Discrete Applied Mathematics 33 (1991) 43-60
North-Holland

New approaches to reduced-complexity decoding

John T. Coffey

University of Michigan, Ann Arbor, MI 48109-2122, USA

Rodney M. Goodman

California Institute of Technology, Pasadena, CA, USA

Patrick G. Farrell

University of Manchester, Manchester, UK

Received 10 August 1989
Revised 11 January 1990

Abstract

Coffey, J.T., R.M. Goodman and P.G. Farrell, New approaches to reduced-complexity decoding, Discrete Applied Mathematics 33 (1991) 43-60.

We examine new approaches to the problem of decoding *general* linear codes under the strategies of full or bounded hard decoding and bounded soft decoding. The objective is to derive enhanced new algorithms that take advantage of the major features of existing algorithms to reduce decoding complexity. We derive a wide range of results on the complexity of many existing algorithms. We suggest a new algorithm for cyclic codes, and show how it exploits all the main features of the existing algorithms. Finally, we propose a new approach to the problem of bounded soft decoding, and show that its asymptotic complexity is significantly lower than that of any other currently known general algorithm. In addition, we give a characterization of the weight distribution of the average linear code and thus show that the Gilbert-Varshamov bound is tight for virtually all linear codes over any symbol field.

1. Introduction

Much attention has been paid to the important and difficult problem of finding decoding algorithms for *general* linear codes [1-18]. These algorithms aim to provide efficient decoders for medium length (80-200) block codes, with the ability to take account of reliability information from the channel. Such decoders can be used in packet based systems or as inner decoders in concatenated coding schemes. In addition to the applications to communications, the analysis of general decoding

algorithms has an application to public key cryptography, as it bears directly on the security of a proposed public key cryptosystem based on the hardness of the decoding problem.

The attention given to the problem has produced a variety of approaches. Almost without exception, however, there has been no attempt to relate proposed new algorithms to previous work in the field, and little attempt to provide rigorous and complete results on the complexity of the algorithms. In this paper we aim to remedy this omission in a number of ways. We provide accurate and complete analyses of the complexities of many existing algorithms. We propose a decoding algorithm for cyclic codes which exploits all the main features of current approaches, and suggest an avenue for generalizing the procedure to all linear codes. Finally, we propose a new algorithm applicable specifically to the (most important) case of bounded soft decision decoding. We prove that this combined algorithm has a complexity significantly less than that of any previously proposed algorithm.

In Section 2, we introduce some decoding terminology and discuss the practicality and applications of these general algorithms. In Section 3, we give rigorous and mostly complete analysis of the complexity of various current approaches. Many of the results are new, such as the analysis of covering polynomials, virtually all the extensions to the cases of bounded soft decoding and decoding over nonbinary symbol fields, and the determination of the weight distribution of the average linear code. Other results are formalizations of existing estimates. The analysis is straightforward; the goal once again is to formalize and compare the results so that combined algorithms can be formed. In Section 4, we propose an algorithm for cyclic codes based on the process of continued division. We suggest an extension (without asymptotic analysis of complexity, however) to general linear codes. In Section 5, we propose a new algorithm for bounded soft decision decoding and show that it has significantly lower complexity than any other known algorithm for virtually all codes.

2. Background

First, we specify the terminology to be used throughout the paper in discussing the decoding strategies. *Bounded hard distance decoding* involves decoding the received word to the nearest codeword, provided the codeword is at distance no greater than t. *Complete hard decision decoding* involves decoding every received word to a nearest codeword. In *bounded soft decoding*, we assume that the receiver specifies the received word as a collection of n real numbers r_i. The *soft distance* between words $c \in C$ and $r = (r_0, \ldots, r_{n-1})$ is defined to be $\sum_{0 \le i \le n-1} |c_i - r_i|$. Chase [8] shows that this is equivalent to maximum likelihood decoding over the additive white Gaussian noise channel, although it is not necessarily optimal for general channels. Decoding to the bounded soft distance is defined to be the decoding algorithm whereby the received word is decoded to the nearest codeword, provided

that the codeword is at soft distance no more than t. This strategy has a performance close to that of maximum likelihood decoding [8].

The decoding problem has long been known to be difficult. The complete hard decision decoding problem has been shown to be NP-complete [5], which "strongly suggests, but does not rigorously imply that no polynomial time algorithm exists for the procedure". Although such results do not exist for the bounded hard and bounded soft decision problems, it is widely believed that there is no polynomial time decoding algorithm for those cases either. As an exponential algorithm is usually taken to be impractical, it is necessary to give some justification for the investigation of such algorithms.

Our justification is one commonly used in such cases: we are interested in solving relatively short instances of the problem — we aim to produce decoders for medium length block codes (up to $n=200$). The trivial algorithms of searching through all codewords or through all syndromes have complexities 2^{nR} and $2^{n(1-R)}$ respectively, an impractical proposition unless the block length of the code is extremely short — $n\approx 30$. However, an algorithm with complexity $2^{nR/m}$, while still exponential, should in theory be practicable for codes m times as long. We demonstrate later that it is possible to achieve at least $m\approx 5$ for bounded soft decoding and $m\approx 9$ for bounded hard decision decoding, for rate 1/2 codes in each case; thus codes of much higher lengths can in principle be decoded.

Furthermore, in coding we are concerned primarily with the decoder error probability, and this *declines* exponentially with increasing block length. This suggests that if the complexity rises at a slower rate than the error probability declines, then decoding is a practical proposition. The situation is exactly the same as for Viterbi decoding of convolutional codes. The algorithm is exponential in the constraint length of the code, which limits the possible constraint lengths that can be used. On the other hand, significant coding gains are achievable with the short constraint length codes used.

Because we are working with exponential algorithms, it is more convenient to measure complexity by the logarithm of the complexity. Given a decoding algorithm of complexity $M(C)$, we define the *complexity coefficient* $F(R)$ to be $(1/n)\log_q(M(C))$, i.e., $M(C)=q^{nF(R)}$.

