there exist codes in the ensemble $\mathscr{C}(n_0, \ell, b)$ for which $a_1 > \alpha_1 \ell W$ for any weight-W error pattern. Thus, the existence of such codes is ensured if

$$\binom{n}{W} P(a_1 \leq \alpha_1 \ell W) < 1.$$

Taking the logarithm and using the inequalities (13) and

$$\binom{n}{\omega n} \lesssim 2^{nh(\omega)}$$

where the asymptotic equality holds for $n \to \infty$, we readily obtain

$$h(\omega) - \ell F(\alpha_1, \omega, n_0) < 0. \tag{14}$$

The largest value of ω which satisfies (14) for a given α_1 is ω_{α_1}. Finally, we have from Lemmas 3 and 4 that for $\alpha_1 > 1/2$, the algorithm $\mathscr{A}$ corrects up to $\omega_{\alpha_1} n/2$ errors with complexity $\mathcal{O}(n \log n)$, which completes the proof. □

Theorem 1 allows us to compute ω_{α_1} numerically for several choices of code parameters. The computations confirm the existence of codes with a nonvanishing ω_{α_1}. First, we consider code ensembles of rates close to 1/2. Figure 3 illustrates the values of ω_{α_1} computed for several such code ensembles $\mathscr{C}(n_0, \ell, b)$.

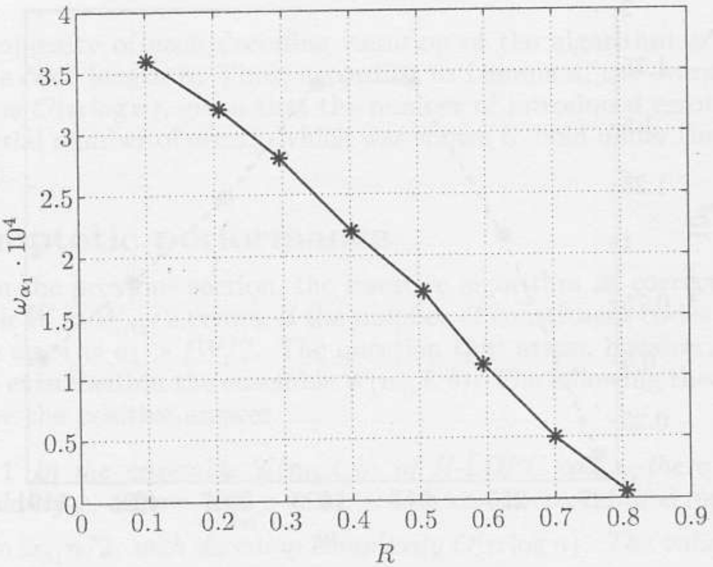

Figure 4: Values of ω_{α_1} computed according to Theorem 1 for several code ensembles of different rates with the fixed constituent code length $n_0 = 511$ and with the number of layers $\ell \in \{51, 45, 40, 34, 28, 23, 17, 11\}$.

With increasing n_0 (and, in order to keep the rate fixed, also with increasing ℓ) the value of ω_{α_1} increases only up to a certain point, $n_0 = 511$, where it reaches its maximum. With further increase of n_0 and ℓ, ω_{α_1} decays quickly.

Next we consider code ensembles of different rates, but with a fixed constituent code. Figure 4 illustrates the values ω_{α_1} for H-LDPC codes with the constituent $(511, 502, 3)$ Hamming code and with different code rates R, obtained by varying ℓ. We have found a nonvanishing ω_{α_1} for a wide range of rates, and its value decreases with increasing rate. Note that all the code ensembles considered in Figures 3 and 4 have minimum distances that almost meet the VG bound, as shown in [14].

5 Conclusions

We have studied the performance of ensembles of Hamming code-based LDPC codes used over the BSC, when the code length n grows to infinity. It was shown that these codes can be decoded with a simple iterative decoding algorithm whose complexity is $\mathcal{O}(n \log n)$, and that there exist H-LDPC codes which, when decoded with such an algorithm, are asymptotically capable of correcting a number of errors that grows linearly with the code length n. Such a property was previously proven to hold for Gallager's LDPC codes and for the expander codes. The maximum fraction of errors correctable with the iterative decoder was computed numerically for two types of code ensembles, which are known to have minimum distances that asymptotically almost meet the VG bound:

codes of rate $R \approx 1/2$ and codes of variable rates with a fixed constituent code.

References

[1] M. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory* 27, 1981, 533-547.

[2] P. Elias, Error-free coding, *IEEE Trans. Inform. Theory* 4, 1954, 29-37.

[3] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, USA, 1963.

[4] M. Sipser, D. A. Spielman, Expander codes, *IEEE Trans. Inform. Theory*, 42, 1996, 1710-1722.

[5] G. Zémor, On expander codes, *IEEE Trans. Inform. Theory* 47, 2001, 835-837.

[6] I. E. Bocharova, B. D. Kudryashov, R. Johannesson, V. V. Zyablov, Asymptotically good woven codes with fixed constituent convolutional codes, *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, June 2007.

[7] I. E. Bocharova, B. D. Kudryashov, R. Johannesson, V. V. Zyablov, Woven graph codes over hyper graphs, *Proc. 7th Int. ITG Conf. Source and Channel Coding*, Ulm, Germany, 2008.

[8] V. V. Zyablov, M. S. Pinsker, Estimation of the error-correction complexity for Gallager low-density codes, *Probl. Inform. Transm.* 11, 1, 1975, 23-26.

[9] A. Barg, G. Zémor, Error exponents of expander codes, *IEEE Trans. Inform. Theory* 48, 2002, 1725-1729.

[10] M. Lentmaier, K. Zigangirov, On generalized low-density parity-check codes based on Hamming component codes, *IEEE Commun. Lett.* 3, 1999, 248-250.

[11] J. Boutros, O. Pothier, G. Zémor, Generalized low density (Tanner) codes, *Proc. IEEE Int. Comm. Conf.*, Vancouver, Canada, June 1999, 1, 441-445.

