The Intractability of Computing the Minimum Distance of a Code

Alexander Vardy, *Senior Member, IEEE*

***Abstract*— It is shown that the problem of computing the minimum distance of a binary linear code is NP-hard, and the corresponding decision problem is NP-complete. This result constitutes a proof of the conjecture of Berlekamp, McEliece, and van Tilborg, dating back to 1978. Extensions and applications of this result to other problems in coding theory are discussed.**

***Index Terms*— Complexity, linear codes, minimum distance, NP-completeness.**

I. Introduction

A PROBLEM Π is said to belong to the class NP if it can be solved by a nondeterministic Turing machine in polynomial time. A problem $\Pi \in$ NP is NP-*complete* if every problem in NP can be transformed to Π in deterministic polynomial time. A problem Π, which is not necessarily in NP, is said to be NP-*hard* if the existence of a deterministic polynomial-time algorithm for Π implies the existence of such an algorithm for every problem in NP. For a more rigorous definition of these terms, see Garey and Johnson [20].

Berlekamp, McEliece, and van Tilborg [9] showed in 1978 that two fundamental problems in coding theory, namely maximum-likelihood decoding and computation of the (nonzero terms in the) weight distribution, are NP-hard for the class of binary linear codes. The formal statement of the corresponding decision problems[1] follows.

Problem: MAXIMUM-LIKELIHOOD DECODING
Instance: A binary $m \times n$ matrix H, a vector $s \in \mathbb{F}_2^n$, and an integer $w > 0$.
Question: Is there a vector $x \in \mathbb{F}_2^n$ of weight $\leq w$, such that $Hx^t = s$?

Problem: WEIGHT DISTRIBUTION
Instance: A binary $m \times n$ matrix H and an integer $w > 0$.
Question: Is there a vector $x \in \mathbb{F}_2^n$ of weight w, such that $Hx^t = \mathbf{0}$?

Berlekamp, McEliece, and van Tilborg [9] proved that both problems are NP-complete using a reduction from THREE-DIMENSIONAL MATCHING, a well-known NP-complete problem [20, p. 50]. They conjectured, but were unable to prove, that the following decision problem:

Problem Π: MINIMUM DISTANCE
Instance: A binary $m \times n$ matrix H and an integer $w > 0$.
Question: Is there a nonzero vector $x \in \mathbb{F}_2^n$ of weight $\leq w$, such that $Hx^t = \mathbf{0}$?

is also NP-complete. It is easy to see that NP-completeness of Π would imply that computing the minimum distance of a binary linear code is NP-hard. Indeed, let $\mathbb{C}$ be a linear code defined by the parity-check matrix H, and let d denote the minimum distance of $\mathbb{C}$. If d is known, then one can answer the question of Π by simply comparing d and w. On the other hand, if one can solve Π, then one can also find d by successively running an algorithm for Π with $w = 1, 2, \cdots$ until the first affirmative answer is obtained.

The MINIMUM DISTANCE problem has a long and convoluted history. To the best of our knowledge, it was first mentioned by Dominic Welsh at an Oxford Conference on Combinatorial Mathematics in 1969. In the printed version [39] of his paper, Welsh calls for an efficient algorithm to find the shortest cycle in a linear matroid over a field $\mathbb{F}$. It is easy to see that for $\mathbb{F} = \mathrm{GF}(q)$, this is equivalent to finding the minimum-weight codeword in a linear code over $\mathrm{GF}(q)$. Hence the NP-completeness of MINIMUM DISTANCE implies that a polynomial-time algorithm for the problem posed by Welsh [39] is unlikely to exist. Following the publication by Berlekamp, McEliece, and van Tilborg [9] of their conjecture, the MINIMUM DISTANCE problem was mentioned as open by Garey and Johnson in [20, p. 280]. Three years later, it was posed by Johnson [23] as an "open problem of the month" in his ongoing guide to NP-completeness column. The problem remained open despite repeated calls for its resolution by Johnson [24] and others.

Determining whether computation of the minimum Hamming distance of a linear code is NP-hard is important not only because this is a long-standing open problem. There are several more compelling reasons. First, for a host of problems in coding theory there is an easy reduction from MINIMUM DISTANCE. A few examples of such problems are presented in Section V. Thus if MINIMUM DISTANCE is computationally intractable, then all these problems are intractable as well. Secondly, it is known that the parameters of almost all linear codes attain the Gilbert–Varshamov bound [34, p. 77]. Hence it is easy to devise randomized algorithms that with high probability yield (long) linear codes with large distance.

Manuscript received November 7, 1996; revised May 20, 1997. This research was supported by the Packard Foundation, the NSF, and the JSEP under Grant N00014–9610129. The material in this paper was presented in part as a Plenary Lecture at the 29th Annual Symposium on Theory of Computing, El Paso, TX, May 1997.

The author is with the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801 USA.

Publisher Item Identifier S 0018-9448(97)07421-X.

[1] The MAXIMUM-LIKELIHOOD DECODING problem was originally termed COSET WEIGHTS in [9]; it is also referred to as DECODING OF LINEAR CODES in [20, p. 280], [29], and as MINIMUM DISTANCE DECODING in [7]. The WEIGHT DISTRIBUTION problem was originally termed SUBSPACE WEIGHTS in [9].

