Construction of Large Constant Dimension Codes With a Prescribed Minimum Distance

Axel Kohnert and Sascha Kurz

Department of Mathematics
University of Bayreuth
95440 Bayreuth
Germany
axel.kohnert@uni-bayreuth.de
sascha.kurz@uni-bayreuth.de

September 24, 2008

Abstract In this paper we construct constant dimension codes with prescribed minimum distance. There is an increased interest in subspace codes in general since a paper [13] by Kötter and Kschischang where they gave an application in network coding. There is also a connection to the theory of designs over finite fields. We will modify a method of Braun, Kerber and Laue [7] which they used for the construction of designs over finite fields to construct constant dimension codes. Using this approach we found many new constant dimension codes with a larger number of codewords than previously known codes. We finally give a table of the best constant dimension codes we found.

network coding, q-analogue of Steiner systems, subspace codes

1 Introduction

1.1 Subspace Codes

In [13] R. Kötter and F. R. Kschischang developed the theory of subspace codes for applications in network coding. We will recapitulate their definitions in a slightly different manner. We denote by $L(GF(q)^v)$ the lattice of all subspaces of the space of dimension v over the finite field with q elements together with the partial order is given by inclusion. A *subspace code* C is a subset of $L(GF(q)^v)$. If all the subspaces in C are of the same dimension then C is a *constant dimension code*.

The subspace distance between two spaces V and W in $L(GF(q)^v)$ is defined as

$$d_S(V,W) := \dim(V+W) - \dim(V \cap W)$$

which is equal to

$$\dim(V) + \dim(W) - 2\dim(V \cap W).$$

This defines a metric on $L(GF(q)^v)$. The minimum (subspace) distance of a subspace code C is defined as

$$D_S(C) := \min\{d_S(V,W) : V, W \in C \text{ and } V \neq W\}.$$

We define now the optimal (subspace) code problem:

> (P1) For a given lattice $L(GF(q)^v)$ (based on inclusion) fix a minimum (subspace) distance d and find the maximal number m of subspaces $V_1, \ldots, V_m$ in $L(GF(q)^v)$ such that the corresponding subspace code $C = \{V_1, \ldots, V_m\}$ has at least minimum distance d.

The following point of view is useful for the study of subspace codes: We first define the *Hamming graph* with parameters v and q by taking as vertex-set the words of length v over the alphabet $GF(q)$ and connecting two vertices u, w by an edge if the minimum distance between u and w is equal to one. One of the classical problems in coding theory can then be stated as follows:

> (P2) Given the Hamming graph of all words of length v and a minimum distance d find a maximal number m of words such that the pairwise minimum distance is at least d.

If we substitute the Hamming graph by the *Hasse diagram* of $L(GF(q)^v)$ (vertices are the subspaces of $GF(q)^v$ and two subspaces are connected by an edge if they are direct neighbors in the partial order arising from inclusion) the problem (P2) becomes problem (P1). Both problems are special cases of a packing problem in a graph. If we start with problem (P2) and use the 'field with one element' we get problem (P1). Because of this property we say (P2) is the $q-$analogue of (P1). This connection is well known (e.g. [1,17]) and will be useful in the following. Since the publication of the paper by Kötter and Kschischang the constant dimension codes as the $q-$analogue of the constant weight codes were studied in a series of papers [10,12,23].

1.2 $q-$Analogues of Designs

A $t-(v,k,\lambda)$ design is a set C of $k-$element subsets (called blocks) of the set $\{1,\ldots,v\}$ such that each $t-$element subset of $\{1,\ldots,v\}$ appears in exactly λ blocks. The special case of $\lambda = 1$ is called a Steiner system.

The same construction which was used to connect problem (P1) to (P2) in the subsection above can be used to define the $q-$analogue of a t-design. A $t-(v,k,\lambda)$ design over the finite field $GF(q)$ is a multiset C of $k-$dimensional subspaces (called q-blocks) of the v-dimensional vector space $GF(q)^v$ such that each $t-$dimensional subspace of $GF(q)^v$ is a subspace of exactly λ $q-$blocks.

