inequality in information theory, and where $I(\zeta,\eta)$ is the mutual information betwen ζ and η. Using a slight modification of a technique of Lipster [7], we obtain

$$I(X,\xi_0^t) = I(X,Y) + \frac{1}{2\sigma_N^2} \cdot \int_0^t \left\{ E\left[A_s^2(X,\xi_0^s)\right] - E\left[\hat{A}_s^2(\xi_0^s)\right] \right\} ds \leqslant \tfrac{1}{2} \log \frac{\sigma_X^2+\sigma_U^2}{\sigma_X^2\sigma_U^2} + \frac{E_0 t}{2\sigma_N^2} \quad (27)$$

with $\hat{A}_t(\xi_0^t) = E[A_t(X,\xi_0^t)|\xi_0^t]$. Hence from (26) and (27) we have

$$P_t \geqslant \frac{\sigma_X^2\sigma_U^2}{\sigma_X^2+\sigma_U^2} \exp(-\beta t).$$

Therefore (24) actually attains this lower bound and thereby is optimal over all admissible nonlinear codings.

References

[1] T. Berger, *Rate-Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[2] S. Ihara, "Optimal coding in white Gaussian channel with feedback," in *Proc. 2nd Japan-USSR Symp. Probability Theory, (Lecture Notes in Math. 330)*, New York: Springer-Verlag, 1973, pp. 120–123.

[3] H. Sakai, T. Soeda, and H. Tokumaru, "A note on feedback communication with noisy side information at the receiver," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 384–386, May 1977.

[4] A. D. Wyner and J. Ziv, "The rate-distortion function for the source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 1–10, Jan. 1976.

[5] A. Segall, "Stochastic processes in estimation theory," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 275–286, May 1976.

[6] A. N. Shiryayev, "Statistics of diffusion type processes," in *Proc. 2nd Japan-USSR Symp. Probability Theory, (Lecture Notes in Math. 330)*, New York: Springer-Verlag, 1973, pp. 397–411.

[7] R. Sh. Lipster, "Optimal encoding and decoding for transmission of a Gaussian Markov signal in a noiseless-feedback channel," *Problems of Information Transmission*, vol. 10, pp. 279–288, 1974.

On the Inherent Intractability of Certain Coding Problems

ELWYN R. BERLEKAMP, FELLOW, IEEE, ROBERT J. McELIECE, MEMBER, IEEE, AND HENK C. A. VAN TILBORG

***Abstract*—The fact that the general decoding problem for linear codes and the general problem of finding the weights of a linear code are both NP-complete is shown. This strongly suggests, but does not rigorously imply, that no algorithm for either of these problems which runs in polynomial time exists.**

I. Introduction

Let C be an (n,k) binary linear code which is to be used on a binary symmetric channel. If y is the received word, then the syndrome is $s=yH$, where H is the $n\times(n-k)$ parity-check matrix, and the receiver's best estimate of the transmitted codeword is $x=y+z_0$, where z_0 is a minimum-weight solution to the equation $s=zH$ [2, Ch. 1]. Unfortunately for a general code it is apparently necessary to search through the entire set of 2^k solutions to $zH=s$ in order to find one of least weight. In other words, the best general algorithm known which, given a matrix H and a vector s, finds a minimum-weight solution to $zH=s$, has a running time which is an exponential function of the number of inputs. The discovery of an algorithm which runs significantly faster than this would be an important achievement.

Manuscript received March 2, 1977; revised September 14, 1977. The contribution of McEliece and van Tilborg to this correspondence represents one phase of research sponsored by the National Aeronautics and Space Administration under Contract No. NAS 7-100, carried out at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA.

E. R. Berlekamp is with the Department of Computer Science and Mathematics, University of California, Berkeley, CA 94720.

R. J. McEliece is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91103.

H. C. A. van Tilborg is with the Department of Mathematics, Technological University of Eindhoven, Eindhoven, The Netherlands.

The code C is the set of vectors satisfying $xH=\mathbf{0}$. In many applications it is desirable to know, for a given weight w, whether C contains any words of weight w, i.e., whether there is a vector of weight w satisfying $xH=\mathbf{0}$. Again, the best general algorithm known for deciding this requires an exponential search, in this case through all 2^k codewords, and a faster algorithm would be highly desirable.