[12] N. Miladinović, M. Fossorier, Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels, *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, St. Louis, MO, USA, Nov. 2005.

[13] M. Lentmaier, K. Zigangirov, Iterative decoding of generalized low-density parity-check codes, *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Aug. 1998.

[14] S. Stiglmayr, V. V. Zyablov, Asymptotically good low-density codes based on Hamming codes, *Proc. XI Int. Symp. Probl. Redund. Inform. Contr. Syst.*, online at http://www.k36.org/redundancy2007/proceedings.php, Saint Petersburg, Russia, July 2007, 98-103.

[15] L. A. Bassalygo, Asymptotic optimal switching systems, *Probl. Inform. Transm.* 17, 1981, 81-88.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 338-347*

On the erasure-correcting capabilities of low-complexity decoded LDPC codes with constituent Hamming codes

VICTOR ZYABLOV zyablov@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences,
Moscow 101447, RUSSIA

MAJA LONČAR maja@eit.lth.se
ROLF JOHANNESSON rolf@eit.lth.se
Dept. of Electrical and Information Technology, Lund University,
P. O. Box 118, SE-22100 Lund, SWEDEN

PAVEL RYBIN* prybin@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences,
Moscow 101447, RUSSIA

Abstract. Low-density parity-check (LDPC) codes can be constructed using constituent block codes other than single parity-check (SPC) codes. This paper considers random LDPC codes with constituent Hamming codes and investigates their asymptotic performance over the binary erasure channel. It is shown that there exist Hamming code-based LDPC codes which, when decoded with a low-complexity iterative algorithm, are capable of correcting any erasure pattern with a number of erasures that grows linearly with the code length. The number of decoding iterations, required to correct the erasures, is a logarithmic function of the code length. The fraction of correctable erasures is computed numerically for various choices of code parameters.

1 Introduction

Gallager's low-density parity-check (LDPC) codes [1] are characterized by a sparse parity-check matrix whose rows specify single parity-check (SPC) codes over small subsets of the code symbols. LDPC codes can be represented by a bipartite Tanner graph [2], whose two disjoint sets of vertices, referred to as the variable nodes and the constraint nodes, correspond to code symbols and SPC constraints, respectively. The adjacency matrix of such a bipartite

This work was supported in part by the Royal Swedish Academy of Sciences in cooperation with the Russian Academy of Sciences, in part by the Swedish Research Council under Grant 621-2007-6281, and in part by the Program of fundamentals scientific researches of DITCS RAS (OITVS RAN), Project no. 5.1.

graph coincides with the code's parity-check matrix $\boldsymbol{H}$; an element $(\boldsymbol{H})_{ij} = 1$ indicates that the jth code symbol participates in the ith SPC code, that is, there is a branch connecting the jth variable node with the ith constraint node. For regular LDPC codes, the corresponding graph is regular: all the variable nodes have the same degree, equal to the number of ones in the each column of $\boldsymbol{H}$, and all the constraint nodes have the degree equal to the number of ones in each row of $\boldsymbol{H}$ (which is the length of the corresponding SPC code).

Alternative constructions of LDPC codes can be obtained by 'replacing' the SPC codes in the code's Tanner graph with different constituent block codes of length equal to the constraint-node degree. The so-obtained LDPC codes are often referred to as the generalized LDPC codes, *cf.*, *e.g.*, [3], [4]. Starting from the sparse adjacency matrix of the underlying Tanner graph, the parity-check matrix of such an LDPC code is obtained by replacing every 1 in the graph's adjacency matrix with a column of the constituent code's parity-check matrix, and every 0 with an all-zero column.

This paper focuses on LDPC codes with constituent Hamming codes and investigates their performance when communicating over the binary erasure channel (BEC). The erasure-correcting capabilities of Gallager's LDPC codes for the BEC were studied in [5], where it was shown that there exist LDPC codes capable of correcting a portion of erasures that grows linearly with the code length n, with decoding complexity $\mathcal{O}(n \log n)$. Hamming code-based LDPC (H-LDPC) codes were first studied in [3]; their distance properties and iterative soft-decision decoding for the AWGN channel were further investigated in [6] and [7]. Recently, it was shown in [8] that the ensemble of H-LDPC codes contains codes with a minimum distance that asymptotically almost meets the Varshamov-Gilbert bound.

In this work, we build upon the results of [5] and we investigate the asymptotic erasure-correcting capabilities of random H-LDPC codes, when the code length n grows to infinity. We will consider a simple iterative decoder whose complexity is $\mathcal{O}(n \log n)$, and prove that there exist H-LDPC codes for which such a decoder corrects any erasure pattern with a number of erasures growing linearly with the code length. The paper is organized as follows: ensembles of H-LDPC codes and their properties are introduced in Section 2. The decoding algorithm is presented in Section 3. The main result is presented in Section 4 and supported by numerical examples. Section 5 summarizes and concludes the paper.

2 Construction and Properties of H-LDPC Codes

For any integer $m \geq 2$, there exists a Hamming code of length $n_0 = 2^m - 1$, dimension $k_0 = n_0 - m$, minimum distance $d_0 = 3$, and rate $R_0 = 1 - m/n_0$. The parity-check matrix $\boldsymbol{H}_0$ of an (n_0, k_0, d_0) Hamming code is an $m \times n_0$ matrix whose columns are all the distinct nonzero binary m-tuples. When a Hamming

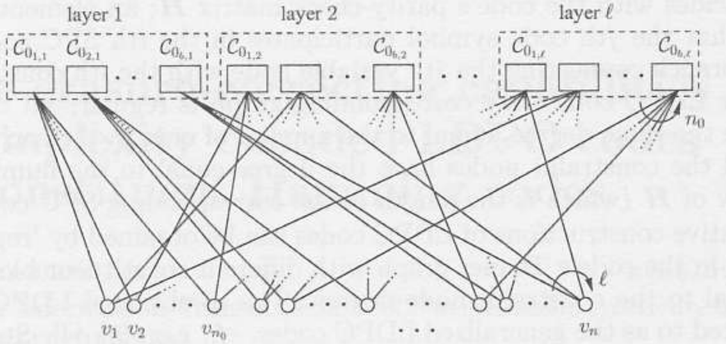

Figure 1: Tanner graph of an H-LDPC code with parity-check matrix given by (2).

code is used for communication over a BEC, it is guaranteed by the code's minimum distance that any erasure pattern with $d_0-1=2$ or fewer erasures will be corrected. Furthermore, it is also possible to correct some erasure patterns with d_0 or more (up to m) erasures, as will be discussed in detail in Section 3.

