FURTHER RESULTS ON THE CLASSIFICATION
OF MDS CODES

JANNE I. KOKKALA AND PATRIC R. J. ÖSTERGYRD

Department of Communications and Networking
School of Electrical Engineering, Aalto University
P.O. Box 13000, 00076 Aalto, Finland

(Communicated by Michael Kiermaier)

ABSTRACT. An unrestricted \( q \)-ary maximum distance separable (MDS) code \( C \) with length \( n \) over an alphabet \( A \) (of size \( q \)) is a set of \( q^k \) codewords that are elements of \( A^n \), such that the smallest Hamming distance between two distinct codewords in \( C \) is \( d = n - k + 1 \). Sets of mutually orthogonal Latin squares of orders \( q \leq 9 \), corresponding to \( q \)-ary MDS codes of size \( q^2 \), and \( q \)-ary one-error-correcting MDS codes for \( q \leq 8 \) have been classified in earlier studies. These results are used here to complete the classification of all \( 7 \)-ary and \( 8 \)-ary MDS codes with \( d \geq 3 \) using a computer search.

1. Introduction

A \( q \)-ary code \( C \) of length \( n \), and size \( M \) is a set of \( M \) elements, called codewords, of \( A^n \), where \( A \) is an alphabet of size \( q \). The minimum distance \( d \) of a code is the smallest Hamming distance between any two distinct codewords. A code with these parameters is called an \( (n, M, d) \) code. If \( A \) is a finite field and \( C \) is a vector subspace, then \( C \) is called linear. A code that is not linear is called nonlinear. Codes that can be either linear or nonlinear are called unrestricted. Unless otherwise mentioned, all codes in this work are unrestricted.

In the unrestricted case, two codes, \( C_1 \) and \( C_2 \), are called equivalent if there is an isometry of the ambient Hamming space that maps \( C_1 \) to \( C_2 \). Such isometries are given by a permutation of coordinates followed by permutations of symbols at each coordinate separately [8, Chapter 3.2.1]. We say that an \( (n, M, d) \) code is unique if all codes with the same parameters are equivalent.

An upper bound for the size of an \( (n, M, d) \) code is the Singleton bound,

\[
M \leq q^{n-d+1}.
\]

Codes meeting this bound are called maximum distance separable (MDS). A \( q \)-ary MDS code with length \( n \) and minimum distance \( d \) is called an \( (n,k) \) MDS code, where \( k = n - d + 1 \) (parentheses are used to make a distinction between these unrestricted MDS codes and linear \( [n,k] \) codes). MDS codes have the property that, given any \( k \) coordinates, each \( k \)-tuple of symbols from \( A \) occurs in the given coordinates in exactly one codeword. It has been conjectured (the MDS conjecture) that an MDS code with parameters \( n, k \), and prime power \( q \) with \( 1 \leq k < n - 1 \)

\[
2010 \text{ Mathematics Subject Classification:} \text{ Primary: 94B25; Secondary: 05B15.}
\]

\text{Key words and phrases:} \text{ MDS code, Classification, Latin hypercube.}

J. I. Kokkala is supported by Aalto ELEC Doctoral School, Nokia Foundation, Emil Aaltonen Foundation, and by Academy of Finland Project 289002. P. R. J. Östergård is supported in part by Academy of Finland Project 289002.
exists if and only if \( n \leq q + 1 \), with the exception that when \( q \) is a power of two, MDS codes with \( n = q + 2 \), and \( k = 3 \) or \( k = q - 1 \) exist [22]. For linear codes, the conjecture has been proven when \( q \) is a prime by Ball [4] and when \( q \) is a power of a prime \( p \) and \( k < 2p - 2 \) by Ball and De Beule [5].

MDS codes with \( d = 1 \) or \( k = 1 \) are unique and they are called trivial; the first is the whole space \( \mathcal{A}^n \) and the latter is a repetition code. The case \( k = 2 \) corresponds to sets of mutually orthogonal Latin squares, which have been classified for \( q \leq 9 \) [7]. MDS codes with minimum distance 2, length \( n \), and alphabet size \( q \) correspond to \((n-1)\)-dimensional Latin hypercubes of order \( q \). For \( q = 2,3 \), the codes are trivially unique. Potapov and Krotov [21] give a recursive formula for the number of \((n,n-1)q\) MDS codes but do not classify them up to equivalence. For small \( n \) and \( q \), \((n,n-1)q\) MDS codes have been classified by McKay and Wanless [17].

