AD NUMBER: AD0226767

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to US Government Agencies and their contractors; Administrative/Operational Use; 1 Jun 1959. Other requests shall be referred to Air Force Cambridge Research Laboratories, Hanscom AFB, MA 01731.

AUTHORITY

AFCRL ltr dtd 14 Apr 1970

AD-226 767

THE USE OF COSET EQUIVALENCE IN THE ANALYSIS AND DECODING OF GROUP CODES

Eugene Prange

June 1959

KEEP UP TO DATE

Between the time you ordered this report—which is only one of the hundreds of thousands in the NTIS information collection available to you—and the time you are reading this message, several *new* reports relevant to your interests probably have entered the collection.

Subscribe to the **Weekly Government Abstracts** series that will bring you summaries of new reports *as soon as they are received by NTIS* from the originators of the research. The WGA's are an NTIS weekly newsletter service covering the most recent research findings in 25 areas of industrial, technological, and sociological interest—invaluable information for executives and professionals who must keep up to date.

The executive and professional information service provided by NTIS in the **Weekly Government Abstracts** newsletters will give you thorough and comprehensive coverage of government-conducted or sponsored research activities. And you'll get this important information within two weeks of the time it's released by originating agencies.

WGA newsletters are computer produced and electronically photocomposed to slash the time gap between the release of a report and its availability. You can learn about technical innovations immediately—and use them in the most meaningful and productive ways possible for your organization. Please request NTIS-PR-205/PCW for more information.

The weekly newsletter series will keep you current. But *learn what you have missed in the past* by ordering a computer **NTISearch** of all the research reports in your area of interest, dating as far back as 1964, if you wish. Please request NTIS-PR-186/PCN for more information.

WRITE: Managing Editor
5285 Port Royal Road
Springfield, VA 22161

Keep Up To Date With SRIM

SRIM (Selected Research in Microfiche) provides you with regular, automatic distribution of the complete texts of NTIS research reports *only* in the subject areas you select. SRIM covers almost all Government research reports by subject area and/or the originating Federal or local government agency. You may subscribe by any category or subcategory of our WGA **(Weekly Government Abstracts)** or **Government Reports Announcements and Index** categories, or to the reports issued by a particular agency such as the Department of Defense, Federal Energy Administration, or Environmental Protection Agency. Other options that will give you greater selectivity are available on request.

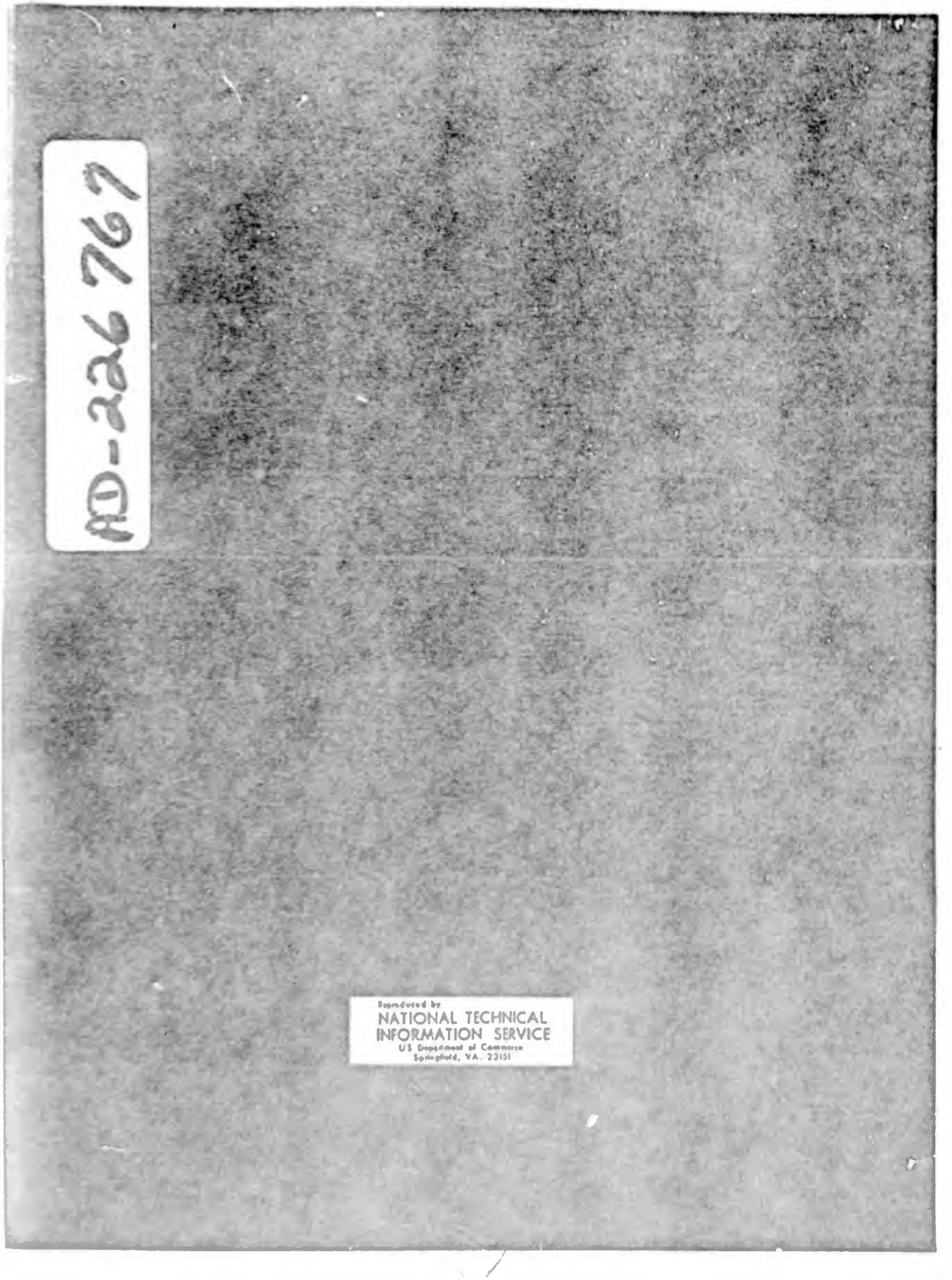
The cost of SRIM service is only 45¢ domestic (60¢ foreign) for each complete microfiched report. Your SRIM service begins as soon as your order is received and processed and you will receive biweekly shipments thereafter. If you wish, your service will be backdated to furnish you microfiche of reports issued earlier.

Because of contractual arrangements with several Special Technology Groups, not all NTIS reports are distributed in the SRIM program. You will receive a notice in your microfiche shipments identifying the exceptionally priced reports not available through SRIM.

A deposit account with NTIS is required before this service can be initiated. If you have specific questions concerning this service, please call (703) 451-1558, or write NTIS, attention SRIM Product Manager.

This information product distributed by

NTIS U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161



AFCRC-TR-59-164
AD226767

THE USE OF COSET EQUIVALENCE IN THE ANALYSIS AND DECODING OF GROUP CODES

EUGENE PRANGE

JUNE 1959

COMMUNICATION SCIENCES LABORATORY
ELECTRONICS RESEARCH DIRECTORATE
AIR FORCE CAMBRIDGE RESEARCH CENTER
AIR RESEARCH AND DEVELOPMENT COMMAND
UNITED STATES AIR FORCE
BEDFORD MASSACHUSETTS

ABSTRACT

For an error-correcting code group A, the group G of coordinate permutations that map A onto itself is used to define an equivalence relation on A-cosets. It is shown that this equivalence relation can be used in the analysis of the error-correcting properties of A, and in the definition and verification of operationally feasible decoding algorithms that satisfy the minimal distance criterion. Decoding algorithms under a (weak) block length criterion are also considered for a special class of group codes. Applications are made to some specific codes, the largest being a code in two symbols whose words have 45 information and 28 check positions.