These algorithms will also be applicable to the McEliece public key cryptosystem [28]. For parameters n, k, and t, the cryptosystem has as private key a $k\times n$ generator matrix G' for a t-error-correcting Goppa code, an $n\times n$ permutation matrix P, and a $k\times k$ nonsingular matrix S. The public key is the $k\times n$ matrix $G=SG'P$. The messages are binary words of length k. To encrypt a message m, we form $c=mG+e$, where e is a randomly chosen word of length n and weight at most t. To decode, we form $c'=cP^{-1}$, apply the algebraic decoding algorithm for the Goppa code to find m' such that $d(m'G',c')\leq t$, and then we have $m=m'S^{-1}$. To crack the system given only the public key, we apparently have to use a procedure capable of bounded hard distance decoding for any linear code; thus the security of the cryptosystem depends on the complexity of these algorithms.

3. Information set and progressive approaches

We examine a number of approaches of two distinct families. This is not an exhaustive survey of general decoding algorithms — for example, we omit any discussion of threshold and majority logic decoding [25]. Instead, we concentrate on determining the exact complexity of the various procedures for bounded hard or soft decision and complete hard decision decoding.

We state, without proof, two results which will be useful later. The first is from our [11]. The second is a recent result of Blinovskii [6]. We follow with an important, and to our knowledge previously unproven result on the weight distribution of the average linear code.

(1) [11] *For any fixed R and α satisfying $0<\alpha<1$, $0<R<1$, and for all sufficiently large values of n, virtually all linear $(n, \lfloor nR \rfloor)$ codes over any symbol field contain no $\lfloor nR \rfloor$-tuple with fewer than $\lfloor nR(1-\alpha) \rfloor$ independent symbols.*

(2) [6] *The covering radius ϱ of virtually every linear code over any symbol field satisfies $\varrho = nH_q^{-1}(1-R)+o(n)$, where $H_q(x) = -x\log_q x-(1-x)\log_q(1-x)+\log_q(q-1)$ is the q-ary entropy function.*

Some of the results on complexity require a knowledge of the weight distribution of the average linear code; computation of the complexity of bounded hard and soft decision decoding algorithms require a knowledge of the average distance of linear codes. The problem of the average distance of *binary* linear codes has been examined many times. Koshelev [21], Pierce [30] and Kozlov [22] have shown that the Gilbert–Varshamov bound is exact for the average binary linear code. Our proof is very much simpler and holds over all symbol fields. It is possible to give a simple estimate of the number of codewords of a given weight in the average linear code: we have $\binom{n}{w}$ binary words of weight w. The code contains 2^k of the 2^n binary words of length n. Thus a reasonable approximation is that there are $\binom{n}{w}2^{-(n-k)}$ codewords of weight w in the average code. The following result formalizes this.

Theorem 3.1. *For any R with $0<R<1$ and any symbol field $GF(q)$, the fraction of linear (n, nR) codes over $GF(q)$ satisfying*

$$A(w) = \lfloor q^{n[H_q(w/n)-(1-R)]+o(n^{1/2})} \rfloor$$

for all w, $0<w\leq n$ is at least $1-q^{-\alpha^2\sqrt{n}+O(\log n)}$ for some constant α. The minimum distance of virtually all linear (n, nR) codes over $GF(q)$ thus satisfies $d = H_q^{-1}(1-R)+o(n)$.

Proof. Assume that the $k\times n$ generator matrix is chosen at random from the uniform distribution (possibly resulting in a matrix with rank less than k). For a given w, define X to be the random variable denoting the number of nonzero codewords of weight w. There are q^k-1 combinations of the k rows if at least one row must be taken. Let X_i, $1\leq i\leq q^k-1$ be a random variable taking the value 1 if the ith combination gives a codeword of weight w, and taking the value 0 otherwise.

Then $X = \sum_i X_i$, so

$$E(X) = \sum_i EX_i = (q^k - 1)\left(\frac{\binom{n}{w}(q-1)^w}{q^n}\right).$$

Thus $EX = q^{n[H_q(w/n)-(1-R)]+O(\log n)}$. For the variance of X, we have $\sigma^2(X) = E(X^2) - E^2(X) = E(X) + (E\sum_i X_i)(E\sum_{j\neq i} X_j) - E^2(X) < EX$. Now from Chebyshev's inequality, $\Pr(|X-\mu| \geq t) \leq \sigma^2/t^2$, so

$$\Pr(|X/\mu - 1| \geq \beta) \leq \sigma^2/(\mu\beta)^2 < (\mu\beta^2)^{-1}.$$

Setting $\beta = q^{\alpha\sqrt{n}}$ gives the required result. Thus the fraction of codes which do not satisfy the average weight condition for a given weight is vanishingly small. The fraction of codes that do not satisfy the average weight condition for *some* w is at most n times greater, and is hence also small. The probability that the generator matrix is of rank k is lower bounded by the probability that a $k \times k$ matrix over $GF(q)$ is nonsingular, which is $\prod_{i=0}^{k-1}(1-q^{-(k-i)})$ [4]. This is lowest for $q=2$, when it converges to 0.288. The set of codes which do not have the average weight distribution must therefore also be a vanishingly small fraction of the codes with rank exactly k. □

3.1. Information set algorithms

We use the term *information set algorithm* to denote algorithms that work by exploiting the redundancy of the code. In an (n, k) linear code, if we know k linearly independent symbols from a codeword, we can reconstruct the codeword. Thus if the received vector contains no errors in the k linearly independent symbols (the "information set"), we can reconstruct the codeword. If the received vector contains no errors in the information set, *any* error pattern in the parity symbols can be corrected. This is where the procedure gains its efficiency: one basic operation (reencoding) is sufficient to correct many error patterns. Of course, if there are errors in the information set, but we know or guess the error values, the same principle applies. We will exploit this fact in two decoding algorithms.

An interesting result is given by Mandelbaum [27].

