Some constant weight codes from primitive permutation groups

Derek H. Smith

Division of Mathematics and Statistics
University of Glamorgan
Pontypridd, CF37 1DL, Wales, U.K.

dhsmith@glam.ac.uk

Roberto Montemanni*

Dalle Molle Institute for Artificial Intelligence (IDSIA)
University of Applied Sciences of Southern Switzerland (SUPSI)
Galleria 2, CH-6928 Manno
Switzerland

roberto@idsia.ch

Submitted: Sep 12, 2012; Accepted: Oct 9, 2012; Published: Oct 18, 2012
Mathematics Subject Classifications: 94B60, 20B15

Abstract

In recent years the detailed study of the construction of constant weight codes has been extended from length at most 28 to lengths less than 64. Andries Brouwer maintains web pages with tables of the best known constant weight codes of these lengths. In many cases the codes have more codewords than the best code in the literature, and are not particularly easy to improve. Many of the codes are constructed using a specified permutation group as automorphism group. The groups used include cyclic, quasi-cyclic, affine general linear groups etc. sometimes with fixed points. The precise rationale for the choice of groups is not clear.

In this paper the choice of groups is made systematic by the use of the classification of primitive permutation groups. Together with several improved techniques for finding a maximum clique, this has led to the construction of 39 improved constant weight codes.

Keywords: constant weight codes; primitive permutation groups

*R. Montemanni acknowledges the support of the Hasler Foundation through grant 11158.

1 Introduction

A *constant weight binary code* is a set of binary vectors of length n, weight w and minimum Hamming distance d. The weight of a binary vector (or *codeword*) $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the number of nonzero x_i in the vector. The Hamming distance $d(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x}$ and $\mathbf{y}$ is the number of positions in which they differ. The minimum distance of a code is the minimum Hamming distance between any pair of codewords. The maximum possible number of vectors in a constant weight code is usually referred to as $A(n, d, w)$. These codes have an important role in the theory of error-correcting codes [14]. They have been used in applications such as the design of demultiplexers for nano-scale memories [13] and the construction of frequency hopping lists for use in GSM networks [16].

Accounts of the theory of constant weight codes can be found in [14, 3]. A detailed account of upper bounds for $A(n, d, w)$ can be found in [1]. A variety of methods for obtaining constructive lower bounds for $A(n, d, w)$ can be found in [3], where tables of best known codes are given for $n \leqslant 28$. In [19] this work was extended to parameter sets $29 \leqslant n \leqslant 63$ and $5 \leqslant w \leqslant 8$ with $d = 2w - 2$, $d = 2w - 4$ and $d = 2w - 6$ appropriate to a frequency hopping application. A small number of improvements to the values in [19] can be found in [8, 9, 10, 22] and more improvements were obtained heuristically in [15]. More recently, most of the lower bounds for $A(n, d, w)$ for $29 \leqslant n \leqslant 63$ have been further improved by Brouwer [2], where more references may be found. The web page [2] supersedes an earlier web page previously maintained by Sloane. The aim of this paper is to further improve some of these lower bounds for $A(n, d, w)$ for $29 \leqslant n \leqslant 63$.

One of the main techniques used in [3, 2] involves permutation groups. A code $C(n, d, w)$ is constructed as a union of orbits of a non-trivial permutation group G permuting the symbols $\{1, 2, \ldots, n\}$. The orbit of a binary vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of weight w under G is the set of all distinct vectors $\mathbf{x}^g = (x_{g(1)}, x_{g(2)}, \ldots, x_{g(n)})$, $g \in G$. It is necessary to choose G and orbits so that the minimum distance between any pair of vectors in a single orbit (referred to here as the *internal minimum distance*) is at least d. It is sometimes convenient to choose a single vector as a representative of each orbit. In this work the lexicographically minimal binary vector has been chosen. Let $\mathbf{x}_i$ denote the representative of O_i. In order to ensure that each pair of orbits is compatible a (vertex) weighted graph $\Gamma(n, d, w)$ is used. The set of vertices $\{v_i\}$ corresponds to the set of orbits of G with internal minimum distance at least d. Given two orbits O_i and O_j corresponding to vertices v_i and v_j of $\Gamma(n, d, w)$, define the distance between the orbits as $\delta(O_i, O_j) = \min\ \delta(\mathbf{x}_i^{g_k}, \mathbf{x}_j^{g_\ell})$ where the minimum is taken over all $g_k \in G$ and all $g_\ell \in G$. Then two vertices v_i and v_j of $\Gamma(n, d, w)$ are adjacent if $\delta(O_i, O_j) \geqslant d$. The weight of a vertex is defined as the size of the orbit it represents. A maximum weighted clique of $\Gamma(n, d, w)$ (complete subgraph with the maximum sum of vertex weights) then corresponds to the largest code $C(n, d, w)$ obtainable by this method from the group G. It should be noted that the actual automorphism group of the code may be larger than the group G used to construct the code.