If there were a polynomial-time procedure for computing the minimum distance of a linear code, these randomized algorithms could be used for code construction. It is, therefore, important to know that such a polynomial-time procedure is unlikely to exist.

Due to these and other reasons, the conjecture of Berlekamp, McEliece, and van Tilborg [9] sparked a remarkable amount of work, most of it unpublished. In particular, MAXIMUM-LIKELIHOOD DECODING was shown to remain hard in more general contexts: with unlimited pre-processing [12], under approximation within a given constant factor [5], [33], and over an arbitrary (fixed) alphabet [7]. Furthermore, in an attempt to establish the NP-completeness of MINIMUM DISTANCE, a great number of closely related problems were proved to be NP-complete. For example, the problems of finding the maximum weight of a codeword, determining the existence of a codeword of weight $n/2$, and computing the minimum weight of a codeword which is nonzero in a specified position, were shown to be NP-hard by Ntafos and Hakimi [29], Calderbank and Shor (see [13]), and Lobstein and Cohen [26], respectively. A brief overview of the plurality of problems of this kind is provided, for completeness, in the next section. All these problems are tantalizingly close to MINIMUM DISTANCE. Nevertheless, the proof of the original conjecture of Berlekamp, McEliece, and van Tilborg [9] remained elusive for almost two decades.

Our main goal in this paper is to prove that MINIMUM DISTANCE is NP-complete. To this end, we exhibit a polynomial transformation to MINIMUM DISTANCE from MAXIMUM-LIKELIHOOD DECODING. Thus we settle the conjecture of [9] in the affirmative, using a reduction from the main result of [9].

We start with some notation and overview of relevant background in Section II. We also show in Section II that MAXIMUM-LIKELIHOOD DECODING remains NP-complete under certain minor restrictions, and reformulate this problem as the finite-field version of SUBSET SUM, a well-known NP-complete problem [20, p. 223]. In Section III, we use certain simple alternants [11], [25], [28] to show that computing the minimum distance for the class of linear codes over a field of characteristic 2 is NP-hard, and the corresponding decision problem MINIMUM DISTANCE OVER GF(2^m), in short MD$_{2^m}$, is NP-complete. Our proof is based on a polynomial transformation from MAXIMUM-LIKELIHOOD DECODING to MD$_{2^m}$. This, however, does not prove that MINIMUM DISTANCE is NP-complete, since the set of possible inputs to MINIMUM DISTANCE is a small subset of the set of possible inputs to MD$_{2^m}$. Therefore, in Section IV, we map the code $\mathbb{C}^{\#}$ over GF(2^m), constructed in Section III, onto a binary code $\mathbb{C}$, in such a way that the minimum distance of $\mathbb{C}^{\#}$ can be determined from the minimum distance of $\mathbb{C}$. The particular mapping used employs a simple construction of low-rate binary codes, which was pointed out to us by Noga Alon [3]. Since the length of $\mathbb{C}$ is bounded by a polynomial in the length of $\mathbb{C}^{\#}$, and the mapping itself can be accomplished in polynomial time, this completes the proof of the NP-completeness of MINIMUM DISTANCE. We conclude the paper in Section V, by showing that MINIMUM DISTANCE is NP-complete for linear codes over an arbitrary, fixed, finite field. Furthermore, several problems are shown to be NP-hard in Section V, using a reduction from MINIMUM DISTANCE. Finally, two important problems in coding theory that are closely related to MINIMUM DISTANCE are also briefly discussed in Section V.

One last remark in this section: we point out that the hardness of MINIMUM DISTANCE can be viewed as an essentially combinatorial question. Indeed, consider the following graph-theoretic decision problem:

Problem: EVEN VERTEX SET
Instance: A graph $G=(V,E)$ and an integer $w>0$.
Question: Is there a nonempty subset $V' \subseteq V$ of at most w vertices, such that every vertex $v \in V$ has an even number of vertices of V' among its neighbors?

It is easy to see that the NP-completeness of MINIMUM DISTANCE immediately implies that EVEN VERTEX SET is NP-complete. In fact, MINIMUM DISTANCE is essentially a restriction of EVEN VERTEX SET to bipartite graphs, obtained by identifying a parity-check matrix H with an adjacency matrix of a bipartite (Tanner) graph G—see [16] for more details. Thus it is interesting to observe that algebraic techniques deeply rooted in coding theory, such as construction of MDS codes via alternants [11], [30] and concatenated coding [17], [19], can be employed to answer a purely combinatorial question.

II. PRELIMINARIES

In the next subsection, we briefly survey some of the prior work motivated by the conjecture of Berlekamp, McEliece, and van Tilborg [9]. In a later subsection, we consider the MAXIMUM-LIKELIHOOD DECODING problem, and show that it remains NP-complete under certain, not too restrictive, conditions.

A. NP-Complete Problems Related to MINIMUM DISTANCE

The following eight problems, closely related to MINIMUM DISTANCE, are known to be NP-complete. These problems are included herein for completeness. They are listed in chronological order, with appropriate references.

First, as noted in [9], MINIMUM DISTANCE is a variation of WEIGHT DISTRIBUTION, obtained by replacing the phrase "of weight w" with the phrase "of weight $\leq w$." It is also easy to see that MINIMUM DISTANCE is a special case of MAXIMUM-LIKELIHOOD DECODING, obtained by restricting the input to $s=\mathbf{0}$ and requiring that x is nonzero. Thus the problem that Berlekamp, McEliece, and van Tilborg [9] conjectured to be NP-complete is in many ways related to the two problems that they proved are NP-complete.

Three other computational tasks, that even more closely resemble MINIMUM DISTANCE, were shown to be NP-hard by Ntafos and Hakimi [29], namely: finding a codeword of maximum weight, finding a codeword of minimum weight which is not a multiple of k, and finding a codeword whose weight is in the range $[w_1, w_2]$. Formally, the problems:

Instance: A binary $m \times n$ matrix H and an integer $w > 0$.
Question: Is there a vector $x \in \mathbb{F}_2^n$ of weight $\geq w$, such that $Hx^t = \mathbf{0}$?

Instance: A binary $m \times n$ matrix H, an integer $w > 0$, and an integer $k \geq 2$.
Question: Is there a nonzero vector $x \in \mathbb{F}_2^n$ of weight $\leq w$, such that $Hx^t = \mathbf{0}$ and $\mathrm{wt}(x) \not\equiv 0 \pmod{k}$?

Instance: A binary $m \times n$ matrix H, integers $w_2 > w_1 > 0$.
Question: Is there a vector $x \in \mathbb{F}_2^n$ such that $Hx^t = \mathbf{0}$ and $w_1 \leq \mathrm{wt}(x) \leq w_2$?

are NP-complete [29]. All the three problems are variations of the WEIGHT DISTRIBUTION problem; they are all somewhat weaker than this problem, in the sense that the existence of a polynomial-time algorithm for WEIGHT DISTRIBUTION directly implies the existence of a polynomial-time algorithm for each of the three problems (of course, the converse is also true indirectly, since all these problems are NP-complete). Along similar lines, Calderbank and Shor (see Diaconis and Graham [13]) showed that

Instance: A binary $m \times n$ matrix H, where n is a positive even integer.
Question: Is there a vector $x \in \mathbb{F}_2^n$ of weight $n/2$, such that $Hx^t = \mathbf{0}$?

is an NP-complete problem. That is, WEIGHT DISTRIBUTION remains NP-complete even if the input is restricted to $w = n/2$. On the other hand, Lobstein and Cohen [26] considered a variation of MAXIMUM-LIKELIHOOD DECODING that is deceptively close to MINIMUM DISTANCE: they showed that finding a codeword of minimum weight among all the codewords that are nonzero on the first position is NP-hard. Formally, the problem:

Instance: A binary $m \times n$ matrix H and an integer $w > 0$.
Question: Is there a vector $x = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$ of weight $\leq w$, such that $Hx^t = \mathbf{0}$ and $x_1 = 1$?

is NP-complete. Lobstein and Cohen [26] also used a polynomial transformation from k-DIMENSIONAL MATCHING (cf. [20, p. 58]) to show that the problem:

Instance: A binary $m \times n$ matrix H, an integer $w > 0$, and an integer $k \geq 3$.
Question: Is there a vector $x = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$ of weight $\leq w$, such that $Hx^t = \mathbf{0}$ and

$$x_1 = x_2 = \cdots = x_{\lfloor wk/(k+1) \rfloor} = 1?$$

is NP-complete. It is pointed out in [26] that all the eight problems are strikingly similar to MINIMUM DISTANCE, and hence provide further evidence to support the conjecture of [9] that MINIMUM DISTANCE is NP-complete. The ensemble of all these problems, however, does not suffice to prove this conjecture.

B. Some Observations on MAXIMUM-LIKELIHOOD DECODING

As mentioned in the Introduction, our proof of the NP-completeness of MINIMUM DISTANCE is based on a polynomial transformation from MAXIMUM-LIKELIHOOD DECODING. The particular transformation we will use places certain minor restrictions on MAXIMUM-LIKELIHOOD DECODING. Hence, our goal herein is to observe that MAXIMUM-LIKELIHOOD DECODING remains NP-complete under these restrictions.

First, we slightly modify the question of MAXIMUM-LIKELIHOOD DECODING by requiring that the solution to $Hx^t = s$ is nonzero. This restriction makes a difference only for $s = \mathbf{0}$, for if $s \neq \mathbf{0}$ then obviously any solution x to $Hx^t = s$ is nonzero. We therefore observe that the proof in [9] of the NP-completeness of MAXIMUM-LIKELIHOOD DECODING, based on the transformation from THREE-DIMENSIONAL MATCHING, uses only the special case where $s = (11 \cdots 1)^t$. Hence the same proof establishes that the minor variation of MAXIMUM-LIKELIHOOD DECODING discussed above is also NP-complete.

Next, as pointed out in [9], one may assume without loss of generality that the $m \times n$ matrix H at the input to MAXIMUM-LIKELIHOOD DECODING has full row rank. This implies that the columns of H contain a basis for $\mathbb{F}_2^m$, and we may further assume w.l.o.g. that $w \leq m - 1$. Indeed, if H is full-rank and $w \geq m$, then the answer to the question of MAXIMUM-LIKELIHOOD DECODING is trivially "Yes."

We also assume w.l.o.g. that the columns of H are distinct. If this is not so, then we can form (in polynomial time) an $m \times n'$ matrix H' by retaining a single representative from each set of equal columns of H. It is easy to see that $Hx^t = s$ has a solution of weight at most w, if and only if so does $H'x^t = s$, providing $s \neq \mathbf{0}$. But the case $s = \mathbf{0}$ may be safely excluded from the input, as discussed above. The assumption that H has distinct columns further implies that $n \leq 2^m$. These are all the assumptions that we will need.

The key idea in the transformation from MAXIMUM-LIKELIHOOD DECODING to MINIMUM DISTANCE is to regard the columns of the $m \times n$ parity-check matrix H as elements $\alpha_1, \alpha_2, \cdots, \alpha_n$ in the finite field $\mathrm{GF}(2^m)$. The syndrome vector $s \in \mathbb{F}_2^m$ may be also regarded as an element β in $\mathrm{GF}(2^m)$. With this notation, taking into account the restrictions discussed in the foregoing paragraphs, we may rephrase MAXIMUM-LIKELIHOOD DECODING as the finite-field version of SUBSET SUM (cf. [20, p. 233]), namely:

Problem: FINITE-FIELD SUBSET SUM
Instance: An integer $m > 0$, a set of $n \leq 2^m$ distinct elements $\alpha_1, \alpha_2, \cdots, \alpha_n \in \mathrm{GF}(2^m)$, a nonzero element $\beta \in \mathrm{GF}(2^m)$, and a positive integer $w \leq m - 1$.
Question: Is there a nonempty subset $\{\alpha_{i_1}, \alpha_{i_2}, \cdots, \alpha_{i_\delta}\}$ of $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$, such that

$$\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_\delta} = \beta$$

and $\delta \leq w$?

According to the discussion in this subsection, the NP-completeness of MAXIMUM-LIKELIHOOD DECODING immediately implies that FINITE-FIELD SUBSET SUM is NP-complete.

III. NP-Completeness for Codes of Characteristic Two

Given the input $\alpha_1, \alpha_2, \cdots, \alpha_n, \beta$, and w to FINITE-FIELD SUBSET SUM, we first construct a series of matrices $A_1, A_2, \cdots, A_w$, which may be thought of as parity-check matrices for the codes $\mathbb{C}_1, \mathbb{C}_2, \cdots, \mathbb{C}_w$ over GF(2^m). These matrices are constructed in such a way (see Lemma 1 below) that the minimum distance of $\mathbb{C}_\delta$ is equal to $\delta+1$ if $\alpha_{i_1}+\alpha_{i_2}+\cdots+\alpha_{i_\delta}=\beta$ for some $i_1, i_2, \cdots, i_\delta$. Otherwise, the minimum distance of $\mathbb{C}_\delta$ is equal to $\delta+2$, and $\mathbb{C}_\delta$ is an MDS code [27, p. 317]. The matrix A_1 is given by

$$A_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n & \beta \end{bmatrix} \tag{1}$$

and it is easy to see that the minimum distance of $\mathbb{C}_1$ is either 2 or 3, according as $\beta = \alpha_i$ for some $i = 1, 2, \cdots, n$ or not. In general, for $\delta = 2, 3, \cdots, w$, the matrix A_δ is given by

$$A_\delta = \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n & 0 \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{\delta-2} & \alpha_2^{\delta-2} & \cdots & \alpha_n^{\delta-2} & 0 \\ \alpha_1^{\delta-1} & \alpha_2^{\delta-1} & \cdots & \alpha_n^{\delta-1} & 1 \\ \alpha_1^{\delta} & \alpha_2^{\delta} & \cdots & \alpha_n^{\delta} & \beta \end{bmatrix}. \tag{2}$$

Notice that for all $\delta = 1, 2, \cdots, w$, the matrix A_δ has $n+1$ columns and $\delta+1$ linearly independent rows. Hence the dimension of $\mathbb{C}_\delta$ is $n-\delta$, and its minimum distance is at most $(n+1) - \dim \mathbb{C}_\delta + 1 = \delta+2$, by the Singleton bound [27, p. 33].

Lemma 1: Let d_δ denote the minimum distance of $\mathbb{C}_\delta$. Then $d_\delta = \delta+1$, if

$$\alpha_{i_1}+\alpha_{i_2}+\cdots+\alpha_{i_\delta}=\beta, \qquad \text{for some } i_1, i_2, \cdots, i_\delta$$

and $d_\delta = \delta+2$ otherwise.

Proof: Let M be a $(\delta+1)\times(\delta+1)$ square matrix consisting of some $\delta+1$ columns of A_δ. If the last column of A_δ, namely, $(0\cdots 01\beta)^t$, is not among the columns of M, then M is a Vandermonde matrix [27, p. 116]. Since $\alpha_1, \alpha_2, \cdots, \alpha_n$ are all distinct, M is nonsingular in this case. Otherwise, assuming w.l.o.g. that $(0\cdots 01\beta)^t$ is the last column of M, we expand along this column to obtain

$$\det M = \beta \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha_{i_1} & \alpha_{i_2} & \cdots & \alpha_{i_\delta} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{i_1}^{\delta-2} & \alpha_{i_2}^{\delta-2} & \cdots & \alpha_{i_\delta}^{\delta-2} \\ \alpha_{i_1}^{\delta-1} & \alpha_{i_2}^{\delta-1} & \cdots & \alpha_{i_\delta}^{\delta-1} \end{vmatrix} - \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha_{i_1} & \alpha_{i_2} & \cdots & \alpha_{i_\delta} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{i_1}^{\delta-2} & \alpha_{i_2}^{\delta-2} & \cdots & \alpha_{i_\delta}^{\delta-2} \\ \alpha_{i_1}^{\delta} & \alpha_{i_2}^{\delta} & \cdots & \alpha_{i_\delta}^{\delta} \end{vmatrix}$$

where $|\cdot|$ with respect to a matrix denotes the determinant. The first determinant on the right-hand side of the above expression is again a Vandermonde determinant, while the second one is a simple first-order alternant [10], [28]. Alternants were studied by Muir [28] and many others. In general, it is well known that

$$\begin{vmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_k \\ \vdots & \vdots & \vdots & \vdots \\ X_1^{j-1} & X_2^{j-1} & \cdots & X_k^{j-1} \\ X_1^{j+1} & X_2^{j+1} & \cdots & X_k^{j+1} \\ \vdots & \vdots & \vdots & \vdots \\ X_1^{k} & X_2^{k} & \cdots & X_k^{k} \end{vmatrix} = \mathcal{S}_{k-j}(\boldsymbol{X}) \prod_{1\le i_1<i_2\le k} (X_{i_2} - X_{i_1})$$

for $j = 1, 2, \cdots, k-1$, where $\mathcal{S}_r(\cdot)$ is the rth elementary symmetric function in the indeterminates $\boldsymbol{X} = X_1, X_2, \cdots, X_k$. A proof of the above expression may be found in Muir [28, vol. III, ch. 5], for instance. The elementary symmetric function $\mathcal{S}_r(\cdot)$ is defined by

$$\mathcal{S}_r(\boldsymbol{X}) \stackrel{\text{def}}{=} \sum_{1\le i_1<i_2<\cdots<i_r\le k} X_{i_1}X_{i_2}\cdots X_{i_r} \tag{3}$$

and in particular $\mathcal{S}_1(\boldsymbol{X}) = X_1 + X_2 + \cdots + X_k$. In our case, we indeed have $r = k-j = \delta-(\delta-1) = 1$ in (3), and the preceding expression for $\det M$ reduces to

$$\det M = -(\alpha_{i_1}+\alpha_{i_2}+\cdots+\alpha_{i_\delta}-\beta) \prod_{1\le a<b\le\delta} (\alpha_{i_b}-\alpha_{i_a}). \tag{4}$$

Since $\alpha_1, \alpha_2, \cdots, \alpha_n$ are distinct, the Vandermonde factor in (4) is nonzero, which implies that $\det M = 0$ if and only if $\alpha_{i_1}+\alpha_{i_2}+\cdots+\alpha_{i_\delta}=\beta$. Thus if no subset of exactly δ elements of $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ sums up to β, then every $\delta+1$ or less columns of A_δ are linearly independent. In this case, $d_\delta = \delta+2$ by the Singleton bound, and $\mathbb{C}_\delta$ is MDS. On the other hand, if $\alpha_{i_1}+\alpha_{i_2}+\cdots+\alpha_{i_\delta}=\beta$ for some $i_1, i_2, \cdots, i_\delta$, then obviously $d_\delta \le \delta+1$. Now, deleting the last row of A_δ, we obtain the parity-check matrix A_δ^* which defines the code $\mathbb{C}_\delta^*$ that contains $\mathbb{C}_\delta$ as a subcode. It is easy to verify (cf. [27, p. 323]) that $\mathbb{C}_\delta^*$ is an MDS code, and hence $d_\delta \ge d_\delta^* = \delta+1$. □

We observe that the MDS codes discussed in Lemma 1 are of independent interest; they were studied by Roth and Lempel in [30]. We also point out that the counterpart of Lemma 1 over the positive integers was proved by Khachiyan in [25]. In our context, it follows immediately from Lemma 1 that if we could find the minimum distance of a linear code over a field of characteristic 2 in polynomial time, we could solve FINITE-FIELD SUBSET SUM in polynomial time. Formally, consider the following problem:

Problem: MINIMUM DISTANCE OVER GF(2^m)
Instance: An integer $m>0$, an $r \times n$ matrix $\mathcal{H}$ over GF(2^m), an integer $\omega>0$.
Question: Is there a nonzero vector x of length n over GF(2^m), such that $\mathcal{H}x^t = \mathbf{0}$ and $\mathrm{wt}(x) \le \omega$?