The connection with the constant dimension codes is given by the following observation in the case of a $q-$analogue of a Steiner system: Given a $q-$analogue of a $t-(v,k,1)$ design C we get a constant dimension code of minimum distance $2(k-t+1)$. As each t-dimensional space is contained in exactly one k-dimensional subspace the intersection between two spaces from C is at most $(t-1)-$dimensional. Therefore the minimum distance of C is at least $2(k-t+1)$. On the other hand given any $(t-1)-$dimensional subspace V we can find two $t-$dimensional spaces U, W with intersection V and then two unique $q-$blocks containing U and W. The minimum distance between these $q-$blocks is $2(k-t+1)$.

q-analogues of designs were introduced by Thomas in 1987 [19]. Later they were studied in a paper by Braun et al. [7] where the authors constructed the first non-trivial $q-$analogue of a 3-design. We will use the methods described in their paper to construct constant dimension codes.

In later papers by Thomas [20] and Etzion and Schwartz [17] it was shown that there are severe restrictions on the possible existence of q-analogues of Steiner systems. We will search for a collection of subspaces satisfying only the conditions given by (P1) and not for the stronger condition satisfied by a q-analogue of a Steiner system. But in general the methods described in this paper can also be used for the search for Steiner systems.

2 Construction of Constant Dimension Codes

In this section we describe how to construct a constant dimension code C using a system of Diophantine linear equations and inequalities. Due to the definition of the subspace distance for all $V, W \in C$ we have $d_S(V,W) = 2k - 2dim(V \cap W)$ where k is the dimension of the code. Thus the minimum subspace distance has to be an even number less or equal to $2k$. To construct a constant dimension code of dimension k and minimum subspace distance $2d$ we have to find n subspaces $\{V_1, \ldots, V_n\}$ of dimension k such that there is no subspace of dimension $k-d+1$ contained in two of the selected k-spaces. We define M as the incidence matrix of the incidence system between the $(k-d+1)$-spaces (labeling the rows of M) and the k-spaces (labeling the columns):

$$M_{W,V} := \begin{cases} 1 \text{ if } V \text{ contains } W, \\ 0 \quad \text{otherwise.} \end{cases}$$

Using M we get the description of a constant dimension code as the solution of a Diophantine system. We denote by s the number of columns of M.

Theorem 1.

There is a constant dimension code with m codewords and minimum distance at least $2d$ if and only if there is a $(0/1)-$solution $x = (x_1, \ldots, x_s)^T$ of the following system of one equation and a set of inequalities:

$$\sum_{i=1}^{s} x_i = m \tag{1}$$

$$Mx \leq \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{2}$$

This set of inequalities has to be read as follows: A solution x has the property that the product of x with a single row of M is 0 or 1. Otherwise if the inner product of x with the row labeled by W is larger than one, then the subspace W is contained in more than one subspaces V. To get the constant dimension code corresponding to a solution we have to use the $(0/1)-$vector x as the characteristic vector of a subset of the set of all $k-$dimensional subspaces of $GF(q)^v$. Theorem 1 is a generalization of the Diophantine system describing the search for a $q-$analogue of a Steiner system which was given in [7].

Corollary 1. *[7]*

There is a $q-$analogue of a $(k-d+1)-(v,k,1)$ design with b blocks if and only if there is a $(0/1)-$solution $x = (x_1, \ldots, x_s)^T$ of the following system of Diophantine linear equations:

$$\sum_{i=1}^{s} x_i = b \tag{3}$$

$$Mx = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{4}$$

The size of these problems is given by the number of subspaces in $GF(q)^v$. In general this number is growing too fast. The number of k-dimensional subspaces of $GF(q)^v$ is given by the q-binomial coefficients:

$$\begin{bmatrix} v \\ k \end{bmatrix}_q := \prod_{j=1..k} \frac{(1-q^{v+1-j})}{(1-q^j)}.$$

Already in the smallest case of a $2-$analogue of the Fano plane ($v=7$, $k=3$, $d=2$) the matrix M has 11811 columns and 2667 rows.

