A Hierarchical-based Greedy Algorithm for Echelon-Ferrers Construction

Xianmang He

November 5, 2019

Abstract

Echelon-Ferrers is one of important techniques to help researchers to improve lower bounds for subspace code. Unfortunately, exact computation of echelon ferrers construction is limited by the computation time. In this paper, we show how to attain codes of larger size for a given minimum distance $d = 4$ or $6$ by the hierarchical-based greedy algorithm for echelon-ferrers introduced in [9]. About 63 new constant-dimension subspace codes are better than previously best known codes.

keywords: Echelon-Ferrers, Constant dimension codes, Projective space, Reduced echelon form, Ferrers diagrams

1 Introduction

Subspace coding was proposed by R.Koetter and F.R.Kschischang in [21] to correct errors and erasures in random network coding. The projective space of order $n$ over the finite field $\mathbb{F}_q$, denoted $\mathcal{P}_q(n)$, is the set of all subspaces of the vector space $\mathbb{F}_q^n$. The set of all $k$-dimensional subspaces of an $\mathbb{F}_q$-vector space $V$ will be denoted by $\mathcal{G}_q(k, n)$. For $n = \dim(V)$, its cardinality is given by the Gaussian binomial coefficient

$$|\mathcal{G}_q(k, n)| = \begin{cases} \frac{(q^n-1)(q^{n-1}-1)\cdots(q^{n-k+1}-1)}{(q^{k-1})(q^{k-1}-1)\cdots(q-1)} & \text{if } 0 \leq k \leq n; \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $\mathcal{P}_q(n) = \bigcup_{0 \leq k \leq n} \mathcal{G}_q(k, n)$.

A widely used distance measure for subspace codes (motivated by an information-theoretic analysis of the Kötter-Kschischang-Silva model, see e.g. [25]) are the subspace distance

$$d_S(U, W) := \dim(U + W) - \dim(U \cap W) = 2 \cdot \dim(U + W) - \dim(U) - \dim(W),$$

1
where \( U \) and \( W \) are subspaces of \( \mathbb{F}_q^n \).

A set \( \mathcal{C} \) of subspaces of \( V \) is called a subspace code. The minimum distance of \( \mathcal{C} \) is given by \( d = \min\{d_S(U, W) \mid U, W \in \mathcal{C}, U \neq W\} \). If the dimension of the codewords, is fixed as \( k \), we use the notation \((n, \#\mathcal{C}, d; k)_q\) and call \( \mathcal{C} \) a constant dimension code (CDC for short). For fixed ambient parameters \( q, n, k \) and \( d \), the main problem of subspace coding asks for the determination of the maximum possible size \( A_q(n, d, k) := M \) of an \((n, M, \geq d, k)_q\) subspace code.

In this paper we give a greedy algorithm for the echelon-ferrers construction. About 127 new constant-dimension subspace codes of larger size for a given minimum distance are illustrated in the table ??.

The remaining part of this paper is structured as follows. The currently implemented lower bounds, constructions, are described in Section ?? The preliminaries are outlined in section ?? Constant dimension codes (CDC) by our algorithm are treated in Section ??, Finally we draw a conclusion in Section ??.

2 Previous constructions

The lower and upper bounds on \( A_q(n, d, k) \) have been intensively studied in the last years, see e.g. ?? The report ?? describes the underlying theoretical base of an on-line database, found at [http://subspacecodes.uni-bayreuth.de](http://subspacecodes.uni-bayreuth.de) and maintained by the research team in the University of Bayreuth that tries to collect up-to-date information on the best lower and upper bounds for subspace codes.

Lifted MRD codes, (we omit the details here, see subsection ??), are one type of building blocks of the Echelon- Ferrers construction, see subsection ?? The latter is a nice interplay between the subspace distance, the rank distance and the Hamming distance. Another construction based on similar ideas is the so-called coset construction ?? The most effective general recursive construction is the linkage construction and its generalization. According the report ??, the lower bound with the highest score is the improved linkage construction, and it yields the best known lower bound in 69.1% of the constant dimension code parameters of the database currently. The linkage construction is to obtain large codes from the subspaces spanned by a given code \( \mathcal{C} \) and choices of an MRD code : rowspace\{(A, Q) \mid A, Q \text{ are sampled from } A_q(n_1, k, d), Q_q(n_2, k, d)\}\). This resulting size of the constructed code is the size of \( \mathcal{C} \) times the size of the MRD code. By performing a tighter analysis of the occurring subspace distances,
papers [24] [13] [?] indicated that codes in a smaller ambient space can be further added.

The expurgation-augmentation method, which starts with a lifted MRD code and then adding and removing codewords, is invented by Thomas Honold. A starting point is possible a computer–free construction for the lower bound \( A_2(7, 4, 3) \geq 329 \), see [22]. The subsequent studies contain \( A_q(6, 4, 3) \geq q^6 + 2q^2 + 2q + 1 \) for \( 3 \leq q \) [19], Theorem 2, \( A_q(7, 4, 3) \geq q^8 + q^5 + q^4 - q - 1 \), [17], and \( A_q(7, 4, 3) \geq q^8 + q^5 + q^4 - 4q - 6 - q^2 \), [13], Theorem 4.

