Group Codes based on Wreath Products of Complex Reflection Groups

J. B. Nation
Department of Mathematics
University of Hawaii
Honolulu, HI 96822
Email: jb@math.hawaii.edu

Catherine Walker
Department of Mathematics
Leeward Community College
Pearl City, HI 96782
Email: walkercl@hawaii.edu

***Abstract*—Group codes based on wreath products of complex matrix groups are constructed. Efficient algorithms for encoding and decoding these codes are described.**

I. Introduction

Group codes, introduced by Slepian [7], [8], all follow the same basic plan. The code is determined by a finite group $\mathbf{G}$ of isometries acting on a vector space V, and an initial vector $\mathbf{x}_0$. The codewords are a subset of the orbit $\mathbf{G}\mathbf{x}_0 = \{g\mathbf{x}_0 : g \in \mathbf{G}\}$. A codeword $\mathbf{x} = g^{-1}\mathbf{x}_0$ is transmitted, and the received vector is $\mathbf{r} = \mathbf{x} + \mathbf{n}$ where $\mathbf{n}$ represents channel noise. Let $\mathbf{r}_0 = \mathbf{r}$. Recursively, given $\mathbf{r}_k$, we can apply a transformation c_{k+1} from some specified set $X_k \subseteq \mathbf{G}$ to obtain $\mathbf{r}_{k+1} = c_{k+1}\mathbf{r}_k$. The transformation c_{k+1} is chosen so that $\|\mathbf{r}_{k+1} - \mathbf{x}_0\| < \|\mathbf{r}_k - \mathbf{x}_0\|$ in some metric. (We use the Euclidean metric throughout.) After a fixed number m of steps, depending on the group, terminate and decode as $c_m \dots c_1$.

Slepian used orthogonal groups on real vector spaces $\mathbb{R}^n$. Among the groups proposed by Slepian are the groups $\mathbf{B}_n$ of signed permutation matrices (a.k.a. variant II). These real reflection groups have the form of a wreath product $\mathbf{C}_2 \wr \mathbf{S}_n$, where $\mathbf{C}_2$ denotes the group $\{1, -1\}$. Fossorier, Nation and Peterson [1] introduced subgroup decoding as an efficient way to decode codes based on real reflection groups. This was extended to certain complex reflection groups $\mathbf{G}(r, 1, n)$ acting on $\mathbb{C}^n$ in Kim, Nation and Shepler [5]. The groups $\mathbf{G}(r, 1, n)$ are wreath products of cyclic groups, with $\mathbf{G}(r, 1, n) = \mathbf{C}_r \wr \mathbf{S}_n$ for the group $\mathbf{C}_r$ of complex r-th roots of unity. For example, $\mathbf{B}_n$ is $\mathbf{G}(2, 1, n)$. This structure allows us to generalize the methods used on real reflection groups. However, subgroup decoding does not work properly on most other types of complex reflection groups. Kim [4] devised an algorithm to decode these groups correctly, which was considerably refined by Walker [10]. Thus for certain relatively small complex reflection group codes, we now have efficient decoding algorithms. Our plan is to take wreath products of these codes, to produce much larger codes that can still be decoded effectively. The authors wish to thank Ian Blake for suggesting this approach.

II. Some exceptional complex reflection groups

The irreducible complex reflection groups were classified by Shephard and Todd [6]. This note describes group codes based on wreath products of some two-dimensional complex reflection groups, because there are efficient decoding schemes for their group codes separately. The following groups are good candidates, with simple presentations (that can also be given as Coxeter diagrams).

- $\mathbf{G}_4$: $\mathbf{A}^3 = \mathbf{B}^3 = \mathbf{I}$, $\mathbf{ABA} = \mathbf{BAB}$ (24 elements)
- $\mathbf{G}_8$: $\mathbf{A}^4 = \mathbf{B}^4 = \mathbf{I}$, $\mathbf{ABA} = \mathbf{BAB}$ (96 elements)
- $\mathbf{G}_{16}$: $\mathbf{A}^5 = \mathbf{B}^5 = \mathbf{I}$, $\mathbf{ABA} = \mathbf{BAB}$ (600 elements)
- $\mathbf{G}_5$: $\mathbf{A}^3 = \mathbf{B}^3 = \mathbf{I}$, $\mathbf{ABAB} = \mathbf{BABA}$ (72 elements)
- $\mathbf{G}_{20}$: $\mathbf{A}^3 = \mathbf{B}^3 = \mathbf{I}$, $\mathbf{ABABA} = \mathbf{BABAB}$ (360 elements)