3 Constant Dimension Codes with prescribed Automorphisms

To handle also larger cases we apply the following method. We no longer look for an arbitrary constant dimension code. We are now only interested in a set of spaces which have a prescribed group of automorphisms. An automorphism φ of set $C = \{V_1, \ldots, V_m\}$ is an element from $GL(v, GF(q))$ such that $C = \{\varphi(V_1), \ldots, \varphi(V_m)\}$. We denote by G the group of automorphisms of C, which is a subgroup of $GL(v, GF(q))$.

The main advantage of prescribing automorphisms is that the size of the system of equations is much smaller. The number of variables will be the number of orbits of G on the k-spaces. The number of equations or inequalities will be the number of orbits on the $(k - d + 1)$-spaces. The construction process will then have two steps:

- In a first step the solution of a construction problem is described as a solution of a Diophantine system of linear equations.
- In a second step the size of the linear system is reduced by prescribing automorphisms.

This construction method is a general approach that works for many discrete structures as designs [3,14], q-analogs of designs [6,7], arcs in projective geometries [8], linear codes [2,4,5,15] or quantum codes [21].

The general method is as follows: The matrix M is reduced by adding up columns (labeled by the k-spaces) corresponding to the orbits of G. Now because of the relation

$$W \text{ subspace of } V \iff \varphi(W) \text{ subspace of } \varphi(V) \tag{5}$$

for any k-space V and $(k - d)$-space W and any automorphism $\varphi \in G$ the rows corresponding to lines in an orbit under G are equal. Therefore the redundant rows are removed from the system of equations and we get a smaller matrix denoted by M^G. The number of rows of M^G is then the number of orbits of G on the $(k-d+1)$-spaces. The number of columns of M^G is the number of orbits of G on the k-spaces. We denote by $\omega_1, \ldots$ the orbits on the k-spaces and by $\Omega_1, \ldots$ the orbits on the $(k-d+1)$-spaces. For an entry of M^G we have:

$$M^G_{\Omega_i, \omega_j} = |\{V \in \omega_j : W \text{ is a subspace of } V\}|$$

where W is a representative of the orbit Ω_i of $(k - d + 1)$-spaces. Because of property (5) the matrix M is well-defined as the definition of $M^G_{\Omega_i,\omega_j}$ is independent of the representative W. Now we can restate the above theorem in a version with the condensed matrix M^G :

Theorem 2.

Let G be a subgroup of $GL(v, GF(q))$. There is a constant dimension code of length m and minimum distance at least $2d$ whose group of automorphisms contains G as a subgroup if, and only if, there is a $(0/1)-$solution $x = (x_1, \ldots)^T$ of the following system of one equation and a set of inequalities:

$$\sum_i |\omega_i|\, x_i = m \tag{6}$$

$$M^G x \le \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{7}$$

There is one further reduction possible. We are looking for a $(0/1)-$solution where each inner product of a row of M^G and the vector x is less or equal to 1. We can remove columns of M^G with entries greater than 1. This gives a further reduction of the size of M^G. After this last removal of columns we again check on equal rows and also on rows containing only entries equal to zero. We remove these all zero rows and all but one copy of the equal rows.

In order to locate large constant dimension codes with given parameters q, k and $2d$ we try do find feasible solutions $x = (x_1, \ldots)^T$ of the system of equations of Theorem 3 for a suitable chosen group G and a suitable chosen length m. Here we remark that we have the freedom to change equation (6) of Theorem 2 into

$$\sum_i |\omega_i| x_i \ge m.$$

For this final step we use some software. Currently we use a variant of an LLL based solver written by Alfred Wassermann [22] or a program by Johannes Zwanzger [24] which uses some heuristics especially developed for applications in coding theory. The advantage of the LLL based solver is that we definitely know whether there exist feasible solutions or not whenever the program runs long enough to terminate. Unfortunately for the examples of Section 5 this never happens so that practically we could only use this solver as a heuristic to quickly find feasible solutions.