We consider H-LDPC codes whose bipartite Tanner graph is regular, with the same Hamming code associated with each constraint node. In order to construct a parity-check matrix of such an LDPC code, we start from a $bm \times bn_0$ block-diagonal matrix $\boldsymbol{H}_{\mathrm{b}}$ with the b constituent parity-check matrices $\boldsymbol{H}_0$ on the main diagonal, that is,

$$\boldsymbol{H}_{\mathrm{b}} = \begin{pmatrix} \boldsymbol{H}_0 & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{H}_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \boldsymbol{H}_0 \end{pmatrix} \tag{1}$$

where b is very large. Let $\pi(\boldsymbol{H}_{\mathrm{b}})$ denote a random column permutation of $\boldsymbol{H}_{\mathrm{b}}$. Then the matrix constructed using $\ell \geq 2$ such permutations as *layers*,

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \\ \vdots \\ \boldsymbol{H}_\ell \end{pmatrix} = \begin{pmatrix} \pi_1(\boldsymbol{H}_{\mathrm{b}}) \\ \pi_2(\boldsymbol{H}_{\mathrm{b}}) \\ \vdots \\ \pi_\ell(\boldsymbol{H}_{\mathrm{b}}) \end{pmatrix} \tag{2}$$

is a sparse $\ell bm \times bn_0$ parity-check matrix of a Hamming code-based LDPC code of length $n=bn_0$, where $n \gg n_0$. For a given constituent Hamming code with parity-check matrix $\boldsymbol{H}_0$, by sampling independently the permutations π_l, $l=1,2,...,\ell$, which are all equiprobable, we obtain the ensemble of H-LDPC codes, which will be denoted by $\mathscr{C}(n_0,\ell,b)$. The rate of a code $\mathcal{C} \in \mathscr{C}(n_0,\ell,b)$

is lower-bounded by [2]

$$R \geq 1 - \frac{\ell b(n_0 - k_0)}{n} = 1 - \ell(1 - R_0) \tag{3}$$

with equality iff the matrix $\boldsymbol{H}$ has full rank. Since the rate must be positive, (3) implies a restriction on the rate of the constituent codes, namely,

$$R_0 > 1 - \frac{1}{\ell}$$

that is, the more layers there are, the higher the rate of the constituent codes must be.

Note that by replacing the Hamming constituent code with the $(n_0, n_0–1, 2)$ SPC code, that is, by setting $\boldsymbol{H}_0 = (1\ 1\ ...\ 1)$, the construction defined by (2) reduces to Gallager's construction [1] of the (ℓ, n_0)-regular LDPC matrices.

An H-LDPC code from the ensemble $\mathscr{C}(n_0, \ell, b)$ contains ℓb constituent Hamming codes; b in each layer. The jth constituent code in the lth layer is denoted by $\mathcal{C}_{0j,l}$, and its parity-check matrix by $\boldsymbol{H}_{0j,l}$, $j = 1, 2, ..., b$, $l = 1, 2, ..., \ell$ (all matrices $\boldsymbol{H}_{0j,l}$ are equal up to column permutations). The Tanner graph representation of such an H-LDPC code is illustrated in Figure 1. There are $n = bn_0$ variable nodes and ℓb constraint nodes. The graph is regular, with the variable-node degree equal to ℓ, and the constraint-node degree equal to n_0. Each variable node is connected to exactly one constraint node in each layer.

3 Decoding Algorithm

Let $\boldsymbol{v}$ be a codeword of an H-LDPC code, transmitted over a BEC with the erasure probability δ, and let $\boldsymbol{r}$ denote the received sequence. The number of erasures in the received sequence $\boldsymbol{r}$ is denoted by W. When the code length is large, $n \to \infty$, the fraction of the erased symbols, $\omega = W/n$, converges to the erasure probability δ of the BEC, $\omega \to \delta$.

Consider an iterative erasure-correcting algorithm $\mathscr{A}$, with two variants denoted by $\mathscr{A}_1$ and $\mathscr{A}_2$, whose iterations i, $i = 1, 2, ..., i_{\max}$, consist of the following steps:

(1) For the tentative sequence $\boldsymbol{r}^{(i)}$, where $\boldsymbol{r}^{(1)}$ is the received sequence $\boldsymbol{r}$, select constituent codes $\mathcal{C}_{0j,l}$ with $\tau_{j,l}$ erasures, $j = 1, 2, ..., b$, $l = 1, 2, ..., \ell$, such that:

(a) $\tau_{j,l} < d_0$ for algorithm $\mathscr{A}_1$

(b) $\tau_{j,l} \leq m$ for algorithm $\mathscr{A}_2$

(2) Assuming 0s on the erased positions, compute the syndromes $\boldsymbol{s}_{j,l}$ for the constituent codes selected in the previous step.

(3) For each selected constituent code $\mathcal{C}_{0j,l}$, construct the $m \times \tau_{j,l}$ matrix $\boldsymbol{M}_{j,l}$ whose columns are the $\tau_{j,l}$ columns of $\boldsymbol{H}_{0j,l}$ which correspond to the erased positions. Note that, in general, $\text{rank}(\boldsymbol{M}_{j,l}) \leq \tau_{j,l} \leq m$.