Lemma (Mandelbaum). *For any linear code C and any coset leader or joint coset leader w of C, there is at least one information set in C that is disjoint from w. Thus a pure information set algorithm is always sufficient to achieve complete minimum distance decoding.*

Proof. Suppose the complement of the support of w contains fewer than k independent symbols. Then there is a nonzero codeword c with all these independent symbols 0. Then c must be 0 on all symbols not in the support of w, i.e., $\text{supp}(c) \subseteq \text{supp}(w)$. Therefore $w + \alpha c$ has weight less than w for appropriate constant α, contradicting the assumption that w is a coset leader. □

Despite the great interest in algorithms based on the information set idea [1–18], no precise estimates of the decoding complexity have been available. We examine two algorithms based on this approach: systematic coset search and the covering polynomials algorithm. We have examined a third algorithm based on the approach in [11]; we report the results here.

3.2. Systematic coset search

Systematic coset search, suggested by Montgomery et al. [29] and by Levitin [18], involves taking one information set and searching through all possible patterns of errors in that set. If we can guess the pattern of errors in the information set, we can recover the error pattern from the re-encoding argument above. We have

Theorem 3.2. *The complexity coefficient for complete hard decision decoding using the systematic coset search algorithm is*

$$\min(R, RH_q(H_q^{-1}(1-R)/R)).$$

This is always less than or equal to $\min(R, 1-R)$.

The complexity coefficient for bounded soft decision decoding is

$$\min(R, RH_2(H_2^{-1}(1-R)/R)).$$

The complexity coefficient for bounded hard decision decoding is

$$\min(R, RH_q(H_q^{-1}(1-R)/2R)).$$

Proof. The maximum number of errors that can occur in a correctable error pattern is the covering radius ϱ of the code. Thus in searching through all patterns of errors in the information set, it is sufficient to search for all patterns of weight ϱ or less. Using Blinovskii's result, cited above, we have $\varrho = nH_q^{-1}(1-R) + o(n)$ for virtually all linear codes over $GF(q)$. If this number is less than k, we achieve some reduction in decoding complexity. We must search through $\sum_{i \le \varrho} \binom{k}{i}(q-1)^i$ instead of q^k possible error patterns. Using $\binom{n}{\lambda n}(q-1)^{\lambda n} = \exp_q(nH_q(\lambda) + o(n))$, we have $(1/n)\log_q(\sum_{i \le \varrho} \binom{k}{i}(q-1)^i) = RH_q(H_q^{-1}(1-R)/R)$ if $\varrho/k \le (q-1)/q$. Note that by the convexity of the entropy function, we have $H_q(xy) > xH_q(y)$ if $0 < x, y < 1$. Thus

$$RH_q(H_q^{-1}(1-R)/R) < R(1/R)H_q(H_q^{-1}(1-R)) = 1-R.$$

In bounded soft decision decoding, the maximum hard weight error pattern has weight $2t = d-1$. Using Theorem 3.1 and Blinovskii's result on covering radius, $2t \approx \varrho$, so bounded soft decision decoding has the same complexity coefficient as complete hard decision decoding. A similar argument holds for the bounded hard decision case, using Theorem 3.1 to show that $t = nH_q^{-1}(1-R)/2$ for virtually all linear codes over any symbol field. □

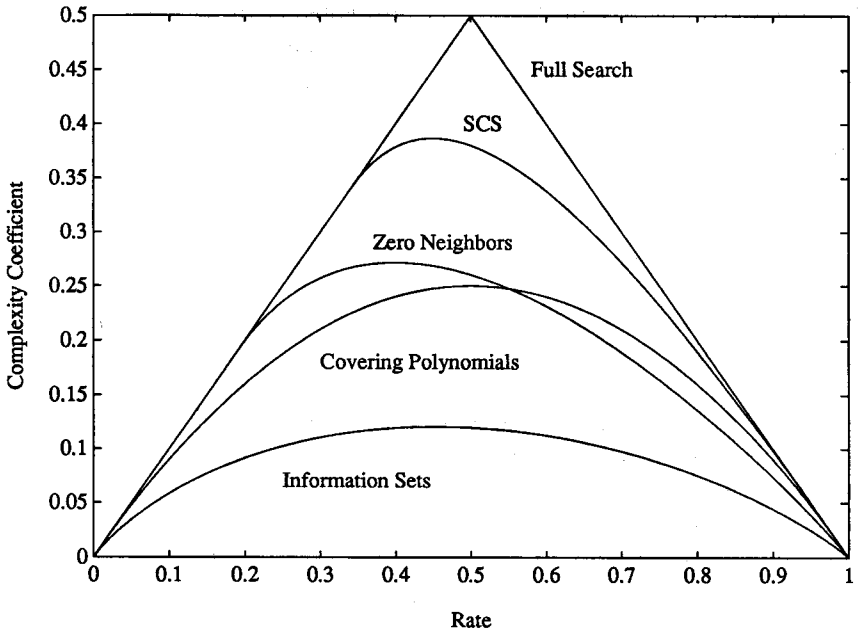

Fig. 1. Comparison of complexity of various schemes.

The function for complete hard decoding (also obtained by Levitin) is plotted in Fig. 1. The bounded hard and soft decision complexity coefficients are new.

3.3. Covering polynomials

Another method in the information set family is to take more than one information set, and attempt to recover the pattern of errors in at least one of the information sets. In each information set I, we search through candidate error patterns (the *covering polynomials*) of up to an appropriate weight w. We correct these postulated errors and use the re-encoding process. If there are in fact w or fewer errors in one of the information sets, the re-encoding principle implies that the entire error pattern will be found. The algorithm was first suggested for cyclic codes [20], with the information sets taken to be the n sets of k consecutive symbols; we follow this convention.

Before computing the complexity coefficient for the procedure, we consider the situation if a selected set of k bits is not an information set, i.e., if the column rank of the corresponding k columns of the generator matrix is $k-c$ for some positive constant c. We can modify the algorithm to take account of this, at the cost of an increase in complexity, as follows. We take $k-c$ linearly independent columns from I and c additional columns from outside I, so that the resulting set of k columns is an information set I'. Now for each postulated error pattern in I, we postulate

every possible error pattern in $I'-I$. If the postulated error pattern in I is the actual error pattern, then because of the exhaustive search through $I'-I$, we will find the actual error pattern in I'. Re-encoding from this information set will then give the error pattern. This procedure multiplies the complexity of the algorithm by q^c, so it is practical only for small c.

Theorem 3.3. *For virtually all linear (n, nR) codes over $GF(q)$, the complexity coefficient for complete hard decision decoding using the covering polynomial algorithm is $R(1-R)$. For bounded soft decision decoding, the complexity coefficient is also $R(1-R)$. For bounded hard decision decoding, the complexity coefficient is $RH_q(H_q^{-1}(1-R)/2)$.*

Proof. From the first result quoted in Section 3, the fraction of codes with *any* set of k symbols of rank less than $k(1-\alpha)$ tends to zero for any $\alpha>0$. Thus we can take as information sets the n sets of k (cyclically) consecutive symbols, find an information set I' containing as many symbols of I as possible, and perform an exhaustive search through $I'-I$; the result shows that the increase in complexity caused by using this procedure will be subexponential.

We need the value of w, the maximum weight information set error pattern that needs to be tried. There are no more than ϱ errors in any correctable pattern, and the n information sets have on average $\varrho(k/n)$ errors each. Thus there is at least one information set with no more than $\varrho(k/n)$ errors, and we can set $w=\varrho R$. The number of patterns to be tried for each information set is then $\sum_{l\le w}\binom{k}{l}(q-1)^l = \sum_{l\le \varrho R}\binom{nR}{l}(q-1)^l = \exp_q(nRH_q(\varrho/n)+o(n))$. To find the overall complexity, we multiply by n (the number of sets) and a subexponential term $q^{o(n)}$ to account for the additional exhaustive searches. Using the fact that $\varrho = nH_q^{-1}(1-R)+o(n)$ for virtually all codes, we have a complexity coefficient of $R(1-R)$ overall. The complexity coefficient is thus independent of the size of the symbol field. For bounded soft decision decoding, the same result holds, because the maximum number of hard errors, $2t$, is the same for virtually all codes as the covering radius. For bounded hard decision decoding, the value of w becomes $tR = n(RH_q^{-1}(1-R)/2+o(1))$, and the complexity coefficient is $RH_q(H_q^{-1}(1-R)/2)$. □

The results for complete hard decision and bounded soft decision — identical for $q=2$ — are plotted in Fig. 1. Clearly, the algorithm represents a great improvement over both the trivial exhaustive search algorithms and systematic coset search.

In generalized information set decoding, we seek to find a large enough number of sets of k symbols so that at least one of the sets is error free. If a set is not an information set, we apply the method used in covering polynomials above. Recently, we proved the following results: the details, too long to be given here, can be found in [11].

Theorem 3.4. *For virtually all linear (n, k) codes C over $GF(q)$, the complexity*

$M(C)$ of complete minimum distance decoding using the generalized information set decoding algorithm satisfies

$$\frac{1}{n}\log_2 M(C) = H_2(H_q^{-1}(1-R)) - (1-R)H_2\left(\frac{H_q^{-1}(1-R)}{1-R}\right) + o(1).$$

Bounded distance decoding has, for virtually all codes, a complexity $M(C)$ satisfying

$$\frac{1}{n}\log_2 M(C) = H_2(H_q^{-1}(1-R)/2) - (1-R)H_2\left(\frac{H_q^{-1}(1-R)}{2(1-R)}\right) + o(1).$$

Bounded soft decision decoding has, for virtually all binary linear codes, a complexity $M(C)$ satisfying

$$\frac{1}{n}\log_2 M(C) = (1-R)\left[1 - H_2\left(\frac{H_2^{-1}(1-R)}{(1-R)}\right)\right] + o(1).$$

The functions for complete hard decoding and bounded soft decoding are identical for $q=2$. They are plotted in Fig. 1. Clearly, they represent a huge improvement over exhaustive search procedures for any fixed rate. For $R=1/2$, generalized information set decoding requires, for complete decoding, less than the fourth root of the number of computations required by a search through all codewords, while it requires less than the ninth root of the number of computations for bounded hard decoding.

3.4. Progressive algorithms

The basis of this family of algorithms is the following simple observation: it is easy in general to locate a word of fairly low weight in a coset. If we can do this, the difference between the low weight word we have and the coset leader will be a codeword of low weight. The average code has a weight distribution approximately equal to a scaled binomial distribution, the low weight words represent the "tails" of the distribution, and so there are relatively few low weight codewords. Our strategy is thus to find a low weight word in the coset, add in turn all sufficiently low weight codewords, and take the lowest weight sum as the coset leader. This is an exhaustive search of a set containing relatively few codewords.

3.4.1. Zero neighbors algorithm

One version of this is the recently proposed zero neighbors algorithm [23]. To summarize this algorithm briefly, the set of codewords used is that required to provide a minimal cover of the *zero domain frame*, which is the set of n-tuples that are not coset leaders, but are at distance 1 from a coset leader. Levitin and Hartmann [23] show that an upper bound on the weight of a codeword in the minimal set is about twice the covering radius. Thus an upper bound on the complexity is given by the number of codewords of weight 2ϱ or less, where ϱ is the covering radius. Thus

Theorem 3.5. *Complete hard decision decoding using the zero neighbors algorithm has, for virtually all linear codes, a complexity of*

$$q^{n[H_q(2H_q^{-1}(1-R))-(1-R)+o(1)]}.$$

Bounded soft decision decoding has, for virtually all binary linear codes, a complexity of

$$2^{n[H_2(2H_2^{-1}(1-R))-(1-R)+o(1)]}.$$

Proof. The complete hard decision result follows immediately from Theorem 3.1. The bounded soft decision result follows from the fact that the hard decision coset leader will, from Blinovskii's result quoted above, have weight no more than $\varrho = nH_2^{-1}(1-R)+o(n)$. The least soft weight error pattern cannot have more than $2t$ hard errors if it is within the bounded soft distance, and from Theorem 3.1, we have $2t = nH_q^{-1}(1-R)+o(n) \approx \varrho$ for virtually all binary linear codes. Then the best hard decision estimate of the error pattern and the best soft decision estimate differ by a codeword of weight at most 2ϱ. Thus after using the zero neighbors algorithm to find the best hard decision estimate, we can add all codewords of weight 2ϱ or less to the estimate and find the best soft decision estimate. But the set of zero neighbors is already assumed to contain all codewords of weight up to 2ϱ, so the second part of the algorithm results in a doubling of complexity, making no difference to the form of the result. □

The result for complete hard decision decoding was given in the original paper by Levitin and Hartmann; the result on the complexity for bounded soft decoding is new to this paper. The function is plotted in Fig. 1 for the binary case. The complexity required is much less than that for exhaustive search (approximately the square root of the number of codewords at rate 1/2), though considerably more than that for generalized information set decoding.

3.4.2. Projecting set decoding

An algorithm with some points in common with the zero neighbors algorithm has been suggested (for the soft decision case) by Hwang [19]. Again, we want to compute a minimal set of codewords so that, given a word in the coset, we can add codewords from the set repeatedly, accepting a sum if it results in a lower weight word, and declaring that the word is a coset leader when no further weight reduction occurs by adding any word of the set. The projecting set is a set of low weight codewords: the set contains all codewords of weight $2d-1$ or less, and for binary codes we need have no word of weight greater than $n-k$ in the minimal set [19]. The second observation follows because any codeword of weight $>n-k$ can be expressed as the sum of two *disjoint* codewords. If the single codeword gave a weight reduction when added to the received word, at least one of its constituent codewords must do so also. Thus we use a subset of the number of codewords, and have a lower complexity than a full search through the codewords. We have (once again a new result)

Theorem 3.6. *The complexity coefficient $f(R)$ for complete (maximum likelihood) soft decision decoding using the projecting set algorithm satisfies, for virtually all binary linear codes,*

$$H_2(2H_2^{-1}(1-R))-(1-R)=f_L(R)\leq f(R)\leq f_U(R)=\max_{R'\leq R} H_2(1-R')-(1-R').$$

The upper bound is higher than $\min(R, 1-R)$.

Proof. Every word of weight $2d-1$ or less is in the set, and from Theorem 3.1, $d=nH_2^{-1}(1-R)+o(n)$. Thus, again from Theorem 3.1, $\sum_{i\leq 2d-1} A(i)\approx 2^{n[H_2(2H_2^{-1}(1-R))-(1-R)]}$. The upper bound follows directly from Theorem 3.1. If $R<1/2$, then Theorem 3.1 shows that an asymptotically insignificant proportion of the codewords have weight greater than $n-k$, so the set includes almost every codeword and $f_U(R)=R$. If $R>1/2$, we have a significant gain over a search through all codewords, but the complexity function, $H_2(1-R)-(1-R)$ is always greater than $1-R$ over this range, and thus the algorithm is inferior to the trellis search proposed by Wolf [34], for which the complexity coefficient is $1-R$. □

Thus the algorithm appears to have no value in the asymptotic sense, although it does have the features that it is valid for complete *soft* decision decoding, and provides an upper bound on the weight of codewords to be included in the set of zero neighbors.

4. Continued division algorithms

The basic procedure of continued division, defined below, was first suggested as an approach to the decoding problem by Farrell [15] on empirical grounds. In this paper, we provide original analysis of the effect of the procedure, and use it as an integral part of a new decoding algorithm. We show that the proposed algorithm is unique in that it exploits features of both information set decoding algorithms and progressive algorithms. It also offers a way of reducing the space complexity of a generalized algorithm at the expense of slightly increased time complexity; this tradeoff is not available with the approaches of Section 3. We discuss the algorithm for the case of cyclic codes; this clarifies the ideas and simplifies the analysis. We show how the procedure may be generalized to linear codes.

The basic procedure is as follows. We have a received word $r(x)$, and a *dividing* codeword $c(x)$. Dividing $r(x)$ by $c(x)$ will produce the remainder $r(x) \bmod c(x)$. By *continued division*, we mean the process of producing $x^i r(x) \bmod c(x)$ for $0\leq i\leq M$ for some appropriate M. This corresponds to performing longhand division with a large number of zeros appended to the right of $r(x)$. Alternatively, we can view the process as involving continued subtractions of cyclic shifts of $c(x)$ that may "wrap around" the end of the word. Table 1 gives an example of the procedure for the

Table 1. Continued division.

$r(x)$	1	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	1	1	1	0	0	0	1
$g(x)$	1	0	1	0	1	1	1	0	0	1	0	1											
	0	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	1	1	1	0	0	0	1
						1	0	1	0	1	1	1	0	0	0	1	1						
$s(x)$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	1
	1												1	0	1	0	1	1	1	0	0	0	1
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0

Golay code. Starting with the syndrome $S(x)=r(x) \bmod g(x)$, the ith remainder in the process is $x^{n-i}(x^i r(x) \bmod c(x)) \bmod (x^n-1)$.

As a first step towards the analysis, we consider division by $g(x)$. This is exactly equivalent to error trapping [25], i.e., information set decoding in which the information sets are the n sets of k consecutive bits (taken cyclically). This is sufficient to detect all error patterns of burst length up to $n-k$: let $r(x)=i(x)g(x)+x^j e(x)$, with $\deg e(x)<n-k$. Then the $(n-j)$th remainder in the continued division process is $x^j(x^n e(x) \bmod g(x)) \bmod (x^n-1)$. But $g(x) \mid x^n-1$ and $\deg g(x)>\deg e(x)$, so $x^n e(x) \bmod g(x)=e(x)$, and the $(n-j)$th remainder is $x^j e(x)$, the error pattern.