If we change equation (6) into a target function

$$f(x) = f(x_1, \ldots) = \sum_i |\omega_i| x_i$$

we obtain a formulation as a binary linear optimization problem. In this case we can apply the commercial ILOG CPLEX 11.1.0 software for integer linear programs. The big advantage of this approach is that at every time of the solution process we have lower bounds, corresponding to a feasible solution with the largest $f(x)$-value found so far, and upper bounds on $f(x)$.

We can even reformulate our optimization problem in the language of graph theory. Here we consider the variable indices i as vertices of a graph $\mathcal{G}$ each having weight $|\omega_i|$. The edges of $\mathcal{G}$ are implicitely given by inequality (7). Therefore let us denote the ith row of M^G by $M^G_{i,\cdot}$. Now the inequality $M^G_{i,\cdot} \le 1$ translates into the condition that the set

$$\mathcal{C}_i := \left\{ j \,:\, M^G_{i,j} = 1 \right\}$$

is an independence set in $\mathcal{G}$. To construct the graph $\mathcal{G}$ we start with a complete graph and for each row $M^G_{i,\cdot}$ we delete all edges between vertices in $\mathcal{C}_i$. Now an optimal solution of the binary linear program corresponds to a maximum weight clique in $\mathcal{G}$. Again there exist heuristics and exact algorithms to determine maximum weight cliques in graphs. An available software package for this purpose is e.g. CLIQUER [16].

This approach allows to use clique bounds from algebraic graph theory to obtain upper bounds on the target function $f(x)$. In the case where we are able to locate large independent sets in $\mathcal{G}$ which are not subsets of the $\mathcal{C}_i$ we can use them to add further inequalities to (7). If those independent sets are large enough and not too many then a solver for integer linear programs highly benefits from the corresponding additional inequalities.

For theoretical upper bounds and practical reasons how to quickly or exhaustively locate solutions of our system of Theorem 2 it is very useful to have different formulations of our problem to be able to apply different solvers.

3.1 Example

We start with the space $GF(2)^7$. We now describe the construction of a subspace code with 304 codewords and constant dimension equal to 3. This code will have minimum subspace distance 4. The matrix M is the incidence matrix between the $3-$dimensional subspaces of $GF(2)^7$ and the $2-$dimensional subspaces. Without further reductions this matrix has $\begin{bmatrix}7\\3\end{bmatrix}_2 = 11811$ columns and $\begin{bmatrix}7\\2\end{bmatrix}_2 = 2667$ rows. We prescribe now a group G of automorphisms generated by a single element:

$$G := \left\langle \begin{pmatrix} 1 & & & & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \\ & 1 & 1 & & 1 & 1 & \\ & & 1 & 1 & & 1 & 1 \\ & 1 & 1 & 1 & & 1 & 1 \end{pmatrix} \right\rangle .$$

This group G has 567 orbits on the 3-spaces and 129 orbits on the 2-spaces. Using Theorem 3 we can formulate the search for a large constant dimension code as a binary linear maximization problem having 129 constraints and 567 binary variables. After a presolving step, automatically performed by the ILOG CPLEX software, there remain only 477 binary variables and 126 constraints with 3306 nonzero coefficients.

After some minutes the software found a $(0/1)-$solution with 16 variables equal to one. Taking the union of the corresponding 16 orbits on the 3-spaces of $GF(2)^7$ we get a constant dimension code with 304 codewords having minimum distance 4. Previously known was a code with 289 codewords obtained from a construction using rank-metric codes ([18] p.28) and another code consisting of 294 subspaces discovered by A. Vardy (private email communication).

In general it is difficult to construct the condensed matrix M^G for an arbitrary group and larger parameters v and k as the number of subspaces given by the q-binomial coefficient $\begin{bmatrix} v \\ k \end{bmatrix}_q$ grows very fast and it becomes difficult to compute all the orbits necessary for the computation of M^G. In the following section we give a method to get a similar matrix in special cases.