Let $\boldsymbol{x}_{j,l}$ denote the $\tau_{j,l}$-tuple of the unknown (erased) transmitted symbols. These symbols can be recovered by solving the equation system

$$\boldsymbol{x}_{j,l}\boldsymbol{M}_{j,l}^{\mathrm{T}} = \boldsymbol{s}_{j,l}. \tag{4}$$

Clearly, the equation system has a unique solution iff the matrix $\boldsymbol{M}_{j,l}$ has full rank, that is, iff $\text{rank}(\boldsymbol{M}_{j,l}) = \tau_{j,l}$. Then the erasure pattern is correctable.

(4) For every constituent code affected by a correctable erasure pattern find the erased tuple $\boldsymbol{x}_{j,l}$ by solving (4). Replace the erasures in $\boldsymbol{r}^{(i)}$ with the so-found code symbols. This yields the updated sequence $\boldsymbol{r}^{(i+1)}$.

As mentioned earlier, when using algorithm $\mathscr{A}_2$, only some erasure patterns with more than d_0 erasures, which affect constituent codes, are correctable. The following lemma allows us to determine the exact number of the correctable patterns:

Lemma 1 *Let $\boldsymbol{M}$ be an $m \times \tau$ matrix whose columns are equal to τ columns of a parity-check matrix $\boldsymbol{H}_0$ of a Hamming code of length n_0, where $1 \leq \tau \leq m$ and $m = \log_2(n_0+1)$. Then the number of matrices $\boldsymbol{M}$ that have full rank,* $\text{rank}(\boldsymbol{M}) = \tau$, *is equal to*

$$M(\tau, m) = \frac{1}{\tau!} \prod_{i=0}^{\tau-1} (2^m - 2^i). \tag{5}$$

Proof: The columns of the parity-check matrix $\boldsymbol{H}_0$ of the Hamming code of length $n_0 = 2^m - 1$ are all nonzero binary m-tuples, which span the m-dimensional binary space. Thus, clearly, the number of matrices $\boldsymbol{M}$, constructed from τ columns of $\boldsymbol{H}_0$, which have $\text{rank}(\boldsymbol{M}) = \tau$, is equal to the number of different bases of τ-dimensional subspaces of the m-dimensional space. Let $\{\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_\tau\}$ denote the set of basis vectors of a τ-dimensional subspace. The number of such sets is determined in the following way:

- First, select the vector $\boldsymbol{b}_1$ as any of the $2^m - 1$ nonzero binary m-tuples;
- Select the nonzero vector $\boldsymbol{b}_2$ different from $\boldsymbol{b}_1$, that is, $\boldsymbol{b}_2 \neq c_1\boldsymbol{b}_1$, $c_1 \in \{0,1\}$. There are $2^m - 2$ choices.
- For $i = 3, 4, ..., \tau$, select the nonzero vector $\boldsymbol{b}_i$ such that it is not equal to a linear combination of the previously chosen $i-1$ basis vectors, that is, $\boldsymbol{b}_i \neq c_1\boldsymbol{b}_1 + c_2\boldsymbol{b}_2 + \cdots + c_{i-1}\boldsymbol{b}_{i-1}$, where $c_1, c_2, ..., c_{i-1} \in \{0,1\}$. Clearly, there are $2^m - 2^{i-1}$ choices for $\boldsymbol{b}_i$, $i = 3, 4, ..., \tau$.

Finally, note that the ordering of the basis vectors in the set $\{\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_\tau\}$ is irrelevant. Thus, the total number of bases of τ-dimensional subspaces of the m-dimensional space is

$$M(\tau, m) = \frac{\prod_{i=0}^{\tau-1} (2^m - 2^i)}{\tau!}$$

■

Clearly, for a constituent Hamming code, an erasure pattern with τ erasures is correctable when the matrix $\boldsymbol{M}$, constructed from the τ columns of $\boldsymbol{H}_0$ corresponding to the erased positions, has the rank equal to τ. Thus, we have the following

Corollary 1 *The number of erasure patterns of length $n_0 = 2^m - 1$, with $\tau \leq m$ erasures, which are correctable by a Hamming code of length n_0, is equal to $M(\tau, m)$ given by (5).*

Thus, the generating function for the number of correctable erasure patterns can be defined as

$$g_1(s, n_0) = \sum_{\tau=1}^{m} M(\tau, m) s^\tau = \sum_{\tau=1}^{m} \frac{\prod_{i=0}^{\tau-1} (2^m - 2^i)}{\tau!} s^\tau .$$

Note that the function $g_1(s, n_0)$ can be written as

$$g_1(s, n_0) = \widetilde{g}_1(s, n_0) + \sum_{\tau=3}^{m} \frac{\prod_{i=0}^{\tau-1} (2^m - 2^i)}{\tau!} s^\tau \tag{6}$$

where

$$\widetilde{g}_1(s, n_0) = \binom{n_0}{1} s + \binom{n_0}{2} s^2 \tag{7}$$

is the generating function of all erasure patterns with less than $d_0 = 3$ erasures, which are all correctable.

For a given erasure pattern of length n with W erasures, let a denote the number of constituent codes which are affected by correctable erasures. In general, $a = \alpha W \ell$, where $\alpha \leq 1$. In the algorithm $\mathscr{A}$ it is assumed that the erasure pattern is such that there is at least one constituent code for which the erasures that affect it are correctable. In other words, we assume that $\alpha > 0$. Then, during the first iteration of the algorithm $\mathscr{A}$, all correctable erasures will be corrected, while the uncorrectable ones will result in the decoding failure. Hence, the new erasure pattern, after one decoding iteration, has fewer erasures than the initial erasure pattern. Clearly, if in each of the following iterations, the number of codes with correctable erasures is larger than zero, then the total number of erasures in $\boldsymbol{r}^{(i)}$ will decrease with the iteration number i and the algorithm $\mathscr{A}$ recovers the transmitted codeword, *i.e.*, $\boldsymbol{r}^{(i_{\max})} = \boldsymbol{v}$. Then, we can state the following