ACKNOWLEDGMENTS

I am indebted to Professor J. M. Wozencraft and to the members of the Coding Theory Seminar at the Massachusetts Institute of Technology for the opportunity to present some of this material in December 1958. The computer programs used were written by me, and run on the Cambridge Computer in the Computer and Mathematical Sciences Laboratory of the Air Force Cambridge Research Center. The large part of the hand computations were done by Mrs. Alice M. Pierce.

THE USE OF COSET EQUIVALENCE IN THE ANALYSIS AND DECODING OF GROUP CODES

INTRODUCTION

A need exists for extending the repertoire of specific error-correcting codes for which an analysis of error-correcting properties has been made, and for which coding and decoding algorithms have been established. The systems designer needs to know not only what a particular code will yield in error-correcting, but what it will cost in coding and decoding operations.

Let A be a group code,[1] and let G be the group of coordinate permutations that map A onto itself. The group G is used to define an equivalence relation on A-cosets. In Part I, it is shown that this equivalence relation on A-cosets can be used in the analysis of the error-correcting properties of the group code A, and in the definition of decoding algorithms for A. The concept of coset equivalence seems particularly useful in the analysis of rather large codes, since the study of one coset in an equivalence class yields information about all cosets in the class.

In Part II, the methods of Part I are applied to some specific binary group codes, the largest of which has 45 information positions and 28 check positions. The decoding algorithms found for these codes are operationally

Received for publication June 2, 1959.

simpler than those considered in Part I and are closely related to those of I.S. Reed.[2-4]

Although we are primarily interested in decoding algorithms satisfying the requirement that each possible received sequence x be mapped onto an element of the code at minimal distance from x, (minimal distance criterion), it is noted that a (weak) block length criterion can be used to simplify decoding for a special class of codes considered: those group codes such that if $(x_0, x_1, \ldots, x_{n-1})$ is in the code, then so is $(x_1, x_2, \ldots, x_{n-1}, x_0)$.

PART I

The vector space J_p^n

Let J_p be the field of integers modulo the prime integer p. An element of J_p is a class of real integers, and we represent such a class by the real integer of smallest absolute value in the class. Thus, if $p = 2k+1$, we write each element of J_p as a symbol from the set $\{0, \pm 1, \ldots, \pm k\}$; the elements of J_2 are written as 0 or 1.

Let J_p^n be the set of all n-tuples $x = (x_0, x_1, \ldots, x_{n-1})$, with coordinates x_i in the field J_p. Then J_p^n can in a natural way be given the structure of a concrete n-dimensional vector space over the field J_p. Since p is prime, every additive subgroup of J_p^n is also a linear subspace.

If A is any k-dimensional subspace of J_p^n, we regard A as a group code of length n in p symbols and of order p^k.

Weight and distance functions on J_p^n

We call a distance function on J_p^n any mapping d of $J_p^n \times J_p^n$ into the real numbers such that

$$(1) \qquad \begin{cases} d(x, x) = 0 \\ d(x, y) > 0 \text{ if } x \neq y, \\ d(x, y) = d(y, x), \\ d(x, y) + d(y, z) \geq d(x, z). \end{cases}$$

We call a weight function on J_p^n any mapping w of J_p^n into the real numbers such that

$$(2)\qquad \begin{cases} w(0) = 0, \\ w(x) > 0 \text{ if } x \neq 0, \\ w(x) = w(-x), \\ w(x) + w(y) \geq w(x+y). \end{cases}$$

If $w(x)$ is a weight function, then a distance function can be defined by

$$(3)\qquad d(x,y) = w(x-y).$$

Given such weight and distance functions, we extend the definitions to over any subsets X and Y of J_p^n by defining

$$(4)\qquad \begin{cases} w(X) = \text{minimum } w(x) \text{ for } x \text{ in } X, \\ d(X,Y) = w(X-Y), \quad \text{where } X-Y = \{x-y\}_{x \text{ in } X,\ y \text{ in } Y}. \end{cases}$$

We now choose specific weight and distance functions suited to our purpose. If x is any element of the vector space J_p^n, we define the weight of x by

$$(5)\qquad w(x) = |x_0| + |x_1| + \ldots + |x_{n-1}|.$$

(Recall that by our convention on notation, an element of J_p is represented by the real integer of smallest absolute value belonging to this element.)

We also define a concept of a neighbor of a subset X of J_p^n. Let $e^{(i)}$ be the vector of J_p^n whose ith coordinate is equal to 1, the remaining coordinates being equal to 0. By a neighbor of X we mean any of the sets $X + e^{(i)}$ or $X - e^{(i)}$, $(0 \leq i \leq n-1)$. In particular, if X contains only one vector x, then x has exactly n or 2n distinct vectors as neighbors, depending on whether p is equal to or greater than 2. The distance of a set X to any of its neighbors is either 0 or 1.

Coset equivalence

Let A be an additive subgroup (and thus a subspace) of J_p^n. Let A' be the subspace of J_p^n containing all vectors y that are orthogonal to every

vector of A; that is,

(6) $$b \cdot y = b_0 a_0 + b_1 y_1 + \dots + b_{n-1} y_{n-1} = 0, \quad \text{all } b \text{ in } A.$$

Let X be any subset of A that contains a basis of A. Let X^* be the subset of J_p^n containing all vectors that can be obtained by applying a coordinate permutation to a vector in X. Let $B = A+b$ be an A-coset. We then have the following

Proposition. For any coordinate permutation P of J_p^n, the following conditions on P are equivalent:

(7) $$AP = A,$$

(8) $$A^{\perp} P = A^{\perp},$$

(9) $$XP \subseteq A,$$

(10) $$XP \subseteq X^* \cap A,$$

(11) BP is an A-coset.

The proof is simple and will be omitted.

Let G be the set of all coordinate permutations P for which $AP = A$. Then G is a group. The equivalence of (7) and (11) allows us to use G to define an equivalence relation for A-cosets: If B and C are A-cosets, call B equivalent to C if $BP = C$ for some P in G. Call an equivalence class of A-cosets a coset type.

Any coordinate permutation Q on J_p^n induces an automorphism of J_p^n that preserves all weight and distance relations. If the permutation Q is restricted to be in G, then Q also induces an automorphism of the factor space J_p^n/A (whose elements are the A-cosets) that preserves all weight and distance relations.

A decoding procedure based on coset equivalence

Suppose that the k-dimensional subspace A of J_p^n possesses a large group G of coordinate permutations that map A onto itself. We sketch a possible use of coset equivalence to define a decoding procedure that satisfies the requirement that any element x of J_p^n be mapped onto an element b of A such that the distance d(x, b) is minimal. We refer to this criterion as the minimal distance criterion for decoding.

We recall the idea of the decoding procedure given by David Slepian.[1] Let L be a linear transformation of J_p^n such that A is the null space of L. Then, given an element x in J_p^n, the value xL determines the A-coset to which the vector x belongs. Suppose that a list of pairs (z, e) for each z in the range of L has been computed and stored in the memory, where e is some vector of minimal weight in the A-coset corresponding to z. To decode x, compute z = xL, find the vector e corresponding to z in the list, and map x onto x-e. The procedure is a beautiful one so long as the number p^{n-k} of A-cosets is not too large.

The somewhat analogous decoding procedure to be described uses coset equivalence to shorten the list necessary. Suppose that, given x in J_p^n, an indicator t of the coset type to which the coset A + x belongs, can be computed. (The problem of how to do this will be considered in the following section.) Suppose further that the weight w of the cosets belonging to the coset type t has been determined for each t. Then, given any vector x, it is possible (by assumption) to compute a coset type indicator t and to find the weight w of the coset A + x by entering a list. Let $e^{(i)}$ be the vector whose ith coordinate is 1, the remaining coordinates being equal to 0.

Given x, determine whether either $x + e^{(0)}$ or $x - e^{(0)}$ belongs to a coset of lower weight than the coset containing x. If it does, make successive changes in the first coordinate of x by one unit in the same direction so long as each such change decreases coset weight. Repeat the procedure for the next coordinate, etc. Stop when x has been altered to a vector x' in the code A or on reaching the (k+1)st coordinate (assuming that the first k coordinates of x are information positions).

In each specific case that we have carried through a complete analysis, we have found a simpler decoding procedure than the one just described. These simpler decoding procedures are related to those of I.S. Reed.[2-4]

Identifying coset types

We do not know a solution to the problem of computing a complete set of invariants for coset type with respect to the group G. The set of invariants defined here has proved adequate in the computations made for specific codes. This set of invariants is computable whether or not the group is known.

Let S be a subset of the space $A^{\perp}$ such that if S contains the vector x, then S also contains every vector y that can be obtained by applying some coordinate permutation in G to x. If the group G is not known, this requirement can be satisfied by including with x all vectors y of $A^{\perp}$ that can be obtained by applying any coordinate permutation whatever to x. We also require that the set S contain a basis for $A^{\perp}$. Subject to these two conditions on S, it is desirable that the number of vectors in S be as small as possible.

A coordinate permutation P is in G if and only if SP = S. Let L be a matrix whose columns are the vectors of S in some order. Then a coordinate permutation P is in G if and only if

(12) $$PLQ = L$$

for some coordinate permutation Q.

Define a linear transformation $x \rightarrow xL$ (over J_p) on J_p^n. Then A is the null space of the linear transformation L, since we have required that the vectors of S be in $A^\perp$ and contain a basis of $A^\perp$. Similarly, let the matrix transpose L^T of L define a linear transformation L^T (over the reals) on the range of L. For P in G, we have

(13) $$LL^TP = PLL^T.$$

Let x and y be vectors of J_p^n that lie in cosets belonging to the same coset type. It follows from (12) that

(14) $$xL = yLQ$$

for some coordinate permutation Q. Thus, the number of coordinates of xL that are equal to any given (real) integer is an invariant for the coset type to which the coset A + x belongs.

Similarly, it follows from (13) that the number of coordinates of xLL^T equal to any given integer is also an invariant for the coset type to which the coset A + x belongs.

This procedure yeilds a set of invariants (not necessarily complete) for coset type. The set is finite, since the set of integers that can actually occur as coordinates of xL or of xLL^T is finite. Call this set of invariants a coset type indicator t.

In some of the codes examined in Part II, the coset type indicators do not completely determine coset type; however, the set of coset type indicators

for the neighbors of a coset in the type proves sufficient to do so in these cases.

Note that if xLL^T is unequal to zero, then xLL^T can be regarded as the real sum of some vectors y in $A^\perp$ such that $x \cdot y \neq 0$.

Counting the number of cosets in a coset type

Suppose that for each coset type α we know how many neighbors of each coset type any coset of type α possesses. Then counting the number of cosets in each coset type can easily be done by a recursive procedure based on the following relation: The number of cosets of type α times the number of neighbors of type β possessed by any coset of type α is equal to the number of cosets of type β times the number of neighbors of type α possessed by any coset of type β. We symbolize this rule by

$$(\alpha)(\beta|\alpha) = (\beta)(\alpha|\beta). \tag{15}$$

To start the counting procedure, use the fact that the group A is the unique coset in its coset type.

The use of coset equivalence in the analysis of a code

Given the code A, we assume that a matrix L is known such that (a) A is the null space of the linear transformation $x \to xL$ over J_p; (b) given P in the group G of coordinate permutations that map A onto itself, there is a coordinate permutation Q such that $PLQ = L$; (c) the coset type indicators t defined above in terms of L are in one-to-one correspondence with coset types. (We know that requirements (a) and (b) can always be satisfied. Requirement (c) allows us to simplify the statements made in this section. In Part II, an analysis is carried through for a specific code in a case where (c) does not hold.)

We wish to determine the coset type indicators t for each coset type, the number of cosets in the coset type, the weight of the cosets in the coset type, and the number of neighbors of each coset type belonging to a coset in any given type.

Recall that if x is a vector, the coset type indicator t of the coset type to which the coset A+x belongs is computed by first computing xL and xLL^T. By an abuse of notation, we also use t to denote coset type. We write t_x to indicate that t can be computed given any vector x belonging to some coset in the coset type t. We say that such a vector x <u>represents</u> the coset type t.

If the vector x represents the coset type t, then the neighbors $x \pm e^{(i)}$ of x represent the coset types that contain cosets that are neighbors of cosets in the coset type t.

We say that a coset type is of weight w if the cosets belonging to the type are of weight w.

Let $X^{(w)}$ be a set of vectors of weight w containing exactly one representative of each coset type of weight w (and containing no other vectors). Then,

(d) The set of coset type indicators t_x for x in $X^{(w)}$ is in one-to-one correspondence with the coset types of weight w, using assumption (c).

(e) The set of coset type indicators t_y, where $y = x \pm e^{(i)}$ for some x in $X^{(w)}$, is the set of coset type indicators of the neighbors of the coset A+x. If t_y is not the coset type indicator for a coset type of weight less than or equal to w, than t_y is the type indicator for a coset type of weight w + 1, and y is a representative of minimal weight w + 1 for this coset type.

(f) We can thus construct $X^{(w+1)}$ by choosing a subset of the neighbors of the vectors in $X^{(w)}$.

(g) To start, we take $X^{(0)}$ as the set containing only the null vector.

(h) We stop when we reach a weight w such that $X^{(w+1)}$ is empty.

(i) We now have enough information to count the number of cosets in each coset type by the method of the preceding section. A check is that the total number of cosets must be p^{n-k}, where k is the dimension of A.

PART II

Cyclic subspaces of J_p^n

We now restrict our attention to subspaces A of J_p^n such that the coordinate permutation Q defined by

(16) $$xQ = (x_1, x_2, \ldots, x_{n-1}, x_0)$$

is in the group G of coordinate permutations mapping the subspace A onto itself. We call such a subspace A a cyclic code space. There is a large body of mathematics, pure and applied, relevant to the study of these subspaces.[5,6] The permutation group G mapping any such code A onto itself is at least singly transitive, since the group generated by Q is singly transitive.

Let c be any vector of J_p^n. Define a linear transformation R_c of J_p^n into itself by

(17) $$xR_c = (x\cdot c, x\cdot(cQ), x\cdot(cQ^2), \ldots x\cdot(cQ^{n-1})).$$

The linear transformation R_c is symmetric; the matrix of R_c equals its

matrix transpose R_c^T.

