On Higher Weights and Code Existence

Hans Georg Schaathun

University of Surrey
Department of Computing
GU2 7XH Guildford, Surrey
H.Schaathun@surrey.ac.uk

Abstract Several open questions in coding theory relate to non-existence or construction of certain optimal codes. Many previous problems of this kind have been solved by studying possible weight enumerators. A couple of authors in this decade have proposed using higher weights (generalised Hamming weights) to a similar effect. In this paper we suggest one approach based on the weight hierarchy, and it allows us to conduct an extremely rapid computer search to prove that there are exactly two inequivalent $[36, 8, 16]$ codes. The technique can also be used to gain new information about the weight hierarchy of the putative $[72, 36, 16]$ code, but not yet enough to say if it exists or not.

1 Introduction

Higher Weights, that is, parameters describing the support weight of subcodes of dimension higher than one, was a very hot topic during the 1990-s. Helleseth, Kløve, and Mykkeltveit [6] had already in 1977 introduced the support weight distributions. From the support weight distribution for a single binary code, they could determine the weight distribution for an infinite class of non-binary codes.

Victor Wei [13] introduced the weight hierarchy, that is, the sequence of minimal support weights of any r-dimensional subcode, which he used to analyse information-theoretic security on the Wire-Tap Channel of Type II [10].

Later it has been suggested to use higher weights to limit searches for putative optimal codes. Dougherty, Gulliver, and Oura [4] showed that the second support weight distribution of the putative $[72, 36, 16]$ code could be calculated by using the MacWilliams-Kløve-Simonis identities [7,12]. Another work [11] found a way to calculate some of the high-order support weight distributions for this code.

More recently Luo, Mitrpant, Han Vinck, and Chen [8] introduced the concept of relative generalised Hamming weights. Their application was analysis of a two-party Wire-Tap Channel of Type II, and the work has drawn little attention in the subsequent literature. Could it help us solve some of the long-standing problems in coding theory?

In the present paper, we discuss the idea of using the weight hieararchy to constrain code searches. We show that the weight hierarchy of any $[36, 8, 16]$ code can be uniquely determined, and that this gives us enough information to run an exhaustive search in less than 2 minutes.

2 Preliminaries

For any vector $\mathbf{c} = (c_1, \dots, c_n) \in F^n$ (where F is a field), the support of $\mathbf{c}$ is defined as

$$\chi(\mathbf{c}) = \{i : c_i \neq 0\}.$$

The (support) weight of $\mathbf{c}$ is $w(\mathbf{c}) = \#\chi(\mathbf{c})$. For a set $D \subset F^n$, the support is defined as

$$\chi(D) = \cup_{\mathbf{c} \in D} \chi(\mathbf{c}),$$

and the (support) weight, as before, is $w(D) = \#\chi(D)$.

Let C be an $[n, k]$ code over F. The weight hierarchy $(d_1, d_2, \dots, d_k)$ is defined by

$$d_i = \min_{D \leq C, \dim D = i} w(D),$$

i.e. d_i is the minimal weight of an i-dimensional subcode. Clearly, $d_1 = d$ is the regular minimal distance, and $d_0 = 0$ for completeness.

The support weight distribution is the set of parameters $A_i^{(r)}$ for $r = 0, 1, \dots, k$ and $i = 0, 1, \dots, n$, where $A_i^{(r)}$ is the number of r-dimensional subcodes $D \leq C$ of weight $w(D) = i$. Like the traditional weigh enumerator, we can form support weight enumerators

$$W_r(Z) = \sum_{i=0}^{n} A_i^{(r)} Z^i.$$

Obviosuly the weight enumerator $W(Z)$ is $W(Z) = W_1(Z) + 1$.

Two binary codes are said to be equivalent if one can be obtained from the other by a combination of permutations of the columns (coordinate positions).

3 The $[36, 8, 16]$ code

The first binary $[36, 8, 16]$ code was discovered by Helleseth and Ytrehus [2], using a computer search detailed in [15]. It had been a long-standing open question whether the optimal minimal distance for a $[36, 8]$ code would be 15 or 16. An exhaustive search was not feasible, and until now, it has not been known whether the code they found is unique. Our technique gives us the following proposition in few minutes on any personal computer.

