Östergård, Patric R.J.

The sextuply shortened binary Golay code is optimal

Published in:
Designs Codes and Cryptography

DOI:
10.1007/s10623-018-0532-z

Published: 15/03/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Östergård, P. R. J. (2019). The sextuply shortened binary Golay code is optimal. *Designs Codes and Cryptography*, *87*(2-3), 341–347. https://doi.org/10.1007/s10623-018-0532-z

The Sextuply Shortened Binary Golay Code is Optimal*

Patric R. J. Östergård
Department of Communications and Networking
Aalto University School of Electrical Engineering
P.O. Box 15400, 00076 Aalto, Finland
patric.ostergard@aalto.fi

Abstract

The maximum size of unrestricted binary three-error-correcting codes has been known up to the length of the binary Golay code, with two exceptions. Specifically, denoting the maximum size of an unrestricted binary code of length n and minimum distance d by $A(n, d)$, it has been known that $64 \leq A(18, 8) \leq 68$ and $128 \leq A(19, 8) \leq 131$. In the current computer-aided study, it is shown that $A(18, 8) = 64$ and $A(19, 8) = 128$, so an optimal code is obtained even after shortening the extended binary Golay code six times.

Keywords: classification; clique; double counting; error-correcting code; Golay code
MSC Codes: 94B25; 94B65; 90C27

1 Introduction

Shannon's seminal paper [25], which appeared in 1948, was a starting point for research on error-correcting and error-detecting codes. Quite soon thereafter, Golay [7] and Hamming [8] presented the remarkable codes that are now so central in coding theory and bear the names of their discoverers. In that very early work, the focus was on systematic codes, and $B(n, d)$ was defined in [8] as the maximum size of a systematic binary code with length n and minimum distance d.

*Supported in part by the Academy of Finland, Project #289002.

Somewhat later, Plotkin realized that also nonsystematic codes are relevant and defined $A(n, d)$ to be the maximum size of any binary code with length n and minimum distance d. Plotkin's first study on $A(n, d)$ was published in his M.Sc. thesis [21] in 1952 and as a journal paper eight (!) years later [22], one of the main results being an upper bound on $A(n, d)$ that we now know as the Plotkin bound. The function $A(n, d)$ later became one of the most studied functions in combinatorial coding theory; the reader may consult [13] for the basic theory of this function. All codes considered here are binary, and the term binary will be used only occasionally for emphasis.

As $A(n, d) = A(n+1, d+1)$ for odd d, it suffices to consider either the odd or the even case. If a code is r-error-correcting, then the minimum distance is at least $2r + 1$, so it is natural to consider odd d in that context. On the other hand, in various computational studies of codes, including the current work, the case of even d has certain advantages. When considering even d, we may also assume that the codes be *even*, that is, that all codewords have even weight. Namely, by first deleting one coordinate and then adding an even parity bit, we can always get a desired even code.

A code with length n, cardinality M, and minimum distance at least d is called an (n, M, d) code. An $(n, A(n, d), d)$ code is called *optimal*. For $n \leq 15$, we currently know not only the size of optimal codes, but we know all optimal codes up to symmetry; see [10, Sect. 7.1.4], [12], and [20]. Computational techniques have played a central role in obtaining those results, but we are now approaching the limits for such work. For $n \leq 28$, there is only a handful of open cases for which it is reasonable to think that the current generation of scientists might experience the determination of $A(n, d)$. Two such cases are $64 \leq A(18, 8) \leq 68$ and $128 \leq A(19, 8) \leq 131$, where the lower bounds follow from shortening the extended binary Golay code six and five times, respectively, and the upper bounds have been proved in [19] and [11], respectively. Here, those two cases will be settled computationally. Indeed, it turns out that $A(18, 8) = 64$ and $A(19, 8) = 128$.

The paper is organized as follows. In Section 2 we survey the development of upper bounds on $A(n, 8)$ for $18 \leq n \leq 24$. These upper bounds have gradually been lowered towards the lower bounds $A(n, 8) \geq 2^{n-12}$ for $n \leq 24$, which come from shortening the extended binary Golay code. The main approach is discussed in Section 3, which is concluded with detailed results. By far the most time-consuming part of the computations is that of classifying the even $(17, 33, 8)$ codes. The core result $A(18, 8) = 64$ follows from the fact that the even $(17, M, 8)$ codes with $33 \leq M \leq 36$ cannot be lengthened to codes of size greater than 64. The classification results are validated using double counting.