New subspace codes from two parallel versions of maximum rank distance codes was introduced by Xu and Chen [26]. The problem asks for the size of \( A_q(2n, 2(n - t), n) \) const dimension codes was turned to find a suitable sufficient condition to restrict the number of roots of \( L_1(L_2(x)) - x \) to \( q^t \), where \( L_1 \) and \( L_2 \) are \( q \)-polynomials over the extension field \( \mathbb{F}_{q^n} \):

\[
If \quad 2t \geq n, then A_q(2n, 2(n - t), n) \geq q^n(t+1) + \sum_{r=n-t}^n Ar(Q(q, n, t)).
\]

Geometric concepts like the Segre variety and the Veronese variety where also used to obtain constructions for constant dimension codes:

**Theorem 1 ([5, Theorem 3.11 and 3.8])** If \( n \geq 5 \) is odd, then \( A_q(2n, 4, n) \geq q^{n^2-n} + \sum_{r=2}^{n-2} A_r(Q(q, n, n-2)) + \prod_{i=1}^{n-1} (q^i + 1) - q^{\frac{n(n-1)}{2}} - [n]_q \)

\[
\left( q^{\binom{n-1}{2}} - q^{\binom{n-1}{3}} \prod_{i=1}^{n-1} (q^{2i-1} - 1) \right) + y(y-1) + 1, \text{ using } y := q^{n-2} + q^{n-4} + \cdots + q^3 + 1.
\]

If \( n \geq 4 \) is even, then \( A_q(2n, 4, n) \geq q^{n^2-n} + \sum_{r=2}^{n-2} A_r(Q(q, n, n-2)) + (q+1) \left( \prod_{i=1}^{n-1} (q^i + 1) - 2q^{\frac{n(n-1)}{2}} + q^{\frac{n(n-2)}{4}} \prod_{i=1}^{n} (q^{2i-1} - 1) \right) - q \cdot |G| + \left[ \frac{q^2}{1} \right] q^2 \left( \left[ \frac{q^2}{1} \right] q^2 - 1 \right) + 1, \text{ using } |G| = 2 \prod_{i=1}^{n/2-1} (q^{2i} + 1) - 2q^{n(n-2)/4} \text{ if } n/2 \text{ is odd and } |G| = 2 \prod_{i=1}^{n/2-1} (q^{2i} + 1) - 2q^{(n-2)/4} + q^{n-4}/8 \prod_{i=1}^{n/4} (q^{4i-2} - 1) \text{ if } n/2 \text{ is even.}
\]

In general, the exact determination of \( A_q(n, d, k) \) is a hard problem, whether in terms of theory or algorithms. The exact calculation for echelon ferrers construction is constrained by the computation time [8] [14] [9]. A greedy-type approach has been considered by Alexander Shishkin, see [23] and also [2]. It is implemented as **greedy_multicomponent**. In [12] [11] the authors considered block designs as skeleton codes. [4] describes an algorithm to tackle the integer linear optimization problems representing the
q-packing design construction by means of a metaheuristic approach, and gives some improvements on the size of $A_2(n, 4, 3)(7 \leq n \leq 14)$. With a stochastic maximum weight clique algorithm and a systematic consideration of groups, authors in [3] gives some new lower bounds on $A_2(n, 4, 3)$ for $8 \leq n \leq 11$.

3 Preliminaries

3.1 Basic Notation

Let $X$ be a $k$-dimensional subspace of $G_q(k, n)$. We represent $X$ by the matrix $EF_q(X)$ in reduced row echelon form, such that the rows of $EF_q(X)$ form a basis of $X$. The identifying vector of $X$, denoted by $v(X)$, is the binary vector of length $n$ and weight $k$, where the $k$ ones of $v(X)$ are exactly in the positions where $EF_q(X)$ has the leading coefficients (the pivots).

In this section we give the definitions for two structures which are useful in describing a subspace in $P_q(n)$. The reduced row echelon form is a standard way to describe a linear subspace. The Ferrers diagram is a standard way to describe a partition of a given positive integer into positive integers.

A matrix is said to be in row echelon form if each nonzero row has more leading zeroes than the previous row.

A $k \times n$ matrix with rank $k$ is in reduced row echelon form if the following conditions are satisfied.

- The leading coefficient of a row is always to the right of the leading coefficient of the previous row.
- All leading coefficients are ones.
- Every leading coefficient is the only nonzero entry in its column.

A $k$-dimensional subspace $X$ of $\mathbb{F}_q^n$ can be represented by a $k \times n$ generator matrix whose rows form a basis for $X$. We usually represent a codeword of a projective space code by such a matrix. There is exactly one such matrix in reduced row echelon form and it will be denoted by $E(X)$.

A Ferrers diagram represents partitions as patterns of dots with the $i$-th row having the same number of dots as the $i$-th term in the partition. A Ferrers diagram satisfies the following conditions.

- The number of dots in a row is at most the number of dots in the previous row.
• All the dots are shifted to the right of the diagram.

The number of rows (columns) of the Ferrers diagram \( \mathcal{F} \) is the number of dots in the rightmost column (top row) of \( \mathcal{F} \). If the number of rows in the Ferrers diagram is \( m \) and the number of columns is \( \eta \) we say that it is an \( m \times \eta \) Ferrers diagram.

Recall that the Hamming metric on \( \mathbb{F}_q^n \) is defined as \( d_H(u, v) \overset{\text{def}}{=} \text{wt}(u - v) \), where \( \text{wt}(w) \) denotes the number of nonzero entries in the vector \( w \). The following results are useful tools for constructions of subspace codes.