In this correspondence we will show that it is unlikely that anyone will ever discover substantially faster algorithms for either of these problems, by showing that both of them belong to a large class of difficult combinatorial problems, the NP-complete problems.

II. NP-Complete Problems

The following discussion will describe in heuristic language a set of results that can be made quite precise. We refer the interested reader to [1, Ch. 10], [3], [4], or [5] for details.

The class P is defined to be the set of computational problems which can be solved by an algorithm which is guaranteed to terminate in a number of steps bounded by a polynomial in the length of the input. Thus P corresponds to the class of *polynomial-time* algorithms. Problems in the class P are generally regarded to be tractable; conversely those not in P are considered intractable. The class P includes such problems as solving linear equations, finding the minimum cut in a flowgraph, certain scheduling problems, etc.

The class NP is defined to be the set of computational problems which can be solved by a backtrack-search algorithm, where the depth of search is bounded by a polynomial in the length of the input. Alternately, NP is the set of problems solvable by a nondeterministic algorithm whose running time is bounded by a polynomial in the length of the input. A nondeterministic algorithm is one which, when confronted with a choice between two alternatives, can create two copies of itself and simultaneously follow the consequences of both courses. This repeated splitting may lead to an exponentially growing number of copies; the algorithm is said to solve the given problem if any one of these copies produces the correct answer. This description explains the notation: NP corresponds to the class of *nondeterministic polynomial-time* algorithms.

The class NP is quite extensive; it contains many classic combinatorial problems, such as the traveling salesman's problem, the 0–1 integer programming problem, and the Hamiltonian circuit problem. It also contains the two coding problems described in Section I, as the reader can easily verify for himself.

The class NP clearly contains the class P as a subclass: $\mathrm{NP} \supseteq P$. It is conversely intuitively evident that NP is "much larger" than P; however, no one has yet succeeded in proving this and the query $\mathrm{NP} \neq P$? is currently one of the central problems of computer science. Recently a circle of results has been developed that strongly suggests, though does not rigorously imply, that $\mathrm{NP} \neq P$. We now describe these results.

In 1971 Cook (see [1, Theorem 10.3]) proved that a certain problem in NP (called the *satisfiability* problem) has the following curious property. Any problem (p) in NP can be reduced to the satisfiability problem, in the sense that if a polynomial-time algorithm could be found for the satisfiability problem, then that algorithm could be modified to yield a polynomial-time algorithm for problem (p). Thus while it is possible that satisfiability *might* possess a polynomial-time algorithm, if it did, then so would the traveling salesman's problem, the integer programming problem, and indeed any problem in NP. But researchers have worked on these NP problems for many years without finding a polynomial-time algorithm for *any* of them. This is strong evidence that satisfiability does not possess a polynomial-time algorithm and that $\mathrm{NP} \neq P$.

In a later paper Karp [4] reversed things and showed that the satisfiability problem can itself be reduced in the sense described above to many other (21, to be exact) NP-problems. Thus if any of these NP-problems possesses a polynomial time algorithm, then so does every NP-problem, and hence $\mathrm{NP} = P$. NP-prob-

0018-9448/78/0500-0384$00.75 ©1978 IEEE

lems with this property are now called *NP-complete problems*. The forthcoming book by Garey and Johnson [3] contains literally hundreds of problems known to be NP-complete; and if the "obvious" assertion NP$\neq P$ is true then no NP-complete problem can have a polynomial-time algorithm.

In the next section we shall show that the two decoding problems described in Section I are NP-complete by reducing to them a known NP-complete problem.

III. New NP-Complete Problems Related to Coding

Here are two decision problems which we will show are NP-complete, and which correspond to the coding problems given in Section I.

A. COSET WEIGHTS

Input: A binary matrix A, a binary vector y, and a nonnegative integer w.

Property: There exists a vector x of Hamming weight $\leqslant w$ such that $xA=y$.

B. SUBSPACE WEIGHTS

Input: A binary matrix A and a nonnegative integer w.

Property: There exists a vector x of Hamming weight w such that $xA=\mathbf{0}$.

In the theory of NP-completeness, problems are always stated in this format. The input must be encoded into a binary string of length N, for example, and fed into a deterministic computing machine, which outputs 0 when the input does not have the property and 1 when it does. If there is such a device for which the time for this computation is bounded by a polynomial in N, the problem is in P; otherwise it is not.

The SUBSPACE WEIGHTS problem corresponds exactly to the problem of deciding whether a linear code has a vector of a given weight, but the exact connection between the minimization problem cited in Section I (i.e., to find the minimum weight among all solutions to $xA=y$) and the decision problem COSET WEIGHTS is perhaps unclear. Note, however, that a polynomial-time algorithm for the minimization problem immediately implies a polynomial-time algorithm for the decision problem (since there exists a solution with weight $\leqslant w$ iff the minimum weight solution has weight $\leqslant w$), and conversely (by running the decision problem for $w=0,1,2,\cdots$ until the first affirmative answer is obtained, the minimum weight solution will be found).

It is easy to see that both COSET WEIGHTS and SUBSPACE WEIGHTS are in NP. We will now show they are NP-complete by reducing the following combinatorial decision problem, which is known to be NP-complete (it is problem 17 on Karp's list in [4]), to each of them.

C. THREE-DIMENSIONAL MATCHING

Input: A subset $U\subseteq T\times T\times T$, where T is a finite set.

Property: There is a set $W\subseteq U$ such that $|W|=|T|$, and no two elements of W agree in any coordinate.

The elements of U are ordered triples from the set T. As an illustration, consider the following two examples, each with $T=\{1,2,3,4\}$ and $|U|=6$:

	Example 1	Example 2
1)	(1, 2, 1)	(1, 2, 1)
2)	(1, 3, 2)	(1, 3, 2)
3)	(2, 1, 4)	(2, 1, 4)
4)	(2, 2, 3)	(2, 2, 3)
5)	(3, 1, 1)	(3, 2, 1)
6)	(4, 4, 4)	(4, 4, 4).

The first set of triples has the desired property, the four triples numbered 2, 4, 5, and 6 being a "matching". The second set does not, as the reader may easily verify.

For our purpose it is convenient to encode the set U of triples into a $|U|\times 3|T|$ binary incidence matrix, in which each row corresponds to one of the triples and has weight three, with each 1 corresponding to a component of the triple. For example, the first set above yields the following 6×12 incidence matrix.

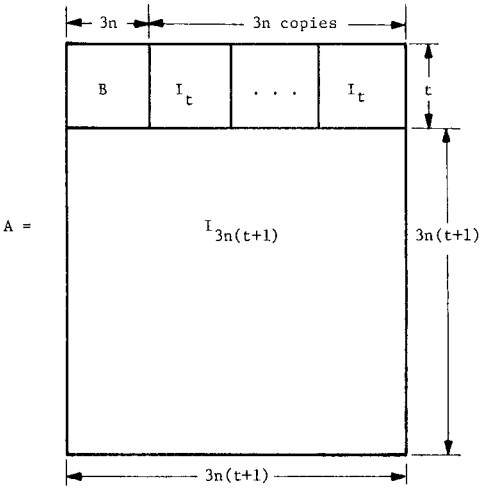
	1	2	3	4	1	2	3	4	1	2	3	4
121:	1	0	0	0	0	1	0	0	1	0	0	0
132:	1	0	0	0	0	0	1	0	0	1	0	0
214:	0	1	0	0	1	0	0	0	0	0	0	1
223:	0	1	0	0	0	1	0	0	0	0	1	0
311:	0	0	1	0	1	0	0	0	1	0	0	0
444	0	0	0	1	0	0	0	1	0	0	0	1

In terms of this matrix, a solution to the THREE-DIMENSIONAL MATCHING problem is the existence of $|T|$ rows whose mod 2 sum is $111\cdots 1111$.

We will now show that our two coding problems are also NP-complete by reducing THREE-DIMENSIONAL MATCHING to each of them. Here are the reductions.

THREE-DIMENSIONAL MATCHING can be reduced to COSET WEIGHTS: Suppose we had a polynomial-time algorithm for COSET WEIGHTS. Now given an input $U\subseteq T\times T\times T$ for the THREE-DIMENSIONAL MATCHING problem, let A be the $|U|\times 3|T|$ incidence matrix described above. Then running the putative COSET WEIGHTS algorithm with inputs A, $y=(111\cdots 111)$, $w=|T|$, we would discover, in polynomial time, whether the matching existed. That is, a polynomial-time algorithm for COSET WEIGHTS implies a polynomial-time algorithm for THREE-DIMENSIONAL MATCHING, which in turn implies a polynomial-time algorithm for every NP-problem. This proves that COSET WEIGHTS is NP-complete.

THREE-DIMENSIONAL MATCHING can be reduced to SUBSPACE WEIGHTS: The idea is again to construct an A for the SUBSPACE WEIGHTS algorithm from the triple incidence matrix, but here the construction is more complicated. Let B be the triple incidence matrix described above, and assume its dimensions are $t\times 3n$. The matrix A is formed from B as shown below.



In words, A is a $(3nt+3n+t)\times(3nt+3n)$ matrix with the top t rows consisting of B followed by $3n$ copies of the $t\times t$ identity matrix, and the lower $3nt+3n$ rows forming a large identity matrix.

Let us assume that we have a polynomial-time algorithm for SUBSPACE WEIGHTS. If we apply this algorithm to the matrix A, and set $w=3n^2+4n$, we will find out in polynomial time whether the original set of triples has a matching. For suppose x is a vector with $xA=\mathbf{0}$. Let x_0 denote the vector of length t formed from the first t components of x, let $y=x_0B$, and let x_1 be the vector formed from the last $3n(t+1)$ components of x. Then clearly $|x_1|=|y|+3n|x_0|$, where $|x|$ denotes the Hamming weight of x. Adding $|x_0|$ to both sides of this equation,

$$|x|=|y|+(3n+1)|x_0|.$$

Since $0\leqslant|y|\leqslant 3n$, this means that $|x_0|$ and $|y|$ can be uniquely determined from $|x|$: they are the remainder and quotient when $|x|$ is divided by $3n+1$. In particular, if $|x|=3n^2+4n$, we get $|x_0|=n$ and $|y|=3n$. So the code with parity-check matrix A has

a word of weight $3n^2+4n$ if and only if the set of triples admits a matching. This shows that SUBSPACE WEIGHTS is NP-complete, as asserted.

IV. Concluding Remarks

1) One can argue that for the coding problem, one always knows that the parity-check matrix has full rank. Thus one might for example wish to consider the following "easier" version of SUBSPACE WEIGHTS.

Input: A binary matrix A with linearly independent columns and a nonnegative integer w.

Property: There exists a vector x of weight w such that $xA=0$.

It is, however, quite easy to show that this problem is also NP-complete. The trick is to find (in polynomial time, by standard methods of linear algebra) a subset of columns forming a basis for the column space of A. If A' denotes the matrix formed from these columns, then $xA'=0$ will have a weight w solution iff $xA=0$ does. Similar considerations show that one may assume that the matrix COSET WEIGHTS has full column rank without disturbing its NP-completeness.

2) The corresponding problems stated in terms of the generator matrix are also NP-complete, since one can go from the parity-check to the generator matrix (or vice versa) in polynomial time.

3) It would be very desirable to replace the phrase "of weight w" in SUBSPACE WEIGHTS with the phrase "of weight $\leqslant w$," for (see the discussion in Section III) this would show that the problem of finding the minimum weight of a general code is NP-complete. While we conjecture that this is the case, we have not been able to prove it, and propose it as a research problem.

Acknowledgment

We wish to thank one of the referees for pointing out to us the relation between the "weight $\leqslant w$" decision problem and the minimum weight problem for linear codes.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Analysis and Design of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.

[2] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: Freeman, 1978.

[4] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. New York: Plenum, 1972, pp. 85–103.

[5] R. M. Karp, "*On the computational complexity of combinatorial problems,*" *Networks* 5, pp. 45–68, 1975.

Interlacing Properties of Shift-Register Sequences with Generator Polynomials Irreducible over GF(p)

FRANZ SURBÖCK AND HANS WEINRICHTER

***Abstract*—Interlacing properties of shift-register sequences with generator polynomials irreducible over GF(p)—herein called elementary sequences—are analyzed. The most important elementary sequences are maximal-length sequences (m-sequences). If the period q of an elementary sequence is not prime, the sequence can always be constructed by interlacing shorter elementary sequences of period q_i, provided q_i divides q. It is proved that all interlaced elementary sequences are generated by one and the same irreducible polynomial. Some relationships between equal-length elementary sequences are derived, including some rather unexpected crosscorrelation properties. As an example of an application of the theory, a new time-division multiplex technique for generating high-speed m-sequences is presented.**

Manuscript received July 7, 1977; revised September 23, 1977. Portions of this correspondence were presented at the IEEE International Symposium on Information Theory, Ronneby, Sweden, June 21–24, 1976.

The authors are with the Institut für Niederfrequenztechnik, Technische Universität Wien, Vienna, Austria.

I. Introduction

Shift-register sequences with elements from GF(p) have proved useful in coding theory and in many other applications. The properties of these sequences have been studied extensively, and many interesting results have been found [1]–[3]. Nevertheless, there are still unsolved problems. In the binary case ($p=2$), for instance, it is difficult to decide which of all the possible maximum-length sequences of a given period simulates a truly random binary sequence best.

In this paper, we present some new properties of the time-division multiplex structure of shift-register sequences with a nonprime period. In particular, we shall be concerned mainly with shift-register sequences with generator polynomials that are irreducible over GF(p). Such sequences will be called *elementary sequences*. The most important class of elementary sequences are *maximum-length sequences* (*m-sequences*) with primitive generator polynomials. In the sequel, a method will be presented to further decompose a certain type of elementary sequence. It will be shown that every elementary sequence with a nonprime period can be constructed by interlacing several shorter elementary sequences, all with the same irreducible generator polynomial.

The concept of interlacing sequences is closely related to the more popular concept of sequence sampling. It is well-known that uniform sampling of an m-sequence will yield either a shifted version of the same m-sequence again or another m-sequence of the same length, if the sample interval is prime to the period of the m-sequence (proper decimation). In contrast, this paper is concerned with improper decimation only, i.e., the sample interval is assumed to be a factor of the period of the sampled sequence. When an elementary sequence is sampled in this way, the resulting sample sequences may be different from each other depending on their sample phase. Lempel and Eastman [4] pointed out that each of these sample sequences can be generated by a linear shift-register with the same, or even a smaller, number of stages as the original shift-register. They described how to generate any given linear shift-register sequence of maximal length at a rate considerably faster than the shift pulse rate generated by interlacing these sample sequences. In this paper, we will show that these sample sequences all have the same generator polynomial and thus can be generated by the same feedback shift register. Similar concepts of interlacing m-sequences have already been used in several cases [5]–[11]. For instance, they have proved valuable in describing the response of digital data scramblers to special m-sequences [5], and in generating high-speed m-sequences by a certain time-division multiplex technique [6], [10], [11].

II. Elementary Sequences Generated by Polynomials Irreducible over GF(p)

The *D-transform* of a sequence $f_0, f_1, f_2, \cdots$, consisting of elements of GF(p) with p prime, is defined by

$$F(D)=\sum_{k=0}^{\infty} f_k D^k. \tag{1}$$

Then any sequence generated by a linear feedback shift-register with a feedback polynomial $N(D)$ with coefficients in GF(p) can be written as

$$F(D)=\frac{H(D)}{N(D)}, \tag{2}$$

where $H(D)$ depends on the initial state of the register. If $N(D)$ is irreducible over GF(p), $F(D)$ will be called an elementary sequence. If $N(D)$ factors into mutually distinct irreducible polynomials $N_1(D)\cdot N_2(D)\cdots$, then the sequence $F(D)$ is merely the sum of elementary sequences with the generator polynomials $N_1(D), N_2(D), \cdots$, as follows from the partial-fraction expansion of (2). If $N(D)$ factors into several irreducible polynomials which are not all mutually distinct, the partial-fraction expansion of (2) is still applicable, resulting in a superposition of elementary sequences some of which may be spread in time [5], [7]. Therefore, our analysis will be concerned primarily with

0018-9448/78/0500-0386$00.75 © 1978 IEEE