Lemma 2 *For any H-LDPC code from the ensemble $\mathscr{C}(n_0, \ell, b)$, if an erasure pattern is such that in each iteration of the algorithm $\mathscr{A}$ the number of constituent codes affected by correctable erasures is larger than zero, then the algorithm $\mathscr{A}$ recovers the transmitted codeword after $\mathcal{O}(\log n)$ iterations, where $n = bn_0$ is the code length.*

Proof: Let ε denote a lower bound on the fraction of erasures that are recovered in each iteration, $0 < \varepsilon < 1$. Then, after x iterations, the number of remaining erasures is at most $\omega n(1-\varepsilon)^x$. The final decoding iteration $i_{\max}$ is reached when

$$\omega n(1-\varepsilon)^{i_{\max}} < 1$$

that is,

$$\log(\omega n) + i_{\max}\log(1-\varepsilon) < 0$$

which yields

$$i_{\max} < \frac{1}{\log\left(\frac{1}{1-\varepsilon}\right)}\log(\omega n). \tag{8}$$

Thus, the number of iterations is a logarithmic function of the code length. ■

The complexity of each iteration of the algorithm $\mathscr{A}$ is proportional to the code length n. Thus, according to Lemma 2, the overall decoding complexity is $\mathcal{O}(n \log n)$.

4 Asymptotic Performance

As shown in the previous section, the iterative algorithm $\mathscr{A}$ corrects any erasure pattern with W or fewer erasures, if in each iteration $\alpha > 0$. The following theorem allows us to confirm the existence of H-LDPC codes for which this condition is fulfilled.

Theorem 1 *In the ensemble $\mathscr{C}(n_0, \ell, b)$ of H-LDPC codes, there exist codes (with probability p, where $\lim_{n\to\infty} p = 1$), which can correct any erasure pattern with up to $\omega_\alpha n$ erasures, with decoding complexity $\mathcal{O}(n \log n)$. The value ω_α is the largest root of the equation*

$$h(\omega) - \ell F(\alpha, \omega, n_0) = 0 \tag{9}$$

where $h(\omega)$ is the binary entropy function, $h(\omega) = -\omega \log_2 \omega - (1-\omega)\log_2(1-\omega)$, and the function $F(\alpha, \omega, n_0)$ is given by

$$F(\alpha,\omega,n_0) \triangleq h(\omega) - \frac{1}{n_0} h(\alpha\omega n_0) + \max\left\{ \omega \log_2 s - \right. \\ \left. -\frac{1}{n_0}\log_2(g_0(s,n_0)) - \alpha\omega\log_2\left(\frac{g_1(s,n_0)}{g_0(s,n_0)}\right)\right\} \quad (10)$$

where $\alpha > 0$ and the maximization is performed over all s such that

$$\frac{\alpha\omega n_0}{1-\alpha\omega n_0} \leq \frac{g_1(s,n_0)}{g_0(s,n_0)}.$$

The function $g_1(s, n_0)$ is the generating function of all the erasure patterns that are correctable by the constituent Hamming codes. It is equal to (7) when the constituent codes correct less than d_0 erasures (algorithm $\mathscr{A}_1$), or equal to (6) when the constituent codes correct up to m erasures (algorithm $\mathscr{A}_2$). $g_0(s, n_0)$ is the generating function of the uncorrectable erasure patterns and it equals

$$g_0(s,n_0) = (1+s)^{n_0} - g_1(s,n_0).$$

The proof of Theorem 1 is omitted here for brevity.

Theorem 1 allows us to compute numerically the fraction of the correctable erasures, ω_α, for several choices of code parameters. The computations confirm the existence of codes with a nonvanishing ω_α. First, we consider code ensembles of rates close to $1/2$. Figure 2 illustrates the values of ω_α computed with $\alpha = 10^{-4}$ for algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$, for several code ensembles of rates $R \approx 1/2$. Using the algorithm $\mathscr{A}_2$ up to 3.5 times more erasures can be corrected than with the algorithm $\mathscr{A}_1$. For both algorithms, with increasing n_0 (and, in order to keep the rate fixed, also with increasing ℓ) the value of ω_α increases only up to a certain point, $n_0 = 127$ for $\mathscr{A}_1$ and $n_0 = 63$ for $\mathscr{A}_2$, where it reaches its maximum. With further increase of n_0 and ℓ, ω_α decays quickly.

Next we consider code ensembles $\mathscr{C}(n_0, \ell, b)$ of different rates, $R \approx \frac{1}{4}$, $R \approx \frac{1}{2}$, and $R \approx \frac{3}{4}$, decoded with the algorithm $\mathscr{A}_2$. Figure 3 illustrates the values ω_α obtained with $\alpha = 10^{-4}$ for several code ensembles of different code rates. We have found a nonvanishing ω_α for different code lengths and rates. With increasing R, the maximum value of ω_α decreases and moves towards longer constituent codes: $n_0 = 31$ for $R \approx \frac{1}{4}$, $n_0 = 63$ for $R \approx \frac{1}{2}$ and $n_0 = 127$ for $R \approx \frac{1}{4}$.

Note that all the code ensembles considered in Figures 2 and 3 have minimum distances that almost meet the Varshamov-Gilbert bound, as shown in [8].

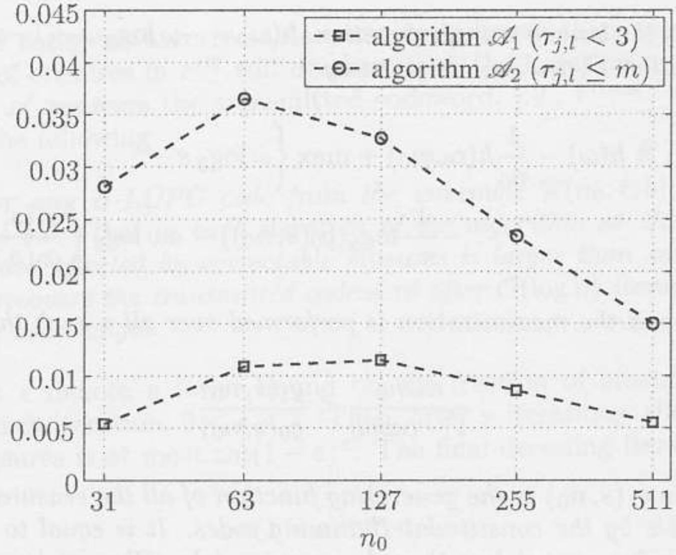

Figure 2: Values of ω_α computed according to Theorem 1 with $\alpha = 10^{-4}$ for decoding algorithms $\mathscr{A}_1$ and $\mathscr{A}_2$, for several code ensembles of rates $R \approx 1/2$.