**Proposition 1** ([9]) For \( X, Y \in \mathcal{G}_q(k, n) \) we have

- \( d_S(X, Y) \geq d_H(v(X), v(Y)) \),
- if \( v(X) = v(Y) \), then \( d_S(X, Y) = 2d_R(\text{EF}_q(X), \text{EF}_q(Y)) \).

### 3.2 Lifted MRD codes

A prominent code construction uses maximum rank distance (MRD) codes. For matrices \( A, B \in \mathbb{F}_q^{m \times n} \) the rank distance is defined via \( d_R(A, B) := \text{rk}(A - B) \).

**Theorem 2** (see [10]) Let \( q \) be prime power, \( m, n \geq d \) are positive integers, and \( C \subseteq \mathbb{F}_q^{m \times n} \) be a rank-metric code with minimum rank distance \( d \). Then,

\[
\#C \leq q^{\max\{n,m\} \cdot (\min\{n,m\} - d + 1)}.
\]

Codes attaining this upper bound are called maximum rank distance (MRD) codes. They exist for all (suitable) choices of parameters. Using an \( n \times n \) identity matrix as a prefix one obtains the so-called lifted MRD codes. For any two MRD code \( A \) and \( B \), the subspaces \( U_A \) and \( U_B \) spanned by rows of \( (I_n, A) \) and \( (I_n, B) \) are the same if and only if \( A = B \). The intersection \( U_A \cap U_B \) is the set \( \{ \alpha A : \alpha A = \alpha B, \alpha \in \mathbb{F}_q^n \} \). Thus \( \dim(U_A \cap U_B) \leq n - \text{rank}(A - B) \leq n - d \). The distance of this CDC is \( 2d \). A CDC constructed as above is called a lifted MRD code.

### 3.3 Echelon-Ferrers

In [9] presented the multi-level construction, which was based on lifted MRD codes. Let us briefly review the construction in the following theorem. Let \( 1 \leq k \leq n \) be integers and \( v \in \mathbb{F}_2^k \) a binary vector of weight \( k \). By \( \text{EF}_q(v) \) we denote the set of all \( k \times n \) matrices over \( \mathbb{F}_q \) that are in row-reduced echelon form.
Theorem 3 (see [9]) For integers $k, n, \delta$ with $1 \leq k \leq n$ and $1 \leq \delta \leq \min\{k, n - k\}$, let $\mathcal{B}$ be a binary constant weight code of length $n$, weight $k$, and minimum hamming distance $2\delta$. For each $b \in \mathcal{B}$ let $\mathcal{C}_b$ be a code in $\text{EF}_q(b)$ with minimum rank distance at least $\delta$. Then, $\cup_{b \in \mathcal{B}} \mathcal{C}_b$ is a constant dimension code of dimension $k$ having a subspace distance of at least $2\delta$.

The code $\mathcal{B}$ is also called skeleton code. For $\mathcal{C}_b$ we have the following upper bound:

Theorem 4 (see [9]) Let $\mathcal{F}$ be the Ferrers diagram of $\text{EF}_q(v)$ and $\mathcal{C} \subseteq \text{EF}_q(v)$ be a subspace code having a subspace distance of at least $2\delta$, then

$$\#\mathcal{C} \leq q^\min\{\nu_i : 0 \leq i \leq \delta - 1\},$$

where $\nu_i$ is the number of dots in $\mathcal{F}$, which are neither contained in the first $i$ rows nor contained in the rightmost $\delta - 1 - i$ columns.

The authors of [9] conjecture that Theorem 4 is tight for all parameters $q$, $\mathcal{F}$, and $\delta$. Constructions settling the conjecture in several cases are given in [8].

Let $c(v)$ denote the maximum size of a known MRD code over $\text{EF}_q(v)$ matching distance $d$. The optimal Echelon-Ferrers construction can be modeled as an ILP:

$$\max \sum_{v \in \mathbb{F}_2^n} c(v) \cdot x_v,$$

s.t.

$$x_a + x_b \leq 1 \quad \forall a \neq b \in \mathbb{F}_2^n : d_H(a, b) < d,$$

$$x_v \in \{0, 1\} \quad \forall v \in \mathbb{F}_2^n.$$

This is implemented as echelon_ferrers. However, the evaluation of this ILP is only feasible for rather moderate sized parameters. The Echelon-Ferrers construction has even been fine-tuned to the pending dots [6].

Now, we are ready to give the formal definition about the problem that will be addressed in this paper.

Definition 1 (Problem Definition) Given $n, k, d, q$, there are total $\binom{n}{k}$ different identifying vectors, and each vector corresponding to a certain dimension. Among these vectors, we need to choose a binary vector $x$ to maximize the size of $A_q(n, d, k) \geq \sum_{v \in \mathbb{F}_2^n} c(v) \cdot x_v.$
4 Greedy Algorithm

In this section, we will present the details of the construction: our greedy algorithm. We first briefly review the classic recursive backtracking procedure that exhaustively enumerates all maximal cliques in an undirected graph $G$. Then we provide the greedy algorithm in the rest of the section.

