Three Approaches to Self-Dual Code Construction

Megan Armentrout[*], Meagan Pitluck[†]
Ranjan Rohatgi [‡], Joseph Thomas[§]
Selin Kalaycioglu [¶], Dr. Klaus Lux[‖]

2008 VIGRE Arizona Summer Program

July 24, 2008

[*]Whitworth University
[†]University of Notre Dame
[‡]Northwestern University
[§]University of Arizona
[¶]University of Arizona
[‖]University of Arizona

Contents

Abstract

In this paper, we investigate experimental, systematic, and probabalistic approaches to the construction of self-dual codes. After a brief overview of pertinent theory, we first examine Philippe Gaborit and Ayoub Otmani's experimental method for finding "good" self-dual codes over finite fields, particularly over $GF(2)$ and $GF(3)$. We find that their method effectively obtains "good" self-dual codes of length $2n$ and consider the theoretical reasons for the procedure's success. Specifically, we consider the action of orthogonal matrix groups on self-dual codes described by Gerald J. Janusz in "Parametrization of self-dual codes by orthogonal matrices." Based on this theory, we construct and analyze two group theoretical approaches to computing "good" self-dual codes.

1 Introduction

Self-dual codes satisfy the relation $C^{\perp} = \left\{d \in GF(q)^{(2n)} | c \cdot d = 0, \forall c \in C\right\} = C$. In finding new self-dual codes, "good" self-dual codes are those that achieve, or come close to achieving, the highest known minimum distance for codes of the same length. The minimum distance of a code is the smallest distance between two different codewords in the code where the distance between two codewords is defined to be the number of positions where the codewords differ. High minimum distances are desirable as they allow the code to detect and correct a larger number of errors.

A code of length l is represented by a $k \times l$ generating matrix, G, where the rows of G form a k-dimensional basis for the code. One strategy for creating "good" self-dual codes is to begin with a simple self-dual code, and then modify it to construct a new self-dual code with better properties. In this paper, we consider three algorithms of this type. Each utilizes, implicitly or explicitly, the properties of the group of $2n \times 2n$ orthogonal matrices, O_{2n}. Accordingly, we present theory pertinent to this group.

2 Underlying Theory: Orthogonal Matrices

Definition 1. *Let M be an $m \times m$ matrix with entries in a field F. We say M is an **orthogonal** matrix if $MM^T = I_m$. We observe that the set O_m of all $m \times m$ orthogonal matrices over F is a group under matrix multiplication. It is clear that I_m is the group identity element and $M^T = M^{-1}$. It is also evident that if $M, N \in O_m$, $MN \in O_m$. ($MN(MN)^T = MNN^TM^T = MI_mM^T = MM^T = I_m$.)*

Given a self-dual code C of length $2n$, we can use the elements of O_{2n} to find other self-dual codes. This is possible due to the following lemma:

Lemma 1. *[3] Let G be a generating matrix of a self-dual code of length $2n$ over a field F, and let M be $2n \times 2n$ matrix satisfying $MM^T = \lambda I$, where λ is an invertible element of F. Then GM generates a self-dual code.*

Proof. Recall that any code $C \subset F^{(2n)}$ has the property that $\dim(C) + \dim(C^{\perp}) = 2n$. If C is a self-dual code, then $C = C^{\perp}$, and hence, $\dim(C) = n$. Because M is invertible, the dimension of C is preserved under M. Thus, if C' is the code generated by GM, $\dim(C') = n$, and hence $\dim(C') = \dim(C'^{\perp})$. As G generates a self-dual code, $GG^T = 0$, it follows $GM(GM)^T = G(MM^T)G^T = \lambda GG^T = 0$. Since the (i, j)th entry in $GM(GM)^T$ is the dot product of rows i and j of GM, the fact $GM(GM)^T = 0$ demonstrates that every pair of rows in GM are orthogonal, hence $C' \subset (C')^{\perp}$. Our earlier observation about $\dim(C')$ ensures $C' = C'^{\perp}$; thus, C' is self-dual.