For \( q = 2 \), nontrivial MDS codes with \( d > 2 \) do not exist. For \( q = 3 \), the only nontrivial MDS code with \( d > 2 \) is the unique \((4,2)3\) MDS code. Alderson [1] showed that the \((6,3)4\) and the \((5,3)4\) MDS codes are unique. The nonexistence of pairs of mutually orthogonal Latin squares of order 6 implies the nonexistence of nontrivial 6-ary MDS codes with \( d \geq 3 \). For \( q = 5,7 \) all MDS codes with \( d \geq 3 \) are equivalent to linear codes, with the exception of the \((4,2)7\) MDS codes [11]. For \( q = 5 \) and given \( n \) and \( k \), all existing \((n,k)q\) MDS codes that are linear are equivalent in terms of a notion of equivalence for linear codes [6], which implies that the existing codes are unique also when treated as unrestricted codes. For \( q = 7,8 \), the MDS codes with \( d = 3 \) were classified in [11, 12]. For \( q = 7 \), some classification results exist for codes with \( d > 3 \) in terms of different notions of equivalence for linear codes; see for example [10].

In this work, we classify all 7-ary and 8-ary MDS codes with \( n > 3 \), \( d > 3 \) by a computer search. This completes the classification of all 7-ary and 8-ary MDS codes with \( d \geq 3 \). In Section 2, we discuss basic properties of MDS codes and computational tools used in this work. Section 3 explains the algorithm used, and the results are given in Section 4. Finally, in Section 5, we discuss the case \( d = 2 \) corresponding to Latin hypercubes for small \( q \).

### 2. Preliminaries

#### 2.1. MDS codes

The operations maintaining equivalence of unrestricted codes form a group of order \((q!)^n n!\), denoted by \( G_n \), that acts on \( \mathcal{A}^n \). The stabilizer of a code \( C \) under this action is called the automorphism group of the code, \( \text{Aut}(C) \).

Let \( C \) be an \((n,k)q\) MDS code. For a symbol \( i \in \mathcal{A} \), we denote by \( C_i \) the \((n-1,k-1)q\) MDS code obtained by removing the last coordinate and retaining the codewords that have symbol \( i \) in that coordinate,

\[
C_i = \{ (c_1, c_2, \ldots, c_{n-1}) : c \in C, c_n = i \}.
\]

Removing a symbol at a given position of each codeword of \( C \) yields an \((n-1,k)q\) MDS code \( C' \). This operation is called puncturing. We say that \( C \) is an extension of \( C' \). An \((n,k)q\) MDS code \( C \) is extendable if there exists an extension of \( C \) that is an \((n+1,k)q\) MDS code.

For an \((n,k)q\) MDS code \( C \) and a symbol \( i \in \mathcal{A} \), we denote by \( Ci \) the code obtained by adding a coordinate with the symbol \( i \) to each codeword,

\[
Ci = \{ (c_1, c_2, \ldots, c_n, i) : c \in C \}.
\]

In general, \( Ci \) is not an MDS code.
The following theorem shows that there is a one-to-one correspondence between extensions of an \((n, k)_q\) MDS code (up to the position of the new coordinate) and labeled partitions of the code into \((n, k-1)_q\) MDS codes.

**Theorem 1.** Let \(C\) be an \((n, k)_q\) MDS code with \(k \geq 2\). (i) Each extension of \(C\) where the new coordinate is the last one can be written as \(\bigcup_{i \in A} C^i\), where the codes \(C^i\) are \((n, k-1)_q\) MDS codes and form a partition of \(C\). (ii) For each labeled partition \(\{C^i\}_{i \in A}\) of \(C\), where the codes \(C^i\) are \((n, k-1)_q\) MDS codes, the code \(\bigcup_{i \in A} C^i\) is an extension of \(C\).

**Proof.** (i) Let \(D\) be an \((n + 1, k)_q\) MDS code that is an extension of \(C\) so that \(C\) is obtained by puncturing \(D\) in the last coordinate. Let \(C^i = D_i\) for \(i \in A\). Now \(D = \bigcup_{i \in A} C^i\), where the codes \(C^i\) are \((n, k-1)_q\) MDS codes. In addition, \(\bigcup_{i \in A} C^i = C\) and \(\sum_{i \in A} |C^i| = \sum_{i \in A} q^{k-1} = q^k = |C|\). Thus the sets \(D_i\) form a partition of \(C\).

(ii) Let \(C^i\) for \(i \in A\) be \((n, k-1)_q\) MDS codes that form a partition of \(C\) and let \(D = \bigcup_{i \in A} C^i\). Now \(C\) can be obtained by puncturing \(D\) in the last coordinate. To show that \(D\) is an extension of \(C\), we show that it is an \((n + 1, k)_q\) MDS code by showing that its minimum distance is at least \(n - k + 2\). Let \(c\) and \(c'\) be distinct codewords of \(D\). If the last symbols of \(c\) and \(c'\) are the same, then \((c_1, \ldots, c_{n-1})\) and \((c'_1, \ldots, c'_{n-1})\) belong to the same \((n, k-1)_q\) MDS code \(C^i\) and their distance is at least \(n - k + 2\). If the last symbols of \(c\) and \(c'\) are different, the distance between \((c_1, \ldots, c_{n-1})\) and \((c'_1, \ldots, c'_{n-1})\) is at least \(n - k + 1\) and the distance between \(c\) and \(c'\) is at least \(n - k + 2\). Therefore, the minimum distance of \(D\) is at least \(n - k + 2\). \(\square\)

As an example of Theorem 1, consider the \((3, 2)_4\) MDS code
\[
C = \{000, 011, 022, 033, 101, 110, 123, 132, 202, 213, 220, 231, 303, 312, 321, 330\}.
\]
This code can be partitioned into four \((3, 1)_4\) MDS codes as
\[
C^0 = \{000, 132, 213, 321\}, \quad C^1 = \{011, 123, 202, 330\},
\]
\[
C^2 = \{022, 110, 231, 303\}, \quad C^3 = \{033, 101, 220, 312\}.
\]
Adding symbol \(i\) to each codeword in \(C^i\), we obtain a \((4, 2)_4\) MDS code
\[
D = \{0000, 1320, 2130, 3210, 0111, 1231, 2021, 3301, 0222, 1102, 2312, 3032, 0333, 1013, 2203, 3123\},
\]
which is an extension of \(C\). On the other hand, the \((3, 1)_4\) MDS codes \(D_0, D_1, D_2,\) and \(D_3\) obtained from \(D\) are precisely the \(C^0, C^1, C^2,\) and \(C^3\) defined above and form a partition of \(C\).

It is known that if a linear code is extendable, it has an extension that is linear [3]. For fixed \(q\) and \(d\) and for sufficiently large \(n\), an extension of a linear MDS code is necessarily equivalent to a linear MDS code [2]. However, very little is known about extending nonlinear codes.

### 2.2. Latin Squares and Hypercubes

A *Latin square* of order \(q\) is a \(q \times q\) array of symbols from an alphabet of size \(q\) such that every row and column contains each symbol exactly once. Two Latin squares of the same order are called *mutually orthogonal* if when they are superimposed, every pair of symbols occurs exactly once. A *Latin hypercube* of order \(q\) and dimension \(k\) is a \(q \times q \times \cdots \times q\) \((k\, \text{times})\) array of symbols from an alphabet of size \(q\).
A set of $r$ mutually orthogonal Latin squares (MOLS) of order $q$ is equivalent to an $(r + 2, 2)_q$ MDS code: index the array coordinates by elements in the alphabet and let $(c_1, c_2, \ldots, c_{r+2})$ be a codeword if and only if $c_{i+2}$ occurs at position $(c_1, c_2)$ in the $i$th latin square. Similarly, a Latin hypercube of dimension $k$ is equivalent to a $(k+1, k)_q$ MDS code: let $(c_1, c_2, \ldots, c_{k+1})$ be a codeword if and only if $c_{k+1}$ occurs at position $(c_1, c_2, \ldots, c_k)$ in the Latin hypercube. Two sets of MOLS or two Latin hypercubes are called paratopic if the corresponding MDS codes are equivalent.

2.3. Tools. We reduce the problem of detecting code equivalence to the graph isomorphism problem and use the software nauty [16] for solving the instances. For an $(n, M, d)_q$ code $C$, we define a colored graph as follows: The graph contains $n$ copies of $K_q$, the complete graph of order $q$, named $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$, colored with color 1. For each $i$, the vertices in $\Gamma_i$ correspond to the elements in $A$. For each codeword $c \in C$, there is an additional vertex, colored with color 2, which is adjacent exactly to the vertex corresponding to the symbol $c_i$ in $\Gamma_i$ for each $i \in \{1, 2, \ldots, n\}$. A graph isomorphism preserving the coloring permutes the copies of $K_q$ and the vertices in each of them separately. Such a mapping corresponds to a permutation of the coordinates and to distinguished permutations of the symbols in each coordinate in the code. Two codes are equivalent if and only if their corresponding graphs are isomorphic. Further, the automorphism group of a code corresponds to the automorphism group of the graph. The software nauty can be used to find the automorphism group of a graph and all isomorphisms between two graphs. For large graphs, we use nauty in the sparse mode with the random Schreier algorithm enabled.

For a finite set $X$ and a family $S$ of subsets of $X$, an exact cover is a subset $S'$ of $S$ that partitions $X$. We use the library libexact [9] for finding all exact covers, given $X$ and $S$.

Finally, we use the software Cliquer [18] to find all cliques of given size in a graph.

3. Generation

Given a set $C$ of equivalence class representatives of $(n, k)_q$ MDS codes, every $(n+1, k)_q$ MDS code is equivalent to a code that is an extension of a code in $C$. This along with Theorem 1 reduces the problem of generating representatives of the equivalence classes of $(n+1, k)_q$ MDS codes to the problem of finding all partitions of all codes in $C$ into $(n, k-1)_q$ MDS codes. For $k = 2$ and $n \geq 3$, this is precisely the method of extending a set of $n-2$ MOLS with a new Latin square by partitioning the Latin squares in the set into common transversals, used for example in [7, 14, 15, 20]; for a discussion of this and similar problems, see for example [19].

Consider a partition of an $(n, k)_q$ MDS code $C$ into $(n, k-1)_q$ MDS codes $C^i$ for $i \in A$,

$$C = \bigcup_{i \in A} C^i.$$ 

We have

$$C_j = \bigcup_{i \in A} C^i_j$$

and

$$C^i = \bigcup_{j \in A} C^i_j.$$
Finding all possible partitions of \( C \) into \((n, k-1)_q\) MDS codes can now be reduced to finding all possible partitions \( \{C'_j\}_{j \in A} \) of \( C_j \) for each \( j \) and combining those partitions in all possible ways. Instead of solving the reduced problem using recursion, our method for finding the partitions of \((n, k)_q\) MDS codes uses the results from finding the partitions of \((n-1, k-1)_q\) MDS codes. In this work, we generate all \( q \)-ary MDS codes with \( k \geq 3 \), \( d \geq 4 \) starting from the MDS codes with \( k = 2 \), their partitions, and the codes with \( d = 3 \).

3.1. Algorithm. As an initial step, we find all partitions of the equivalence class representatives of \((n, 2)_q\) MDS codes into \((n, 1)_q\) MDS codes in the following way. Let \( C_n \) be the set of representatives of \((n, 2)_q\) MDS codes. For a given \( n \), we loop over all \( \hat{C} \in C_{n+1} \) and over all \( n+1 \) punctured codes \( C' \) of \( \hat{C} \). Now, \( \hat{C} \) is an extension of \( C' \) so it induces a partition of \( C' \) into \((n, 1)_q\) MDS codes. We find the equivalence class representative \( \hat{C}' \in C_n \) for which \( \hat{C}' \cong C' \) and find all \( g \in G_n \) for which \( gC' = \hat{C}' \). Considering how the codewords are mapped by \( g \), we get partitions of \( C' \) into \((n, 1)_q\) MDS codes. All partitions of each element in \( C_n \) are obtained in this way, as every such partition corresponds to an \((n+1, 2)_q\) MDS code \( C \) that is equivalent to some \( \hat{C} \in C_{n+1} \).