4.1 Classic Maximum Clique Enumeration (MCE)

A classic Maximum Clique Enumeration (MCE) algorithm relies on recursive calls to procedure $MCE$, which is illustrated in Algorithm 1. We denote the set of neighbors of a vertex $v$ by $N(v)$. The algorithm takes a graph $G$ as input and initially invokes $MCE(\emptyset, V, \emptyset)$. In Algorithm 1, the basic idea is to recursively backtrack to add a vertex from the set of candidate vertices in $T$ to grow the current clique $C$. A vertex $v$ is a candidate to $C$ if and only if $v$ is a neighbor of all vertices in $C$. Each time when $C$ is augmented by a vertex $v$, we refine $T$ by keeping only the vertices that are also neighbors of $v$. When $T$ becomes empty, $C$ cannot be further grown. At this point, we need to check whether $C$ is indeed maximal. Towards this, we maintain a set $D$ which keeps the set of vertices that are neighbors of all vertices in $C$ and have been outputted as part of some maximal clique earlier, i.e., the recursive procedure has outputted some maximal clique $C \supseteq (C \cup \{v\})$ earlier, where $v \in D$. Thus, if $D$ is not empty, $C$ is not a maximal clique; otherwise, we output $C$ as a maximal clique.

In the worst case, the algorithm can be achieved in $O(3^{|V|/3})$ time complexity. The time taken to compute and output the set of all maximal cliques is acceptable when the $|V|$ is small. The following algorithm makes use of this feature. On a normal PC machine, when the size of $V$ is under 80, the classic maximum clique enumeration algorithm can be calculated in a few minutes.

4.2 Algorithm

As mentioned in Section 3.3, the optimal Echelon-Ferrers construction of code $\mathcal{B}$ can be modeled as an Integer Linear Programming (ILP). Consider that the evaluation of this ILP is only feasible for rather moderate sized parameters, we present a hierarchical-based greedy algorithm as illustrated in the following. The greedy algorithm iteratively maintains a set $S_v$ of identifying vectors. The algorithm starts by initializing a set of all the \( \binom{n}{k} \) identifying vectors denoted by $V_{set}$, and computing its corresponding
Algorithm 1 MCE($C, T, D$)

1: if $T = \emptyset$ and $D = \emptyset$ then
2: output $C$ as a maximal clique;
3: return;
4: end if
5: choose a pivot vertex $v_p$ from $T \cup D$
6: $T' \leftarrow T - N(v_p)$
7: for each $v \in T'$ do
8: call MCE($C \cup \{v\}, T \cap N(v), D \cap N(v)$)
9: $D \leftarrow D \cup \{v\}$
10: end for

In the above algorithm, the way to choose the clique is critical for the resulting solution. Suppose that cliques $(c_1, \cdots, c_m)$ were calculated from
the previous step. We pick the click with largest codes into $S_v$. If there exists serval clicks with same largest codes, we need to evaluate the impact on the subsequent selection after joining the result set $S_v$. Towards this, suppose that $c_i$ was added to $S_v$, we choose the vectors with dimensions from $i - 1$ to $i - \text{depth} - 1$, which were compatible to the new result set $S_v$, we invoke Algorithm MCE again to generate all the possible cliques. Among all the $m$ cliques, we pick the one that maximizes the total number of codes. The parameter $\text{depth}$ makes the MCE can be finished in acceptable time.

**Example 1** Let $q$ be any prime power, $C$ be $A_q(13, 4, 5)$, we observe that $(\binom{13}{5}) = 1287$ total identifying vectors. After apply the greedy algorithm, we obtain 100 identifying vectors, 24 of which are illustrated in table 1. With this, the codes of $A_q(13, 4, 5)$ have the cardinalities $A_q(13, 4, 5) \geq q^{32} + q^{28} + q^{26} + 8q^{24} + 3q^{23} + 3q^{22} + q^{21} + 4q^{20} + 4q^{19} + 4q^{18} + 4q^{17} + 4q^{16} + 6q^{15} + 12q^{14} + 7q^{13} + 6q^{12} + 5q^{11} + 2q^{10} + 8q^{9} + 4q^{8} + 3q^{7} + q^{6} + 4q^{4} + q^{3} + 3q^{2} + q + 1$.

Table 1 gives some new lower bounds for codes $A_q(13, 4, 5)$.

| identifying vector | dimension | identifying vector | dimension |
|--------------------|-----------|--------------------|-----------|
| 1 1111100000000    | 32        | 13 0110101010000  | 22        |
| 2 1110011000000    | 28        | 14 0110110001000  | 22        |
| 3 1101010100000    | 26        | 15 0110010010000  | 22        |
| 4 1011001100000    | 24        | 16 1010101001000  | 21        |
| 5 1001111000000    | 24        | 17 1110000001100  | 20        |
| 6 1100110010000    | 24        | 18 0111110001000  | 20        |
| 7 1010110100000    | 24        | 19 0101100110000  | 20        |
| 8 1110000110000    | 24        | 20 0111000101000  | 20        |
| 9 1100101100000    | 24        | 21 0011100101000  | 19        |
| 10 0111010010000   | 24        | 22 0011101001000  | 19        |
| 11 1101001010000   | 24        | 23 0011110000100  | 19        |
| 12 1011010001000   | 23        | 24 1011000010100  | 19        |

It has been proved that for general diagrams $F$, the bound of Theorem 4 is attained for $\delta = 2, 3$ (see [9, 8] for more details). The improvements on CDC codes are given in Table 2-3, achieved by our greedy algorithm. All the codes are attached in the Supplementary material.
5 Discussion

The echelon-ferrers construction is an important method to construct the const dimension code. One of the outstanding advantages is that this method can be applied to various parameters. In this paper, we give a greedy algorithm for the echelon-ferrers construction. About 63 improvements are given by our greedy algorithm. It is also interesting if the greedy algorithm of this paper can be improved to get larger codes.