The above proof ensures we can use the elements of O_{2n} to find self-dual codes. Since we are particularly interested in the case where $F = GF(2)$ the following definition and theorem allow us to restrict our attention solely to O_{2n}.

Definition 2. *Given C, a self-dual code over $GF(2)^{(2n)}$, a basis $x_1, \ldots, x_n, y_1, \ldots, y_n = \{x_i, y_j\}$ of $GF(2)^{(2n)}$ is called a **C-orthogonal basis** if the following holds:*

- *$x_1, \ldots, x_n$ is a basis of C*
- *$x_i \cdot y_j = 1$ if $i = j$, 0 otherwise*
- *$y_i \cdot y_j = 1$ if $i = j$, 0 otherwise*

Theorem 1. *[5] Let C and C' be self-dual codes of length $2n$ over $GF(2)$. Then there is an orthogonal matrix B with $CB = C'$.*

Proof. Let the V-orthogonal basis be given by $\{v_i, e_i\}$ with $v_i = e_i + e_{n+i}$, such that the v_i form a basis for the self-dual code V. V is of the form $[I_n|I_n]$. Let C be another self-dual code with $\{u_i, w_i\}$ a C-orthogonal basis. Consider a matrix A that satisfies: $v_iA = u_i$ and $e_iA = w_i$. Then we know $u_i = v_iA = [e_i + e_{n+i}]A = e_iA + e_{n+i}A = w_i + e_{n+i}A$ so $e_{n+i}A = u_i - w_i$. It follows A is a $2n$ x $2n$ matrix of the form:

$$A = \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ u_1 - w_1 \\ \vdots \\ u_n - w_n \end{pmatrix}$$

Since v_i and e_i are vectors of length $2n$, u_i and w_i must be vectors of length $2n$. Applying the definition of a C-orthogonal basis, $AA^T = I_{2n}$. Therefore, we can find orthogonal matrices A and B such that $VA = C$ and $VB = C'$. It follows $C(A^{-1}B) = C'$ and $A^{-1}B \in O_{2n}$.

The theorem in conjunction with Lemma 1 implies that O_{2n} acts transitively on self-dual codes of length $2n$.

Definition 3. *For a code C of length $2n$, let H be the **orthogonal stabilizer** of C under O_{2n}. Then, $H(C) = St(C) \cap O_{2n}$ where*
$St(C) = \{A \in GL(2n, GF(2)) | CA = C\}$.

We can utilize following result then to calculate the number of inequivalent (up to vector space equality) self-dual codes in $F^{(2n)}$.

Theorem 2. *[5] For $n \in \mathbb{N}$, $[O_{2n} : H] = \prod_{i=1}^{n-1}(2^i - 1)$.*

Clearly, the number of cosets of H in O_{2n} increases very quickly. We can reduce the problem further by observing that the properties we are interested in (minimum distance, weight, etc.), are independent of the ordering of the entries in the code words. Consequently, we say two self-dual codes C and C' are *isomorphic* if there exists a generator matrix G for C and a permutation matrix Π such that $G\Pi$ is a generator matrix for C'.

The next result allows us to construct O_m. First, however, we must introduce the concept of a transvection and its properties as well as two important properties of the orthogonal group.

Definition 4. *Let $u \in GF(2)^n$ be a vector with even weight. The **transvection** determined by u is the transformation $T_u : x \to x + (x \cdot u)u$ for all $x \in GF(2)^n$. In terms of matrices, $T_u = I_n + u^T u$.*

Lemma 2. *Let u be a vector of even weight. If $A \in O_m$, then $A^{-1}T_u A = T_{uA}$.*

Proof. Let v be a vector.
$$\begin{aligned} v(A^{-1}T_u A) &= (vA^{-1})T_u A \\ &= [vA^{-1} + ((vA^{-1}) \cdot u)u]A \\ &= v + (vA^{-1} \cdot u)uA \\ &= v + (vA^{-1}A \cdot uA)uA \\ &= v + (v \cdot uA)uA \\ &= (v)T_{uA}. \end{aligned}$$
* Note: $(u \cdot v) = uv^T = uAA^T v^T = (uA)(vA)^T = (uA \cdot vA)$ (i.e. orthogonal matrices preserve the dot product.