The algorithm for finding all partitions of an \((n, k)_q\) MDS code \( C \) into \((n, k-1)_q\) MDS codes consists of three parts.

1. For each \( j \in A \), find all \((n-1, k-2)_q\) MDS codes that occur as a part in a partition of \( C_j \) into \((n-1, k-2)_q\) MDS codes. Because \( C_j \cong \hat{C} \), where \( \hat{C} \) is a representative whose partitions into \((n-1, k-2)_q\) MDS codes are known, this can be solved by finding a \( g \in G_n \) for which \( C_j = g\hat{C} \) and applying \( g \) directly to each partition of \( \hat{C} \). Denote the set of \((n-1, k-2)_q\) MDS codes found in this step by \( S_j \).

2. Find all subsets \( D \) of \( C \) that are \((n, k-1)_q\) MDS codes for which \( D_j \in S_j \) for each \( j \). This is done by finding all \( q \)-cliques in the \( q \)-partite graph where, for each \( j \), each element of \( S_j \) corresponds to a vertex in the \( j \)th part, and between two vertices in different parts there is an edge if the minimum distance between them is at least \( n-k+1 \).

3. Find all partitions of \( C \) into \((n, k-1)_q\) MDS codes. This is done by solving an exact cover problem, where \( q \) sets are selected from the family created in the previous phase such that each codeword in \( C \) is covered exactly once.

We run the algorithm for one representative \( C \) from each equivalence class of \((n, k)_q\) MDS codes, and finally perform isomorph rejection for all \((n+1, k)_q\) MDS codes corresponding to the obtained partitions.

3.2. Example. To illustrate the principle of the algorithm, consider the case \( n = 4 \), \( k = 3 \), \( q = 4 \), and the \((4, 3)_4\) MDS code \( C \) whose \( 4^3 \) codewords are listed in Table 1.

In phase 1, \( C \) is split into 4 parts according to the last symbol. These parts are shown as the quadrants in the table. Ignoring the last symbol, each part is an \((3, 2)_4\) MDS code, denoted by \( C_j \), where \( j \) is the symbol ignored. For each \( j \), all partitions of \( C_j \) into \((3, 1)_4\) MDS codes are obtained from previous results, and all parts that occur in such a partition are inserted into the set \( S_j \). An example of an element in \( S_j \) is the set of bolded codewords in the \( j \)th quadrant in the table.

The clique search in phase 2 finds all \((4, 2)_4\) MDS codes that are subsets of \( C \). An example of a found clique is the set of \( 4^2 \) bolded words in the table; each vertex of the clique corresponds to a set of 4 words in one quadrant of the table.
Phase 3 of the algorithm finds all partitions of $C$ into $(4,2)_4$ MDS codes based on the cliques found in phase 2. By Theorem 1, such a partition corresponds to an extension of $C$ to a $(5,3)_4$ MDS code.

3.3. Consistency check. We check the consistency of each phase of the generation by double counting. The total number of $(n+1,k)_q$ MDS codes is the sum of the sizes of the equivalence classes,

$$
\sum_{C \in \mathcal{C}_{n+1}} \frac{|G_{n+1}|}{|\text{Aut}(C)|},
$$

where $\mathcal{C}_{n+1}$ is the obtained set of equivalence class representatives of $(n+1,k)_q$ MDS codes. On the other hand, the same count can be obtained as follows. Let $N(C)$ be the number of partitions of an $(n,k)_q$ MDS code $C$ into $(n,k-1)_q$ MDS codes. The number of $(n+1,k)_q$ MDS codes is

$$
\sum_{C} N(C)! = \sum_{\hat{C} \in \mathcal{C}_n} \frac{|G_n|N(\hat{C})}{|\text{Aut}(\hat{C})|} q!,
$$

where the first sum is taken over all $(n,k)_q$ MDS codes, $\mathcal{C}_n$ is a set containing exactly one representative from each equivalence class of $(n,k)_q$ MDS codes, and $N(\hat{C})$ is the number of partitions of $\hat{C}$ obtained by the algorithm.