There is a straightforward way to explicitly construct generators for these groups, denoted $\mathbf{A}$ and $\mathbf{B}$, using the method of Householder [2], [3]. The eigenvalues for the generators are $\lambda = e^{\frac{2\pi i}{3}}$ for $\mathbf{G}_4$, $\mathbf{G}_5$ and $\mathbf{G}_{20}$; $\lambda = i$ for $\mathbf{G}_8$; and $\lambda = e^{\frac{2\pi i}{5}}$ for $\mathbf{G}_{16}$. We can use the matrices $\mathbf{A} = \mathbf{I} - (1-\lambda)\mathbf{u}\mathbf{u}^H$ where $\mathbf{u} = [0, 1]^T$ and $\mathbf{B} = \mathbf{I} - (1-\lambda)\mathbf{v}\mathbf{v}^H$ with $\mathbf{v} = [v_1, v_2]^T$ real and $v_1^2 + v_2^2 = 1$. In practice, we want $v_1 > 0$ and $v_2 < 0$. Computing using the relations $\mathbf{ABA} = \mathbf{BAB}$ etc., we obtain the following values for v_2.

- For $\mathbf{G}_4$, $v_2 = -\sqrt{\frac{1}{3}}$.
- For $\mathbf{G}_8$, $v_2 = -\sqrt{\frac{1}{2}}$.
- For $\mathbf{G}_{16}$, $v_2 = -\sqrt{\frac{5+\sqrt{5}}{10}}$.
- For $\mathbf{G}_5$, $v_2 = -\sqrt{\frac{2}{3}}$.
- For $\mathbf{G}_{20}$, $v_2 = -\sqrt{\frac{3+\sqrt{5}}{6}}$.

By symmetry, it makes sense to choose $\mathbf{x}_0$ so that the distances satisfy $\|\mathbf{A}^{-1}\mathbf{x}_0 - \mathbf{x}_0\| = \|\mathbf{B}^{-1}\mathbf{x}_0 - \mathbf{x}_0\|$. Now

$$\begin{aligned}\|\mathbf{A}^{-1}\mathbf{x}_0 - \mathbf{x}_0\| &= \|\mathbf{x}_0 - \mathbf{A}\mathbf{x}_0\| = \|(\mathbf{I} - \mathbf{A})\mathbf{x}_0\| \\ &= \|(1-\lambda)\mathbf{u}\mathbf{u}^H\mathbf{x}_0\| = |1-\lambda|(\mathbf{x}_0^H\mathbf{u}\mathbf{u}^H\mathbf{x}_0)^{\frac{1}{2}} \\ &= |1-\lambda||\mathbf{u}^H\mathbf{x}_0|.\end{aligned}$$

So the required condition is $|\mathbf{u}^H\mathbf{x}_0| = |\mathbf{v}^H\mathbf{x}_0|$. Again it is convenient to choose $\mathbf{x}_0$ real. For the groups we are considering, one obtains the following non-normalized initial vectors.

- For $\mathbf{G}_4$, $\mathbf{x}_0 = (\frac{\sqrt{2}+\sqrt{6}}{2}, 1)$.
- For $\mathbf{G}_8$, $\mathbf{x}_0 = (1+\sqrt{2}, 1)$.
- For $\mathbf{G}_{16}$, $\mathbf{x}_0 = (\frac{1+\sqrt{5}+\sqrt{10+\sqrt{20}}}{2}, 1)$.

- For $\mathbf{G}_5$, $\mathbf{x}_0 = (\sqrt{2}+\sqrt{3}, 1)$.
- For $\mathbf{G}_{20}$, $\mathbf{x}_0 = (\frac{3+\sqrt{5}+\sqrt{18+\sqrt{180}}}{2}, 1)$.

Thus we can easily construct the generator matrices and initial vector for these $\mathbf{G}_k$. With a little more work, we can identify all the reflections, which are conjugates of the generators. Encoding is easy: choose some canonical form for each $g \in \mathbf{G}$, and transmit $g^{-1}\mathbf{x}_0$.