Table 2: New constant subspace codes in the case $A_q(n,4,k)$

| $A_q(n,4,k)$ | New        | Old        |
|--------------|-------------|------------|
| $A_2(13,4,5)$ | 4796417559  | 4794061075 |
| $A_3(13,4,5)$ | 1880918023783990 | 1853306869495369 |
| $A_4(13,4,5)$ | 18525690479132333173 | 18447026753270989253 |
| $A_5(13,4,5)$ | 23322304248923865096456 | 2328312407045023029131 |
| $A_7(13,4,5)$ | 1104898620939789578683671514 | 110442786590684544176605829 |
| $A_8(13,4,5)$ | 7924784616391565520842806985 | 7922816523780456983985067529 |
| $A_9(13,4,5)$ | 3434214279120353599762054717228 | 3433683900071278242100477868743 |
| $A_2(14,4,5)$ | 76745404672 | 76641774536 |
| $A_3(14,4,5)$ | 152354354408240436 | 150117856399907497 |
| $A_4(14,4,5)$ | 474257675714559102457 | 47224388488166554449 |
| $A_5(14,4,5)$ | 1457644015479485212082050 | 1455195254051476527718901 |
| $A_7(14,4,5)$ | 265286158875282767080909163052 | 2651731306042334943557302058601 |
| $A_8(14,4,5)$ | 3245991778738338095324360943616 | 324518573006045405876299328553537 |
| $A_9(14,4,5)$ | 2253187988530838997185267308928888 | 2252840000683676576253819820897489 |
| $A_2(13,4,6)$ | 3832515314657 | 38325127529 |
| $A_3(13,4,6)$ | 50782209101569336 | 50031831779643235 |
| $A_4(13,4,6)$ | 1185639430145591024577 | 1180591903396972741061 |
| $A_5(13,4,6)$ | 291528642712172039794126 | 29103831053784642146631 |
| $A_7(13,4,6)$ | 378980216844611802379704124332 | 378818692457327706478178392571 |
| $A_8(13,4,6)$ | 4057489691045648256842730905344 | 4054819212026880567284943689225 |
| $A_9(13,4,6)$ | 250354220254502772509319406311282 | 25031555050732020367253014614865311 |
| $A_2(14,4,6)$ | 1227203232293 | 12340234566810426241 |
| $A_3(14,4,6)$ | 12340234566810426241 | 1214095649227435312865809 |
| $A_4(14,4,6)$ | 911027032108553578429954401 | 911027032108553578429954401 |
| $A_7(14,4,6)$ | 6369520523834151727821917674342793 | 6369520523834151727821917674342793 |
| $A_8(14,4,6)$ | 13295822304541472914409793499570241 | 13295822304541472914409793499570241 |
| \(A_q(n, 4, k)\) | New | Old |
|-----------------|-----|-----|
| \(A_0(14, 4, 6)\) | 1478316635555770209444761739522908440329 | |
| \(A_2(15, 4, 6)\) | 1243233943362040432057180581 | 39267675031563 |
| \(A_3(15, 4, 6)\) | 39267675031563 | 2998676636295383433055 |
| \(A_4(15, 4, 6)\) | 2849596349440811610995019309681 | 2849596349440811610995019309681 |
| \(A_5(15, 4, 6)\) | 10705253144414985109399576183764 9623043 | |
| \(A_7(15, 4, 6)\) | 87293119013046541123889721019479 42905023381 | 4356696385275145145553741586485 379191945 |
| \(A_8(15, 4, 6)\) | 1256703351587805 | 1256703351587805 |
| \(A_9(15, 4, 6)\) | 728678523483522880513165 | 728678523483522880513165 |
| \(A_2(16, 4, 6)\) | 1273071559584674249524907514705 | 1273071559584674249524907514705 |
| \(A_3(16, 4, 6)\) | 889674885949215909452997378522 6401 | 889674885949215909452997378522 6401 |
| \(A_4(16, 4, 6)\) | 1799231895987079405217983712681 67518489205 | 1799231895987079405217983712681 67518489205 |
| \(A_5(16, 4, 6)\) | 1427602271526982761008273790284 715743116587585 | 1427602271526982761008273790284 715743116587585 |
| \(A_7(16, 4, 6)\) | 5154571384601397730287591300333 30465293577720745 | 5154571384601397730287591300333 30465293577720745 |
| \(A_8(16, 4, 6)\) | 40210734642430233 | 40210734642430233 |
| \(A_9(16, 4, 6)\) | 177068857538981556600415147 | 177068857538981556600415147 |
| \(A_2(17, 4, 6)\) | 1303625275416014562978042328889 121 | 1303625275416014562978042328889 121 |
| \(A_3(17, 4, 6)\) | 2780234018500650513000010339634 26380051 | 2780234018500650513000010339634 26380051 |
| \(A_4(17, 4, 6)\) | 3023969047576656762486652218945 3324986367024243 | 3023969047576656762486652218945 3324986367024243 |