2 On Subcodes of the Binary Golay Code

The extended binary Golay code [7] shows that $A(24, 8) = 4096$. By *shortening* a binary code—removing one coordinate and taking one of the two subcodes induced by the value in the removed coordinate—it follows that $A(n-1, d) \geq A(n, d)/2$, so $A(n, 8) \geq 2^{n-12}$ for $n \leq 24$. The inverse operation of shortening is called *lengthening* (with some freedom to define how the new codewords are chosen).

Johnson [9] carried out extensive calculations to obtain upper bounds on $A(n, d)$ for specific small values of n and d including those of interest here. Major progress was made by Best *et al.* [2], who used Delsarte's linear programming method [4] to get many new upper bounds and even prove that $A(21, 8) = 512$, from which it follows that $A(n, 8) \leq 2^{n-12}$ for $21 \leq n \leq 24$. (The paper [2] lacks some details, which was a challenge for later studies [1]. Such details were later provided by Brouwer [3], a central inequality for proving $A(21, 8) = 512$ being $A_{16} + 12A_{18} + 21A_{20} \leq 21$.)

In [2] it is further shown that $A(18, 8) \leq 74$, $A(19, 8) \leq 144$, and $A(20, 8) \leq 279$. The first of these bounds was later improved by van Pul [23] to $A(18, 8) \leq 72$, and the other two were improved by Schrijver [24]—using novel techniques based on semidefinite programming—to $A(19, 8) \leq 142$ and $A(20, 8) \leq 274$. After developing the technique based on semidefinite programming even further, Gijswijt, Mittelmann, and Schrijver [6] were able to prove $A(19, 8) \leq 135$ and even optimality of the quadruply shortened extended binary Golay code, $A(20, 8) = 256$. The current author showed [18] that $A(17, 8) = 36$, which implies $A(18, 8) \leq 72$, and later [19] $A(18, 8) \leq 68$. The result $A(19, 8) \leq 131$ by Kim and Toan [11], based on semidefinite programming, completes this brief historical survey of bounds on $A(n, 8)$ for $18 \leq n \leq 24$. (For smaller values of n, $A(17, 8)$ was mentioned above, and $A(n, 8)$ for $n \leq 16$ are easy cases.)

3 Classifying Error-Correcting Codes

3.1 Background and Preliminaries

The current work is a continuation of work carried out by the author in [18] and [19]; see also [12]. The general approach in all these studies is essentially the same and the core task is about classifying even $(17, M, 8)$ codes for gradually smaller values of M.

The concept of symmetry is essential when discussing classification of combinatorial objects. Two binary codes are *equivalent* if one can be ob-

tained from the other by a permutation of the coordinates followed by a transposition of the coordinate values in a subset of the coordinates (the latter operation can also be viewed through addition of a binary vector). Such a mapping from a code onto itself is an *automorphism*. The set of automorphisms of a code C form a group under composition, the *automorphism group* of C, $\mathrm{Aut}(C)$. We denote the group of all possible mappings by G; this group has order $n!2^n$, where n is the length of the codewords. In group-theoretic terms, classification is about finding a transversal of the orbits of codes under the action of G. See [10] for the theory of classifying combinatorial objects.

The even $(17, 37, 8)$ codes were classified in [18]: there are none, so $A(17, 8) \leq 36$, and since $A(17, 8) \geq 36$—attained by a conference matrix code (see [13, Sect. 2.4])—it follows that $A(17, 8) = 36$. In [18] there is a remark that the next step would be to classify the $(17, 36, 8)$ codes, but no application or motivation for such work could then be anticipated. However, later it turned out that the conference matrix code, if it is the unique code attaining $A(17, 8) = 36$, would prove a result on the distribution of coordinate values of optimal codes. Specifically, this would show that there are parameters for which no optimal codes have a balanced coordinate (meaning that the number of 0s and 1s differ by at most one). Indeed, that particular optimal code is unique.

The classification of even $(17, 35, 8)$ codes can be obtained with roughly the same effort as for even $(17, 36, 8)$ codes, so those were also classified in [19]. However, for two reasons the work was not extended to even $(17, 33, 8)$ and $(17, 34, 8)$ codes. First, necessary computational resources for classifying the even $(17, 33, 8)$ and $(17, 34, 8)$ codes were not available to the author at that time. Second, the increasing number of intermediate codes posed a challenge for the approach used in [18] and [19].

3.2 General Approach

The general algorithm used in [18, 19] and here proceeds in a bottom-up fashion: to classify the (n, M, d) codes, the $(n-1, M', d)$ codes with $M' \geq \lceil M/2 \rceil$ are lengthened and equivalent copies amongst the constructed codes are removed (this is called *isomorph rejection*). Let us now briefly consider the two subtasks of lengthening and isomorph rejection.

The task of lengthening error-correcting codes fits perfectly into the framework of finding cliques in graphs. The vertices of such a graph correspond to the candidate words that are at distance at least d from all codewords in the code from which the search starts, and there is an edge in the graph exactly when the mutual distance between the corresponding words is at least d. When lengthening an $(n-1, M', d)$ code to an (n, M, d) code, we

specifically want to find all cliques of size $M - M'$ in this graph. In the current work, we have used the Cliquer software [16], which is based on the algorithm presented in [17].

Whereas the implementation of the lengthening part poses little challenge with dedicated software routines available, the isomorph rejection part is somewhat more involved. The general technique employed here is *canonical augmentation* [14], which has established itself as the standard method for classification [10, Sect. 4.2.3]. A useful tool in this work is the *nauty* graph isomorphism software [15], which can be employed after a proper mapping of codes into graphs [10, p. 89].

The set of $(n-1, M', d)$ codes with $M' \geq \lceil M/2 \rceil$ are called *seeds* when classifying (n, M, d) codes. Notice that an (n, M, d) code C can be obtained from n to $2n$ seeds (some of which may be equivalent): C can be shortened in n coordinates, and when shortening in some coordinate we get two seeds if there are equally many 0s and 1s in that coordinate and one seed otherwise.

In our implementation of canonical augmentation, a code C obtained by lengthening a code C' is subject to two tests, both of which must be passed for C to be accepted:

(i) Identify a canonical $\mathrm{Aut}(C)$-orbit of seeds contained in C, and check whether the seed from which C was extended is in that orbit.

(ii) If C is obtained by extending a seed C', check whether C is the (lexicographic) minimum of its $\mathrm{Aut}(C')$-orbit.

Test (i) is conveniently handled via *nauty*, see [12]. Notice, however, that all shortened subcodes are seeds in [12] but not here. Proper coloring of the graph that is fed to *nauty* ensures that the canonical subcode is indeed a seed. Invariants—which are, for example, based on value distributions in coordinates—can be utilized as a part of determining whether C' is in the canonical orbit of seeds. Invariants speed up the algorithm in two ways: Test (i) can often be handled without calling *nauty* at all, and proper invariants speed up *nauty*. Care must be taken to use the same invariants outside and inside of *nauty*.

In [12, 18, 19], Test (ii) is simply handled by instead comparing canonical labellings (produced by *nauty*) of all codes that pass Test (i) starting from some code C'. However, Test (ii) as stated here is faster and is also immune against the number of codes that pass the tests. As an example, one code in the current work led to more than 10 million accepted lengthened codes.

3.3 Results

The numbers of equivalence classes of even $(n, M, 8)$ codes that have been classified are given in Table 1, including old [18, 19] as well as new results. The new entries are for even $(16, 17, 8)$, $(17, 33, 8)$, and $(17, 34, 8)$ codes and are given in bold.

The even $(16, 17, 8)$ and $(17, 33, 8)$ codes were classified using the approach discussed earlier. Since there are only 459 equivalence classes of even $(17, 33, 8)$ codes, the even $(17, 34, 8)$ codes are most conveniently classified from those, by adding one codeword in all possible ways and carrying out isomorph rejection.

Table 1: Equivalence classes of even $(n, M, 8)$ codes

n	M	#	n	M	#
11	1	1	16	17	**1 554 638 339**
11	2	2	16	18	152 962 983
12	2	3	16	19	24 872 526
12	3	1	16	20	4 904 647
12	4	1	16	21	1 022 642
13	3	2	16	22	223 793
13	4	4	16	23	50 808
14	5	7	16	24	12 708
14	6	11	16	25	3 239
14	7	4	16	26	936
14	8	4	16	27	251
15	9	30 490	16	28	102
15	10	10 688	16	29	30
15	11	886	16	30	15
15	12	139	16	31	5
15	13	25	16	32	5
15	14	14	17	33	**459**
15	15	5	17	34	**44**
15	16	5	17	35	5
			17	36	1

Most of the even $(17, M, 8)$ codes with $33 \leq M \leq 36$ are subsets of codewords of the (unique) even $(17, 36, 8)$ conference matrix code: 418 of the 459 equivalence classes for length 33 and 42 of the 44 equivalence classes for length 34 have this property. The even $(17, 33, 8)$ codes have automorphism

groups of order 96 (1 code), 16 (3 codes), 8 (7 codes), 3 (1 code), 2 (8 codes), and 1 (439 codes). The even $(17, 34, 8)$ codes have automorphism groups of order 272 (1 code), 16 (1 code), 8 (4 codes), 2 (2 codes), and 1 (36 codes).

In the final step, we need to lengthen the even $(17, M, 8)$ codes with $M \geq 33$.

Theorem 3.1. $A(18, 8) = 64$.

Proof. Codes obtained by shortening the extended binary Golay code six times prove that $A(18, 8) \geq 64$.

By [19], we know that an even $(18, M, 8)$ code with $M \geq 65$ cannot be shortened to a $(17, M', 8)$ code with $M' \geq 35$. Hence, if such a code exists, it can be shortened to a $(17, M', 8)$ code with $\lceil 65/2 \rceil = 33 \leq M' \leq 34$. However, when lengthening the classified even $(17, 33, 8)$ and $(17, 34, 8)$ codes, the largest number of codewords that can be added is 7 and 3, respectively. Consequently, $A(18, 8) < 65$, and the theorem follows. □

Corollary 3.1. $A(n, 8) = 2^{n-12}$ *for* $18 \leq n \leq 24$.

The intermediate results were validated by double counting [10, Sect. 10.2]. Consider the step of classifying the even (n, M, d) codes, $\mathcal{C}$, from representatives of even $(n-1, M', d)$ codes with $M' \geq M/2$, $\mathcal{C}'$. Let $N(C)$ denote the total number of codes found when lengthening a code $C \in \mathcal{C}'$, before isomorph rejection. Further, let $S(C) = 1$ if $|C| = M/2$ and $S(C) = 2$ otherwise (in the former case, an (n, M, d) code can be obtained from two subcodes, but in the latter case only from one). Utilizing the orbit–stabilizer theorem, we may now in two ways count the number of *labelled* (n, M, d) codes with only even-weight or only odd-weight codewords:

$$\sum_{C \in \mathcal{C}} \frac{2^n n!}{|\mathrm{Aut}(C)|} = \sum_{C \in \mathcal{C}'} \frac{2^{n-1}(n-1)! N(C) S(C)}{|\mathrm{Aut}(C)|}. \tag{1}$$

When classifying the even $(16, 17, 8)$ codes from the even $(15, M', 8)$ codes with $M' \geq 9$, both sides of (1) give

$$2\,124\,460\,504\,747\,223\,745\,822\,720\,000$$

and when classifying the even $(17, 33, 8)$ codes from the even $(16, M', 8)$ codes with $M' \geq 17$ (by [18] we may also assume that $M' \leq 20$), both sides of (1) give

$$20\,718\,513\,827\,648\,372\,736\,000.$$

The computations were carried out on a compute cluster with 256 cores with 2.4-GHz Intel Xeon E5-2665 processors and 192 cores with 2.5-GHz Intel

Xeon E5-2680v3 processors, with two virtual cores for each physical core. The total amount of compute time for the virtual cores was approximately 37 years, practically all of which was spent on extending the even $(16, 17, 8)$ codes to even $(17, 33, 8)$ codes by clique search.

It is known that the optimal $(23, 2048, 8)$ and $(24, 4096, 8)$ codes are unique [26]; see [5] for a later and simpler proof. A classification of the optimal $(n, 2^{n-12}, 8)$ codes for $18 \leq n \leq 22$ does not seem too far away but would still require at least hundreds of years of core time with the current approach. One way to speed up the search could be to make use of theoretical results, which were crucial for making the classification in [12] possible.

References

[1] E. Agrell, A. Vardy, and K. Zeger, A table of upper bounds for binary codes, *IEEE Trans. Inform. Theory* **47** (2001), 3004–3006.

[2] M. R. Best, A. E. Brouwer, F. J. MacWilliams, A. M. Odlyzko, and N. J. A. Sloane, Bounds for binary codes of length less than 25, *IEEE Trans. Inform. Theory.* **24** (1978), 81–93.

[3] A. E. Brouwer, Disclosure, Available at ⟨`http://www.win.tue.nl/~aeb/codes/lpdetail.html`⟩.

[4] P. Delsarte, Bounds for unrestricted codes, by linear programming, *Philips Res. Rep.* **27** (1972), 272–289.

[5] P. Delsarte and J.-M. Goethals, Unrestricted codes with the Golay parameters are unique, *Discrete Math.* **12** (1975), 211–224.

[6] D. C. Gijswijt, H. D. Mittelmann, and A. Schrijver, Semidefinite code bounds based on quadruple distances, *IEEE Trans. Inform. Theory* **58** (2012), 2697–2705.

[7] M. J. E. Golay, Notes on digital coding, *Proc. IRE* **37** (1949), 657.

[8] R. W. Hamming, Error detecting and error correcting codes, *Bell System Tech. J.* **29** (1950), 147–160.

[9] S. M. Johnson, On upper bounds for unrestricted binary error-correcting codes, *IEEE Trans. Inform. Theory* **17** (1971), 466–478.

[10] P. Kaski and P. R. J. Östergård, *Classification Algorithms for Codes and Designs*, Springer, Berlin, 2006.

[11] H. K. Kim and P. T. Toan, Improved semidefinite programming bound on sizes of codes, *IEEE Trans. Inform. Theory* **59** (2013), 7337–7345.

[12] D. S. Krotov, P. R. J. Östergård, and O. Pottonen, On optimal binary one-error-correcting codes of lengths $2^m - 4$ and $2^m - 3$, *IEEE Trans. Inform. Theory* **57** (2011), 6771–6779.

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

[14] B. D. McKay, Isomorph-free exhaustive generation, *J. Algorithms* **26** (1998), 306–324.

[15] B. D. McKay and A. Piperno, Practical graph isomorphism, II, *J. Symbolic Comput.* **60** (2014), 94–112.

[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, 2003.

[17] P. R. J. Östergård, A fast algorithm for the maximum clique problem, *Discrete Appl. Math.* **120** (2002), 197–207.

[18] P. R. J. Östergård, On the size of optimal three-error-correcting binary codes of length 16, *IEEE Trans. Inform. Theory* **57** (2011), 6824–6826.

[19] P. R. J. Östergård, On optimal binary codes with unbalanced coordinates, *Appl. Algebra Engrg. Comm. Comput.* **24** (2013), 197–200.

[20] P. R. J. Östergård and O. Pottonen, The perfect binary one-error-correcting codes of length 15. I. Classification, *IEEE Trans. Inform. Theory* **55** (2009), 4657–4660.

[21] M. Plotkin, Binary Codes with Specified Minimum Distance, M.Sc. thesis, Moore School of Electrical Engineering, University of Pennsylvania, 1952.

[22] M. Plotkin, Binary codes with specified minimum distance, *IRE Trans. Inform. Theory* **6** (1960), 445–450.

[23] C. L. M. van Pul, On Bounds on Codes, M.Sc. Thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, 1982.

[24] A. Schrijver, New code upper bounds from the Terwilliger algebra and semidefinite programming, *IEEE Trans. Inform. Theory* **51** (2005), 2859–2866.

[25] C. E. Shannon, A mathematical theory of communication, *Bell System Tech. J.* **27** (1948), 379–423, 623–656.

[26] S. L. Snover, The Uniqueness of the Nordstrom–Robinson and the Golay Binary Codes, Ph.D. Thesis, Department of Mathematics, Michigan State University, 1973.