4. Results

The algorithm was run for $q = 7$ and $q = 8$, for each $k = 3, 4, \ldots, q - 1$, and for each $n$ starting from $n = k + 2$ and increasing $n$ until no more codes were found. The numbers of inequivalent extendable codes are shown in Tables 2 and 4, for $q = 7, 8$, respectively, and the numbers of equivalence classes are shown in Tables 3 and 5. In the tables, the numbers marked with * are obtained in this work using the method described above. The previously known numbers are obtained from the classification of one-error-correcting MDS codes and sets of MOLS, and in addition it is known that 7-ary MDS codes with $n > 8$ do not exist [11], and the nonexistence of $(11,3)_8$ MDS codes follows from the nonexistence of $(10,2)_8$ MDS codes.

For $q = 8$, 221 calls to Cliquer were made with graph sizes ranging from 64 to 1280. The number of cliques found, which is the number of sets in the exact cover problem in the next step, was 0 in 162 cases and between 8 and 160 in the remaining 59 cases. The largest instance of the exact cover problem was for extending the $(9,7)_8$ MDS code that is linear; there were 80 sets of size 8th over the
Table 2. Number of inequivalent extendable nontrivial \((n,k)_7\) MDS codes

| \(n \setminus k\) | 2   | 3   | 4   | 5   | 6   | 7   |
|-----------------|-----|-----|-----|-----|-----|-----|
| 3               | 6   |     |     |     |     |     |
| 4               | 2   | 1   |     |     |     |     |
| 5               | 1   | *1  | 1   |     |     |     |
| 6               | 1   | *1  | *1  | 1   |     |     |
| 7               | 1   | *1  | *1  | *1  | 1   |     |
| 8               | 0   | 0   | 0   | 0   | 0   | 0   |

Table 3. Number of equivalence classes of nontrivial \((n,k)_7\) MDS codes

| \(n \setminus k\) | 2   | 3   | 4   | 5   | 6   | 7   |
|-----------------|-----|-----|-----|-----|-----|-----|
| 3               | 147 |     |     |     |     |     |
| 4               | 7   |     |     |     |     |     |
| 5               | 1   | 1   |     |     |     |     |
| 6               | 1   | *3  | 1   |     |     |     |
| 7               | 1   | *1  | *1  | 1   |     |     |
| 8               | 1   | *1  | *1  | *1  | 1   |     |
| 9               | 0   | 0   | 0   | 0   | 0   | 0   |

Table 4. Number of inequivalent extendable nontrivial \((n,k)_8\) MDS codes

| \(n \setminus k\) | 2   | 3   | 4   | 5   | 6   | 7   |
|-----------------|-----|-----|-----|-----|-----|-----|
| 3               | 2024|     |     |     |     |     |
| 4               | 38  | 4470|     |     |     |     |
| 5               | 1   | *44 | 36  |     |     |     |
| 6               | 1   | *2  | *1  | 12  |     |     |
| 7               | 1   | *2  | *1  | *1  | 7   |     |
| 8               | 1   | *2  | *1  | *1  | 4   |     |
| 9               | 0   | *2  | *0  | *0  | *0  | *1  |
| 10              |     |     |     | *2  |     |     |

8^7 elements to be covered. The computations lasted 43 seconds for \(q = 7\) and 21 hours for \(q = 8\) using a single core of an Intel Xeon E5-2665 processor, including 18 hours for extending the \((9,7)_8\) MDS codes.

By comparing the equivalence class representatives with known linear codes, we see that all 8-ary MDS codes with \(d \geq 4\), except when \(d = 4\) and \(k = 2, 3\), are equivalent to a linear code.

5. MDS codes with \(d = 2\)

The preceding part of this work completes the classification of all MDS codes with minimum distance at least 3 for alphabet sizes at most 8. In this section, we discuss the remaining case of minimum distance 2. We use a simple construction
Table 5. Number of equivalence classes of nontrivial \((n, k)_q\) MDS codes

| \(n \setminus k\) | 2   | 3   | 4   | 5   | 6   | 7   | 8   |
|-------------------|-----|-----|-----|-----|-----|-----|-----|
| 3                 | 283657 |
| 4                 | 2165  |
| 5                 | 39 12484 |
| 6                 | 1 *39 14 |
| 7                 | 1 *2 *2 8 |
| 8                 | 1 *2 *1 *2 4 |
| 9                 | 1 *2 *1 *1 *2 4 |
| 10                | 0 *1 *0 *0 *1 0 |
| 11                | 0 *0 |

to give a lower bound for the number of equivalence classes of \((n, n-1)_q\) MDS codes, or equivalently the paratopy classes of \((n-1)\)-dimensional Latin hypercubes of order \(q\), for small \(q\).

We denote by \(N_n\) the number of \((n, n-1)_q\) MDS codes and by \(M_n\) the number of equivalence classes of \((n, n-1)_q\) MDS codes. Because each class contains at most \(|G_n| = n!(q!)^n\) elements, we get

\[
M_n \geq \frac{N_n}{n!(q!)^n}.
\]

For small values of \(n\) and \(q\), we find lower bounds for \(N_n\) and \(M_n\) by the following construction. This construction resembles Construction X4 given in [13, Chapter 18.7]. We denote by \(CC'\) the direct sum of codes \(C\) and \(C'\) of lengths \(n\) and \(n'\), respectively,

\[
CD = \{ (c_1, c_2, \ldots, c_n, c'_1, c'_2, \ldots, c'_{n'}) : c \in C, c' \in C' \}.
\]

**Theorem 2.** Let \(C\) and \(C'\) be \((n_1, n_1-1)_q\) and \((n_2, n_2-1)_q\) MDS codes, respectively. Then

\[
C'' = \bigcup_{i \in A} C_i C'_i
\]

is an \((n_1 + n_2 - 2, n_1 + n_2 - 3)_q\) MDS code.

**Proof.** The code \(C''\) contains \(qq^{n_1-2}q^{n_2-2} = q^{n_1+n_2-3}\) codewords of length \(n_1+n_2-2\). Because the minimum distance of each \(C_i\) and \(C'_i\) is 2, the minimum distance of each \(C_i C'_i\) is at least 2. For distinct \(i, j\), any codeword in \(C_i C'_i\) differs from any codeword in \(C_j C'_j\) at least one position in the first \(n_1 - 1\) coordinates and in at least one position in the last \(n_2 - 1\) coordinates. Therefore, \(C''\) has minimum distance 2.

Because the construction in Theorem 2 maps exactly \(q!\) pairs of \((C, C')\) to the same code, this yields a lower bound for the number of \((n, n-1)_q\) MDS codes,

\[
N_n \geq \max_{n'} \frac{N_{n'} N_{n-n'+2}}{q!}.
\]

This lower bound along with the known numbers of \((n, n-1)_q\) MDS codes and (1) allows us to find lower bounds for the number of equivalence classes of \((n, n-1)_q\) MDS codes for small \(n\).
Classification of MDS codes

| n \ q | 4   | 5   | 6   | 7   | 8   |
|-------|-----|-----|-----|-----|-----|
| 3     | 2   | 2   | 12  | 147 | 283657 |
| 4     | 5   | 15  | 264248 | ≥ 4.8 × 10^7 | ≥ 4.6 × 10^{15} |
| 5     | 26  | 86  | ≥ 4.8 × 10^7 | ≥ 2.3 × 10^{13} | ≥ 6.2 × 10^{25} |
| 6     | 4785 | 3102 | ≥ 1.4 × 10^{13} | ≥ 9.5 × 10^{18} | ≥ 6.9 × 10^{35} |
| 7     | ≥ 3.5 × 10^9 | ≥ 1.5 × 10^3 | ≥ 3.1 × 10^{15} | ≥ 3.3 × 10^{24} | ≥ 6.6 × 10^{45} |

Table 6. Number of equivalence classes of \((n, n-1)_q\) MDS codes

Table 6 lists the known numbers of equivalence classes of \((n, n-1)_q\) MDS codes for \(4 \leq q \leq 8\) given in [17] and lower bounds obtained by (1) and (2) for small values of \(n\). Since the lower bounds for the unknown values are rather large, except for a few entries, obtaining further classification results by explicitly constructing representatives from each equivalence class is not feasible in general.

Finally, we note that [21] gives a double exponential lower bound for \(N_n\) when \(q \geq 4\):

\[
\log_2 N_n \geq \begin{cases} 
\left(\frac{q}{2}\right)^{n-1}, & \text{for even } q, \\
\left(\frac{(q-3)(q-1)}{4}\right)^{(n-1)/2}, & \text{for odd } q.
\end{cases}
\]

From (1), it follows that that the number of equivalence classes \(M_n\) for \(q \geq 4\) also has a lower bound that is double exponential in \(n\).

References

[1] T. L. Alderson, (6,3)-MDS codes over an alphabet of size 4, Des. Codes Cryptogr., 38 (2006), 31–40.
[2] T. L. Alderson, A. A. Bruen and R. Silverman, Maximum distance separable codes and arcs in projective spaces, J. Combin. Theory Ser. A, 114 (2007), 1101–1117.
[3] T. L. Alderson and A. Gács, On the maximality of linear codes, Des. Codes Cryptogr., 53 (2009), 59–68.
[4] S. Ball, On sets of vectors of a finite vector space in which every subset of basis size is a basis, J. Eur. Math. Soc., 14 (2012), 733–748.
[5] S. Ball and J. De Beule, On sets of vectors of a finite vector space in which every subset of basis size is a basis II, Des. Codes Cryptogr., 65 (2012), 5–14.
[6] A. Betten, M. Braun, H. Fripertinger, A. Kerber, A. Kohnert and A. Wassermann, Error-Correcting Linear Codes: Classification by Isometry and Applications, Springer, Berlin, 2006.
[7] J. Egan and I. M. Wanless, Enumeration of MOLS of small order, Math. Comp., 85 (2016), 799–824.
[8] P. Kaski and P. R. J. Östergård, Classification Algorithms for Codes and Designs, Springer, Berlin, 2006.
[9] P. Kaski and O. Pottonen, Libexact User’s Guide, version 1.0, Technical Reports TR 2008-1, Helsinki Inst. Inform. Techn., Helsinki, 2008.
[10] G. Kéri, Types of superregular matrices and the number of \(n\)-arcs and complete \(n\)-arcs in \(PG(r, q)\), J. Combin. Des., 14 (2006), 363–390.
[11] J. I. Kokkala, D. S. Krotov and P. R. J. Östergård, On the classification of MDS codes, IEEE Trans. Inf. Theory, 61 (2015), 6485–6492.
[12] J. I. Kokkala and P. R. J. Östergård, Classification of Graeco-Latin cubes, J. Combin. Des., 23 (2015), 509–512.
[13] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes, North-Holland, Amsterdam, 1977.
[14] B. M. Mønch and I. M. Wanless, Atomic Latin squares of order eleven, J. Combin. Des., 12 (2004), 12–34.
[15] B. D. McKay, A. Meynert and W. Myrvold, Small Latin squares, quasigroups, and loops, J. Combin. Des., 15 (2007), 98–119.
[16] B. D. McKay and A. Piperno, Practical graph isomorphism, II, J. Symbolic Comput., 60 (2014), 94–112.
[17] B. D. McKay and I. M. Wanless, A census of small Latin hypercubes, SIAM J. Discrete Math., 22 (2008), 719–736.
[18] S. Niskanen and P. R. J. Östergård, Cliquer User’s guide, Version 1.0, Technical Report T48, Communications Laboratory, Helsinki Univ. Techn., Espoo, 2003.
[19] P. R. J. Östergård, Constructing combinatorial objects via cliques, in Surveys in Combinatorics 2005 (ed. B. S. Webb), Cambridge Univ. Press, Cambridge, 2005, 57–82.
[20] E. T. Parker, Computer investigation of orthogonal Latin squares of order ten, in Proc. Sympos. Appl. Math., Amer. Math. Soc., Providence, 1963, 73–81.
[21] V. N. Potapov and D. S. Krotov, On the number of n-ary quasigroups of finite order, Discrete Math. Appl., 21 (2011), 575–585.
[22] B. Segre, Curve razionali normali e k-archi negli spazi finiti, Ann. Mat. Pura Appl., 39 (1955), 357–379.

Received April 2015; revised June 2016.
E-mail address: janne.kokkala@aalto.fi
E-mail address: patric.ostergard@aalto.fi