In order to find larger codes $C(n, d, w)$ than those presented in [2] three main enhancements can be attempted:

1. A wider choice of permutation groups can be used.

2. Improved maximum clique algorithms can be applied for cases where maximum clique algorithms do not terminate. Alternatively, more intensive use of existing algorithms can give some improved results.

3. Different heuristic post-processing of the codes can give some larger codes.

When a new best code is found it may also give further new best codes by shortening.

2 Primitive Permutation Groups

The groups used in [2] include cyclic, quasi-cyclic, affine general linear groups etc., with 0, 1, 2 or more fixed symbols. The precise rationale for the selection of groups considered in [2] is unclear. In [20, 21] the database of transitive permutation groups in the computer algebra system Magma[1], based on the classifications in [11, 5], was used to carry out similar tasks for permutation codes with $n \leqslant 18$. While the use of all transitive permutation groups, applied to n, $n-1$ or $n-2$ symbols with 0, 1 or 2 fixed symbols, might be an ideal choice for a systematic approach, it is not feasible for $29 \leqslant n \leqslant 63$. The classifications only extend to $n = 32$ and for larger n there would be far too many groups for the systematic approach presented here to be feasible. A more realistic choice is given by primitive permutation groups.

A permutation group acting on the symbols $\{1, 2, \ldots, n\}$ is *primitive* if it acts transitively on $\{1, 2, \ldots, n\}$ and preserves no nontrivial partition of $\{1, 2, \ldots, n\}$. Primitive permutation groups are known for $n < 4096$ [7], and Magma contains a database for $n < 2500$. It is this database that is used in the current work.

3 Maximum Clique Algorithms

A similar approach to that used in [20, 21] for permutation codes can be adopted. However, for constant weight codes the orbit lengths are not constant and $\Gamma(n, d, w)$ is a vertex weighted graph. In consequence an algorithm suitable for vertex weighted graphs must be used.

Any efficient algorithm can be used if the clique search terminates. If the problem is too large for the algorithm to terminate a variety of heuristic approaches are available. It can be helpful to try several approaches. Algorithms that allow vertex orderings based on vertex degrees are effective for these problems and will be described first.

The software system FASoft used for radio frequency assignment [12] contains a maximum clique algorithm based on that described in [6]. A weighted version of the algorithm is available. The effectiveness of this algorithm can depend strongly on the initial order in which the vertices of the graph are presented. This is true both for the speed of termination of the algorithm, and the quality of the solution available if the algorithm does

[1] http://magma.maths.usyd.edu.au/magma/

not terminate. The order of vertices can be defined for the original degrees of $\Gamma(n, d, w)$ and also for a generalized degree (sum of the weights of adjacent vertices). The vertex orderings used can include the following:

- **Initial ordering**: The algorithm in [6] is applied with the order of vertices as presented by the problem.
- **Initial ordering reversed**: The reverse of the above ordering.
- **Largest degree first (LF1)**: The vertices are sorted in decreasing order of their degrees before the algorithm is applied.
- **LF1 reversed**: The reverse of the above ordering.
- **Largest degree first (LF2)**:] The vertices of largest degree are successively removed from the graph and added to a list. This time the degree calculation excludes vertices that have already been ordered and removed from the graph.
- **LF2 reversed**: The reverse of the above ordering.
- **Smallest degree last (SL)**:] The vertices of smallest degree are successively removed from the graph and added to a list. Again the degree calculation excludes vertices that have already been ordered and removed from the graph. When all vertices have been removed the list is reversed.
- **SL reversed**: The reverse of the above ordering.
- **Largest degree first (LF1) using generalized degree**.
- **LF1 reversed using generalized degree**.
- **Largest degree first (LF2) using generalized degree**.
- **LF2 reversed using generalized degree**.
- **Smallest degree last (SL) using generalized degree**.
- **SL reversed using generalized degree**.
- **Nonincreasing order of vertex weights**.
- **The reverse of the above ordering**

A good approach appears to be to run all available orderings for say 120 seconds. The ordering giving the largest clique can then be run again for as long as is practical. The FASoft maximum clique algorithm was modified to allow this.

Some other algorithms, were also used. The weighted version of the algorithm *Cliquer* [17, 18] did terminate in a small number of cases when FASoft did not, demonstrating optimality. The algorithm of Busygin [4] was also used in some of the most promising cases (new best results and cases needing only small improvement to become new best results). The Busygin algorithm never improved a result obtained with FASoft using the above approach.

In comparison with the results in [20], where different algorithms were best in different cases, the increased number of vertex orderings available for these weighted problems appears to make the FASoft based approach a particularly effective option.

4 Heuristic Post-processing

The *Clique Search (CS)* procedure presented in [15] can be used to heuristically improve a code obtained by any method. If the original code is constructed using a permutation group the new code may have more codewords. The advantage of the method is that the maximum clique problem can be kept to a manageable size for large codes.

Initially the working code is the given starting code. Repeatedly a random subset of the codewords of the working code is removed, leaving a partial code. All the binary vectors of weight w compatible with those already in the code are identified, and a graph is built from these vectors, where the vectors are represented by vertices. Two vertices are adjacent if and only if the Hamming distance between the corresponding vectors is at least d. It is then possible to run a maximum clique algorithm on the graph in order to complete the partial code in the best possible way. Fuller details can be found in [15].

5 Shortening

A shortening procedure can also be used, as is done in [3]. In this way a new best code $C(n, d, w)$ can give rise to a new best code $C(n-1, d, w)$. If *Clique Search* has been used the position where the maximum number n_0 of codewords have a 0 in that position must be identified. Otherwise the first position can be used. These n_0 codewords are selected and the position is removed from all of the selected codewords.

6 The Constant Weight Codes Constructed

For codes of length n all primitive permutation groups of degree n, $n-1$ and $n-2$ were considered. For good codes obtained the *Clique Search* and *Shortening* techniques were applied, and sometimes combinations of these techniques were applied. For $w = 5$ and 6, $d = 2w-2$, $d = 2w-4$ and $d = 2w-6$ a comprehensive set of constructions were carried out for $29 \leqslant n \leqslant 63$ and 38 improved codes were found. For $w = 7$, $d = 2w-2$, $d = 2w-4$ and $d = 2w-6$ the constructions were restricted to $29 \leqslant n \leqslant 48$ to avoid excessive run times, and only one improved code was found. For $w = 8$, only $d = 2w-6$ appeared promising, and the construction was restricted to $29 \leqslant n \leqslant 44$ with no improvements found.

Details of the constructions of the improved codes are given in Table 1. The primitive permutation group used is identified by the degree, number and name used in the Magma database. Also given in the table are the old and new lower bounds, an upper bound, the number of orbits found, the maximum clique algorithm used and, if CS is used, the number of codewords before CS is applied. This is the size of the code obtained with the permutation group alone.

(n,d,w)	Old lower bound	New lower bound	Upper bound	Group from Magma database	Order	Number of orbits	Max. clique alg.	CS Used?	Size before CS
(32,4,5)	6582	6758	6944	(31,7)=31:15	465	16	FASoft		
(42,4,5)	20671	21320	21781	(41,8)=AGL(1,41)	1640	16	FASoft		
(43,4,5)	22728	23478	24647	(43,8)=AGL(1,43)	1806	17	FASoft		
(50,4,5)	42920	43341	45080	(49,24)=AGL(1,49)	2352	23	FASoft	yes	42924
(55,4,5)	63973	64447	68156	(55,4)=M_{11}	7920	16	FASoft	yes	62964
(58,4,5)	79330	79866	83311	*Shorten code below*					
(59,4,5)	85728	87261	91025	(59,4)=AGL(1,59)	3422	32	FASoft		
(63,4,5)	112457	113337	119133	(63,3)=PSU(3,3).2	12096	30	FASoft		
(41,6,5)	930	943	1066	(41,4)=41:5	205	7	FASoft		
(45,6,5)	1172	1175	1386	(43,3)=43:3	129	13	FASoft	yes	1161
(47,6,5)	1293	1363	1607	(47,2)=D($2*47$)	94	17	FASoft		
(48,6,5)	1360	1452	1689	*Shorten code below*					
(49,6,5)	1500	1617	1764	(49,1)=49:4	196	11	FASoft		
(53,6,5)	1946	2067	2341	(53,5)=53:26	1378	2	FASoft		
(32,6,6)	1612	1643	2213	(31,4)=31:5	155	13	Cliquer		
(33,6,6)	1798	1829	2673	(31,4)=31:5	155	15	Cliquer		
(35,6,6)	2146	2170	3249	*Shorten code below*					
(36,6,6)	2427	2604	3864	*Shorten code below*					
(37,6,6)	2702	3108	4261	(37,6)=37:9	333	10	Cliquer		
(38,6,6)	3112	3330	4636	(37,6)=37:9	333	12	Cliquer		
(42,6,6)	4774	5002	7462	(41,6)=41:10	410	18	Cliquer		
(43,6,6)	5516	5719	8005	(43,7)=43:21	903	7	FASoft		
(45,6,6)	6387	6840	9832	(45,1)=M_{10}	720	16	Cliquer	yes	6810
(46,6,6)	7084	7494	10626	(45,1)=M_{10}	720	20	Cliquer		
(52,6,6)	11764	12220	17680	*Shorten code below*					
(53,6,6)	13091	13780	18735	(53,5)=53:26	1378	13	Cliquer		
(52,8,6)	754	791	1057	[a]				yes	777
(53,8,6)	847	888	1095	[b]				yes	887
(54,8,6)	950	995	1224	[c]				yes	980
(55,8,6)	974	1098	1283	(55,1)=PSL(2,11)	660	5	FASoft	yes	1045
(56,8,6)	1000	1126	1334	(55,1)=PSL(2,11)	660	5	FASoft	yes	1045
(57,8,6)	1038	1166	1377	(55,1)=PSL(2,11)	660	5	FASoft	yes	1045
(58,8,6)	1061	1166	1527	*From code above*					
(59,8,6)	1104	1189	1593	*Shorten code below*					
(60,8,6)	1144	1320	1650	*Shorten code below*					
(61,8,6)	1220	1464	1708	(61,10)=61:20	1220	3	FASoft		
(62,8,6)	1237	1464	1891	*From code above*					
(63,8,6)	1338	1483	1953	(61,10)=61:20	1220	3	FASoft	yes	1464
(42,8,7)	1355	1394	2952	(41,3)=41:4	164	15	FASoft		

[a] *Shorten (54,8,6) (980 codewords) in position 47 and shorten the 874 codewords found in position 7. Apply CS to the 777 codewords found.*

[b] *Shorten (54,8,6) (995 codewords) in position 48 and apply CS to the 887 codewords found.*

[c] *Shorten (55,8,6) (1098 codewords) in position 4 and apply CS to the 980 codewords found.*

Table 1: Improved constant weight binary codes. All upper bounds are taken from [2] or [19]. Groups are identified as (i,j) ="name" where i is the degree, j is the number and "name" is the name in the Magma database. If CS gives an improvement the last column gives the number of codewords of the code from the permutation group or shortened code.

7 Conclusion

The computations confirm that the results in [2] are fairly hard to improve using primitive groups, but the 39 improved codes found show that the enhancements outlined at the end of Section 1 have been successful. Codeword files, together with files of orbit representatives and details of the groups used can be found on the authors' web pages at:

http://www.idsia.ch/~roberto/constant_weight_codes_2012.zip

http://data.research.glam.ac.uk/constant_weight_codes

Permutation generators for the groups can be found in these files or in the Magma database.

References

[1] E. Agrell, A. Vardy, and K. Zeger, Upper bounds for constant-weight codes. *IEEE Trans. Information Theory*, 46(7): 2373–2395, 2000.

[2] A. E. Brouwer. Bounds for binary constant weight codes. http://www.win.tue.nl/∼aeb/codes/Andw.html [Accessed 3rd September 2012].

[3] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith. A new table of constant weight codes. *IEEE Trans. Information Theory*, 36(6): 1334–1380, 1990.

[4] S. Busygin. A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics*, 154: 2080–2096, 2006.

[5] J. J. Cannon and D. F. Holt. The transitive permutation groups of degree 32. *Experiment. Math.*, 17: 307–314, 2008.

[6] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9: 375–382, 1990.

[7] H. J. Coutts, M. Quick, and C. M. Roney-Dougal. The primitive permutation groups of degree less than 4096. *Communications in Algebra*, 39(10): 3526–3546, 2011.

[8] I. Gashkov, J. A. O. Ekberg and D. Taub. A geometric approach to finding new lower bounds of $A(n, d, w)$. *Designs, Codes and Cryptography*, 43(2-3): 85–91, 2007.

[9] I. Gashkov and D. Taub. New optimal constant weight codes. *Electronic Journal of Combinatorics*, 14(1): N13, 2007.

[10] G. Ge. Group divisible designs. In *Handbook of Combinatorial Designs - Second Edition* (eds. C. J. Colbourn and J. H. Dinitz) CRC, Boca Raton, Florida, IV.4: 255–260, 2007.

[11] A. Hulpke. Constructing transitive permutation groups. *J. Symbolic Comput.*, 39(1): 1–30, 2005.

[12] S. Hurley, D. H. Smith and S. U. Thiel. FASoft: A system for discrete channel frequency assignment. *Radio Science*, 32(5): 1921–1939, 1997.

[13] P. J. Kuekes, W. Robinett, R. M. Roth, G. Seroussi, G. S. Snider, and R. S. Williams. Resistor-logic demultiplexers for nanoelectronics based on constant-weight codes. *Nanotechnology*, 17: 1052–1061, 2006.

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

[15] R. Montemanni and D. H. Smith. Heuristic algorithms for constructing binary constant weight codes. *IEEE Trans. Information Theory*, 55(10): 4651-4656, 2009.

[16] J. N. J. Moon, L. A. Hughes and D. H. Smith. Assignment of frequency lists in frequency hopping networks. *IEEE Trans. on Vehicular Technology*, 54(3): 1147–1159, 2005.

[17] P. R. J. Östergård. A new algorithm for the maximum-weight clique problem. *Nordic Journal of Computing*, 8(4): 424–436, 2001.

[18] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Math.*, 120: 197–207, 2002.

[19] D. H. Smith, L. A. Hughes and S. Perkins. A new table of constant weight codes of length greater than 28. *The Electronic Journal of Combinatorics*, 13(1): A2, 2006.

[20] D. H. Smith and R. Montemanni. A new table of permutation codes. *Designs, Codes and Cryptography*, 63(2): 241-253, 2012.

[21] D. H. Smith and R. Montemanni. Permutation codes with specified packing radius. *Designs, Codes and Cryptograph*, Online First: doi: 10.1007/s10623-012-9623-4

[22] D. R. Stinson, R. Wei and J. Yin. Packings. In *Handbook of Combinatorial Designs - Second Edition* (eds. C. J. Colbourn and J. H. Dinitz) CRC, Boca Raton, Florida, IV.40: 550–556, 2007.