One might argue that the operations in MINIMUM DISTANCE over GF(2^m), in short MD$_{2^m}$, are over the finite field GF(2^m), whereas the operations in MAXIMUM-LIKELIHOOD DECODING are over GF(2). If one were to implement the operations in GF(2^m) using a table of the field, for example, then this would require exponential memory. However, if we

implement the operations in GF(2^m) as polynomial addition and multiplication modulo an irreducible polynomial $g(x)$ of degree m, then only linear memory is required, and each operation in GF(2^m) can be carried out in polynomial time using operations in GF(2).

Proposition 2: Existence of a polynomial-time algorithm for MD$_{2^m}$ implies the existence of a polynomial-time algorithm for FINITE-FIELD SUBSET SUM.

Proof: Suppose that Φ is a polynomial-time algorithm for MD$_{2^m}$. Then, given the input to FINITE-FIELD SUBSET SUM, we construct the matrices $A_1, A_2, \cdots, A_w$ as in (1) and (2). We then run Φ with $\mathcal{H} = A_\delta$ and $\omega = \delta + 1$, for $\delta = 1, 2, \cdots, w$. It follows from Lemma 1 that if Φ returns "Yes" in at least one of these queries, then the answer to the question of FINITE-FIELD SUBSET SUM is "Yes," otherwise the answer is "No."

It is also easy to see that in each of the w queries, the length of the input to MD$_{2^m}$ is bounded by a polynomial in the length of the input to FINITE-FIELD SUBSET SUM. If the input $\alpha_1, \alpha_2, \cdots, \alpha_n$ and β to FINITE-FIELD SUBSET SUM takes $m(n+1) = O(n^2)$ bits, then the number of bits required to specify each matrix A_δ is $O(n^3)$, and the number of bits required to specify all of them is at most $O(n^4)$. Furthermore, each of these matrices can be obviously constructed in polynomial time from $\alpha_1, \alpha_2, \cdots, \alpha_n$ and β, using operations in GF(2^m). The only thing that is not entirely obvious is that GF(2^m) itself, namely, an irreducible polynomial $g(x)$ of degree m that defines GF(2^m), can be constructed in deterministic polynomial time. However, Shoup [32] provides a deterministic algorithm for this purpose, whose complexity is strictly less than $O(m^5)$ operations in GF(2). □

The procedure used in the proof of Proposition 2 is called "Turing reduction" in Garey and Johnson [20]. It uses a polynomial number (namely w, in our case) of queries to an oracle Φ for MD$_{2^m}$. Loosely speaking, a polynomial transformation is different from a Turing reduction in that it allows only a *single* query to an oracle. Turing reduction is sufficient to show that a problem is NP-hard, but not necessarily NP-complete, at least according to how this terminology is used in Garey and Johnson [20]. To prove that MD$_{2^m}$ is NP-complete, we need a polynomial transformation.

There are at least two alternative ways to convert our proof of Proposition 2 into a polynomial transformation. One way is to reduce directly from THREE-DIMENSIONAL MATCHING. The key observation here is that the reduction from THREE-DIMENSIONAL MATCHING to MAXIMUM-LIKELIHOOD DECODING in Berlekamp, McEliece, and van Tilborg [9] holds without change if we replace the phrase "of weight $\leq w$" with the phrase "of weight exactly w" in the question of MAXIMUM-LIKELIHOOD DECODING. This eliminates the need for multiple queries to Φ in the proof of Proposition 2, and establishes a polynomial transformation from THREE-DIMENSIONAL MATCHING to MD$_{2^m}$. However, we find some intrinsic merit in reducing to MD$_{2^m}$, and hence also to MINIMUM DISTANCE, from MAXIMUM-LIKELIHOOD DECODING rather than from THREE-DIMENSIONAL MATCHING. Therefore, we now describe a simple construction which shows that a single query to Φ would suffice to solve FINITE-FIELD SUBSET SUM, and hence also MAXIMUM-LIKELIHOOD DECODING.

As before, given $\alpha_1, \alpha_2, \cdots, \alpha_n, \beta$, and w, we first construct the matrices $A_1, A_2, \cdots, A_w$, given by (1) and (2), which define the codes $\mathbb{C}_1, \mathbb{C}_2, \cdots, \mathbb{C}_w$. Next, for $\delta = 1, 2, \cdots, w$, we let $\mathbb{C}'_\delta$ denote the linear code obtained by repeating each codeword of $\mathbb{C}_\delta$ exactly l_δ times. A parity-check matrix for $\mathbb{C}'_\delta$ is given by

$$H'_\delta = \left[\begin{array}{c|c|c|c|c} A_\delta & & & & \\ \hline I_{n+1} & -I_{n+1} & & & \\ \hline I_{n+1} & & -I_{n+1} & & \\ \hline \vdots & & & \ddots & \\ \hline I_{n+1} & & & & -I_{n+1} \end{array}\right] \quad (5)$$

where I_{n+1} is the $(n+1) \times (n+1)$ identity matrix and blanks denote zeros. Clearly, the length of $\mathbb{C}'_\delta$ is $l_\delta(n+1)$, its dimension is $n - \delta$, and its minimum distance is $d'_\delta = l_\delta d_\delta$ which is equal to either $l_\delta(\delta+1)$ or $l_\delta(\delta+2)$ by Lemma 1. The integers $l_1, l_2, \cdots, l_w$ are defined, recursively, as follows:

$$l_w = 2 + 3 + \cdots + w = \frac{w(w+1)}{2} - 1 \quad (6)$$

and

$$l_\delta = \left\lceil \frac{l_{\delta+1}(\delta+2)}{\delta+1} \right\rceil, \qquad \text{for } \delta = w-1, w-2, \cdots, 1. \quad (7)$$