Define a polynomial f_c in λ by

$$f_c(\lambda) = c_0 + c_1\lambda + \ldots + c_{n-1}\lambda^{n-1}. \tag{18}$$

Let k be the degree of the polynomial f_c, that is, k is the largest coordinate index for which $c_k \neq 0$. For the case $k > 1$, define a linear recursion ρ_c over J_p by

$$\rho_c : c_k x_{j+k} = c_0 x_j + \ldots + c_{k-1} x_{j+k-1} \quad (j = 0, 1, \ldots). \tag{19}$$

The subspace A of J_p^n is a cyclic subspace if and only if A is the null space of some linear transformation R_c. If A is a cyclic subspace of dimension $k (0 < k < n)$, there is a unique vector c in J_p^n such that

a) A is the null space of R_c;

b) f_c divides $1-\lambda^n$ over J_p;

c) f_c is of degree k, $c_k = 1$;

d) the period of the linear recursion ρ_c divides n.

The linear recursion ρ_c can be used to code J_p^k onto the subspace A of J_p^n.

Decoding cyclic codes under a weak block length criterion

Coding for group codes is relatively simple since the check digits are linear functions (over J_p) of the message digits. Coding by a linear recursion is particularly simple. Such coding is possible for a class of group codes including, but larger than, the class of cyclic codes.

In general, it seems difficult to define and verify feasible decoding algorithms that satisfy the minimal distance criterion if both the number k of message positions and the number n-k of check positions are relatively large. In this section we give a general decoding algorithm for cyclic codes

such that the work of decoding a vector is at most (and on the average, less than) n times the work of coding a vector. The procedure has the disadvantage that the minimal distance criterion may not be satisfied on all A-cosets.

If a decoding procedure maps the vector y of J_p^n onto the vector x of A, we call the vector z = y-x the (assumed) error vector. The vector z must lie in the coset A+y. If the decoding is to satisfy the minimal distance criterion, z must be of minimal weight in this coset.

In a decoding procedure that maps y onto x, the requirement that x agree with y in a circular block of at least k consecutive coordinates is here called the weak block length criterion. By a circular block of a vector $x = (x_0, \cdots, x_{n-1})$ we mean a block of consecutive coordinates where x_0 is considered the successor of x_{n-1}.

If A is a cyclic code space, then for every vector y in J_p^n at least one and at most n vectors x in A agree with y on a circular block of at least k coordinates. Equivalently, at least one and at most n vectors z in every A-coset have a circular block of at least k consecutive coordinates equal to zero. This follows from the fact that any k consecutive coordinate positions in A are linearly independent. Any element x of A is completely determined by a circular block of k consecutive coordinates (coordinate positions being known), and can be computed from this block by using the linear recursion that defines coding into A.

Let $y^{(i)}$ be the unique vector in A that agrees with a given vector y in J_p^n on the coordinates $y_i, y_{i+1}, \cdots, y_{i+k-1}$ (where the coordinate indices are regarded as integers modulo n.) The vectors $y^{(i)}$ for i from 0 to n-1 need

not all be distinct, since any such vector $y^{(i)}$ can agree with y on a circular block of more than k consecutive coordinates or on several blocks of at least k consecutive coordinates.

For cyclic codes we state a decoding algorithm that uses the weak block length criterion as primary rule and the minimal distance criterion as secondary rule. A numerical example is given at the end of this section.

Algorithm. Given y in J_p^n. Construct the (distinct) solution vectors $y^{(i)}$. At each stage, remember the indices i for which the solutions $y^{(i)}$ have not yet been examined; and from among those so far constructed, remember one solution at minimal distance from y. Stop either on finding $y^{(i)}$ for which the distance from y to $y^{(i)}$ is less than or equal to the packing integer for A, or when all possible values of $y^{(i)}$ have been examined.

Some trivial facts about the relation of this procedure to the minimal distance criterion are summarized in the following

Proposition. Let A be a cyclic code of length n and dimension k over J_p. Let α be the packing integer for A, that is, the largest integer such that all vectors of weight α or less lie in distinct A-cosets. Let β be the smallest integer greater than or equal to n/k. Then: (a) If $\alpha \geq w \geq \beta$, there are cosets of weight w such that for y in any of these cosets, the algorithm does not yield a minimal distance solution. (b) All error vectors of weight less than or equal to the minimum of α and $\beta-1$ are corrected by the algorithm. (c) If the distance of y from the best solution $y^{(i)}$ is less than or equal to the maximum of α and β, then the solution is necessarily a minimal distance solution.

Proof. All vectors of weight w in J_p^n have a circular block of at least k coordinates equal to zero if and only if w is less than n/k. Thus, the decoding algorithm yields all solutions at distances less than n/k from y if any such solution exists. It follows that (b) and (c) hold. If w is an integer greater than or equal to n/k, we can construct a vector z of weight w such that z has no block of k consecutive zeros. If w is less than or equal to the packing integer α for A, the coset A+z contains no vector other than z of weight less than $\alpha+1$. If y is in A+z, for no solution $y^{(i)}$ can $y-y^{(i)} = z$. Thus, the solution yielded by the algorithm must be at a distance of at least $\alpha+1$ from y, and (a) holds.

Numerical example. Let p = 2, n = 17, k = 8, and let ρ be the linear recursion

$$x_{j+8} = x_j + x_{j+3} + x_{j+4} + x_{j+5} \qquad \text{over } J_2.$$

Let y = (000 110 100 111 010 00). Using ρ, we compute:

$y^{(i)}$	$y-y^{(i)}$	i
(000 110 100 010 110 00)	(000 000 000 101 100 00)	0, 1, 13, 14, 15, 16;
(010 110 100 110 001 10)	(010 000 000 001 011 10)	2, 3;
(110 010 100 111 000 01)	(110 100 000 000 010 01)	4, 5;
(001 011 100 111 010 00)	(001 101 000 000 000 00)	6, 7, 8, 9, 10, 11;
(000 010 011 110 010 00)	(000 100 111 001 000 00)	12.

The algorithm yields the solution $y^{(0)}$, although $y^{(6)}$ is equally good. They are both minimal distance solutions by (c) above since β equals 3.

The decoding algorithms in the following sections satisfy the mi-

nimal distance criterion. If more than one element of minimal weight exists in a coset B, we allow for a choice between two alternatives: (a) the selection of a fixed but arbitrary error vector e for the coset; (b) the decision to detect rather than correct errors for the coset.

Codes for $p = 2$, $n = 73$, $k = 45$ or 46

For the case $p = 2$, a vector c in J_2^n can be described by giving the set Γ_c of coordinate indices j such that $c_j = 1$. Let $n = 73$, and let

$$\Gamma_c = \{0, 2, 10, 24, 25, 29, 36, 42, 45\}. \tag{20}$$

Then c is a vector of weight 9 in J_2^{73}. The polynomial $f_c(\lambda)$ divides the polynomial $1 - \lambda^{73}$ over J_2. The period of the linear recursion ρ_c therefore divides 73. This period must equal 73 since the period is greater than one and 73 is a prime. The linear recursion ρ_c codes the space J_2^{45} into a 45-dimensional cyclic subspace A of J_2^{73}. All the vectors of A are of even weight since c is of odd weight. The linear transformation R_c satisfies the requirements imposed on L above since all the columns cQ^i of its matrix are in $A^\perp$, since the columns include a basis of $A^\perp$, and since (as will be shown below) the columns can be described as the set of all elements of weight 9 in $A^\perp$. The transformation R_c will be called L or L^T in what follows, L being over J_2 and L^T being over the reals.

A 46-dimensional cyclic subspace A' can be obtained from A by adjoining the vector $h = (1, 1, \ldots, 1)$. If B is an A-coset, then the set union of B and B+h is an A'-coset. We call B and B+h dual A-cosets. The neighbors of an A-coset B differ in weight from B by plus or minus 1, and are thus always of different coset type than B. A neighboring coset of an

A'-coset B' can be of the same weight and even of the same coset type as B'.

Since an analysis of the linear space A' follows readily from an analysis of A, we will concentrate on A. The error-correcting properties of the two code spaces are summarized in Table A of the Appendix. Both codes correct all errors of weight 4 or less, and A [A'] corrects some errors through weight 9 [7]. The 2^{28} [2^{27}] cosets fall into 56 [28] coset types.

The analysis of the code space A is rather completely summarized in Table B. Superscripts have been affixed to the weight of xL in Table B, thus making it possible to use these weights as names of coset types or of coset type indicators. Note that if the weight of xL is even [odd], then all coordinates of xLL^T are even [odd]. The coset type indicator t can be written (redundantly) as a twelve-digit decimal number. For example, for coset type 41^1 we write t = (41;04,00,64,00,05) to indicate that xL has 41 coordinates equal to 1, and xLL^T has 4 coordinates equal to 1, 64 coordinates equal to 5, and 5 coordinates equal to 9.

Note that six pairs of coset types (for example, types 33^{1a} and 33^{1b}) in Table B have the same coset type indicator for each type in the pair. Nevertheless, a decoding algorithm can be defined that satisfies the minimal distance criterion on every A-coset and uses the 50 coset type indicators. It is given in Table C.

This algorithm is based on the fact that the coset type of a neighbor $B + e^{(i)}$ of a coset B is related (but not in a one-to-one way) to the value of the ith coordinate of xLL^T for x in B. To decode $x = x^{(0)}$ in J_2^{73}, the decoding procedure is recursively defined. Given $x^{(i)}$, $x^{(i)}$ is in the code

A if and only if $x^{(i)} L = 0$. If $x^{(i)}$ is not in the code, compute $x^{(i)} LL^T$ and the coset type indicator t. Use t to enter a list (see Table C) and there find instructions to operate in one of the following modes:

Mode I. Change all coordinates in $x^{(i)}$ that correspond to coordinates of xLL^T that equal one of the integers j, ...

Mode II. Change the first coordinate of $x^{(i)}$ for which the corresponding coordinate of xLL^T is equal to the integer j.

Mode III. Proceed as in Mode II, but if the coset type of $x^{(i+1)}$ does not have one of the coset type indicators t', ..., restore the coordinate of $x^{(i)}$ just altered and alter the next coordinate of $x^{(i)}$ for which the corresponding coordinate of $x^{(i)}LL^T$ is equal to the integer j.

These modes are listed in increasing order of operational difficulty. Mode I allows several simultaneous steps toward the code A. Mode II allows one certain step toward A. Mode III allows one step; if this turns out to be mistaken, it will be known at once. For completeness, we add Mode IV for the decision not to decode x. This may be a reasonable decision for cases where there is more than one element of minimal weight in the coset A + x.

In defining the decoding algorithm of Table C, we try to minimize the probability of entering Mode III at any stage of the descent to the code A. For example, consider coset type 32^4. In Table C we find that for x in a coset of type 32^4, the change of a coordinate of x such that the corresponding coordinate of xLL^T is equal to 6 [8] leads to a coset of type 29^1 $[25^2]$. We choose a Mode II operation for 32^4 with j equal to 6 rather than

to 8 since in the first case Mode III can never be entered in the decoding process; whereas in the second case, coset type 25^2 requires a Mode III operation.

For half of the 50 coset type indicators, decoding a vector x with such an indicator will not involve a Mode III operation at any stage. For the other 25 indicators, there is a positive probability (which may equal 1) that at least one Mode III operation will be involved in decoding a vector x with such an indicator. For only 3 of the indicators is a Mode III operation required at the first stage.

The algorithm given in Table C can be verified from the information given in Table B. It is best to do so for coset types ordered by weight, starting with the coset type whose unique member is the code A of weight 0. (The order used in Tables B through D emphasizes the relation between dual A-cosets.)

Table D gives one representative vector x of minimal weight for each coset type. The computations were done in part by computer. For any given vector x, the vectors xL and xLL^7 were printed, as were the coset type indicators for the neighbors $x+e^{(i)}$ $(0 \leq i \leq 72)$ of the vector x. The machine computations were done for a representative vector x of minimal weight in some coset of each new coset type found. The rest of the computations were done by hand. The number of cosets in each coset type was computed. A final check made certain that the total number of cosets was 2^{28}.

Since the columns of the matrix of L are in $A^{\perp}$, and since L is a symmetric linear transformation, the range of L is equal to $A^{\perp}$. Thus, Table B

incidentally yields a count of how many vectors of each possible weight occur in $A^{\perp}$. In particular, there are only 73 vectors of weight 9 in $A^{\perp}$, verifying a statement made above.

(Two codes for p = 2, n = 21, k = 11 or 12, have been analyzed[7] by the above methods, and may be of interest as an example suited to hand computation. For both the cases n = 21, 73, the basic matrix R_c may be retrieved as the incidence matrix[8] of a finite Desarguesian projective plane, and the permutation group G as the collineation group of this plane.)

Short decoding algorithm (n = 73, k = 45).

The following decoding algorithm for A does not make use of a list of coset type indicators. Decoding that satisfies the minimal distance criterion is accomplished for all cosets containing unique elements of minimal weight with the exception of the cosets of weight 7 belonging to coset types 41^3 and 45^1. The linear transformation $L = R_c$ is that of the preceding section.

To decode $x = x^{(0)}$, note that x is in A if and only if $xL = 0$. If $x^{(i)}$ is not in A, compute $x^{(i)}LL^T = z^{(i)}$. If more than 7 coordinates of $z^{(i)}$ are maximal, reject $x^{(i)}$. If 7 or fewer coordinates of $z^{(i)}$ are maximal, change all coordinates of $x^{(i)}$ that correspond to coordinates of maximal value in $z^{(i)}$. If $x^{(i)}$ is not in A for some i less than or equal to 3, reject.

The Golay code (p = 2, n = 23, k = 12) as a cyclic subspace

Marcel J.E. Golay[9] defined a code space in J_2^{23} having the property that all error vectors of weight 3 or less, and only these, can be corrected. Lowell J. Paige[10] showed the relation of an equivalent code space to the

4-transitive Mathieu group M_{23}. Prange[11] showed that two equivalent cyclic codes exist. One possible decoding algorithm for such a code is given here.

Let $\Gamma_c = \{0, 2, 4, 5, 6, 10, 11\}$. Then $f_d(\lambda) = (1 + \lambda)(f_c(\lambda))$ is a polynomial of degree 12 that divides $1-\lambda^{23}$ over J_2. The linear recursion ρ_d codes the space J_2^{12} onto a 12-dimensional cyclic subspace A' of J_2^{23}.

Let Q_1 be the coordinate permutation of J_2^{23} that takes the coordinate x_i of x into the coordinate x_{2i}, the coordinate indices being regarded modulo 23. The permutation Q_1 is of order 11. Let C be the set of 11 vectors obtained from the vector c by applying Q_1, and let S be the set of $253 = 23 \cdot 11$ vectors obtained from the set C by applying the coordinate permutation Q of (16). Let L be a matrix whose columns are the vectors in S.

Let A' be regarded as a code space. To decode x, note that x is in A' if and only if xR_c equals 0 or $(1, 1, \ldots, 1)$. If x is not in A', compute

$$z = xLL^T = \sum_{a \text{ in } C} xR_a R_a^T.$$

The coordinates of z will have two values, one of which (say j) occurs at most three times. Changing the coordinates of x that correspond to the coordinates of z that equal j, yields the unique element in A' that is at minimal distance from x.

There are only four coset types for A', one of each weight from 0 to 3.

It may be of interest to indicate the concrete form that the Mathieu group M_{23} takes in the present context. The following four permutations on coordinate indices are a redundant set of generators:

$Q = Q_0 = (0, 1, 2, \ldots, 21, 22)$,

$Q_1 = (1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12)(5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14)$,

$Q_2 = (2, 9, 8, 16, 6)(3, 13, 4, 12, 18)(5, 11, 20, 17, 15)(7, 22, 19, 21, 10)$,

$Q_3 = (2, 8)(3, 18)(5, 17)(6, 16)(7, 19)(10, 21)(11, 20)(12, 13)$.

These permutations satisfy the following relations:

$$I = Q_1^{-1} Q_0 Q_1 Q_0^{21} \quad Q_2^{-1} Q_1 Q_2 Q_1^6 = (Q_3 Q_2)^2 = Q_0^{23} = Q_1^{11} = Q_2^5 = Q_3^2,$$

where I is the identity permutation. We have followed K. D. Fryer[12] in the choice of permutations to display.

Conclusions

The two general types of decoding algorithms discussed in Part II are here referred to as minimal distance criterion (MDC) and block length criterion (BLC) algorithms. These terms would not be satisfactory in a wider context. We give some comparisons between the two types of algorithms.

The BLC algorithms are defined for all cyclic codes. The MDC algorithms are worked out code by code.

The general method used to obtain an MDC algorithm for a specific code also yields complete information on the error- correcting properties of the code. It seems that much more work would be involved in obtaining similar information for the BLC algorithms. Certainly the set of error vectors corrected under a BLC algorithm is not closed (in general) under the permutation group G.

The BLC algorithms do not satisfy the minimal distance criterion everywhere on J_p^n. The situation is particularly unfavorable in this respect if the ratio n/k is small.

Operationally, the BLC algorithms are simple and economical. The MDC algorithms for the specific codes discussed in Part II are also operationally rather good, but there is some reason to suppose that these particular codes are special in one way or another.

REFERENCES

1. DAVID SLEPIAN, A Class of Binary Signalling Alphabets. Bell System Tech. J. 35:203-234, 1956.
2. I.S. REED, A Class of Multiple-Error-Correcting Codes and the Decoding Scheme. IRE Trans. on Information Theory IT4:38-49, 1954.
3. PAUL B. YALE, Error Correcting Codes and Linear Recurring Sequences. Lincoln Laboratory Group Report 34-77, 1958.
4. NEAL ZIERLER, On a Variation of the First Order Reed-Muller Codes. Lincoln Laboratory Group Report 34-80, 1958.
5. A. ADRIAN ALBERT, Fundamental Concepts of Higher Algebra. Univ. of Chicago Press, 1956.
6. NEAL ZIERLER, Linear Recurring Sequences. Lincoln Laboratory Group Report 34-63, 1958.
7. EUGENE PRANGE, Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms. Tech. Note AFCRC-TN-58-156, Air Force Cambridge Research Center, 1958.
8. H.J. RYSER, Geometries and Incidence Matrices, Contributions to Geometry, Slaught Memorial Papers, Am. Math. Mthly Vol. 62, 1955.
9. MARCEL J.E. GOLAY, Notes on Digital Coding. Proc. IRE 37:657, 1949.
10. LOWELL J. PAIGE, A Note on the Mathieu Groups. Canadian J. Math. 9:15-18, 1957.
11. EUGENE PRANGE, Cyclic Error-Correcting Codes in Two Symbols. Tech. Note AFCRC-TN-57-103, Air Force Cambridge Research Center, 1957.
12. K.D. FRYER, A Class of Permutation Groups of Prime Degree. Canadian J. Math. 7:24-34, 1955.

APPENDIX

Note to Table B

Table B summarizes the basic information for the case $p = 2$, $n = 73$, $k = 45$. The second column of this table gives the weight of the vector xL for x in a coset of the given type. Superscripts are added to these weights so that they can be used as names of coset types and of coset type indicators. The coset types have been ordered by these weights to emphasize coset duality. If 25^2 is a type name, then 48^2 is the name for the corresponding dual type ($48 = 73 - 25$). The number of cosets in a type is equal to the number of cosets in the dual type.

The remaining entries in Table B can be explained by an example. Consider the coset type 25^2. The cosets of this type are of weight 5. Each contains more than one element of minimal weight (the contrary case being indicated by an asterisk). For any vector x in such a coset, the vector xLL^7 has 15 coordinates equal to 1, 40 coordinates equal to 3, and 18 coordinates equal to 5. In any coset of type 25^2, consider the set of 18 neighboring cosets obtained by altering a coordinate such that the corresponding coordinate of xLL^7 is equal to 5. This set of 18 cosets contains 3 cosets of type 24^1 and 15 cosets of type 24^2; the plus or minus signs indicate that the weight of these neighboring cosets is respectively one greater or one less than the cosets of the given type 25^2.

Note to Table C

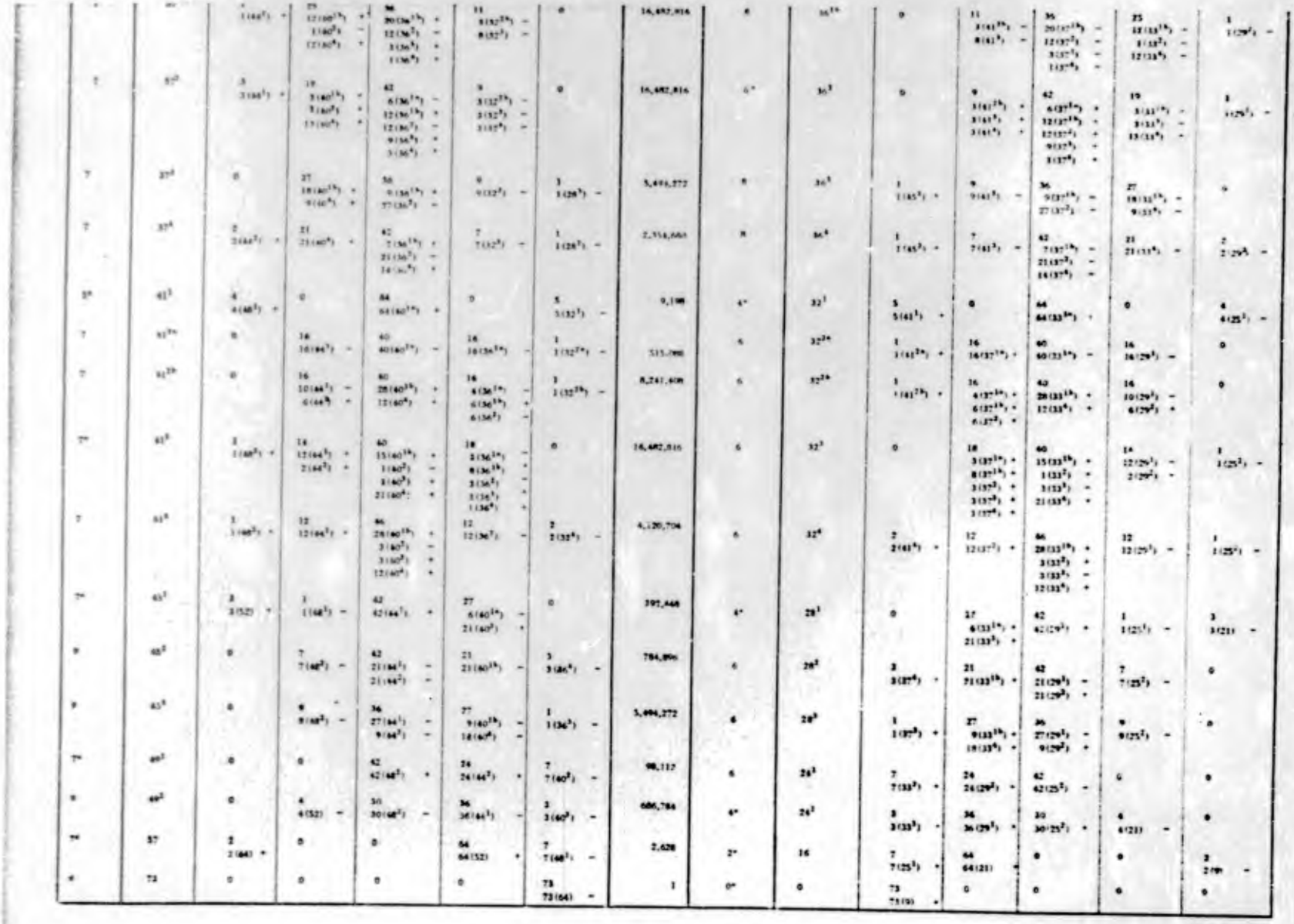
The columns headed t give the name of the coset type indicator. The columns headed w give the weight of the cosets for this coset type indicator. The columns headed + have an entry + if a Mode III operation never occurs during decoding that starts with a vector in a coset of the given type indicator. The columns headed Mode give the mode of operation for the given type indicator. The columns headed j and t' give the constants belonging to the given mode of operation for the given type indicator.

TABLE A. Error-correcting properties of two codes ($p = 2$, $n = 73$, $k = 45$ or 46)

	$k = 45$			$k = 46$		
Weight w	Cosets of Weight w	Cosets ÷ Vectors	Cosets Types	Cosets of Weight w	Cosets ÷ Vectors	Cosets Types
0	1	1.00	1	1	1.00	1
1	73	1.00	1	73	1.00	1
2	2,628	1.00	1	2,628	1.00	1
3	62,196	1.00	2	62,196	1.00	2
4	1,088,430	1.00	3	1,088,430	1.00	3
5	12,273,198	0.82	5	12,264,000	0.82	4
6	61,743,108	0.36	12	61,148,304	0.36	10
7	114,916,308	0.07	16	59,652,096	0.04	6
8	71,383,561	0.01	11	–	–	–
9	6,965,953	0.00	4	–	–	–

TABLE B. Type indicators and neighbor relations for coset types (p = 2, n = 73, k = 45)

Coset Weight	αL	αLL^T					Number of Cosets	Coset Weight	αL	αLL^T				
	1	1	3	5	7	9			1	0	2	4	6	8
1*	*	72 $72(1^{73})$ +	0	0	0	1 $1(0)$ −	73	*	64	1 $1(73)$ +	0	0	0	72 $72(1^{73})$ −
0*	21	21 $21(28^2)$ +	49 $49(24^2)$ +	0	3 $3(16)$ −	0	36,064	*	52	0	3 $3(37)$ −	0	49 $49(49^2)$ +	21 $21(41^2)$ −
0*	21^2	4 $4(32^2)$ +	64 $64(28^2)$ +	0	0	1 $1(16)$ −	6,132	0*	40^2	3 $3(37)$ +	0	0	64 $64(45^2)$ +	6 $6(41^2)$ −
5	29^2	15 $12(32^2)$ + $3(32^2)$ +	60 $4(28^2)$ + $56(28^2)$ +	38 $5(24^2)$ + $...(24^2)$ −	0	0	1,373,168	*	48^2	0	0	38 $3(49^2)$ − $35(49^2)$ +	60 $4(45^2)$ + $56(45^2)$ +	35 $12(41^2)$ − $7(41^2)$ −
5*	29^2	7 $3(36^{24})$ + $4(36^2)$ +	41 $1(32^{24})$ + $10(32^{24})$ + $24(32^2)$ + $6(32^4)$ +	22 $2(28^2)$ − $2(28^2)$ + $18(28^2)$ +	1 $1(24^2)$ −	0	8,241,408	*	44^2	0	1 $1(49^2)$ +	22 $2(45^2)$ − $2(45^2)$ + $18(45^2)$ +	41 $1(41^{24})$ − $10(41^{24})$ − $24(41^2)$ − $6(41^4)$ −	7 $1(37^{24})$ − $6(37^2)$ −
7	29^2	9 $7(36^{120})$ − $2(36^4)$ +	55 $21(32^{24})$ − $34(32^4)$ −	28 $7(28^2)$ − $21(28^2)$ −	1 $1(24^2)$ −	0	2,354,688	*	44^2	0	1 $1(49^2)$ −	28 $7(45^2)$ + $21(45^2)$ +	35 $21(41^{24})$ − $14(41^2)$ −	8 $7(37^{24})$ − $2(37^4)$ −
5*	33^{24}	5 $5(60^{24})$ +	28 $28(36^{24})$ +	36 $1(32^4)$ − $35(32^{24})$ +	4 $4(28^2)$ −	0	589,872	6*	40^{24}	0	4 $4(45^2)$ +	36 $1(41^4)$ − $35(41^{24})$ +	28 $28(37^{24})$ +	5 $5(33^{24})$ −
7	33^{24}	5 $2(60^{24})$ + $3(60^4)$ +	28 $7(36^{24})$ − $12(36^{24})$ + $3(36^4)$ − $6(36^4)$ +	36 $14(32^{24})$ − $15(32^4)$ − $7(32^8)$ −	4 $1(28^2)$ − $3(28^4)$ −	0	36,482,816	8	40^{24}	0	4 $3(45^4)$ + $3(45^8)$ +	36 $14(41^{24})$ − $15(41^4)$ − $7(41^8)$ −	28 $7(37^{24})$ − $12(37^{24})$ − $3(37^4)$ − $6(37^8)$ −	5 $3(33^{24})$ − $3(33^8)$ −
7	33^2	6 $6(60^2)$ +	28 $28(36^{24})$ +	42 $28(32^4)$ − $28(32^8)$ −	0	1 $1(24^2)$ −	590,784	6*	40^2	1 $1(49^2)$ +	0	42 $28(41^4)$ − $28(41^8)$ +	28 $28(37^{24})$ +	6 $6(33^4)$ −
5*	33^2	2 $2(60^2)$ +	36 $12(36^{24})$ + $24(36^4)$ +	30 $24(32^4)$ + $6(32^8)$ +	4 $4(28^2)$ −	1 $1(24^2)$ −	2,060,352	8	40^2	1 $1(49^2)$ +	4 $4(45^4)$ −	30 $24(41^4)$ − $6(41^8)$ −	36 $12(37^{24})$ − $24(37^4)$ −	2 $2(33^4)$ −
7	33^2	2 $2(60^{24})$ +	34 $5(36^{24})$ − $12(36^{24})$ + $12(36^4)$ − $3(36^4)$ + $3(36^8)$ +	30 $6(32^{24})$ − $21(32^4)$ − $3(32^8)$ −	5 $6(28^2)$ −	0	36,482,816	8	40^2	0	6 $6(45^4)$ +	30 $6(41^{24})$ − $21(41^4)$ − $3(41^8)$ −	34 $5(37^{24})$ − $12(37^{24})$ − $12(37^4)$ − $3(37^4)$ − $3(37^8)$ −	3 $3(33^{24})$ −
7	37^{24}	1 $1(44^2)$ +	25 $2(60^{24})$ − $16(60^4)$ + $3(60^8)$ + $4(60^8)$ +	36 $24(36^{24})$ − $12(36^4)$ −	11 $3(32^{24})$ − $4(32^{24})$ − $4(32^8)$ −	0	9,241,408	6*	36^{24}	0	11 $3(41^{24})$ + $4(41^{24})$ + $4(41^8)$ +	36 $24(37^{24})$ + $12(37^4)$ +	25 $2(33^{24})$ − $16(33^{24})$ + $3(33^8)$ − $4(33^8)$ +	1 $1(29^2)$ −
7	37^{24}	1 $1(44^2)$ +	25 $12(60^{24})$ + $1(60^4)$ − $12(60^8)$ +	36 $20(36^{24})$ + $12(36^4)$ − $3(36^8)$ + $1(36^8)$ +	11 $3(32^{24})$ − $8(32^8)$ −	0	36,482,816	8	36^{24}	0	11 $3(41^{24})$ − $8(41^8)$ −	35 $20(37^{24})$ − $12(37^4)$ − $3(37^8)$ − $1(37^8)$ −	25 $12(33^{24})$ − $1(33^4)$ − $12(33^8)$ −	1 $1(29^2)$ −
7	37^2	3 $3(44^2)$ +	29 $3(60^{24})$ + $3(60^4)$ + $23(60^8)$ +	42 $6(36^{24})$ − $12(36^{24})$ + $12(36^4)$ − $9(36^8)$ +	9 $3(32^{24})$ − $3(32^8)$ − $3(32^8)$ −	0	36,482,816	6*	36^2	0	9 $3(41^{24})$ + $3(41^8)$ + $3(41^8)$ +	42 $6(37^{24})$ + $12(37^{24})$ + $12(37^4)$ + $...$ −	19 $3(33^{24})$ − $3(33^8)$ − $13(33^8)$ +	3 $3(29^2)$ −