Lemma 3. *Let u be a vector of even weight. Then $T_u^2 = I$.*

Proof. $T_u^2 = T(T_u) = [x + (x \cdot u)u] + ([x + (x \cdot u)u] \cdot u)u$.
There are 2 cases:

- For $(x \cdot u) = 0$, $T(T_u) = (x+0u)+[(x+0u)\cdot u]u = x+(x\cdot u)u = x+0u = x$.
- For $(x \cdot u) = 1$, $T(T_u) = (x + 1u) + [(x + 1u) \cdot u]u$
  $= (x + u) + [(x \cdot u) + (u \cdot u)]u$
  $= (x + u) + (1 + 0)u$
  $= x + 2u = x$.

Lemma 4. *All row and column vectors of $A \in O_m$ have odd weight.*

Proof. Assume the ith row of A, a_i, has even weight. Then AA^T has a zero in the iith entry, so $AA^T \neq I_m$, which means $A \notin O_m$. A similar argument shows that the columns of A cannot have even weight.

Theorem 3. *[6] For $m = 2k > 0$, $|O_m| = 2^{k^2} \prod_{i=1}^{k-1} (2^{2^i} - 1)$.*
For $m = 2k + 1 > 0$, $|O_m| = 2^{k^2} \prod_{i=1}^{k} (2^{2^i} - 1)$.

We are now ready to state and prove a theorem which gives us the tools to construct O_m. Note that the actual theorem as written by Janusz is more extensive but we state and prove only those parts pertinent to this paper.

Theorem 4. *[5][Generators of O_m] Let K_u denote the subgroup $\langle P_m, T_u \rangle$ of O_m generated by all $m \times m$ permutation matrices and the transvection T_u for some vector u of even weight. It follows:*
(a) If $1 \leq m \leq 3$, then $K_u = O_m = P_m$.
(b) If $wt(u) = m$ and $m \geq 4$ then $K_u \cong P_m \times \langle T_u \rangle$ has order $2 \cdot m!$.
(c) If $wt(u) = 4$ and $m > 4$, then $K_u = O_m$.

Proof of (a) This result follows from the facts $P_m \subseteq K_u \leq O_m$ and $|O_m| = |P_m|$ for $1 \leq m \leq 3$.
Proof of (b) As $wt(u) = m$, u must be of the form $[1, ..., 1] = 1^m$. Therefore $PT_u = T_u P$ for all $P \in P_m$. Since $T_u \notin P_m$, $K_u \cong P_m \times \langle T_u \rangle$, which has order $2 \cdot m!$.
Proof of (c) We first consider the case where $wt(u) = 4$ and $m > 4$. We know $K_u \subseteq O_m$ and show now that $O_m \subseteq K_u$. By Lemma 1, we know $P^{-1} T_v P = T_{vP}$ for $P \in P_m$. This fact, combined with the transitivity of P_m on vectors of a given weight, implies that if $T_v \in K_u$, with $wt(v) = s$, K_u must contain T_w for all vectors w of weight s.

WLOG, we choose our vector u with $wt(u) = 4$ to be of the form $u = e_1 + e_2 + e_3 + e_4$. Let $e_1 K_u = \{e_1 g | \ g \in K_u\}$ be the orbit of e_1 in K_u. We now show by induction that the orbit contains a vector of weight s for all odd $s < m$, proving that K_u must contain all vectors of odd weight $s < m$. If we take $g = I_m \in K_u$, it is evident that the orbit $e_1 K_u$ contains e_1, a vector of length 1. The orbit also must also contain a vector of weight 3, namely $e_1 T_u = e_1 + u = e_2 + e_3 + e_4$.

Assume that $e_1 K_u$ contains a vector, v such that $wt(v) = 2t - 1 < m - 2$. We show that $e_1 K_u$ must then contain a vector of weight $2t + 1 < m$. We can choose $v = e_4 + e_5 + \cdots + e_{2t+2}$, so that $wt(v) = 2t - 1$. As u and v share one non-zero coordinate (from e_4), $u \cdot v = 1$, and

$$vT_u = v + u = e_1 + e_2 + e_3 + 0e_4 + e_5 + \cdots + e_{2t+2}.$$

Then vT_u is a vector of weight $2t + 1$. By induction, $e_1 K_u$ contains a vector of weight s for all odd $s < m$. Therefore K_u contains T_w for all vectors w of odd weight s.

Now we prove $O_m \subseteq K_u$. Let $A \in O_m$. The first row of A, e_1A, must be a vector of odd weight $s < m$. Since K_u contains T_w for all vectors w of weight s, there must be a $g \in K_u$ such that $e_1 g = e_1 A$. Multiplying on the right by g^{-1}, results in $e_1 = e_1 A g^{-1}$ and so the first row of Ag^{-1} must be e_1. Because the dot product of row 1 with any other row is zero (due to the fact that $Ag^{-1} \in O_m$) , the first column of Ag^{-1} must have zeros in every row but the first. Therefore, Ag^{-1} is of the form:

$$Ag^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & A_1 \end{pmatrix}, \qquad A_1 \in O_{m-1}.$$

We show by induction that for $m-1 \geq 4$, O_m equals the group generated by P_m and a transvection T_u with $wt(u) = 4$. If $m-1 = 4$, we must show $O_4 = \langle P_4, T_w \rangle$, where $wt(w) = 4$. By the proof of b, $|\langle P_4, T_w \rangle| = 2 \cdot 4! = |O_4|$. If $m-1 > 4$, assume $O_{m-1} = \langle P_{m-1}, T_{u'} \rangle = K_{u'}$, where $wt(u') = 4$. Given a matrix of the form Ag^{-1}, we know $A_1 \in O_{m-1} = K_{u'}$ (by assumption). Recall that if $T_v \in K_v$ for a vector v with $wt(v) = c$, K_v contains all T_w such that $wt(w) = c$. For this reason, we can choose specific u and u' WLOG. Let:

$$u' = [111100 \cdots 0] \in F_2^{(m-1)}$$

$$u = [011110 \cdots 0] \in F_2^{(m)}$$

where u' is a vector of length $m-1$ and u is a vector of length m. Let $A_1 = T_{u'}$. Then

$$T_u = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & & \cdots & & 0 & & & \\ \vdots & & & & \vdots & & I_{m-5} & \\ 0 & & \cdots & & 0 & & & \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & T_{u'} & \\ 0 & & & \end{pmatrix}$$

Basically, we are showing that an m-dimensional transvection can be made into an $(m+1)$-dimensional transvection simply by adding a row of zeros to the top and a column of zeros to the left and placing a 1 on the diagonal at their intersection as shown above. Thus, $\mathrm{Diag}[1, T_{u'}] \in K_u$. Now, suppose $A_1 = P$ for some $P \in P_{m-1}$. It follows that $\mathrm{Diag}[1, A_1] \in P_m \subset K_u$. Then, $Ag^{-1} \in K_u$ and since K_u is a subgroup with $g \in K_u$, $A \in K_u$. By induction we have shown $O_m = \langle P_m, T_u \rangle = K_u$ for $wt(u) = 4$ and $m - 1 \geq 4$.

The above theorem shows that given a simple transvection, T_u and a few generators for the permutation group, P_m, we can construct the entire orthogonal group, O_m, which we can in turn use to generate self-dual codes, the goal

of this paper. We now have the tools necessary to fully explain the algorithm created by Gaborit and Otmani for constructing new self-dual codes from an initial self-dual code generating matrix, G. We then proceed by analyzing its overall effectiveness by examining observed advantages and disadvantages of the method.