In dividing by a codeword $d(x)=i(x)g(x)$ other than the generator, we distinguish between the two cases $d(x) \mid x^n-1$ and $d(x) \nmid x^n-1$. If $d(x) \mid x^n-1$, then $d(x)$ is itself the generator of a cyclic code C_S of length n; this eases the analysis considerably, and we concentrate on this case. Every codeword in C_S is of the form $a(x)d(x)=a(x)i(x)g(x) \in C$, so C_S is a subcode of C. Continued division by $d(x)$ thus corresponds to error trapping *in the subcode*. If the transmitted codeword $c(x) \in C$ is also in C_S, then the received word can be written as $r(x)=b(x)d(x)+e(x)$. In this case, if the error pattern has burst length less than the redundancy *of the subcode*, we achieve correction. The redundancy of the subcode is $\deg d(x)=n-k+\deg i(x)$. This is higher than the redundancy of the code C, and so many more error patterns can be trapped. In general, however, the transmitted codeword does not belong to the subcode. In this case, the received word is of the form $r(x)=c(x)+e(x)=c_s(x)+e_s(x)$, where $c_s(x)$ is the nearest codeword of C_S to $r(x)$, and $e_s(x)$ is in the same coset as $e(x)$. We refer to $e_s(x)$ as the *subcode* coset leader. Three situations are possible when we begin division of $r(x)$ by $d(x)$:

(i) $r(x)$ is a subcode coset leader;

(ii) $r(x)$ is not a subcode coset leader, and all words of lower weight in the same subcode coset have burst length $>n-k+\deg i(x)$;

(iii) $r(x)$ is not a subcode coset leader, and at least one word of lower weight in the same coset has burst length $\leq n-k+\deg i(x)$.

In the first case, no reduction in weight is possible on division by $d(x)$. In the second case, we can only achieve weight reduction in the exceptional case that the received

word is of the form $x^l a(x)d(x)+w(x)$, where $\mathrm{wt}(w(x))<\mathrm{wt}(r(x))$ and $\mathrm{len}\ w(x)<n-k+\deg i(x)+l$. In the third case, a word of lower weight in the coset is found. This suggests the following algorithm.

Continued division algorithm

(1) Select M codewords $c_i(x)$, $0\le i<M$, with $c_i(x) \mid x^n-1$.
(2) Construct the table T of adjustment codewords (see below).
(3) Given received word $r(x)$, perform continued division for two cycles by each $c_i(x)$.
(4) Let $w(x)$ be the lowest weight word resulting from step (2). If $\mathrm{wt}(w(x))<\mathrm{wt}(r(x))$, let $r(x)=w(x)$ and go to step (3); otherwise, go to step (5).
(5) For each codeword $a(x)$ in T, compute $a(x)+w(x)$.
(6) Take the lowest weight such word as the coset leader.

The basis for the algorithm is the assumption that case (ii) is rare, and that cases (i) and (iii) dominate. Then if there is a word $w(x)$ in the coset such that $r(x)-w(x)$ is contained in *any* of the subcodes, we find $w(x)$ through the continued division process. We eventually finish with a $w(x)$ which is a coset leader in each of the subcodes. The table of "adjustment" codewords T consists of all those codewords that are representable as the difference between a word that is a coset leader in all the subcodes and its coset leader in the code C. As in the case of the zero neighbors algorithm, only codewords of relatively low weight will be required. We will not, however, have to store any such low weight codewords that are contained in any one of the cyclic subcodes. We thus achieve a form of compression of the required decoding codewords: if many zero neighbors lie in a single subcode, then instead of having to store them all explicitly, we can store the generator of the subcode only.

To generalize the process to any linear code C, we select an information set of C, and a subset of the information set. The set of codewords whose nonzero information bits are confined to the subset of the information set define a subcode C_S of the code. Given a received word $r(x)$, we subtract the codeword in C_S which has the same pattern in the subset as $r(x)$. As in the case of cyclic codes, this is sufficient to recover the error pattern if the transmitted codeword belongs to the subcode and the error pattern is disjoint from the subset. Given any received word, a lower weight word in the same coset will be found if the codeword equal to the difference between the two words is in the subcode, and the lower weight word is disjoint from the subset. We now draw a distinction between two types of errors: ones turned to zeros (type I) and zeros turned to ones (type II). A necessary condition for decoding is that no type I errors are located in the information set: if a type I error is in the subset, then the error pattern is not trapped even in the subcode, while if it is in the remainder of the information set, the difference codeword cannot be in the subcode. However, decoding is still possible even if type II errors are in the information set, but not the subset. The asymptotic complexity of the procedure will be determined by the manner of selection of the information sets and subsets. The above argument

indicates that the best strategy is to select the information sets so that the ones of the received word are more likely to be in an information set than the zeros. Thus the type I errors are more likely to be confined to the parity check bits.

5. An improved combined algorithm for bounded soft decoding

Recall that in bounded soft decision decoding, we are concerned with getting the nearest codeword provided that the soft distance is less than t. This means that the total number of hard errors cannot be greater than $2t$. The complexity coefficients obtained in Section 3 above assumed that this many hard errors had been made. However, if $2t$ hard errors have been made and the soft distance is still not greater than t, then the received vector must have $2t$ components with soft value 1/2 (i.e., erasures) and $n-2t$ components that are 0 or 1. If we know that there are exactly $2t$ errors, then we know that all the erasures represent hard errors, for otherwise the soft distance is greater than t. Thus the error pattern is easy to compute. If the number of errors is much lower than $2t$, the information set decoding algorithm will perform well. If, on the other hand, the number of errors e is high (close to $2t$) we can perform a sequential search for the error pattern through the words with hard weight e that are closest in the soft weight sense to the received word. To do this, we order the received bits in increasing order of reliability, and define a partial ordering on the e-tuples: let the word $a=(a_0,\ldots,a_{n-1})$ have support $\{a^1,\ldots,a^e\}$ and define $\{b^1,\ldots,b^e\}$ similarly. Then $a \gg b$ if and only if $a^i \leq b^i$ for all i. A full ordering is given by the rule $a>b$ if and only if $a_i<b_i$ and $a_j=b_j$ for $j<i$. Clearly, if (soft) $\mathrm{dist}(r,a)>t$ and $a \gg b$, then $\mathrm{dist}(r,b)>t$. Using this ordering, and beginning with 1^e0^{n-e}, we examine $\mathrm{dist}(r,a)$ for each a in turn. If $\mathrm{dist}(r,a)>t$, we need not examine any b such that $a \gg b$.

