

Decoding the Mathieu Group M_{12}

Robert Bailey

Queen Mary, University of London

`r.f.bailey@qmul.ac.uk`

`http://www.maths.qmul.ac.uk/~rfb/`

Joint work with P. J. Cameron

*1st North Eastern Postgraduate Pure Mathematics Workshop
28th July 2003*

Can write a permutation in various ways, e.g.

Two-line	$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 4 & 3 & 2 \end{pmatrix}$
Cycle	$(1) (2 \ 5) (3 \ 4)$
Passive	1 5 4 3 2

In passive form, we can define the *Hamming distance* between two permutations in the usual way:

e.g. $d(1 \ 5 \ 4 \ 3 \ 2, 2 \ 5 \ 4 \ 1 \ 3) = 3$

If, for a group (or set) of permutations, the minimum distance is known, it can be used as an error-correcting code, using the permutations as the transmitted words.

In general, this is not easy to calculate! However, for one family of groups it is straightforward.

Suppose G is a group acting on $\Omega = \{1, \dots, n\}$.

G is *transitive* if, for any two symbols $x, y \in \Omega$, $\exists g \in G$ such that $g(x) = y$.

G is *k-transitive* if for any two k -tuples of distinct symbols in Ω , $\exists g \in G$ mapping one to the other.

If this element g is *unique*, then we say G is *sharply k-transitive*.

It follows that if k positions of a permutation are known, then this determines a permutation *uniquely*.

$g, h \in G$ can agree in *at most* $k - 1$ positions, so the minimum distance is $n - k + 1$.

Groups are all known for $k \geq 2$ (Zassenhaus 1936).

Examples:

- the symmetric group S_n , which is sharply n and $n - 1$ transitive;
- the alternating group A_n , which is sharply $n - 2$ transitive;
- the affine groups $AGL(1, q)$, which are sharply 2-transitive;
- the projective group $PGL(2, q)$, which are sharply 3-transitive;
- the Mathieu groups M_{11} and M_{12} , which are sharply 4 and 5-transitive respectively.

Blake (1974) first suggested using sharply k -transitive groups as codes.

We'll concentrate on the sharply 5-transitive Mathieu group M_{12} .

M_{12} : sharply 5-transitive simple group of degree 12, order 95040.

M_{12} is generated by the permutations

$$(1\ 2)(3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12) \text{ and} \\ (1\ 3\ 2)(4\ 7\ 5)(8\ 9\ 11).$$

Standard result in coding theory: a code with minimum distance d can correct $\lfloor \frac{d-1}{2} \rfloor$ errors.

Here the minimum distance is $12 - 5 + 1 = 8$
 \Rightarrow can correct 3 errors.

How to decode?

Since the group is sharply 5-transitive, need to know 5 correct symbols to uniquely determine a group element.

If received word has 3 errors, in unknown positions, then we need to find a set of 5-subsets of $\{1, \dots, 12\}$ such that any 3-subset is *disjoint* from at least one 5-set.

Call this a $(12, 5, 3)$ -*uncovering*.

Example:

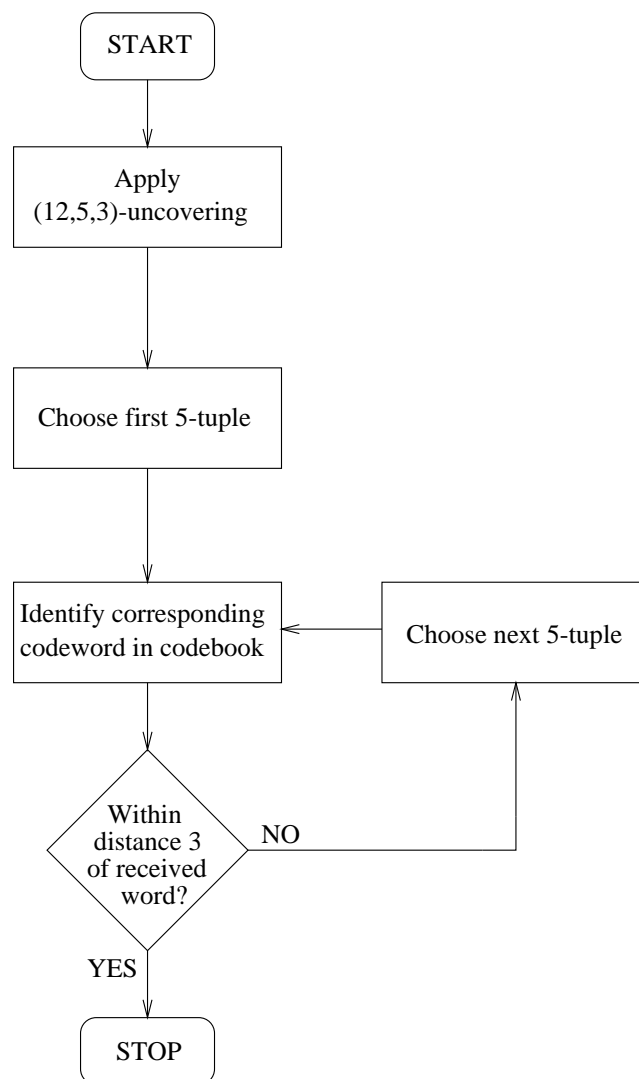
1	2	3	4	5
1	2	6	11	12
1	3	7	8	9
1	4	6	7	10
1	5	8	9	11
2	4	8	9	12
2	5	7	10	11
3	4	7	11	12
3	5	6	10	12
3	6	8	9	11
6	7	8	9	10