3 Three Strategies for Finding Self-Dual Codes

3.1 Gaborit and Otmani's Algorithm

Gaborit and Otmani's strategy for constructing self-dual codes is to devise a self-dual code C with a small minimum distance, and then, using Lemma 1, apply a series of linear maps to C to produce a self-dual code with greater minimum distance. Their procedure for constructing a self-dual code of length $2n$ is as follows:

For a small code-word length m, where $m|2n$, construct a self-dual code with generator matrix g. Then construct a generator matrix G for a code of length $2n$ by placing $2n/m$ copies of g along the diagonal of G. Over $GF(2)$, for example, $g = (\ 1 \quad 1\ )$ defines a self-dual code of length 2, while

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & & & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{pmatrix}$$

defines a self-dual code of length $2n$.

To use Lemma 1, one must construct a matrix M with the property that $MM^T = \lambda I_{2n}$ where λ is a non-zero element in the base field. For small dimensions, one can easily construct a $b \times b$ square matrix B such that $BB^T = \lambda I_b$. Once such a small B is found, one forms a $2n \times 2n$ matrix M as a block matrix consisting of $\lfloor 2n/b \rfloor$ copies of B along the diagonal, with the remaining $2n - b\lfloor 2n/b \rfloor$ spaces along the diagonal filled by the matrix $\beta I_{2n-b\lfloor 2n/b \rfloor}$, where $\beta^2 = \lambda$. Thus, M has the form:

$$M = \begin{pmatrix} B & 0 & \cdots & 0 \\ 0 & B & & \\ \vdots & & \ddots & \\ 0 & & & \beta I \end{pmatrix}$$

and has the property that $MM^T = \lambda I_{2n}$.

Since every $2n \times 2n$ permutation matrix Π has the property that $\Pi\Pi^T = I_{2n}$, the matrix $(M\Pi)^r$ has the property that $(M\Pi)^r((M\Pi)^r)^T = \lambda^r I_{2n}$. Thus, once one has calculated M, one can produce many matrices L with the property that LL^T is a scalar matrix; using Lemma 1, one can then produce many generator matrices for self-dual codes.

One goal of the Gaborit-Otmani algorithm is to provide integer parameters j, a, and r that can be varied to produce new generator matrices. The j and a parameters determine which permutation matrices are used, while the r parameter determines how many times G is multiplied by a matrix of the form $M\Pi$, where Π is among the selected permutation matrices.

To construct $2n \times 2n$ permutation matrices, Gaborit and Otmani specify that the a parameter must be co prime to $2n$ and the j parameter must select f_j from one of three families of invertible functions on $\mathbb{Z}_{2n}$:

$$\begin{aligned} f_1 &= \{\pi_i(z) : z \mapsto a \cdot (z+1) | 1 \leq i \leq r\} \\ f_2 &= \left\{\pi_i(z) : z \mapsto a^i \cdot (z+1) | 1 \leq i \leq r\right\} \\ f_3 &= \left\{\pi_i(z) : z \mapsto a^i \cdot (z+i) | 1 \leq i \leq r\right\} \end{aligned}$$

The $2n \times 2n$ permutation matrix Π_i then describes the action of $\pi_i \in f_j$ on the points $\{0, 1, 2, \ldots, 2n-1\}$. For example, if $2n = 4$, $a = 1$, and $j = 1$, then $f_1(z) = 3(z+1)$, and the corresponding permutation matrix is:

$$\Pi_i = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Having selected parameters j, a, and r, the final formula for the new generator matrix G' is

$$G' = G \prod_{i=1}^{r} (M\Pi_i)$$

where for each i, Π_i corresponds to the function $\pi_i \in f_j$.

Using the GUAVA package provided in GAP, we implemented Gaborit and Otmani's method. We executed this program (see attached for the source code) on a four-core Linux server with 16 gigabytes of random-access memory. The program verified results given in their paper. We give sample tables for $2n \leq 54$ over $GF(2)$ and $2n \leq 60$ over $GF(3)$ on the following pages.

Table 1: Self-dual codes in GF(2)

$2n$	d	(f_i, a, r)	time (h:m:s)
2	2	(f_1,1,1)	0
4	2	(f_1,1,1)	0
6	2	(f_1,1,1)	0
8	2	(f_1,1,1)	0
10	2	(f_1,1,1)	0
12	4	(f_1,1,4)	0
14	4	(f_1,1,4)	0
16	4	(f_1,1,1)	0
18	4	(f_1,1,4)	0
20	4	(f_1,1,4)	0
22	6	(f_1,1,20)	0
24	6	(f_1,1,4)	0
26	6	(f_1,1,76)	0
28	6	(f_1,1,4)	0
30	6	(f_1,1,17)	0
32	8	(f_1,3,3)	0
34	6	(f_1,1,10)	0
36	8	(f_1,1,8)	0
38	8	(f_1,1,116)	0
40	8	(f_1,1,8)	00:00.1
42	8	(f_1,1,94)	00:00.2
44	8	(f_1,1,8)	00:00.4
46	8	(f_1,1,10)	00:00.8
48	8	(f_1,1,12)	00:01.5
50	8	(f_1,1,12)	00:03.1
52	10	(f_1,1,19)	00:06.1
54	8	(f_1,5,9)	00:12.3

Table 2: Self-dual codes in $GF(3)$

$2n$	d	(f_i, a, r)	time (h:m:s)
4	3	(f_1,1,1)	0
8	3	(f_1,1,1)	0
12	6	(f_1,5,2)	0
16	6	(f_1,1,5)	0
20	6	(f_1,1,5)	0
24	9	(f_1,1,8)	00:00.2
28	9	(f_1,1,8)	00:01.8
32	9	(f_1,1,9)	00:17.1
36	9	(f_2,29,22)	02:28.9
40	12	(f_1,1,59)	23:13.0
44	12	(f_1,1,12)	03:41:48.5
48	12	(f_1,1,14)	-
52	12	(f_3,7,187)	-
56	15	(f_1,43,49)	-
60	15	(f_1,1,16)	-

Note that in Gaborit and Otmani's results for $GF(2)$, $r = 3$ not 4 for $n = 24$ and for $n = 54$, a separate B matrix was used for the construction of M.

As a small experiment, we enumerated all possible choices of B for a small dimension, m, and selected a length $2n$ for the self-dual codes we would examine. We found that for some B, it was difficult to find a choice of f, a, and r that would yield a code with high minimum distance. In some cases, no such code could be obtained. Although not thorough enough to be conclusive, the experimental results suggest a theoretical motivation behind Gaborit and Otmani's selection of B. It appears B's that are simply permutation matrices are less successful than B's that are transvections or permutations of transvections. In fact, if we examine their choice of B over $GF(2)$, we see that B is a transvection where $u = [1111]$.

If we restrict our attention to self-dual codes over $GF(2)$, we can see that

Gaborit and Otmani's algorithm is designed to select an element from O_{2n} and apply it to G in the hope of that the resulting matrix describes a self-dual code with greater minimum distance. Examining the listing of self-dual codes provided on [Gaborit's Website] we can see that this procedure can find codes with high minimum distances, but not necessarily codes with the highest minimum distance for their weight. Indeed, one deficiency of the algorithm is that without a good theoretic explanation for its design, it is difficult to judge whether examining some triples (f, a, r) will be useful or merely redundant. It is not even apparent from the construction that it is possible to find every self-dual code of length $2n$ with a choice of parameters (f, a, r); potentially, this approach misses codes with high minimum distance. In an attempt to correct this problem and better utilize the group-theoretic structure of O_n, we devised the following algorithm.

3.2 An Orbit Algorithm

From Theorem 1, we know that O_{2n} acts transitively on the set of all self-dual codes over $GF(2)$. By Theorem 4 we also know a generating set for O_{2n}. Using Theorem 2, we know the total number of self-dual codes of length $2n$. Consequently, we can use the basic orbit algorithm to systematically apply the elements of O_{2n} to an initial self-dual code C, and thereby find every self-dual code of length $2n$. In fact, since we are only interested in self-dual codes up to isomorphism, we can exit the orbit algorithm once we have found one representative for each class of equivalent codes. Once we have one representation for each class of equivalent self-dual codes of a given length, we can be certain the minimum distances found are the highest possible for self-dual codes of that length.