5 Summary

We have investigated the asymptotic erasure-correcting capabilities of random LDPC codes with constituent Hamming codes, used over the binary erasure channel. A simple iterative decoding algorithm was considered, which can recover the transmitted codeword after $\mathcal{O}(\log n)$ iterations, where n is the code length. It was shown that there exist H-LDPC codes which, when decoded with such an algorithm, are capable of correcting a number of erasures that grows linearly with the code length n. The maximum fraction of correctable erasures was computed numerically for several code ensembles with different code rates and constituent-code lengths.

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, USA, 1963.

[2] M. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory* 27, 1981, 533-547.

[3] M. Lentmaier, K. Zigangirov, Iterative decoding of generalized low-density parity-check codes, *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Aug. 1998.

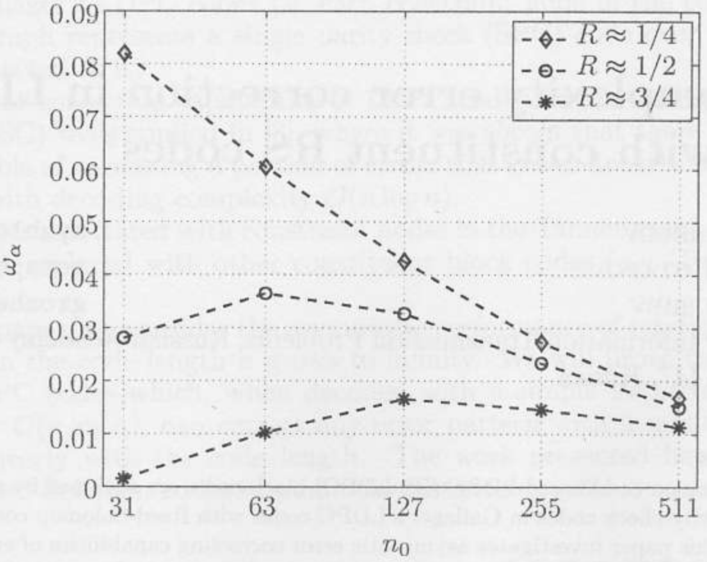

Figure 3: Values of ω_α computed according to Theorem 1 with $\alpha = 10^{-4}$ for the decoding algorithm $\mathscr{A}_2$, for several code ensembles of different rates.

[4] N. Miladinović, M. Fossorier, Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels, *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, St. Louis, MO, USA, 2005.

[5] V. V. Zyablov, M. S. Pinsker, Decoding complexity of low-density codes for erasure channel, *Probl. Inform. Transm.* 10, 1974, 15-28.

[6] M. Lentmaier., K. Zigangirov, On generalized low-density parity-check codes based on Hamming component codes, *IEEE Commun. Lett.* 3, 1999, 248-250.

[7] J. Boutros, O. Pothier, G. Zémor, Generalized low density (Tanner) codes, *Proc. IEEE Int. Comm. Conf.*, Vancouver, Canada, 1, 1999, 441-445.

[8] S. Stiglmayr, V. V. Zyablov, Asymptotically good low-density codes based on Hamming codes, *Proc. XI Int. Symp. Problems of Redundancy in Inform. and Control Systems, available online at http://www.k36.org/redundancy2007/proceedings.php*, Saint Petersburg, Russia, 2007, 98-103.

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria *pp. 348-353*

Low-complexity error correction in LDPC codes with constituent RS codes[1]

Victor Zyablov zyablov@iitp.ru
Vladimir Potapov potapov@iitp.ru
Fedor Groshev groshev@iitp.ru
Institute for Information Transmission Problems, Russian Academy of Sciences,
Moscow 101447, Russia

Abstract.
Reed-Solomon code-based LDPC (RS-LDPC) block codes are obtained by replacing single parity-check codes in Gallager's LDPC codes with Reed-Solomon constituent codes. This paper investigates asymptotic error correcting capabilities of ensembles of random RS-LDPC codes, used over the binary symmetric channel and decoded with a low-complexity harddecision iterative decoding algorithm. The number of required decoding iterations is a logarithmic function of the code length. It is shown that there exist RS-LDPC codes for which such iterative decoding corrects any error pattern with a number of errors that grows linearly with the code length. The results are supported by numerical examples, for various choices of code parameters.

1 Introduction

Long block codes can be obtained by combining one or more simpler codes in various types of concatenated structures. Such constructions are of interest since they can yield powerful codes with good error-correcting capabilities, which are decodable with low complexity, using simple constituent decoders as separate modules.

A method for constructing long codes from short constituent codes, based on bipartite graphs, was introduced by Tanner in [1]. In this method, one of the two sets of nodes in a bipartite graph is associated with code symbols, while the other set is associated with constituent block codes of length equal to the node degree. These two sets of nodes are hereinafter referred to as variable nodes and constraint nodes, respectively. Tanner's general code construction unifies many known code families that can be obtained by choosing different underlying bipartite graphs and associating different constituent codes with its constraint nodes. For example, Gallager's Low-Density Parity-Check (LDPC) codes [2], graph-based approach.

[1] This work was supported by the Program of fundamental scientific reseaches of ONIT RAN (the Department of Nanotechnologies and Information Technologies)

For Gallager's LDPC codes [2], each constraint node in the corresponding bipartite graph represents a single parity-check (SPC) code over the variable nodes connected to it.