4 Using Singer Cycles

A special case of the above method is the use of a Singer cycle. We use for the reduction a Singer subgroup of $GL(v, GF(q))$ which acts transitively on the one-dimensional subspaces of $GF(q)^v$. Singer cycles have been used in many cases for the construction of interesting geometric objects [9]. We will now describe a method to construct a set C of k-subspaces of $GF(q)^v$ with the following two special properties:

1. C has the Singer subgroup as a subgroup of its group of automorphisms.
2. The dimension of the intersection of two spaces from C is at most one.

Of course such a set C is a constant dimension subspace code of minimum distance $2(k-1)$. This is a special case of the situation of Theorem 3. We now fix one generator $g \in GL(v, GF(q))$ of a Singer subgroup G and a one-dimensional subspace V of $GF(q)^v$. As G acts transitively on the one-dimensional subspaces we can label any one-dimensional subspace W by the unique exponent i between 0 and $l := \begin{bmatrix} v \\ 1 \end{bmatrix}_q - 1$ with the property that $W = g^i V$. Given a k-space U we can describe it by the set of one-dimensional (i.e. numbers between 0 and l) subspaces contained in U. Given such a description of a k-space it is now easy to get all the spaces building the orbit under the Singer subgroup G. Adding one to each number results in the complete orbit by performing it l times.

Example 1. $q = 2, v = 5, k = 2$

A two-dimensional binary subspace contains three one-dimensional subspaces. We get a two-dimensional space by taking the two one-dimensional spaces labeled $\{0, 1\}$ and the third one given by the linear combination of these two will have a certain number, in this example $\{14\}$. Therefore we have a two dimensional space described by the three numbers $\{0, 1, 14\}$. To get the complete orbit under the Singer subgroup we simply have to increase the numbers by one for each multiplication by the generator g of the Singer subgroup. The orbit length of the Singer subgroup is 31 and the orbit is built by the 31 sets: $\{0, 1, 14\}, \{1, 2, 15\}, \ldots, \{16, 17, 30\}, \{0, 17, 18\}, \ldots \{12, 29, 30\}$, $\{0, 13, 30\}$.

To describe the different orbits of the Singer subgroup we build the following set of pairwise distances:

Let s be the number of one-dimensional subspaces in k-space. Let $\{v_1, \ldots, v_s\} \subset \{0, 1, \ldots, l\}$ be the set of s numbers describing a fixed k-space U. Denote by $d_{\{i,j\}}$ the distance between the two numbers v_i and v_j modulo the length of the Singer cycle. $d_{\{i,j\}}$ is a number between 1 and $l/2$. We define the multiset $D_U := \{d_{\{i,j\}} : 1 \leq i < j \leq s\}$. We call D_U the distance distribution of the subspace U. All the spaces in an orbit of a Singer subgroup have the same distance distribution and on the other hand different orbits have different distance distribution. We therefore also say that D_U is the distance distribution of the orbit.

We use these distance distribution to label the different orbits of the Singer subgroup of the k-spaces. The first observation is:

Lemma 1. *A Singer orbit as a subspace code*

An orbit $C = \{V_0, \ldots, V_l\}$ of a Singer subgroup on the k-subspaces of $GF(q)^v$ is a subspace code of minimum distance $2(k-1)$ if and only if the distance distribution of the orbit has no repeated numbers.

Proof. We have to show that the intersection of any pair of spaces in C has at most one one-dimensional space in common. Having no repeated entry in the distance distribution means that a pair of numbers (i.e. pair of one-dimensional subspaces) in a $q-$block b of C can not be built again by shifting the numbers in b using the operation of the Singer subgroup on b.

The same is true if we want to construct a subspace code by combining several orbits of the Singer subgroup. We have to check that the intersection between two spaces is at most one-dimensional. For this we define the matrix S, whose columns are labeled by the orbits Ω_j of the Singer subgroup on the k-dimensional subspaces of $GF(q)^v$ and the rows are labeled by the possible numbers $i \in \{0, \ldots, l/2\}$ in the distance distribution of the k-spaces. Denoting by D_{Ω_j} the distance distribution of the $j-$th orbit, we define an entry of the matrix S by

$$S_{i,\Omega_j} := \begin{cases} 1 \text{ if } i \in D_{\Omega_j} \\ 0 \text{ otherwise.} \end{cases} .$$

Using this matrix S we have the following characterization of constant dimension codes with prescribed automorphisms:

Theorem 3.

There is a constant dimension code C with $n \cdot (l+1)$ codewords and minimum distance at least $2(k-1)$ whose group of automorphisms contains the Singer subgroup as a subgroup if and only if there is a $(0/1)-$solution $x = (x_1, \ldots)^T$ of the following system of one equation and a set of inequalities:

$$\sum_i x_i = n \tag{8}$$

$$Sx \leq \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} . \tag{9}$$

This is the final system of one Diophantine linear equation together with $l/2 + 1$ inequalities which we successfully solved in several cases.

5 Results

As mentioned in the introduction there is an increased interest on constant dimension codes with a large number of codewords for a given minimum subspace distance. There are (very) recent ArXiV-preprints [10,11,18] giving some constructions for those codes.

Here we restrict ourselves on the binary field $q = 2$ and dimension $k = 3$ and minimum subspace distance $d_S = 4$.

Using the approach described in Section 4 it was possible to construct constant dimension codes using the Singer cycle with the following parameters. We denote by n the number of orbits used to build a solution, by d_S we denote the minimum space distance of the corresponding constant dimension code:

v	k	n = number of used orbits	total number of orbits	number of codewords	best known	$d_S = 2d$
6	3	1	19	$1 \cdot 63 = 63$	71[18]	4
7	3	2	93	$2 \cdot 127 = 254$	294	4
8	3	5	381	$5 \cdot 255 = 1275^*$	1164[18]	4
9	3	11	1542	$11 \cdot 511 = 5621^*$	4657[18]	4
10	3	21	6205	$21 \cdot 1023 = 21483^*$	18631[18]	4
11	3	39	24893	$39 \cdot 2047 = 79833^*$	74531[18]	4
12	3	77	99718	$77 \cdot 4095 = 315315^*$	298139[18]	4
13	3	141	399165	$141 \cdot 8191 = 1154931$	1192587[18]	4
14	3	255	1597245	$255 \cdot 16383 = 4177665$	4770411[18]	4

In [11] the authors defined the number $A_q(v, d_S, k)$ as the maximal number of codewords in a constant dimension code of minimum distance d_S. They derived lower and upper bounds. We have implemented the construction method described in [18] to obtain the resulting code sizes which give the lower bounds for $A_q(v, d_S, k)$ for $v \geq 9$. In the above table we marked codes which improved the lower bounds on $A_q(v, d_S, k)$ with an $*$. We would like to remark that for $6 \leq v \leq 8$ our results are optimal for the

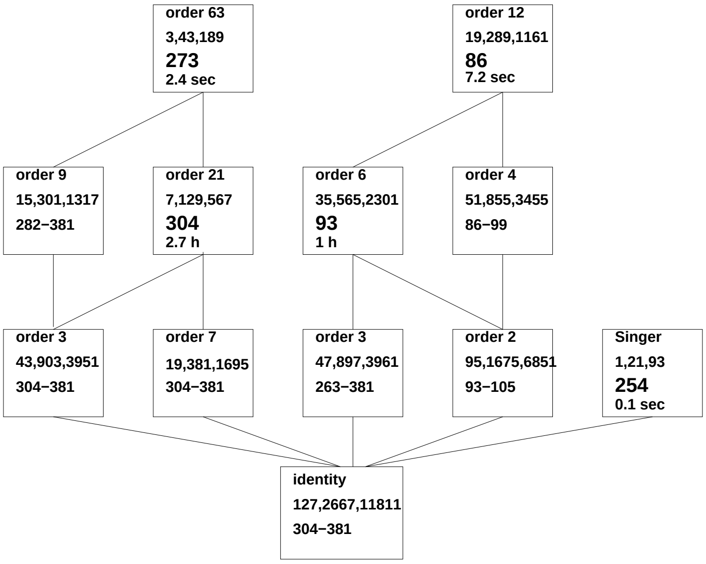
Singer cycle as a subgroup of the group of automorphisms (using the formulation as a binary linear program). So far, for $v = 9$ a code size of $n = 12$ is theoretically possible. (In this case the corresponding binary linear program was not solved to optimality.)

Since for $v = 6, 7$ the method using the Singer cycle was not capable of beating the best known constant dimension code we tried the more general approach described in Section 3. In both cases we improved the cardinality of the best known constant dimension codes as shown in this small table:

v	k	number of codewords	best known	$d_S = 2d$
6	3	77	71[18]	4
7	3	304	294	4

For $v = 6$ even the original incidence matrix M or $M^G$where G is the identity group results in only 1395 binary variables and 651 constraints having 9765 nonzero entries. Using the ILOG CPLEX 11.1.0 solver directly on this problem yields a constant dimension code of cardinality $n = 77$ which beats the example of [10,18] by 6. The best known upper bound in this case is given by 81, where as the upper bound given by the linear relaxation is give by 93. Marcus Grassl (private communication) also found codes of cardinality $n = 77$ using some heuristics together with the CLIQUER software [16].

As mentioned in Example 3.1 the original incidence matrix M is quite large. Here the direct approach has not led to any improvements. Although in general it is difficult to construct the condensed matrix M^G for an arbitrary group and larger parameters we were able to conquer the difficulties for $v = 7, k = 3$, $d_S = 4$ and some groups. The group resulting in the code having 304 three-dimensional subspaces of $GF(2)^7$ such that the intersection of two codewords has dimension at most one was already given in Example 3.1. We have tried several groups before ending up with this specific group. More details can be shown using the following diagram:



This picture shows part of the subgroup lattice of the automorphism group $PGL(7,2)$ of the $L(GF(2)^7)$. It only shows cyclic groups and in the top row we give the order of the group. In the second row we give the number of orbits on the points, lines and planes. In the third row of each entry we give the size lb of a constant dimension6 code and the best found upper bound ub in the format $lb - ub$. As described in Section 3 for a given group our problem corresponds to several versions of feasibility or optimization problems. To obtain the lower bounds we have used the LLL based algorithm, the coding theoretic motivated heuristic and the ILOG CPLEX solver for integer linear programs. The upper bounds were obtained by the CPLEX solver stopping the solution process after a reasonable time. Whenever the lower and the upper bound meet we have written only one number in bold face. In each of these cases we give the necessary computation time to prove optimality in the forth row.

As we can split orbits if we move to a subgroup we can translate a solution found for a group G into a solution for a subgroup of G. E.g. for the groups of order smaller than 21 we did not find codes of size 304 directly. This fact enables us to perform a restricted search in systems corresponding to subgroups by only considering solutions which are in some sense *near* to such a translated solution. We have tried this for the subgroups of the group of order 21 - unfortunately without success.

We would like to remark that solving the linear relaxation can prevent other heuristics from searching for good solutions where no good solutions can exist. E.g. we can calculate in a second that every code in the case of the third group in the third row can contain at most 105 codewords. Since we know better examples we can skip calculations in this group and all groups which do contain this group as a subgroup.

Finally we draw the conclusion that following the approach described in Section 3 it is indeed possible to construct good constant dimension codes for given minimum subspace distance. Prescribing the Singer cycle as a subgroup of the automorphism group has some computational advantages. The resulting codes are quite competitive for $v \geq 8$. The discovered constant dimension codes for $v = 6, 7$ show that it pays off to put some effort in the calculation of the condensed matrix M^G for other groups.