Proposition 1. *There are exactly two distinct* $[36, 8, 16]$ *codes up to equivalence.*

3.1 The Weight Hiearchy

Lemma 1 (Ytrehus). *Every binary* $[36, 8, 16]$ *code has weight enumerator*

$$A(Z) = 1 + 153Z^{16} + 72Z^{20} + 30Z^{24}.$$

$$
\begin{aligned}
A^{(2)}_{24} &= 2420 + A^{(2)}_{36} \\
A^{(2)}_{26} &= 3228 - 6A^{(2)}_{36} \\
A^{(2)}_{28} &= 2640 + 15A^{(2)}_{36} \\
A^{(2)}_{30} &= -20A^{(2)}_{36} + 1832 \\
A^{(2)}_{32} &= 615 + 15A^{(2)}_{36} \\
A^{(2)}_{34} &= 60 - 6A^{(2)}_{36} \\
A^{(2)}_{36} &= A^{(2)}_{36}
\end{aligned}
$$

$$
\begin{aligned}
A^{(3)}_{28} &= A^{(3)}_{28} \\
A^{(3)}_{29} &= 48A^{(2)}_{36} + 36240 - 8A^{(3)}_{28} \\
A^{(3)}_{30} &= 28A^{(3)}_{28} - 288A^{(2)}_{36} - 77040 \\
A^{(3)}_{31} &= 720A^{(2)}_{36} + 203184 - 56A^{(3)}_{28} \\
A^{(3)}_{32} &= 70A^{(3)}_{28} - 960A^{(2)}_{36} - 213645 \\
A^{(3)}_{33} &= 720A^{(2)}_{36} + 201840 - 56A^{(3)}_{28} \\
A^{(3)}_{34} &= 28A^{(3)}_{28} - 288A^{(2)}_{36} - 82080 \\
A^{(3)}_{35} &= -8A^{(3)}_{28} + 48A^{(2)}_{36} + 31056 \\
A^{(3)}_{36} &= A^{(3)}_{28} - 2400
\end{aligned}
$$

$$
\begin{aligned}
A^{(4)}_{30} &= 320 + 4A^{(2)}_{36} \\
A^{(4)}_{31} &= 7008 - 24A^{(2)}_{36} \\
A^{(4)}_{32} &= 25815 + 60A^{(2)}_{36} \\
A^{(4)}_{33} &= 41600 - 80A^{(2)}_{36} \\
A^{(4)}_{34} &= 55560 + 60A^{(2)}_{36} \\
A^{(4)}_{35} &= 49056 - 24A^{(2)}_{36} \\
A^{(4)}_{36} &= 21428 + 4A^{(2)}_{36}
\end{aligned}
$$

Table 1. The second through the fourth support weight distribution of a $[36, 8, 16]$ binary code.

	5	6	7	8
32	225	—	—	—
33	6240	—	—	—
34	19620	630	—	—
35	37152	3312	36	—
36	33918	6853	219	1

Table 2. The fifth through the eighth support weight distribution for a $[36, 8, 16]$ binary code are uniquely determined by the MacWilliams-Kløve identities.

The weight enumerator was found in [15], showing clearly that the code is doubly-even and consequently self-orthogonal.

Using the MacWilliams-Kløve-Simonis identities [7,12], and using the generalised Griesmer bound to fix some zero coefficients for small weights, we can get most of the coefficients in the support weight distribution as well, as seen in Tables 1 and 2.

Lemma 2. *Every binary* $[36, 8, 16]$ *code has weight hierarchy*

$$(16, 24, 28, 30, 32, 34, 35, 36).$$

Proof. Note in particular that $A_{24}^{(2)}$ and $A_{30}^{(4)}$ are positive, so $d_2 = 24$ and $d_4 = 30$. It follows from the generalised Griesmer bound that $d_3 = 28$.

The values for d_5, d_6, d_7, and d_8 may be read directly from Table 2.

The chain condition, introduced by Wei and Yang [14], states that there exists a sequence of subcodes

$$\{0\} < D_1 < D_2 < \ldots < D_{k-1} < C$$

such that $w(D_r) = d_r$ for each r, where $<$ denotes a proper subgroup (subspace). Note that it follows that $\dim D_r = r$.

Lemma 3. *Every binary* $[36, 8, 16]$ *code satisfies the chain condition.*

Proof. Consider a subcode D_{30}^4 of dimension 4 and weight 30. Any $[30, 4, 16]$ code is equivalent to the two-fold replication of the $[15, 4, 8]$ simplex code. All the codewords in such a subcode has weight zero or 16, and in particular, every three-dimensional subcode $D^3 < D_{30}^4$ contains seven words of weight 16, and thus $w(D^3) = 28$. Hence there are $15A_{30}^4 \geq 4800$ pairs $(D_{28}^3 < D_{30}^4)$.