The orbit algorithm takes as input, $g = \{g_1, g_2, g_3\}$, generators of O_{2n}, and a self dual code C; as output, it returns a list of inequivalent codes, one for each class of isomorphic codes. A list Δ, representing the orbit, is initialized to contain C. The generators act on the elements in Δ to come up with other self-dual codes C'_1, C'_2, C'_3. When C'_i is not found in Δ, it is added to the orbit to be later acted on by each of the generators. Throughout the procedure, we also maintain a list Ω of non-isomorphic codes. When a new code is found in the standard orbit procedure, we check whether it is isomorphic to any of the codes in Ω. If no such isomorphism is found, the code is added to Ω; using Theorem 2, we can then check whether we have found representatives for every class of self-dual codes, to see if the algorithm can be terminated. We implement the algorithm below in pseudo code.

```
Δ := [C]; Ω := [C];
  for δ ∈ Δ do
    for i ∈ {1, ..., m} do
      γ := δ^{g_i};
      if γ ∉ Δ then
         Append γ to Δ;
```

```
            if ∀ω ∈ Ω : γ ≇ ω
              Append γ to Ω;
              if Σ_{ω∈Ω} 2n!/Aut(|ω|) = Π_{i=1}^{2n/2−1} 2^i − 1 then
                return Ω;
              fi;
            fi;
          fi;
        od;
      od;
    return Δ;
```

In terms of implementation, one can optimize this algorithm by sorting Δ into smaller lists according to code equivalence; this minimizes the number of vector-space comparisons required to determine whether a code C' is in Δ.

Compared to Gaborit and Otmani's algorithm, this procedure has the advantage that it will find a representative for every equivalence class of self-dual codes. Thus, we do not need to worry that we have missed a code with desirable properties. On the other hand, the algorithm really solves a different problem—it classifies all codes of a given length $2n$, while we are only interested in codes with high minimum distances. Because of this unnecessary thoroughness, as $|O_{2n}|$ becomes large it may be necessary to store many self-dual codes to find only a handful of inequivalent representatives. Our implementation was able to verify the table of inequivalent self-dual codes found in [1] up to $2n = 14$. For $2n \geq 14$, however, the computations became more time consuming than we were willing to complete (upwards of six hours).

3.3 A Probabilistic Algorithm

Given the computational demands of the orbit-based strategy, we considered a probabilistic approach. In addition, rather than trying to classify all self-dual codes of length $2n$, we returned to Gaborit and Otmani's focus: finding self-dual codes with high minimum distance.

In principle, the probabilistic algorithm is simple. Given a generator matrix G for a self-dual code, one randomly selects a matrix M from O_{2n}, and then considers the properties of the self-dual code defined by GM. If the new code does not have the desired properties, we choose another element from O_{2n} and repeat the procedure.

Randomly selecting an element from O_{2n} when one is only given the group's generators is a nontrivial problem, however. To accomplish this task, we used the product-replacement algorithm described in [2]. This algorithm is not guaranteed to produce elements from O_{2n} with uniform randomness, but it has proved random enough for our purposes.

The product replacement algorithm takes the generators of O_{2n}, g_0, g_1, and g_2, and two parameters, K and N, as input. An array Ω with N entries is

Table 3: Sample Output of Results from Probabilistic Algorithm

$2n$	d^a	Type I^b	Type II ^{c}	Total	Total that Exist
8	2	0	0	0	1
8	4	0	1	1	1
10	2	1	-	1	2
12	4	1	0	1	1
14	4	1	-	1	1
16	4	1	1	2	3
18	4	2	-	2	2
20	4	3	0	3	7
22	6	1	-	1	1
24	6	1	0	1	1
24	8	0	0	0	1
26	6	1	-	1	1
28	6	2	0	2	3
30	6	4	-	4	13
32	8	0	0	0	8

[a] N=10, K=200, tries=100
[b] A Type I code is a singly even code
[c] A Type II code is a doubly even code

then initialized so that the mth entry of Ω is $g_{(m \mod 3)}$; consequently after any number of spins, Ω always contains a generating set for O_{2n}.