| \(A_5(17, 4, 6)\) | 4677967123339541192988109893334 6105696388708610177 | 4677967123339541192988109893334 6105696388708610177 |
| \(A_7(17, 4, 6)\) | 3043722856893271957593877642162 4397010040042569507531 | 3043722856893271957593877642162 4397010040042569507531 |
| \(A_8(17, 4, 6)\) | 313923840120169 | 313923840120169 |
| \(A_9(17, 4, 6)\) | 8096287333738514962426 1328764310870435650696 | 8096287333738514962426 7976649310870435650696 |
| \(A_2(15, 4, 7)\) | 79566863724904828874349525569 79228163694856240990516691397 | 79566863724904828874349525569 79228163694856240990516691397 |
| $A_q(n, 4, k)$ | New | Old |
|---------------|-----|-----|
| $A_5(15, 4, 7)$ | 3558699030750375431966367668488876 | 3552713681710884034734089759357256 |
| $A_7(15, 4, 7)$ | 3671901811166948536632605172636 | 3670336821767294433687538738511835372 |
| $A_8(15, 4, 7)$ | 223062854654900138243771190583555006005322241 | 22300745198571198796074774515321564640669641 |
| $A_9(15, 4, 7)$ | 6363668373167010339661177001673180121197368344 | 6362685441138455139799017829057872405081632 |
| $A_2(16, 4, 7)$ | 12857807555925958656 | 1285807203784040529 |
| $A_3(16, 4, 7)$ | 5902159190709809623868717 | 58149848582764764023592067 |
| $A_4(16, 4, 7)$ | 325005875503996895183219736444928 | 324518573106546434155786844197589 |
| $A_5(16, 4, 7)$ | 55604672369921077974140644073486328125 | 555115145910058648807177885740702975 |
| $A_7(16, 4, 7)$ | 4319955761830991620286839274911971343679804101 | 4318114567708564276760733012970457755883719 |
| $A_8(16, 4, 7)$ | 584745889706780730982259844569690677571597238272 | 5846006549408706324349690039073222774820452937 |
| $A_9(16, 4, 7)$ | 6363668373167010339661177001673180121197368344 | 6362685441138455139799017829057872405081632 |
| $A_2(17, 4, 7)$ | 12857807555925958656 | 1285807203784040529 |
| $A_3(17, 4, 7)$ | 3136648139010930771095048570592 | 3090321523481915926684665395596 |
| $A_4(17, 4, 7)$ | 54677932180604048199931056825130150758912 | 5444518195253588521960427874695135580885 |
| $A_5(17, 4, 7)$ | 13575359458996180137997725978493690490722656250 | 1355252721157995688419374498460502822925406 |
| $A_7(17, 4, 7)$ | 17972879091075079063143558397209055898619209028734124332 | 1797010299920793895843182750235849414381414783356188033 |
| $A_8(17, 4, 7)$ | 153287626511294330205972486322567416153287626511294330205972486322567416 | 153249554088818960355691821511056921795476833271817 |
| $A_9(17, 4, 7)$ | 17972879091075079063143558397209055898619209028734124332 | 1797010299920793895843182750235849414381414783356188033 |
| $A_2(18, 4, 7)$ | 3136648139010930771095048570592 | 3090321523481915926684665395596 |
| $A_3(18, 4, 7)$ | 13575359458996180137997725978493690490722656250 | 1355252721157995688419374498460502822925406 |
| $A_4(18, 4, 7)$ | 54677932180604048199931056825130150758912 | 5444518195253588521960427874695135580885 |
| $A_5(18, 4, 7)$ | 13575359458996180137997725978493690490722656250 | 1355252721157995688419374498460502822925406 |
| $A_7(18, 4, 7)$ | 59737438394835040265326746482178302037689516829976915564 | 5976826389847406387498683542286352299560719286565368 |
| $A_q(n, d, k)$ | New                      | Old                      |
|----------------|--------------------------|--------------------------|
| $A_9(15, 6)$   | 462517456081269363814355839720905589 | 452617456081269363814355839720905589 |
| $A_9(16, 6)$   | 6366805691948346661720218673975  | 6366805691948346661720218673975  |
| $A_9(17, 6)$   | 13292280570887931451837029829242  | 13292280570887931451837029829242  |
| $A_9(18, 6)$   | 14780882979641273909118047876478  | 14780882979641273909118047876478  |
| $A_9(19, 6)$   | 17637885117217               | 17637885117217               |