Solving the system of inequalities $A_i^{(r)} \geq 0$ for $r = 2, 3$ and all i, we get that $A_{28}^3 \leq 3732$, and consequently there must be two four-dimensional subcodes E_1 and E_2 of weight 30 that intersect in a three-dimensional subcode D_{28}^3 of weight 28. The span $D_{32}^5 = \langle E_1, E_2 \rangle$ must be a five-dimensional subcode of weight 32. Hence we have a chain of subcodes

$$\{0\} < D_{16}^1 < D_{24}^2 < D_{28}^3 < D_{30}^4 < D_{32}^5 < D_{36}^8 = C.$$

By puncturing C on an arbitrary coordinate not in $\chi(D^5_{32})$ we obtain a seven-dimensional subcode D^7_{35} of weight 35, and by puncturing on a second coordinate, also a six-dimensional subcode D^6_{34} of weight 34, completing the chain

$$\{0\} < D^1_{16} < D^2_{24} < D^3_{28} < D^4_{30} < D^5_{32} < D^6_{34} < D^7_{35} < D^8_{36} = C. \qquad (1)$$

Ergo, any $[36, 8, 16]$ binary code satisfies the chain condition.

$$G_{\mathrm{Y}} = \begin{bmatrix} 111111111111111100000000000000000000 \\ 111111110000000011111111000000000000 \\ 111100001111000011110000111100000000 \\ 110011001100110011001100110011000000 \\ 101110111011101110111011101110110000 \\ 000000000001110100011101110011111100 \\ 000100010000001101110111100110010110 \\ 000000000110100101011010111111000011 \end{bmatrix}$$

$$G_{\mathrm{new}} = \begin{bmatrix} 111111111111111100000000000000000000 \\ 111111110000000011111111000000000000 \\ 111100001111000011110000111100000000 \\ 110011001100110011001100110011000000 \\ 101110111011101110111011101110110000 \\ 000000000001110100011101110011111100 \\ 000100010000010111101011010101010110 \\ 000000000100011101110100001100111111 \end{bmatrix}$$

Table 3. Two inequivalent $[36, 8, 16]$ codes, where G_{Y} is equivalent to Ytrehus' code, and G_{new} is new.

3.2 The code search

The $[30, 4, 16]$ subcode is clearly unique, where all non-zero codewords have weight 16. It may be possible to construct the possible $[32, 5, 16]$ codes analytically as well, but there is little point as a computer search can be made in less than a minute.

Suppose we have constructed a $[N, K]$subcode D, and need a $[N + t, K + 1]$ subcode. We construct candidates for Row $K + 1$, as the set

$$S := \{\mathbf{x} \in D^{\perp} : \forall \mathbf{c} \in c, w := w(\mathbf{x} + \mathbf{c}) + t, d \leq w \leq m \wedge w \mod 4 \cong 0\},$$

where d and m are the minimal and maximal weights (16 and 24 for the $[36, 8, 16]$ code). Obviously, the conditions can be modified to allow for singly-even codes, or even codes which are not self-orthogonal.

All possible codes are constructed by appending t zero columns, and all possible rows $\mathbf{x}||(1 \ldots 1)$ for $\mathbf{x} \in S$ to the generator matrix of D.

In order to rule out equivalent codes, we use the nauty library of Brendan McKay [9]. Nauty works on coloured graphs, so we use a standard technique to represent codes as graphs. We need a set S of codewords which is invariant under all automorphisms, and which spans the code. Usually, the set of minimal weight codewords will do, but if this does not span the code, we add codewords of the next higher weight, until we span the code.

Each codeword in S corresponds to a black vertex in the graph. There is a white vertex for each coordinate position, and there is an edge between a black vertex B and a white vertex W, if the codeword corresponding to B is one in the position corresponding to W.

The graphs are represented by incidence matrices, and every graph corresponding to a linear code C has an incidence matrix of the form

$$I = \begin{bmatrix} 0 & M^{\mathrm{T}} \\ M & 0 \end{bmatrix},$$

where the rows of M are the necessary low weight codewords of C.

From an incidence matrix I, nauty can produce a canonical incidence matrix which is common for all isomorphic graphs (equivalent codes). We can form a canonical generator matrix for the corresponding code by Gaussian elimination on the matrix M. Since this algorithm is deterministic, the same canonical graph will always give the same canonical generator matrix.

When we want to reject equivalent codes, we keep a hash table using the canonical generator matrix as a key. For each code we generate, we try to insert it in the hash table. If it is already there, we have already searched this code and proceed immediately to the next one. If it is successfully inserted, we continue by searching this code.