Finally, we define the code $\mathbb{C}^\#$ over GF(2^m) as the direct sum of the codes $\mathbb{C}'_1, \mathbb{C}'_2, \cdots, \mathbb{C}'_w$. Thus a parity-check matrix for $\mathbb{C}^\#$ is given by

$$H^\# = \begin{bmatrix} H'_1 & & & \\ & H'_2 & & \\ & & \ddots & \\ & & & H'_w \end{bmatrix} \quad (8)$$

where $H'_1, H'_2, \cdots, H'_w$ are given by (5), and blanks again denote zeros. Clearly, the length of $\mathbb{C}^\#$ is

$$n^\# = (l_1 + l_2 + \cdots + l_w)(n+1)$$

its dimension is

$$k^\# = (n-1) + \cdots + (n-w) = O(n^2)$$

and its minimum distance is given by

$$d^\# = \min\{l_1 d_1, l_2 d_2, \cdots, l_w d_w\}.$$

We now show that the number of bits required to specify $H^\#$ is bounded by a polynomial in n. It is easy to see from (7) that $l_1 > l_2 > \cdots > l_w$, and, therefore,

$$n^\# = (l_1 + l_2 + \cdots + l_w)(n+1) \leq w l_1 (n+1) \leq l_1 n^2.$$

Using the relation

$$(\delta+1) l_\delta < (\delta+2) l_{\delta+1} + (\delta+1)$$

which follows from (7), it can be readily verified by (reverse) induction that for all $\delta = w-1, w-2, \cdots, 1$, we have

$$(\delta+1)l_\delta < (w+1)l_w + (\delta+1) + (\delta+2) + \cdots + w. \quad (9)$$

Substituting $\delta = 1$ in (9) yields

$$2l_1 < (w+1)l_w + (2+3+\cdots+w) = (w+2)l_w \le w^3 \quad (10)$$

where the last two inequalities follow from (6). Hence $n^\# \le l_1 n^2 \le w^3 n^2 = O(n^5)$, and the number of bits required to specify $H^\#$ is at most $n^\#(n^\# - k^\#)m = O(n^{11})$. Since the expressions in (5) and (8) are straightforward, this argument is all we need to prove that $H^\#$ can be constructed from $\alpha_1, \alpha_2, \cdots, \alpha_n, \beta$, and w in polynomial time.

We are now ready to prove that FINITE-FIELD SUBSET SUM can be solved using only a single query to an oracle for MD_{2^m}.

Theorem 3: MD_{2^m} is NP-complete.

Proof: Clearly, MD_{2^m} is in NP, since given a putative solution x, we can verify $\mathcal{H}x^t = \mathbf{0}$ and $\mathrm{wt}(x) \le \omega$ in polynomial time. We exhibit a polynomial transformation from FINITE-FIELD SUBSET SUM to MD_{2^m} as follows. Given the input to FINITE-FIELD SUBSET SUM, we construct in polynomial time the matrix $H^\#$ in (8), and then run the oracle Φ for MD_{2^m} with $\mathcal{H} = H^\#$ and $\omega = 2l_1$. By the definition of the integers $l_1, l_2, \cdots, l_w$ in (7), we have $(\delta+1)l_\delta \ge (\delta+2)l_{\delta+1}$ for all $\delta = 1, 2, \cdots, w-1$. This implies

$$2l_1 \ge 3l_2 \ge \cdots \ge (w+1)l_w \quad (11)$$

$$3l_1 \ge 4l_2 \ge \cdots \ge (w+2)l_w. \quad (12)$$

Now, suppose that the answer to the question of FINITE-FIELD SUBSET SUM is "Yes." Then it follows from Lemma 1 that $d_\delta = \delta + 1$ for at least one $\delta = 1, 2, \cdots, w$. Therefore,

$$\begin{aligned} d^\# &= \min\{l_1 d_1, l_2 d_2, \cdots, l_w d_w\} \\ &\le \max\{2l_1, 3l_2, \cdots, (w+1)l_w\} \\ &= 2l_1 \end{aligned} \quad (13)$$

in view of (11), and Φ will necessarily return "Yes." On the other hand, suppose that the answer to the question of FINITE-FIELD SUBSET SUM is "No." Then, by Lemma 1, we have $d_\delta = \delta + 2$ for all $\delta = 1, 2, \cdots, w$, and

$$\begin{aligned} d^\# &= \min\{l_1 d_1, l_2 d_2, \cdots, l_w d_w\} \\ &= \min\{3l_1, 4l_2, \cdots, (w+2)l_w\} \\ &= (w+2)l_w \\ &> 2l_1 \end{aligned} \quad (14)$$

where the third equality follows from (12), and the last inequality is precisely (10). Hence, in this case, Φ will necessarily return "No." □

Obviously, the NP-completeness of MD_{2^m} is a weaker result than the NP-completeness of MINIMUM DISTANCE, since the set of inputs to MINIMUM DISTANCE is a special case of the set of inputs to MD_{2^m}. However, Theorem 3 is a useful stepping stone in the proof of the NP-completeness of MINIMUM DISTANCE, which is the subject of the next section.