These exceptional reflection groups generally do not admit subgroup decoding [4]. However, they can be efficiently decoded by the *snowflake* algorithm [10]. Let $X \subset \mathbf{G}$ consist of all the reflections in $\mathbf{G}$, the identity element $\mathbf{I}$, the element in the presentation of $\mathbf{G}$, and its inverse. For example, for $\mathbf{G}_4$, $\mathbf{G}_8$, and $\mathbf{G}_{20}$, X would include the reflections, identity, $\mathbf{ABA}$, and the element $\mathbf{A}^2\mathbf{B}^2\mathbf{A}^2$, $\mathbf{A}^3\mathbf{B}^3\mathbf{A}^3$, or $\mathbf{A}^4\mathbf{B}^4\mathbf{A}^4$, respectively. For larger dimensional groups, additional relation elements may also be included. Let W_0 be the set of all codewords, i.e., $W_0 = \mathbf{G}\mathbf{x}_0$. Now define

$$d_1 = \max_{\mathbf{w}\in W_0} \min_{h\in X} \|h\mathbf{w} - \mathbf{x}_0\|$$

and let $X_1 \subseteq X$ be a minimal-sized set such that for all $\mathbf{w} \in W_0$ there exists $h \in X_1$ with $\|h\mathbf{w} - \mathbf{x}_0\| \le d_1$. Then set $W_1 = \{\mathbf{w} \in W_0 : \|\mathbf{w} - \mathbf{x}_0\| \le d_1\}$.

Recursively, let

$$d_{j+1} = \max_{\mathbf{w}\in W_j} \min_{h\in X_j} \|h\mathbf{w} - \mathbf{x}_0\|$$

and choose $X_{j+1} \subseteq X$ to be a minimal-sized set such that for all $\mathbf{w} \in W_j$ there exists $h \in X_{j+1}$ with $\|h\mathbf{w} - \mathbf{x}_0\| \le d_{j+1}$. Then set $W_{j+1} = \{\mathbf{w} \in W_j : \|\mathbf{w} - \mathbf{x}_0\| \le d_{j+1}\}$. It is known that this process terminates in m steps, where $m \le 2n$ for a reflection group on $\mathbb{C}^n$, with $d_1 > d_2 > \cdots > d_m = 0$.

To decode a received vector $\mathbf{r}$, set $\mathbf{r}_0 = \mathbf{r}$. Then recursively find $c_{k+1} \in X_{k+1}$ such that $\|c_{k+1}\mathbf{r}_k - \mathbf{x}_0\| \le d_{k+1} + \varepsilon_{k+1}$, where $\varepsilon_{k+1} > 0$ is determined below, and let $\mathbf{r}_{k+1} = c_{k+1}\mathbf{r}_k$. Interpret the message as the group element $c_m \dots c_1$.

While ideally we are trying to solve $\|c_{k+1}\mathbf{r}_k - \mathbf{x}_0\| \le d_{k+1}$, some tolerance must be built in to account for channel noise. For $j \ge 1$, let

$$e_j = \min_{\substack{\mathbf{w}\in W_{j-1} \\ \|\mathbf{w}-\mathbf{x}_0\|>d_j}} \|\mathbf{w} - \mathbf{x}_0\|$$

and then $\varepsilon_j = \frac{1}{2}(e_j - d_j)$.

In [10], methods for finding and ordering the sets $X_1, \dots, X_m$ are given. While these sample groups, and hence the corresponding codes, are fairly small, they can also be decoded in a small number of steps. One way to compute the efficiency of this type of algorithm is to count the average number of comparisons γ for decoding, and then calculate $\delta = \frac{\gamma}{\log_2|\mathbf{G}|}$, the average number of comparisons per bit.

For $\mathbf{G}_4$, we have $m = 3$ rounds of comparisons, with $\gamma = 3.7$ and $\delta = .81$. For $\mathbf{G}_5$, there are $m = 3$ rounds of comparisons, with $\gamma = 5.5$ and $\delta = .90$. For $\mathbf{G}_8$, it is $m = 4$, with $\gamma = 5.6$ and $\delta = .85$. The group $\mathbf{G}_{16}$ did not work as well, with $m = 4$, $\gamma = 12.9$ and $\delta = 1.4$. The group $\mathbf{G}_{20}$ is in between, with $m = 4$, $\gamma = 8.9$ and $\delta = 1.04$.

III. $\mathbf{G} \wr \mathbf{S}_n$

For a group $\mathbf{G}$ of $k \times k$ unitary matrices, the group $\mathbf{G} \wr \mathbf{S}_n$ consists of all $kn \times kn$ block permutation matrices with entries from $\mathbf{G}$. That is, thinking of the elements of $\mathbf{G} \wr \mathbf{S}_n$ as $n \times n$ matrices of $k \times k$ matrices from $\mathbf{G}$, each member of $\mathbf{G} \wr \mathbf{S}_n$ would have one nonzero entry in each row and each column, and those nonzero entries would be members of $\mathbf{G}$.