Using this search algorithm, we find two inequivalent $[32, 5, 16]$ codes, in half a minute on a mid-range laptop. Growing the code from $[32, 5]$ to $[36, 8]$ it is advantageous to use the same candidate set S for all the three rows required. Because we know that the $d_7 = 35$, we know that the $[36, 8, 16]$ code is equivalent to one with generator matrix on the form

$$G = \begin{bmatrix} G' & 0 \\ \mathbf{s}_1 & 1100 \\ \mathbf{s}_1 & 1010 \\ \mathbf{s}_1 & 1001 \end{bmatrix}, \tag{2}$$

where G' is a generator matrix of a $[32, 5, 16]$ code. This search takes 70-80 seconds, yielding two inequivalent codes as shown in Table 3.

4 Partial results on the $[72, 36, 16]$ code

Inspired by our success with the relatively small $[36, 8]$ code, we give some preliminary results for the $[72, 36, 16]$ code. In this section, we let C denote an arbitrary $[72, 36, 16]$ Type II code. We know from [3] that $d_1 = 16$ and $d_2 = 24$.

4.1 Further preliminaries

We need the well-known Johnson bounds for some of the proofs. Let $A(n,d,w)$ denote the maximum size of a (non-linear) code with constant weight w and minimum distance d.

Lemma 4 (Johnson bounds). *We have*

$$\begin{aligned} A(n,2w,w) &= \lfloor \frac{n}{w} \rfloor, \\ A(n,d,w) &\leq \lfloor \frac{n}{w} A(n-1,d,w-1) \rfloor, \\ A(n,d,w) &\leq \lfloor \frac{n}{n-w} A(n-1,d,w) \rfloor. \end{aligned}$$

Forney [5] discussed a series of duality results for higher weights, some of which could be traced back to Wei [13]. We summarise a few key points which we will use. Let $I = \{1,\ldots,n\}$ be the co-ordinate index set. For any $J \subset I$, let C_J denote code C shortened on $I \backslash J$, i.e.

$$C_J = \{\mathbf{c} \in C : \forall i \notin J, c_i = 0\}.$$

It is known that if $\dim C_J = r$, then $\dim(C^\perp)_{I\backslash J} = r + n - k - \#J$. Clearly, for each r, there is $J \subset I$ such that $w(C_J) = d_r$. Then, $w((C^\perp)_{I\backslash J}) = d^\perp_{r+n-k-d_r}$.

Let $P_J(C)$ be the code punctured on $I\backslash J$, i.e. the code

$$P_J(C) = \{(c_1,\ldots,c_n) : \exists (c_1',\ldots,c_n') \in C, \forall i \in I\backslash J, c_i = c_i'; \forall i \in J, c_i = 0\}.$$

Clearly $\dim P_J(C) + \dim C_J = \dim C$.

We define the past subcode $P_i = C_{1,\ldots,i}$ and the future subcode $F_i = C_{i+1,\ldots,n}$.

4.2 The third, fourth, and fifth weight

Lemma 5. *Any* $[72,36,16]$ *code has* $d_3 = 28$ *or* $d_3 = 29$*.*

Proof. We get $d_3 \geq 28$ from the Griesmer bound. Consider a shortened code C_J of weight $w(C_J) = 24$ and dimension 2. Then $P_{I\backslash J}(C)$ would be a $[48,34]$ code, which has minimum distance 6 or less by Brouwer's tables [1]. Hence $d_3 \leq 30$.

Suppose for a contradiction that $d_3 = 30$.

Assume a coordinate ordering such that P_{16} has dimension one and P_{24} dimension two. Now F_{16} is a doubly-even $[56,21,16]$ code containing the all-one word, and thus $F_{16}^\perp$ is a $[56,35,8]$ even code. Solving the MacWilliams identities, we find that $F_{16}^\perp$ has 1155 words of weight 8.

Since C has $d_3 = 30$, $F_{16}^\perp$ has $d_2 \geq 14$, and thus two words of weight 8 must have distance at least 12. This results in a $(56,1155,12)$ constant weight code with $w = 8$. However, this is impossible because

$$\begin{aligned} A(56,12,8) &\leq \lfloor \frac{56}{8} A(55,12,7) \rfloor \leq \lfloor \frac{56}{8} \lfloor \frac{55}{7} A(54,12,6) \rfloor \rfloor \\ &\leq \lfloor \frac{56}{8} \lfloor \frac{55}{7} \lfloor \frac{54}{6} \rfloor \rfloor \rfloor = 490, \end{aligned}$$

by Lemma 4.