An (n, k, r) -uncovering is the complement of an $(n, n - k, r)$ covering design.

These are well-studied, and a database is available of the smallest known designs for certain parameters.

Once we have such an uncovering, for each 5-set we determine the *unique* group element agreeing with received word in these 5 positions.

If distance is ≤ 3 , then stop. Otherwise, go to next 5-set and repeat.



Example:

Suppose we transmit

$$g = 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12$$

and receive

$$w = 6\ 2\ 1\ 4\ 6\ 6\ 7\ 8\ 9\ 10\ 11\ 12$$

w has errors in positions 1, 3 and 5.

As the algorithm works through the uncovering, it outputs:

Error (repeated symbol);

Error (repeated symbol);

6 3 1 4 12 2 7 8 9 5 10 11, which is distance 6 from w and is rejected;

Error (repeated symbol);

Error (repeated symbol);

1 2 3 4 5 6 7 8 9 10 11 12, which is distance 3 from w and is accepted.

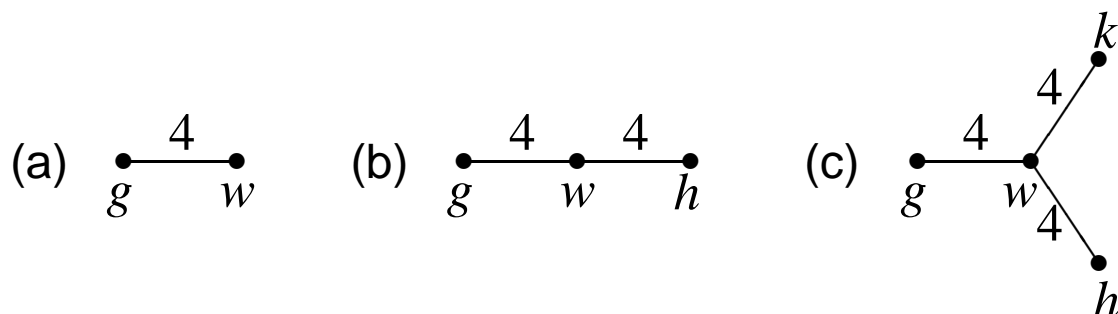
Why stop at 3 errors?

Algorithm only requires 5 correct symbols, so (in theory) there are cases where up to 7 errors are correctable.

What if there are 4 errors?

Minimum distance 8 \Rightarrow some words with 4 errors have *more than one* nearest neighbour in the code.

Our work has shown that a word with 4 errors can occur in one of the following configurations:



The number of configurations of types (b) and (c) is very small. In fact (after some equation-solving) only 3.3% of possible received words occur in these.

\Rightarrow 96.7% of possible received words are uniquely decodable!

5 or more errors?

Problem: some words will definitely decode incorrectly
e.g. if we have two codewords g, h with $d(g, h) = 8$, then
5 errors in g could move us within distance 3 of h
 \Rightarrow decoding failure.

Useful notion: *colouring* possible received words:

Green: will definitely decode correctly (and uniquely);

Yellow: will decode correctly but *not uniquely* (algorithm returns several possible solutions, one of which is the correct one);

Red: will definitely decode incorrectly.

For M_{12} , we have the following:

