M. Hanan
F. P. Palermo

An Application of Coding Theory to a File Address Problem

Abstract: In this paper a file address problem is proved to be equivalent to a problem in coding theory. Results in coding theory can thus be used in solving this file addressing problem. It is shown, in particular, how the theory of Bose-Chaudhuri codes can be applied. A simple transformation from the input information to its address is given. This method can be easily implemented using either a computer or shift registers.

Introduction

The file addressing problem treated in this paper can be described as follows: An information item to be stored in a file is identified by a part of its content which is called the information item's *key*. The set K of keys is a set of k-tuples from an alphabet S which contains s symbols. This file address problem is to find a rule which enables one to calculate the location of an information item, called its *address*, from its key. The address is an a-tuple from the same alphabet S.

In some applications the keys to be stored form a subset N of the s^k elements of K and the number of keys in N is less than s^a, the number of storage locations. Since $a < k$, several keys may have the same address. The problem is to minimize this occurrence, i.e., this rule or function which maps the keys to the addresses should be a "randomizing" function so that the addresses cover the keys evenly. In order to accomplish this, some assumption must be made about the information to be stored. One such assumption (which appears to have merit and is the assumption under which the problem is solved in this paper) is that "clusters" of the keys are common. In clustering, the information tends to appear in groups where all members of a particular group are close together. If there are not too many clusters, a transformation which breaks up any clusters will be a "good randomizing" function. The function to be found, then, is one that breaks up these clusters, that is, assigns different addresses to keys which are close together using the definition of distances introduced by Hamming.

In this paper the above problem is treated as a problem in modern algebra and in this context, it is shown to be equivalent to a problem in coding theory. Some well-known results in coding theory are then used to give a simple construction of this function. The method has the added advantage that it is easily implemented.

Mathematical formulation

A set of items in an information system is identified by a set K of k-tuples from an alphabet S consisting of s symbols. These k-tuples are the *keys*. The storage locations A for the items, identified by their keys, are the *addresses* and are a-tuples from the same alphabet. The problem is to find an easily computable function which assigns to each key an address so that any two keys which are close together go into distinct addresses. The distance between two keys is defined to be the number of places in which the keys differ. Thus, the problem reduces to finding a maximum distance w and a function $T{:}K \longrightarrow A$ so that for all distinct u and v in K, $T(u) = T(v)$ implies the distance $d(u, v)$ between u and v is greater than w. Thus, T separates any two keys which are closer together than w units.

In this context, we shall make the following mathematical assumptions. The alphabet S is a ring with s elements. (This can be accomplished by choosing a ring with s elements and setting up a 1:1 correspondence between the elements of the ring and the alphabet.) Then the set of keys K is a free module over S of rank k. (Actually, one can think of K as being the set of all k-tuples over S.) The set of addresses A is a free module (all a-tuples) over S of rank a. The problem, then, is to find a module homomorphism $T{:}K \longrightarrow A$ which is onto A and with the property that for any u and v in K, if $T(u) = T(v)$, and $u \neq v$, then

the distance $d(u, v)$ between u and v is greater than w.

Recall that a code C of dimension $k - a$ over S is a free submodule of rank $k - a$ of the free module K of rank k over S. The weight of a code is the minimum weight of the nonzero code words of the code. The weight of a code word is the number of nonzero positions when the code word is represented as a k-tuple.

The problem considered above is equivalent to the following problem is coding theory: *Find a code C of dimension $k - a$ in the free module of rank k over the ring S whose weight w is maximum.* We will prove that this coding problem is equivalent to the file address problem which can be formulated as follows: *Find the maximum w and a module homomorphism T of K onto A with the property that $T(u) = T(v)$ and $u \neq v$ implies $d(u, v) > w$ where $d(u, v)$, the distance between u and v, is defined to be the weight $w(u - v)$ of $u - v$.*

First, we show that if C is a code of dimension $k - a$ in K whose weight is w, then we can define $T{:}K \rightarrow A$ which is onto A, with the property that $T(u) = T(v)$ and $u \neq v$ implies $d(u, v) > w$. Let M be the matrix of a basis for the null space of C. M is an $a \times k$ matrix. Define $T{:}K \rightarrow A$ by the formula $T(v) = M \cdot v^t$ for any v in K. Note v^t is the transpose of the row vector v and the multiplication is matrix multiplication. Then if $T(v_1) = T(v_2)$ with $v_1 \neq v_2$ and $d(v_1, v_2) \leq w$, we have $w(v_2 - v_1) \leq w$ and $T(v_2 - v_1) = 0$. Hence $v_2 - v_1 \in C$ and must either have weight greater than w or be zero. Since $w(v_2 - v_1) \leq w$, we conclude that $v_2 - v_1 = 0$ but this contradicts the hypothesis $v_1 \neq v_2$.