Table 3: New constant subspace codes in the case $A_q(n, 6, k)$
| $A_q(n, d, k)$ | New                     | Old                     |
|----------------|-------------------------|-------------------------|
| $A_2(14, 6)$  | 94823809349898855629    | 98477255302827825304    |
| $A_3(17, 6)$  | 309486213904426891598430721 | 309485028268160807488274522 |
| $A_5(17, 6)$  | 568434482353194399432042062 | 56843419093638778802784421 |
| $A_7(17, 6)$  | 152867010119499398022264943 | 15286700633047038954609367 |
| $A_9(17, 6)$  | 54451791138356967089577848 | 54451787081424357851974081 |
| $A_9(18, 6)$  | 96977373294263981177227448 | 96977329790957286705989818 |
| $A_2(18, 6)$  | 282206174721269          | 282206169223861          |
| $A_3(18, 6)$  | 797702857541262831049    | 7977052899429655194991    |
| $A_4(18, 6)$  | 7922847075953498436223522065 | 7922846521355437618551984193 |
| $A_5(18, 6)$  | 35527155147074650443812457129881 | 3552715498605378031730651854454 |
| $A_8(18, 6)$  | 36703369129691805465310215681701 | 3670336912690482475539679096987 |
| $A_9(18, 6)$  | 22300745365027101371993243436762 | 22300745365027101371993243436762 |
| $A_5(19, 6)$  | 3636257928719988539679096987 | 3636257928719988539679096987 |
| $A_8(19, 6)$  | 88124789287124625593704300569118173 | 88124789287124625593704300569118173 |
| $A_9(19, 6)$  | 913438530151510072190068510908 | 913438530151510072190068510908 |
| $A_7(19, 6)$  | 22204471966921656798893547819953 | 22204471966921656798893547819953 |
| $A_9(19, 6)$  | 4174557928719988539679096987 | 4174557928719988539679096987 |
| $A_2(14, 7)$  | 6461429901931719995258781459 | 6461429901931719995258781459 |
| $A_3(14, 7)$  | 2028488514440621893555948060816 | 2028488514440621893555948060816 |
| $A_4(14, 7)$  | 2204471966921656798893547819953 | 2204471966921656798893547819953 |
| $A_5(14, 7)$  | 916105199764544          | 916105199764544          |
| $A_7(14, 6)$  | 1474557928719988539679096987 | 1474557928719988539679096987 |
| $A_9(14, 6)$  | 34432185344          | 34432185344          |
| $A_2(14, 7)$  | 50034101937449940    | 50034101937449940    |
| $A_3(14, 7)$  | 118059612953618814976 | 118059612953618814976 |
| $A_4(14, 7)$  | 291038456341002757812500 | 291038456341002757812500 |
| $A_7(14, 7)$  | 37881870165372653972364518864 | 37881870165372653972364518864 |
| $A_q(n, d, k)$ | New          | Old                        |
|----------------|--------------|----------------------------|
| $A_8(14, 6, 7)$ | 40564819509544287726043985346560 | 4056481920730334085292565865025 |
| $A_9(14, 6, 7)$ | 2503155511454435281384458579675756 | 2503155504993241601338448821594903 |
| $A_2(15, 6, 7)$ | 2954463963046489945809 | 29543127928813990579 |
| $A_4(15, 6, 7)$ | 12379447686748014242944778240625 | 1237940039341260356943320718751 |
| $A_5(15, 6, 7)$ | 28421723988664230355836181640625 | 2842170943041260356943320718751 |
| $A_7(15, 6, 7)$ | 1070069070755212446299883649 | 1070069042359893429042904249699254898347 |
| $A_8(15, 6, 7)$ | 4355614329041038258627293015 | 43556142965880142666125207741487691071489 |
| $A_9(15, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_2(16, 6, 7)$ | 2954463963046489945809 | 29543127928813990579 |
| $A_3(16, 6, 7)$ | 12379447686748014242944778240625 | 1237940039341260356943320718751 |
| $A_4(16, 6, 7)$ | 28421723988664230355836181640625 | 2842170943041260356943320718751 |
| $A_5(16, 6, 7)$ | 1070069070755212446299883649 | 1070069042359893429042904249699254898347 |
| $A_7(16, 6, 7)$ | 4355614329041038258627293015 | 43556142965880142666125207741487691071489 |
| $A_8(16, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_9(16, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_2(17, 6, 7)$ | 2954463963046489945809 | 29543127928813990579 |
| $A_3(17, 6, 7)$ | 12379447686748014242944778240625 | 1237940039341260356943320718751 |
| $A_4(17, 6, 7)$ | 28421723988664230355836181640625 | 2842170943041260356943320718751 |
| $A_5(17, 6, 7)$ | 1070069070755212446299883649 | 1070069042359893429042904249699254898347 |
| $A_7(17, 6, 7)$ | 4355614329041038258627293015 | 43556142965880142666125207741487691071489 |
| $A_8(17, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_9(17, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_2(18, 6, 7)$ | 2954463963046489945809 | 29543127928813990579 |
| $A_3(18, 6, 7)$ | 12379447686748014242944778240625 | 1237940039341260356943320718751 |
| $A_4(18, 6, 7)$ | 28421723988664230355836181640625 | 2842170943041260356943320718751 |
| $A_5(18, 6, 7)$ | 1070069070755212446299883649 | 1070069042359893429042904249699254898347 |
| $A_7(18, 6, 7)$ | 4355614329041038258627293015 | 43556142965880142666125207741487691071489 |
| $A_8(18, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_9(18, 6, 7)$ | 8727963590616552703971769305 | 8727963568087712949239031857923967131259815 |
| $A_q(n, d, k)$ | New                                      | Old                                      |
|---------------|------------------------------------------|------------------------------------------|
| $A_4(18, 6, 7)$ | 1298079177949694792623525109 694464     | 1298074215842632726751651324 815360      |
| $A_5(18, 6, 7)$ | 2775558983641348099570285579 71191406250 | 2775555761653840821236376762 39308865625 |
| $A_7(18, 6, 7)$ | 3022680272092038061709235593 39677220999693638844 | 30226801970400796071 249833927949957043 |
| $A_8(18, 6, 7)$ | 4676805274306089197937511219 163163439410138513408 | 4676805239459022261051369956 2803183808164427005952 |
| $A_9(18, 6, 7)$ | 3043257230025904195116850962 59196026892667460249974 | 304325722170468489520140759 7453596033986399175840325 |
| $A_{2}(19, 6, 7)$ | 42393335521752447820465426254 | 42391161229528932943108423780 |
| $A_{3}(19, 6, 7)$ | 132923078240589709748976035 041378304 | 132922797022855912261245205 660897281 |
| $A_{5}(19, 6, 7)$ | 867362182401526117754879617 69104003906250 | 8673617380168252566364035010 5289689859376 |
| $A_{7}(19, 6, 7)$ | 5080218733305001542512879685 5761201122579285830766 | 5080218607397303722266117961 45559406022133779734348 |
| $A_{8}(19, 6, 7)$ | 1532495552284619388500421614 87392054424385745181425664 | 153249554086593241501312907 29297530826480609874673665 |
| $A_{9}(19, 6, 7)$ | 1797010304552996180275709594 710174698404463023145829263594 | 179701029991443993876747917 22255427876333551504982366216 |

References

[1] E. A. Akkoyunlu. The enumeration of maximal cliques of large graphs. *Siam Journal on Computing*, 2(1):1–6, 1973.

[2] Ernst M. Gabidulin, Alexander Shishkin, and Nina I. Pilipchuk. On cardinality of network subspace codes. *Proceedings of the Fourteenth Int. Workshop on Algebraic and Combinatorial Coding Theory (ACCT-XIV)*, 7, 2014.

[3] Michael Braun, Patric R. J. Östergård, and Alfred Wassermann. New lower bounds for binary constant-dimension subspace codes. *Experimental Mathematics*, 27(2):179–183, 2018.
[4] Michael Braun and Jan Reichelt. q-analogs of packing designs. *Journal of Combinatorial Designs*, 22(7):306–321, 2014.

[5] Antonio Cossidente and Francesco Pavese. Subspace codes in pg(2n -1, q). *Combinatorica*, 60(1):1–23, 2014.

[6] T. Etzion and N. Silberstein. Codes and designs related to lifted mrd codes. *IEEE Transactions on Information Theory*, 59(2):1004–1017, 2013.

[7] Tuvi Etzion. Problems on q-analogs in coding theory. *Computer Science*, 2013.

[8] Tuvi Etzion, Elisa Gorla, Alberto Ravagnani, and Antonia Wachter-Zeh. Optimal ferrers diagram rank-metric codes. *IEEE Transactions on Information Theory*, 62(4):1616–1630, 2016.

[9] Tuvi Etzion and Natalia Silberstein. Error-correcting codes in projective spaces via rank-metric codes and ferrers diagrams. *IEEE Transactions on Information Theory*, 55(7):2909–2919, 2009.

[10] E. M. Gabidulin. Theory of codes with maximum rank distance. *Probl.inform.transm*, 21(1):3C16, 1985.

[11] E. M. Gabidulin and N. I. Pilipchuk. Rank subcodes in multicomponent network coding. *Problems of Information Transmission*, 49(1):40–53, 2013.

[12] Ernst M Gabidulin and Nina I Pilipchuk. New multicomponent network codes based on block designs. *Proc. Int. Mathematical Conf.50 years of IITP*, 2011.

[13] Heide Gluesing-Luerssen and Carolyn Troha. Construction of subspace codes through linkage. Advances in Mathematics of Communications, 10(3):525–540, 2016.

[14] Elisa Gorla and Alberto Ravagnani. Subspace codes from ferrers diagrams. *Mathematics*, 2014.

[15] Daniel Heinlein, Michael Kiermaier, Sascha Kurz, and Alfred Wassermann. Tables of subspace codes. 2016.

[16] Daniel Heinlein and Sascha Kurz. Coset construction for subspace codes. *IEEE Transactions on Information Theory*, PP(99):1–1, 2015.
[17] Thomas Honold. Remarks on constant-dimension subspace codes. In Talk at ALCOMA15, March 16, 2015.

[18] Thomas Honold and Michael Kiermaier. On Putative q-Analogues of the Fano Plane and Related Combinatorial Structures. 2015.

[19] Thomas Honold, Michael Kiermaier, and Sascha Kurz. Optimal binary subspace codes of length 6, constant dimension 3 and minimum distance 4. Mathematics, 2013.

[20] Wang Jia, James Cheng, and Wai Chee Fu. Redundancy-aware maximal cliques. In Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2013.

[21] Ralf Koetter and Frank R. Kschischang. Coding for errors and erasures in random network coding. IEEE Transactions on Information Theory, 54(8):3579–3591, 2008.

[22] Thomas Honold Liu Haiteng. A new approach to the main problem of subspace coding. In Communications and Networking in China (CHINACOM), 2014 9th International Conference on, pages 676–677, 2014.

[23] Alexander Shishkin. A combined method of constructing multicomponent network codes. (in Russian), https://mipt.ru//upload/medialibrary/4fe/188-194.pdf, 6(2):188–194, 2014.

[24] Natalia Silberstein and Anna Lena Trautmann. Subspace codes based on graph matchings, ferrers diagrams, and pending blocks. IEEE Transactions on Information Theory, 61(7):3937–3953, 2015.

[25] Danilo Silva, Frank R. Kschischang, and Ralf Koetter. A rank-metric approach to error control in random network coding. IEEE Transactions on Information Theory, 54(9):3951–3967, 2008.

[26] L. Xu and H. Chen. New constant-dimension subspace codes from maximum rank-distance codes. IEEE Transactions on Information Theory, 64(9):6315–6319, 2018.