The error-correcting capabilities of LDPC codes for the binary symmetric channel (BSC) were studied in [3], where it was shown that there exist LDPC codes capable of correcting a portion of errors that grows linearly with the code length n, with decoding complexity $\mathcal{O}(n \log n)$.

The codes associated with constraint nodes in the Tanner graph of an LDPC code can be replaced with other constituent block codes (*e.g.* Reed-Solomon codes [4]).

In this paper, we consider the asymptotic performance of random RS-LDPC codes, when the code length n grows to infinity. We will prove that there exist RS-LDPC codes which, when decoded with a simple iterative decoder of complexity $\mathcal{O}(n \log n)$, can correct any error pattern with a number of errors growing linearly with the code length. The work presented here, with constituent Reed-Solomon codes of minimum distance $d_0 = 3$.

2 Construction and properties of RS-LDPC codes

An (n_0, k_0, d_0) extended Reed-Solomon code has length $n_0 = 2^q$, dimension $k_0 = n_0 - d_0 - 1$, code rate $R_0 = 1 - (d_0 - 1)/n_0.$. We will consider single-error correcting extended RS code with minimum distance $d_0 = 3$,

A parity-check matrix H_0 of a Reed-Solomon code is an $(d_0 - 1) \times n_0$ matrix whose columns are all nonzero q-nary $(d_0 - 1)$-tuples. We will consider RS-LDPC codes with identical constituent codes. Let H denote a block-diagonal matrix with the b constituent parity-check matrices H_0 on the main diagonal, that is,

$$H = \begin{pmatrix} H_0 & 0 & 0 & \cdots & 0 \\ 0 & H_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & H_0 \end{pmatrix} \tag{2}$$

where b is very large. The matrix H is of size $b(d_0 - 1) \times bn_0$. Let $\pi(H)$ denote a random column permutation of H. Then the matrix constructed using $\ell \geq 2$ such permutations as *layers*,

$$H = \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_\ell \end{pmatrix} = \begin{pmatrix} \pi_1(H) \\ \pi_2(H) \\ \vdots \\ \pi_\ell(H) \end{pmatrix} \tag{3}$$

is a sparse $\ell b(d_0 - 1) \times bn_0$ parity-check matrix which characterizes the ensemble of Reed-Solomon code-based LDPC codes of length $n = bn_0$, where $n \gg n_0$.

Let $C(n_0, \ell, b)$ denote this ensemble. For a given constituent Reed-Solomon code with parity-check matrix H_0, the elements of the ensemble $C(n_0, \ell, b)$ are obtained by sampling independently the permutations π_l, $l = 1, 2, ..., \ell$, which are all equiprobable. The rate of a code $\mathcal{C} \in C(n_0, \ell, b)$ is lower-bounded by [1]

$$R \geq 1 - \frac{\ell b(n_0 - k_0)}{n} = 1 - \ell(1 - R_0) \tag{4}$$

with equality if the matrix H has full rank.

The RS-LDPC codes from the ensemble $C(n_0, \ell, b)$ contain ℓb constituent Reed-Solomon codes; b in each layer. Such RS-LDPC codes can be represented by a Tanner graph [1] with $n = bn_0$ variable nodes, and ℓb constraint nodes. Each constraint node comprises $n_0 - k_0$ parity-check constraints specified by the rows of the corresponding constituent parity-check matrix. If a code symbol is checked by a constituent code (that is, by at least one row of its parity-check matrix), there is a branch connecting the corresponding variable node and the constraint node.. Each code symbol is checked by exactly one Reed-Solomon code in each layer. The graph is regular, with the variable-node degree equal to ℓ, and the constraint-node degree equal to n_0.

Let $\vec{v}$ be the transmitted codeword and $\vec{e}$ be the error pattern. Then the received sequence is given by $\vec{r} = \vec{v} + \vec{e}$. The weight of the error sequence is $W = |\vec{e}|$ and the fraction of erroneous symbols is $\omega = W/n$ for code length $n \to \infty$.

For a given error pattern with W errors, we introduce the ℓ-tuple $\vec{a} = (a_1\ a_2\ ...\ a_\ell)$, where a_l, $l = 1, 2, ..., \ell$, denotes the number of constituent codes at the lth layer whose codewords are affected by errors. Note that $\vec{a}$ contains realizations of ℓ independent random variables that are integer-valued in the range $0 \leq a_l \leq b$, $l = 1, 2, ..., \ell$. Furthermore, let a denote the total number of constituent codes affected by errors, that is,

$$a = |\vec{a}| = \sum_{l=1}^{\ell} a_l.$$

In other words, a is the number of constraint nodes in the Tanner graph that are connected to at least one variable node with an erroneously received value.

3 Decoding algorithm

Consider an iterative hard-decision decoding algorithm A, whose decoding iterations i, $i = 1, 2, ..., i_{\max}$, consist of the following two steps:

1) For the tentative sequence $r^{(i)}$, where $r^{(1)}$ is the received sequence r, decode independently ℓb constituent Reed-Solomon codes (that is, compute their syndromes $s_{j,l}$, $j = 1, 2, ..., b$, $l = 1, 2, ..., \ell$, and if the value

is nonzero, output the n_0-tuple where the position indicated by the syndrome is flipped). This yields ℓ independent decisions for each of the n symbols.

2) Flip every symbol $r_k^{(i)}$, $k = 1, 2, ..., n$, in the sequence $r^{(i)}$, for which *at least one* of the ℓ decisions requires that. This yields the updated sequence $r^{(i+1)}$.

Assume that the error pattern e is such that the number of errors that can be corrected by the constituent codes is larger than the number of uncorrectable errors. Then, during the first iteration of the algorithm A, all correctable errors will be corrected. Since, in our case, Reed-Solomon codes are single-error correcting codes, each erroneous decoding will added one new error. Hence, the new error pattern, resulting from one decoding iteration has fewer errors than the initial error pattern. Clearly, if in each of the following iterations, the number of correctable errors is larger than the number of uncorrectable ones, then the total number of errors in $r^{(i)}$ will decrease with the iteration number i and the algorithm yields the correct decision, *i.e.*, $r^{(i_{\max})} = v$. Then, we can state the following

Lemma 1 For any RS-LDPC code from the ensemble $C(n_0, \ell, b)$, if an error pattern is such that in each iteration of algorithm A the number of errors correctable by the constituent codes is larger in $(1 + \varepsilon)$ times than the number of added errors, then algorithm A yields a correct decision after $\mathcal{O}(\log n)$ iterations, where $n = bn_0$ is the code length.