The group $\mathbf{G} \wr \mathbf{S}_n$ acts on $\mathbb{C}^{kn}$, and we think of vectors in $\mathbb{C}^{kn}$ as n-vectors of k-vectors.

If $\mathbf{G}$ is generated by $\mathbf{A}$ and $\mathbf{B}$, then $\mathbf{G} \wr \mathbf{S}_n$ is generated by $a_1, b_1, t_2, t_3, \dots, t_n$ where

$$(a_1)_{ij} = \begin{cases} \mathbf{A} & \text{if } i = j = 1 \\ \mathbf{I} & \text{if } i = j \ne 1 \\ \mathbf{O} & \text{otherwise.} \end{cases}$$

The matrix b_1 is defined similarly, with $\mathbf{B}$ replacing $\mathbf{A}$. The block permutation matrices t_i for $2 \le i \le n$ are given by

$$(t_i)_{jk} = \begin{cases} \mathbf{I} & \text{if } j+1 = k = i \\ \mathbf{I} & \text{if } i = j = k+1 \\ \mathbf{I} & \text{if } j = k \ne i, i-1 \\ \mathbf{O} & \text{otherwise.} \end{cases}$$

Thus, for example, when $n = 3$ the generators are

$$a_1 = \begin{bmatrix} A & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \qquad b_1 = \begin{bmatrix} B & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$t_2 = \begin{bmatrix} 0 & I & 0 \\ I & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \qquad t_3 = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix}.$$

In decoding, one also uses the block diagonal matrices $a_2, \dots, a_n, b_2, \dots, b_n$ defined analogously, but these are not needed as generators. Note that the size of $\mathbf{G} \wr \mathbf{S}_n$ is $|\mathbf{G}|^n n!$, which is considerably larger than either group code, $\mathbf{G}$ or $\mathbf{S}_n$, individually.

IV. Initial vector

The initial vector for $\mathbf{G} \wr \mathbf{S}_n$ can be regarded as an n-vector of k-vectors for a k-dimensional group $\mathbf{G}$, where $k = 2$ in our examples. Let $\mathbf{v}_0$ denote the initial vector for $\mathbf{G}$. Mimicking the form of the initial vector for $\mathbf{G}(r, 1, n)$ in [5], let

$$\mathbf{x}_0 = \langle a\mathbf{v}_0, (a+b)\mathbf{v}_0, (a+2b)\mathbf{v}_0, \dots, (a+(n-1)b)\mathbf{v}_0 \rangle$$

where the ratio $\frac{b}{a}$ of the positive real numbers a and b is chosen (as below) to minimize the decoding error.

In a wreath product code based on $\mathbf{G} \wr \mathbf{S}_n$ with an initial vector $\mathbf{x}_0$ of the given form, the parameters a, $b > 0$ are at our disposal to minimize the error due to noise. For a normalized system with $\|\mathbf{x}_0\| = \|\mathbf{v}_0\| = 1$, there is really only one parameter, say b/a. Here is one reasonable way to choose b/a.

It is easy to see that the threshold for permutation errors, i.e., for incorrectly interchanging two entries, is $b/2$. Suppose the threshold for errors in the normalized inner code is p. Then

the threshold for error in the copy of length $a+jb$ is $p(a+jb)$. The least of these is of course pa.

A reasonable approach is to choose b/a so that these two thresholds are equal, yielding $b/2 = pa$ or $b/a = 2p$. With this choice, if the noise in each component has magnitude below $b/2$, then the decoding is always correct; if the noise is larger, then either type of decoding error is possible.

For the snowflake decoding algorithm, p is the minimum of the values $\varepsilon_1, \dots, \varepsilon_m$ determined in Section II. As an example, in our simulations using the reflection group $\mathbf{G}_8$, with the standard initial vector $\mathbf{v}_0$ and distances d_j we have chosen, $p = .069$ and $b/a = .137$.

V. Encoding

Let $\mathbf{H}_i$ denote the subgroup of $\mathbf{G} \wr \mathbf{S}_n$ consisting of those matrices that act on the ith coordinate only. Thus, for example, $\mathbf{H}_1$ is all block diagonal matrices $\mathrm{diag}(\mathbf{M}, \mathbf{I}, \dots, \mathbf{I})$ with $M \in \mathbf{G}$. Use $\mathbf{S}_n$ to denote the subgroup of block permutation matrices. Then every element of $\mathbf{G} \wr \mathbf{S}_n$ can then be written uniquely as $g = \pi h_n \dots h_1$ with $\pi \in \mathbf{S}_n$ and each $h_i \in \mathbf{H}_i$. Variations of this canonical form are possible, and indeed other forms were used in [1], [5] and our simulations, but this order is easy to describe and has good error control properties; see [5].