Conversely, let $T{:}K \rightarrow A$ be a module homomorphism with the property $T(v_1) = T(v_2)$ and $v_1 \neq v_2$ implies $d(v_1, v_2) > w$. Then let C be the kernel of T, i.e., $C = \{v \in K \mid T(v) = 0\}$. Then the weight of the code C is greater than w. For if $x \in C$, $x \neq 0$, $T(x) = 0 = T(0)$. Hence $d(x, 0) > w$, i.e., $w(x) > w$. Q.E.D. (For those readers accustomed to thinking in coding theory terms this mapping can be regarded as the mapping of a received message onto the set of "syndromes.")

Thus, we have established the equivalence of the file address problem to a problem in coding theory. Although the latter is a very difficult problem, some partial results have been obtained to determine maximum weight codes for the case where this ring S is a Galois field. If the ring S is a Galois field $GF(s)$, then K and A are, respectively, k and a dimensional vector spaces over S and the module homomorphism T is a linear transformation.

Application of coding theory to the problem

We shall now apply some results in coding theory to the file address problem. First, we shall describe the Bose-Chaudhuri codes, which are the best constructive codes known for large values of n (the length of code words), and then show how they can be applied to solve the file address problem.

Let the alphabet S be the Galois field of s elements $GF(s)$ and let the a-tuples of A be represented by polynomials in x with the coefficients over $GF(s)$. A cyclic code is an ideal I in the algebra of polynomials modulo $x^n - 1$. If $g(x)$ is the generator of I, then the code words are multiples of $g(x)$. Bose-Chaudhuri codes are a class of cyclic codes which are described in the following theorem.

Theorem.[2] *Let a be an element of $GF(q^m)$. The polynomial $g(x)$ over $GF(q)$ generates a code with minimum distance at least w if $a, a^2, \cdots, a^{w-1}$ are roots of $g(x)$. The length of the code n equals e, the order of a.*

The generator $g(x)$ can be constructed as follows: If $m_i(x)$ is the minimum polynomial over $GF(q)$ of a^i, then

$$g(x) = LCM[m_1(x), m_2(x), \cdots, m_{w-1}(x)].$$

If the degree of $g(x)$ is c, then the number of check symbols is c and the number of information symbols is $n - c$.

Clearly, the length of the code n corresponds to the length of the key k and the number of check symbols c corresponds to the length of the address a. In any practical problem, we could take $n \geq k$. Hence, for certain values of k, c, and w we can apply the theory of Bose-Chaudhuri codes to the file address problem.

Since the file address problem is essentially the inverse of the coding theory problem, the application of the Bose-Chaudhuri theorem is rather simple. We represent the k-tuples (i.e., the keys) as polynomials $K(x)$ of degree less than or equal to $k - 1$ over $GF(s)$. We divide $K(x)$ by $g(x)$ and examine the remainder $A(x)$. The remainder $A(x)$, which is a polynomial of degree less than or equal to $a - 1$, represents the a-tuple of the address of the key $K(x)$. In this way, all keys which are distance w or less apart have distinct remainders, that is, distinct addresses.

The implementation of this method is simple. The division $K(x)$ by $g(x)$ can be performed easily by a computer. (Since we are only interested in the remainder, it may be easier to program the computer to find $K(x)$ modulo $g(x)$.) In fact, a large scale computer is not necessary. The division of $K(x)$ by $g(x)$ can easily be performed by an a-stage shift register.

In most practical problems, the alphabet S consists of 2^p elements. In this case, the Reed-Solomon codes[3] (which are a special case of Bose-Chaudhuri codes with $m = 1$) are of particular interest.

If a is a primitive element of $GF(2^p)$, that is $a^{2^p-1} = 1$ but $a^i \neq 1$ for $0 < i < 2^p - 1$, then the code generated by $g(x) = (x - a)(x - a^2) \cdots (x - a^w)$ has a minimum distance $w + 1$ and length $n = 2^p - 1$. Since the degree of $g(x)$ is w, the length of the address is also w.

Conclusions

The equivalence of the file address problem and the coding theory problem is proved. It is shown, in particular, how the theory of Bose-Chaudhuri codes can be applied. A simple transformation from the input information to its address is given. This method is easily implemented using either a computer or shift registers.

Acknowledgments

The authors wish to thank G. Schay for the many stimu-

lating discussions during which independent proofs of the equivalence of the file address problem and the coding problem were obtained. Our proof is given above while G. Schay's proof appears in this issue of the Journal.[4]

Note added in proof

S. Muroga[5] has also considered the notion of using group codes for file addressing. His work was called to the attention of the authors after the present manuscript was written.

References and footnotes

1. For definitions of the mathematical terms used here, see C. Chevalley, *Fundamental Concepts of Algebra,* Academic Press, New York, 1956, ch. 3.
2. W. W. Peterson, *Error Correcting Codes,* John Wiley and Sons, New York, 1961, page 162.
3. *Ibid.,* p. 168.
4. G. Schay and N. Raver, this issue, page 121.
5. S. Muroga, unpublished report, April 1961.

Received Nov. 1, 1962