We now perform the following "mixing" procedure K times: First, integers j and k are selected at random from $\{1, 2, \ldots, N\}$, so that $j \neq k$. The jth entry in Ω is then multiplied by the k-th entry in Ω, and the result is stored in the jth component of Ω.

Finally, we return a random element from Ω. Our results for $2n \leq 32$ are given in Table 3. Table 4 on the following page compares the performance of our random algorithm with Galborit and Otmani's results.

Table 4: Method Comparisons of Distances over GF(2)

$2n$	Gaborit & Otmani	Probabilistic	Highest Known d
2	2	-	2
4	2	-	2
8	2	4	4
10	2	2	2
12	2	2	4
14	4	4	4
16	4	4	4
18	4	4	4
20	4	4	4
22	6	6	6
24	6	6	8
26	6	6	6
28	6	6	6
30	6	6	6
32	8	6	8
34	6	6	6
36	8	6	8
38	8	8	8
40	8	8	8
42	8	8	8
44	8	8	8
46	8	8	10
48	8	8	12
50	8	8	10

From Table 4 above, one can see that in some cases, the probabilistic algorithm performs as well as Gaborit and Otmani's construction. In addition, because our algorithm selects elements randomly from O_{2n}, we can be sure that it is possible for it to output any of the possible self-dual codes of that length. Unfortunately, one can also see from the entries for $2n > 30$ that it is sometimes difficult to randomly find codes with high minimum distance.

4 Conclusion

To some extent, it is difficult to compare the strengths and weaknesses of the three algorithms discussed above because they attempt to solve different problems. Gaborit and Otmani's scheme was constructed to find self-dual codes with high minimum distances. From an engineering perspective, their procedure is an efficient solution to the problem; mathematically, however, we would like to more fully understand what underlying structure allows this strategy to be so successful.

In contrast, our orbit-based approach to the problem is grounded in the group-theoretic properties of O_{2n}. In addition to locating good self-dual codes, this algorithm attempts to solve the more difficult problem of classifying all self-dual codes of a given length $2n$. Given the size of O_{2n} for $2n > 14$, it is unsurprising that this computationally demanding approach could not duplicate Otmani and Gaborit's results.

Our probabilistic strategy also uses constructions from group theory to more efficiently compute codes with high minimum distances, but requires further tuning to make the procedure successful for larger values of $2n$.

Gaborit and Otmani's original goal was to "quickly and easily" find self-dual codes with high minimum distances for large values of $2n$. From this perspective, their solution is more effective than either of the algorithms we devised due to its ability to locate self-dual codes with lengths prohibitively large for the other two procedures. It would be interesting to determine what percentage of the inequivalent codes the experimental strategy can find. Still, in light of the modest success of the probabilistic algorithm, as well as its simplicity of implementation and strong theoretical basis, we would also like to know whether this strategy could be optimized to find codes of longer lengths.

References

[1] Bilous, R.T, An Enumeration of Binary Self-Dual Codes of Length 32, Designs, Codes, and Cryptography 26 (2002) 61-86.

[2] Celler, Frank, Leedham-Green, Charles R., Murray, Scott H., Niemeyer, Alice C. and O'brien, E.A. Generating random elements of a finite group, Communications in Algebra, 23:13 (1995) 4931-4948.

[3] Gaborit, P. and A. Otmani, Experimental Constructions of Self-Dual Codes, Finite Fields and Their Applications 9 (2003) 372-394.

[4] Gaborit, P. and A. Otmani, Tables of Self-Dual Codes. Website, 1998. `http://www.unilim.fr/pages_perso/philippe.gaborit/SD/index.html`.

[5] Janusz, Gerald J., Parametrization of Self-Dual Codes by Orthogonal Matrices, Finite Fields and Their Applications 13 (2007) 450-491.

[6] MacWilliams, J., Orthogonal Matrices Over Finite Fields, The American Mathematical Monthly 76 (1969) 152-164.