Some canonical form for the permutation π as a product of transpositions should also be chosen. Two versions of this are given in [1] and also used in [5]. One uses the adjacent transpositions t_i $(2 \le i \le n)$ mentioned above, while the second uses a larger set of transpositions for more efficient decoding.

As usual, the vector $g^{-1}\mathbf{x}_0 = h_1^{-1} \dots h_n^{-1}\pi^{-1}\mathbf{x}_0$ is transmitted through a noisy channel, and some corrupted vector $\mathbf{r}$ is received.

VI. Decoding

To decode $\mathbf{r}$, we reverse the process. Let $\mathbf{r} = \langle \mathbf{u}_1, \dots, \mathbf{u}_n \rangle$ and $\mathbf{x}_0 = \langle \mathbf{x}_{01}, \dots, \mathbf{x}_{0n} \rangle$. First, using for example the snowflake algorithm, find $h'_1, \dots, h'_n$ with $h'_i \in \mathbf{H}_i$ to minimize each $\|h'_i\mathbf{u}_i - \mathbf{x}_{0i}\|$. Applying these, we obtain $h'_n \dots h'_1\mathbf{r} = \mathbf{s} = \langle \mathbf{s}_1, \dots \mathbf{s}_n \rangle$.

Now find a permutation π' that sorts $\mathbf{s}$ according to size. That is, because $\mathbf{v}_0$ has the property $\|x_{01}\| < \|x_{02}\| < \dots < \|\mathbf{x}_{0n}\|$, we find π' such that $\pi'\mathbf{s} = \mathbf{t} = \langle \mathbf{t}_1, \dots, \mathbf{t}_n \rangle$ has $\|t_1\| < \|t_2\| < \dots < \|\mathbf{t}_n\|$. This can be done in any number of ways. Two different insertion sorts are suggested in Fossorier, Nation and Peterson [1], which have the advantage that they yield a canonical form for π'. Wadayama and Hagiwara suggest using linear programming methods for permutation codes [9].

Finally, decode the received vector as $g' = \pi' h'_n \dots h'_1$.

VII. Discussion

A standard measure of the efficiency of decoding methods for group codes is the average number of comparisons (here of distances to the initial vector) used in decoding a received vector. For wreath product codes, this will be the sum of the number of comparisons for the outer permutation code, i.e., the number of comparisons required to find π', and n times the average number of comparisons required to decode the inner matrix code, i.e., the time required to find $h'_1, \dots, h'_n$. The space required for sorting can also be a practical issue. While insertion sorts are asymptotically inefficient, using $\mathcal{O}(n^2)$ steps, for small n (say $n \le 20$) they are pretty good, using close to $n \log n$ steps; see [1].

So far we have only begun simulating and testing these group codes. For the wreath product codes $\mathbf{G}(r, 1, n)$, better performance with respect to errors can be obtained by using only a subset of the code, while maintaining the ease of decoding. The same may well hold for these larger wreath product codes, and this should be investigated.

References

[1] M. Fossorier, J. Nation and W. Peterson, *Reflection group codes and their decoding*, IEEE Trans. on Information Theory, **56** (2010), 6273–6293.

[2] A.S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell, New York, 1964.

[3] A.S. Householder, Lectures on Numerical Algebra, MAA, 1972.

[4] H.J. Kim, Decoding Complex Reflection Groups, Master's project, University of Hawaii, 2011.

[5] H.J. Kim, J.B. Nation and A. Shepler, *Group coding with complex permutation groups*, in preparation.

[6] G.C. Shephard and J.A. Todd, *Finite unitary reflection groups*, Canad. J. Math., **6** (1954), 274–304.

[7] D. Slepian, *Permutation modulation*, Proc. IEEE, **53** (1965), 228–236.

[8] D. Slepian, *Group codes for the Gaussian channel*, Bell Syst. Tech. J., **47** (1968), 575–602.

[9] T. Wadayama and M. Hagiwara, *LP decodable permutation codes based on linearly constrained permutation matrices*, Proc. of ISIT 2011, pp.139-143.

[10] C. Walker, The Snowflake Decoding Algorithm, Master's project, University of Hawaii, 2012.