The following is our proposed new strategy: at step j of the algorithm, we assume that there are j hard errors. If j is less than some appropriate threshold, we try to decode using information sets. If it is higher than the threshold, we try the lexicographic search outlined above. The overall complexity coefficient will be determined by the value of the threshold. To find the optimum setting, we determine an upper bound on the number of operations in the lexicographic search as a function of j, and find the point at which this equals the complexity of an information set algorithm that seeks to decode j hard errors.

We have the following result:

Theorem 5.1. *The combined algorithm of information set decoding and lexicographic search achieves bounded soft distance decoding with, for virtually all binary linear codes, a complexity coefficient of at most*

$$H_2(\xi)-(1-R)H_2(\xi/(1-R))$$

where ξ is the solution to the equation

$$H_2(\xi)-(1-R)H_2(\xi/(1-R))=2(2\tau-\xi)+(1-4\tau+2\xi)H_2\left(\frac{(2\tau-\xi)}{(1-4\tau+2\xi)}\right)$$

where $\tau=t/n=H_2^{-1}(1-R)/2$.

We plot this function in Fig. 2. Clearly, it represents a significant improvement on the information set decoding algorithm. As it is an upper bound, further improvement may be possible using this approach.

Proof. To prove the result, we bound the maximum distance two words can be apart if they both need to be examined in the systematic search. Assume that we compute the reliability of each received bit (the soft distance from the nearer of 0 and 1) and arrange the bits in increasing order of reliability. Take any two words of weight j that are within soft distance t of the received word. Label the bits where both words are one as E_1, the bits where both are zero as C_1, and the bits in which they differ as D_1 and D_2 (where D_1 is the set of locations which have a one in the first word and a zero in the second, etc.). We have

$$\sum_{E_1}(1-\alpha_i)+\sum_{D_1}(1-\alpha_i)+\sum_{D_2}\alpha_i+\sum_{C_1}\alpha_i<t$$

and

$$\sum_{E_1}(1-\alpha_i)+\sum_{D_1}\alpha_i+\sum_{D_2}(1-\alpha_i)+\sum_{C_1}\alpha_i<t.$$

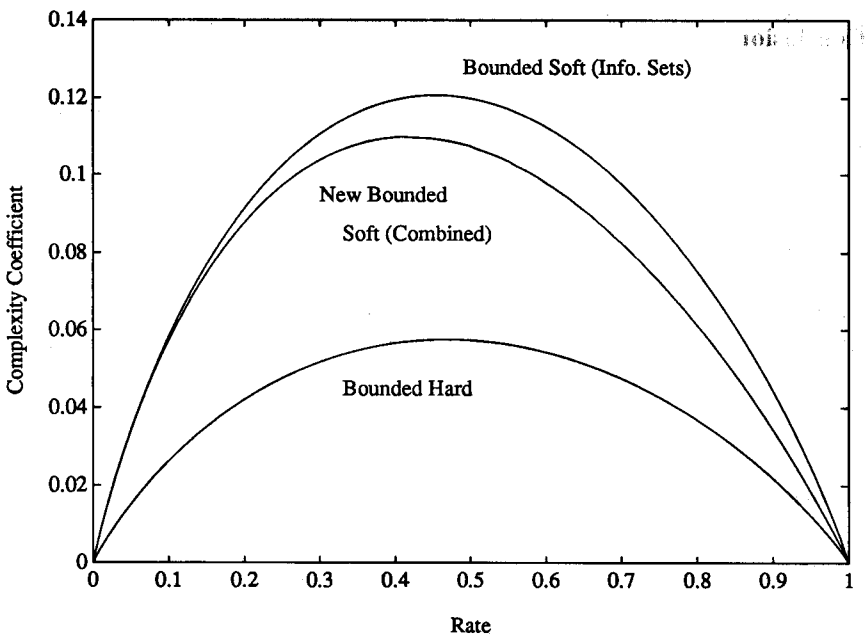

Fig. 2. Complexity for bounded hard and soft decoding.

Summing these, we have

$$2\sum_{E_1}(1-\alpha_i)+m+2\sum_{C_1}\alpha_i<2t$$

where m is the distance between the two sequences. As $1-\alpha_i \geq 1/2$, we must have $m+|E_1|<2t$; now $|E_1|\geq j-m/2$, so $m\leq 2(2t-j)$. Note that if j is close to $2t$, we must have m close to zero. We need to find the maximum number of words of weight j of distance no more than m apart. This amounts to trying to maximize the number of codewords in a constant weight anticode. We find an upper bound by determining the maximum number of words of any weight that are all within distance m of two words that are distance m apart. Without loss of generality, let one of the words be the zero codeword. The second word then has weight m. We seek the maximum number of words of weight m or less that are distance no more than m from the second word. This is given by $\sum_a \sum_b \binom{m}{a}\binom{n-m}{b}$ with $m-a+b\leq m$ and $a+b\leq m$. Thus $a\geq b$ and the complexity is dominated by the term $\binom{m}{m/2}\binom{n-m}{m/2}$. Taking the logarithm of this and dividing by n gives a complexity coefficient of $m/n+(1-m/n)H_2(m/n)/(2(1-m/n)))$. Substituting $m=2(2t-j)$ yields the expression in the theorem. □

Clearly this is an upper bound, so we pose the obvious questions: what is the exact solution, and what is the maximum number of codewords in a constant weight anticode?