The complexity of each decoding iteration of the algorithm A is proportional to the code length n. Thus, according to Theorem 1, the overall decoding complexity is $\mathcal{O}(n \log n)$, given that the number of correctable errors in the error pattern is larger than the number of the uncorrectable ones. The following lemma formulates a condition under which this holds.

Lemma 2 If for any error pattern with $w \leq W$ errors, the number of constituent Reed-Solomon codes of an RS-LDPC code from the ensemble $C(n_0, \ell, b)$ that are affected by errors is $a = \alpha w \ell$ with $\alpha \geq 2/3 + \varepsilon$, then the number of correctable errors in any such error pattern is always larger than the number of uncorrectable errors.

In other words, $\alpha \geq 2/3 + \varepsilon$ specifies the necessary expansion of the Tanner (expander) graph of the code, which ensures that the number of errors decreases in each iteration of algorithm A.

4 Asymptotic performance

As shown in the previous section, the iterative algorithm A corrects any error pattern with W or fewer errors, if the code's Tanner graph has the expansion coefficient $\alpha \geq 2/3 + \varepsilon$. The question that arises, however, is whether such a code exists in the ensemble $C(n_0, \ell, b)$. The following theorem allows us to receive the positive answer.

Theorem 1 In the ensemble $C(n_0, \ell, b)$ of RS-LDPC codes, there exist codes (with probability p, where $\lim\limits_{n\to\infty} p = 1$), which can correct any error pattern of weight up to $\omega_\alpha n$, with decoding complexity $\mathcal{O}(n \log n)$. The value ω_α is the largest root of the equation

$$h(\omega) + \omega log_2(q-1) - \ell F(\alpha, \omega, n_0) = 0 \tag{5}$$

where $h(\omega) = -\omega \log_2 \omega - (1-\omega)\log_2(1-\omega)$ and the function $F(\alpha, \omega, n_0)$ is given by

$$\begin{aligned} F(\alpha, \omega, n_0) \triangleq{} & h(\omega) + \omega \log_2(q-1) - \frac{1}{n_0} h(\alpha\omega n_0) \\ & + \max\Big\{\omega \log_2 s - \alpha\omega \log_2\Big(\left(1 + s(q-1)\right)^{n_0} - 1\Big)\Big\} \end{aligned} \tag{6}$$

where $\alpha \geq 2/3 + \varepsilon$ and the maximization is performed over all s such that

$$(1 + s(q-1))^{n_0} < 1 + \frac{1 - \alpha\omega n_0}{\alpha\omega n_0}$$

Theorem 1 allows us to compute ω_α numerically for several choices of code parameters. The computations confirm the existence of codes with a nonvanishing ω_α. We use $\alpha = 0.67$, which is slightly above the limit value of $2/3$. First, we consider code ensembles of a rate close to $1/2$. Figure 1 illustrates the values of ω_α computed for several code ensembles $C(n_0, \ell, b)$ of rates approximately $1/2$. With increasing n_0 (and, in order to keep the rate fixed, also with increasing ℓ) the value of ω_α increases only up to a certain point, $n_0 = 128$, where it reaches its maximum. With further increase of n_0 and ℓ, ω_α decays quickly.

Next we consider code ensembles of different rates, but with a fixed constituent code. Figure 2 illustrates the values ω_α for RS-LDPC codes with the constituent $(128, 126, 3)$ Reed-Solomon code and with different code rates R, obtained by varying the choice of ℓ. We have found a nonvanishing ω_α for a wide range of code rates, and its value decreases with increasing code rate.

References

[1] M. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory* 27,1981, 533-547.

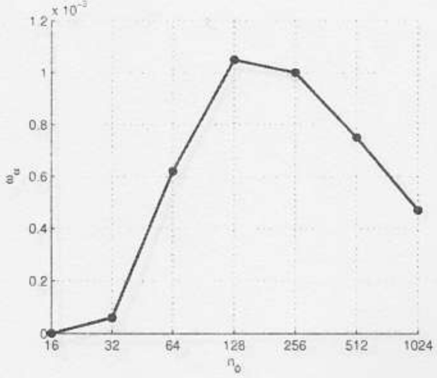

Figure 1: Values of ω_α computed for $\alpha = 0.67$ according to Theorem 1 for several code ensembles of rates approximately $R \approx 1/2$. The maximum is achieved with the constituent code length $n_0 = 128$.

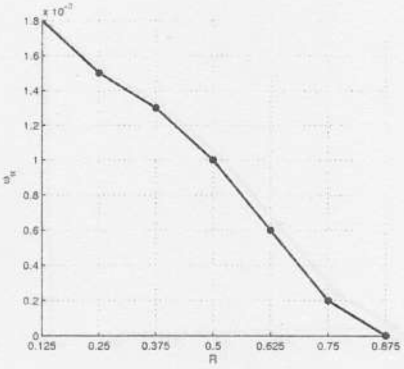

Figure 2: Values of ω_α computed for $\alpha = 0.67$ according to Theorem 1 for several code ensembles of different rates with the fixed constituent code length $n_0 = 128$.

[2] R. G. Gallager, *Low-Density Parity-Check Codes*, Ph.D. thesis, MIT Press, Cambridge, MA, USA, 1963.

[3] V. V. Zyablov, M. S. Pinsker, Estimation of the error-correction complexity for Gallager low-density codes, *Problems of Inform. Transmission* 11, 1975, 23-26.

[4] N. Miladinović, M. Fossorier, Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels, in *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, St. Louis, MO, USA, Nov. 2005.

ISSN 1313-423X