See note to Table B, page 25

TABLE C. Decoding algorithm (p = 2, n = 73, k = 45)

t	w	+	Mode	j	t'	t	w	+	Mode	j	t'
9	1*	+	I	9		64	8	+	II	8	
21	3*	+	I	7		52	8	+	II	2	
25^1	3*	+	I	9		48^1	6*	+	I	8	
25^2	5		III	5	24^2	48^2	8		II	8	
29^1	5*	+	I	7		44^1	8		II	8	
29^2	7		I	7		44^2	8	+	I	2	
33^1	5*, 7		II	7		40^1	6*, 8		II	8	
33^2	7		II	5		40^2	6*	+	I	8	
33^3	5*	+	I	7, 9		40^3	8	+	II	2	
33^4	7		II	7		40^4	8		II	8	
37^1	7,7		II	7		36^1	6*, 8		I	8	
37^2	7		II	7		36^2	6*	+	I	6	
37^3	7		I	9		36^3	8		II	6	
37^4	7		I	9		36^4	8		II	8	
41^1	5*	+	I	9		32^1	4*	+	I	8	
41^2	7, 7		I	9		32^2	6, 6		III	6	29^1
41^3	7*		III	7	36^2	32^3	6		I	8	
41^4	7	+	II	9		32^4	6	+	II	6	
45^1	7*	+	I	8		28^1	4*	+	I	6, 8	
45^2	9		II	5		28^2	6		II	6	
45^3	9		II	5		28^3	6		II	6	
49^1	7*	+	I	9		24^1	6		II	4	
49^2	9	+	II	9		24^2	4*	+	I	6	
57	7*	+	I	9		16	2*	+	I	8	
73	9	+	II	9		0	0*	+			

See note to Table C, page 25

TABLE D. Vector c of minimal weight in a coset of each coset type ($p = 2$, $n = 73$, $k = 45$)

Coset Type	Γ_c	Coset Type	Γ_c
0	0	64	0, 2, 10, 24, 25, 29, 36, 42
21	0, 1, 2	52	0, 1, 2, 10, 24, 25, 29, 36
25^1	0, 1, 5	48^1	0, 2, 10, 24, 25, 29
25^2	0, 1, 2, 3, 4	48^2	0, 2, 8, 10, 11, 22, 24, 50
29^1	0, 1, 2, 3, 5	44^1	0, 1, 2, 3, 10, 24, 25, 29
29^2	0, 1, 2, 3, 5, 8, 12	44^2	0, 1, 2, 3, 7, 10, 24, 25
33^{1a}	0, 1, 2, 5, 12	40^{1a}	0, 1, 2, 10, 24, 25
33^{1b}	0, 2, 8, 10, 14, 22, 24	40^{1b}	0, 1, 2, 3, 10, 11, 24, 25
33^2	0, 1, 2, 5, 9, 10, 37	40^2	0, 22, 23, 24, 42, 72
33^3	0, 1, 2, 5, 10	40^3	0, 1, 2, 4, 10, 24, 25, 29
33^4	0, 2, 5, 8, 10, 22, 24	40^4	0, 1, 2, 3, 6, 10, 24, 25
37^{1a}	0, 1, 2, 4, 10, 24, 25	36^{1a}	0, 2, 8, 10, 22, 24
37^{1b}	0, 1, 2, 3, 5, 13, 59	36^{1b}	0, 1, 2, 3, 10, 24, 25, 34
37^2	0, 2, 4, 8, 10, 22, 24	36^2	0, 22, 23, 24, 70, 72
37^3	0, 4, 22, 23, 24, 70, 72	36^3	0, 2, 8, 10, 11, 14, 22, 24
37^4	0, 17, 22, 23, 24, 70, 72	36^4	0, 2, 8, 10, 11, 16, 22, 24
41^1	0, 1, 5, 12, 18	32^1	0, 1, 5, 12
41^{2a}	0, 2, 8, 10, 22, 24, 25	32^{2a}	0, 1, 2, 3, 49
41^{2b}	0, 1, 2, 3, 10, 24, 25	32^{2b}	0, 1, 2, 3, 5, 59
41^3	0, 2, 8, 10, 11, 22, 24	32^3	0, 1, 24, 27, 42, 71
41^4	0, 4, 22, 23, 24, 43, 72	32^4	0, 1, 2, 5, 9, 10
45^1	0, 1, 2, 10, 24, 25, 29	28^1	0, 1, 2, 5
45^2	0, 20, 22, 23, 24, 45, 46, 47, 64	28^2	0, 1, 2, 3, 5, 8
45^3	0, 20, 22, 23, 24, 45, 46, 47, 69	28^3	0, 1, 2, 3, 4, 10
49^1	0, 22, 23, 24, 42, 68, 72	24^1	0, 1, 2, 3, 4, 9
49^2	0, 1, 2, 3, 10, 24, 25, 29, 36	24^2	0, 1, 2, 3
ρ^0	0, 2, 10, 24, 25, 29, 36	16	0, 1
91	0, 2, 10, 24, 25, 29, 36, 42, 68	0	Empty set

17

END