6. Conclusions

Our motivation is to synthesize some of the various approaches to the general decoding problem and to derive accurate measures of complexity for such schemes. We give accurate results for many schemes, covering the cases of full and bounded hard decision decoding and bounded soft decision decoding, and show that a vast improvement over brute force algorithms is possible from many different approaches. We propose a decoding algorithm for cyclic codes that takes advantage of the main features of both types of algorithm studied, and suggest a generalization to all linear codes. Finally, we propose an algorithm for the case of bounded soft decision decoding that is significantly better than any algorithm currently known.

References

[1] L.A. Bassalygo, V.V. Zyablov and M.S. Pinsker, Problems of complexity in the theory of correcting codes, Problemy Peredachi Informatsii 13 (1977) 5–17.

[2] L.D. Baumert, R.J. McEliece and G. Solomon, Decoding with multipliers, JPL Deep Space Network Progress Report 42 (34), Jet Propulsion Laboratory (1976) 42–46.

[3] A. Benyamin-Seeyar, S.G.S. Shiva and V.K. Bhargava, Capability of error-trapping technique in decoding cyclic codes, IEEE Trans. Inform. Theory 32 (1986) 166–180.
[4] E.R. Berlekamp, The technology of error-correcting codes, Proc. IEEE 68 (1980) 564–593.
[5] E.R. Berlekamp, R.J. McEliece and H.C.A. van Tilborg, On the inherent intractability of certain coding problems, IEEE Trans. Inform. Theory 24 (1978) 384–386.
[6] V.M. Blinovskii, Lower asymptotic bound on the number of linear code words in a sphere of given radius in F_q^n, Problemy Peredachi Informatsii 23 (1987) 50–53.
[7] A.H. Chan and R.A. Games, (n, k, t)-covering systems and error-trapping decoding, IEEE Trans. Inform. Theory 27 (1981) 643–646.
[8] D. Chase, A class of algorithms for decoding block codes with channel measurement information, IEEE Trans. Inform. Theory 18 (1972) 170–182.
[9] G.C. Clark and J.B. Cain, Error-Correcting Coding for Digital Communications (Plenum Press, New York, 1981).
[10] J.T. Coffey, On complexity and efficiency in encoding and decoding error-correcting codes, Ph.D. Dissertation, Department of Electrical Engineering, California Institute of Technology, Pasadena, CA (1989).
[11] J.T. Coffey and R.M.F. Goodman, The complexity of information set decoding, IEEE Trans. Inform. Theory 36 (1990) 1031–1037.
[12] O.F. Dmitriev, An algorithm for the correction of independent errors by cyclic codes, Problemy Peredachi Informatsii 3 (1967) 102–104.
[13] G.S. Evseev, Complexity of decoding for linear codes, Problemy Peredachi Informatsii 19 (1983) 3–8.
[14] P.G. Farrell, M. Rice and F. Taleb, Minimum weight decoding for cyclic codes, in: Proceedings IMA Conference on Cryptography and Coding, Cirencester, UK (1986).
[15] P.G. Farrell, M. Rice and F. Taleb, Division algorithms for hard and soft decision decoders, in: Proceedings International Conference on Digital Signal Processing, Florence, Italy (1987).
[16] T.J. Goblick Jr, Coding for a discrete information source with a distortion measure, Ph.D. Dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA (1962).
[17] R.M.F. Goodman and A.D. Green, Microprocessor-controlled permutation decoding of error-correcting codes, in: Proceedings IERE Conference on Microprocessors in Automation and Communications 41, Kent, UK (1978) 365–376.
[18] C.R.P. Hartmann and L.B. Levitin, An improvement of the zero-neighbors minimum distance decoding algorithm: the zero-guard algorithm, 1988 IEEE International Symposium on Information Theory, Kobe, Japan, 1988.
[19] T.-Y. Hwang, Decoding linear block codes for minimizing word error rate, IEEE Trans. Inform. Theory 25 (1979) 733–737.
[20] T. Kasami, A decoding procedure for multiple-error-correcting cyclic codes, IEEE Trans. Inform. Theory 10 (1964) 134–138.
[21] V.N. Koshelev, On some properties of random group codes of great length, Problemy Peredachi Informatsii 1 (1965) 35–38.
[22] M.V. Kozlov, The correcting capacities of linear codes, Soviet Phys. Dokl. 14 (1969) 413–415.
[23] L.B. Levitin and C.R.P. Hartmann, A new approach to the general minimum distance decoding problem—the zero neighbors algorithm, IEEE Trans. Inform. Theory 31 (1985) 378–384.
[24] F.J. MacWilliams, Permutation decoding of systematic codes, Bell Syst. Tech. J. 43 (1964) 485–505.
[25] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error-Correcting Codes (North-Holland, Amsterdam, 1977).
[26] D. Mandelbaum, On vote-taking and complete decoding of certain error-correcting codes, Inform. and Control 43 (1979) 195–197.
[27] D. Mandelbaum, On complete decoding of linear error-correcting codes, Inform. and Control 47 (1980) 195–200.

[28] R.J. McEliece, A public-key cryptosystem based on algebraic coding theory, JPL Deep Space Network Progress Report 42 (44), Jet Propulsion Laboratory (1978).
[29] B.L. Montgomery, H. Diamond and B.V.K. Vijaya Kumar, A general minimum distance decoding procedure for binary linear block codes, in: Proceedings IEEE International Symposium on Information Theory, Ann Arbor, MI (1986).
[30] J.N. Pierce, Limit distribution of the minimum distance of random linear codes, IEEE Trans. Inform. Theory 13 (1967) 595–599.
[31] E. Prange, The use of information sets in decoding cyclic codes, IRE Trans. Inform. Theory 8 (1962) S5–S9.
[32] S.G.S. Shiva and K.C. Fung, Permutation decoding of certain triple-error-correcting binary codes, IEEE Trans. Inform. Theory 18 (1972) 444–446.
[33] V.K. Wei, An error-trapping decoder for nonbinary cyclic codes, IEEE Trans. Inform. Theory 30 (1984) 538–541.
[34] J.K. Wolf, Efficient maximum likelihood of linear block using a trellis, IEEE Trans. Inform. Theory 24 (1978) 76–80.