References

1. R. Ahlswede, H. K. Aydinian, and L. H. Khachatrian. On perfect codes and related concepts. *Des. Codes Cryptography*, 22(3):221–237, 2001.
2. A. Betten, M. Braun, H. Fripertinger, A. Kerber, A. Kohnert, and A. Wassermann. *Error-correcting linear codes. Classification by isometry and applications. With CD-ROM.* Algorithms and Computation in Mathematics 18. Berlin: Springer. xxix, 798 p., 2006.
3. A. Betten, A. Kerber, A. Kohnert, R. Laue, and A. Wassermann. The discovery of simple 7-designs with automorphism group PΓL$(2, 32)$. Cohen, Gérard (ed.) et al., Applied algebra, algebraic algorithms and error-correcting codes. 11th international symposium, AAECC-11, Paris, France, July 17-22, 1995. Proceedings. Berlin: Springer-Verlag. Lect. Notes Comput. Sci. 948, 131–145, 1995.
4. M. Braun. Construction of linear codes with large minimum distance. *IEEE Transactions on Information Theory*, 50(8):1687–1691, 2004.

5. M. Braun, A. Kohnert, and A. Wassermann. Optimal linear codes from matrix groups. *IEEE Transactions on Information Theory*, 51(12):4247–4251, 2005.
6. M. Braun. Some new designs over finite fields. *Bayreuther Math. Schr.*, 74:58–68, 2005.
7. M. Braun, A. Kerber, and R. Laue. Systematic construction of q-analogs of t-(v, k, λ)-designs. *Des. Codes Cryptography*, 34(1):55–70, 2005.
8. M. Braun, A. Kohnert, and A. Wassermann. Construction of (n, r)-arcs in $PG(2, q)$. *Innov. Incidence Geom.*, 1:133–141, 2005.
9. K. Drudge. On the orbits of Singer groups and their subgroups. *Electronic Journal Comb.*, 9(1):10p., 2002.
10. T. Etzion and N. Silberstein. Construction of error-correcting codes for random network coding. *submitted (in arXiv 0805.3528)*, 2008.
11. T. Etzion and A. Vardy. Error-Correcting codes in projective space. *ISIT Proceedings*, 5p., 2008.
12. M. Gadouleau and Z. Yan. Constant-rank codes and their connection to constant-dimension codes. *submitted (in arXiv 0803.2262)*, 2008.
13. R. Kötter and F. Kschischang. Coding for errors and erasures in random network coding *IEEE Transactions on Information Theory*, 54(8):3579–359, 2008.
14. E. S. Kramer and D. M. Mesner. t-designs on hypergraphs. *Discrete Math.*, 15:263–296, 1976.
15. T. Maruta, M. Shinohara, and M. Takenaka. Constructing linear codes from some orbits of projectivities. *Discrete Math.*, 308(5-6):832–841, 2008.
16. S. Niskanen and P. R. J. Östergård. Cliquer user's guide, version 1.0. Technical Report T48, Communications Laboratory, Helsinki University of Technology, Espoo, Finland, 2003.
17. M. Schwartz and T. Etzion. Codes and anticodes in the Grassman graph. *J. Comb. Theory, Ser. A*, 97(1):27–42, 2002.
18. N. Silberstein. Coding theory in projective space. *Ph.D. proposal (in arXiv 0805.3528)*, 2008.
19. S. Thomas. Designs over finite fields. *Geom. Dedicata*, 24:237–242, 1987.
20. S. Thomas. Designs and partial geometries over finite fields. *Geom. Dedicata*, 63(3):247–253, 1996.
21. V. D. Tonchev. Quantum codes from caps. *to appear in Discrete Math.*, 2008.
22. A. Wassermann. Lattice point enumeration and applications. *Bayreuther Math. Schr.*, 73:1–114, 2006.
23. S.-T. Xia and F.-W. Fu. Johnson type bounds on constant dimension codes. *submitted (in arXiv 0709.1074)*, 2007.
24. J. Zwanzger. A heuristic algorithm for the construction of good linear codes. *IEEE Transactions on Information Theory*, 54(5):2388 – 2392, 2008.