Lemma 6. *If C has $d_3 = 28$, then* $30 \leq d_4 \leq 32$.

Proof. We have $d_4 \geq 30$ by the Griesmer bound, and $d_4 \leq 33$ because $d(44, 33) \leq 5$. Suppose $d_4 = 33$ for a contradiction. Assume a coordinate ordering such that P_{28} has dimension 3. Now $F_{28}^{\perp}$ is a $[44, 33, 5]$ code containing the all-one word, and F_{28} is doubly-even without the all-one word. The MacWilliams identities for this code pair has no integer solutions, so the codes cannot exist.

Lemma 7. *If C has $d_3 = 29$, then* $32 \leq d_4 \leq 33$.

Proof. The lower bound follows from Griesmer and the upper bound follows from the fact that $d(43, 33) = 4$.

Lemma 8. *We have* $d_5 < 37$.

Proof. We know that $d_4 \leq 33$, so $d_5 \leq 37$ follows from Brouwer's tables. Suppose for a contradiction that $d_5 = 37$. Then $d_6 \geq 39$ by Griesmer, and $d_6 \leq 39$ by Brouwer's tables. Thus we get top-down greedy weights $\tilde{e}_4 = 35$ and $\tilde{e}_3 = 33$. We know that $\tilde{e}_2$ is even, so it must be 30 or 32, but then there is no possible choice for $\tilde{e}_1$ and we therefore conclude that $d_5 < 37$.

5 Conclusion

We have presented a novel approach to constraining code searches. This approach proved very effective in the case of $[36, 8, 16]$ where an exhaustive search can be done in less than two CPU-minutes, and show exactly two distinct codes up to equivalence.

Hopefully, this can inspire renewed interest in some of the legendary problems of coding theory, and combining the present techniques with others, one might just see some solutions in the foreseeable future.

References

1. A. E. Brouwer. Bounds on the minimum distance of linear codes, 2002. http://www.win.tue.nl/ aeb/voorlincod.html.
2. S. M. Dodunekov, T. Helleseth, N. Manev, and Ø. Ytrehus. New bounds on binary linear codes of dimension eight. *IEEE Trans. Inform. Theory*, 33(6):917–919, 1987.
3. Steven Dougherty and Aaron Gulliver. Higher weights of self-dual codes. In Daniel Augot, editor, *Workshop on Coding and Cryptography*, pages 177–188, January 2001.
4. Steven Dougherty, Aaron Gulliver, and Manabu Oura. Higher weights and graded rings for binary self-dual codes. *Discrete Applied Mathematics*, 128:251–261, 2003. Special issue for WCC 2001.
5. G. David Forney, Jr. Dimension/length profiles and trellis complexity of linear block codes. *IEEE Trans. Inform. Theory*, 40(6):1741–1752, 1994.
6. Tor Helleseth, Torleiv Kløve, and Johannes Mykkeltveit. The weight distribution of irreducible cyclic codes with block lengths $n_1((q^l - 1)/n)$. *Discrete Math.*, 18:179–211, 1977.

7. Torleiv Kløve. Support weight distribution of linear codes. *Discrete Math.*, 106/107:311–316, 1992.
8. Y. Luo, C. Mitrpant, A.J.H. Vinck, and K. Chen. Some new characters on the wire-tap channel of type II. *Information Theory, IEEE Transactions on*, 51(3):1222–1229, March 2005.
9. Brendan D. McKay. The nauty page, 2002. http://cs.anu.edu.au/people/bdm/nauty/.
10. L. H. Ozarow and A. D. Wyner. Wire-tap channel II. *AT&T Bell Laboratories Technical Journal*, 63(10):2135–2157, December 1984.
11. H.G. Schaathun. Duality and support weight distributions. *Information Theory, IEEE Transactions on*, 50(5):862–867, May 2004.
12. Juriaan Simonis. The effective length of subcodes. *Appl. Algebra Engrg. Comm. Comput.*, 5(6):371–377, 1994.
13. Victor K. Wei. Generalized Hamming weights for linear codes. *IEEE Trans. Inform. Theory*, 37(5):1412–1418, 1991.
14. Victor K. Wei and Kyeongcheol Yang. On the generalized Hamming weights of product codes. *IEEE Trans. Inform. Theory*, 39(5):1709–1713, 1993.
15. Øyvind Ytrehus. Code-buster: A software tool for characterizing abstract codes. Technical report, Dept. of Informatics, University of Bergen, March 1987.