IV. NP-COMPLETENESS FOR BINARY CODES

Given the transformation from FINITE-FIELD SUBSET SUM to MD_{2^m} in Theorem 3, the NP-completeness of MINIMUM DISTANCE would follow if we could map, in polynomial time, the code $\mathbb{C}^\#$ constructed in (8) onto a binary linear code $\mathbb{C}$ in such a way that the minimum distance $d^\#$ of $\mathbb{C}^\#$ could be determined from the minimum distance d of $\mathbb{C}$. A mapping of this kind is exhibited in this section.

Certain simple mappings from codes over $\mathrm{GF}(2^m)$ to binary codes are well known [27, pp. 207–209]; however, none of these mappings is adequate for our purposes. For example, we could let $\mathbb{C}$ be the binary subfield subcode of $\mathbb{C}^\#$, as is commonly done in obtaining BCH codes from Reed–Solomon codes. In this case $d \ge d^\#$. Alternatively, one could let $\mathbb{C}$ be the trace code (cf. [27, p. 208]) of $\mathbb{C}^\#$, in which case $d \le d^\#$. Yet another option is to represent each element of $\mathrm{GF}(2^m)$ as a binary m-tuple (cf. [27, p. 298]), using a fixed basis for $\mathrm{GF}(2^m)$ over $\mathrm{GF}(2)$. In this case, we again have $d \ge d^\#$. All these mappings establish bounds on $d^\#$, and it can be shown that these bounds are reasonably tight. However, such mappings are not sufficient to determine the value of $d^\#$ exactly, which is what we need in the present context.

Instead, we will employ a concatenated, or multilevel, coding scheme [17], [19], using $\mathbb{C}^\#$ as the outer code. We let $\mathbb{C}^*$ denote the (n^*, k^*, d^*) binary linear code used as the inner code in the concatenation: namely, we require that $k^* = m$ and represent each element of $\mathrm{GF}(2^m)$ by a codeword of $\mathbb{C}^*$. Specifically, fix a basis $\beta_1, \beta_2, \cdots, \beta_m$ for $\mathrm{GF}(2^m)$ over $\mathrm{GF}(2)$ and a generator matrix G^* for $\mathbb{C}^*$. Then each element $\beta = b_1\beta_1 + b_2\beta_2 + \cdots + b_m\beta_m$ of $\mathrm{GF}(2^m)$ is mapped onto

$$\varphi(\beta) = (b_1, b_2, \cdots, b_m)G^* \quad (15)$$

which is a binary n^*-tuple. When this mapping is applied to $\mathbb{C}^\#$, the result is a binary linear code $\mathbb{C} = \varphi(\mathbb{C}^\#)$ of length $n^* n^\#$ and dimension $mk^\#$. It is obvious that a parity-check matrix H for $\mathbb{C}$ can be constructed in polynomial time from a parity-check matrix for $\mathbb{C}^\#$ and a generator matrix for $\mathbb{C}^*$. Henceforth, we let d denote the minimum distance of the concatenated code $\mathbb{C}$ constructed in this manner. The following lemma provides an upper bound on d in terms of $d^\#, n^*$, and $k^* = m$.

Lemma 4:

$$d \le n^* d^\# \frac{2^{m-1}}{2^m - 1}.$$

Proof: Since $\mathbb{C}^\#$ is a linear code over $\mathrm{GF}(2^m)$, if it contains a codeword c of weight $d^\#$, then it contains $2^m - 1$ such codewords, namely, all the multiples of c by the nonzero elements of $\mathrm{GF}(2^m)$. Let $c_1, c_2, \cdots, c_{2^m-1} \in \mathbb{C}^\#$ denote these $2^m - 1$ codewords, and consider the $(2^m - 1) \times n^\#$ matrix M having $c_1, c_2, \cdots, c_{2^m-1}$ as its rows. It is obvious that each of the $d^\#$ nonzero columns of M contains each of the $2^m - 1$ nonzero elements of $\mathrm{GF}(2^m)$ exactly once. Now let $c_1', c_2', \cdots, c_{2^m-1}' \in \mathbb{C}$ be the images of $c_1, c_2, \cdots, c_{2^m-1}$ under the mapping $\varphi(\cdot)$ and consider the

$(2^m-1)\times n^*n^\#$ matrix M' having $c'_1, c'_2, \cdots, c'_{2^m-1}$ as its rows. If some n^* columns of M' correspond to a nonzero position of $c \in \mathbb{C}^\#$, then every nonzero codeword of $\mathbb{C}^*$ appears exactly once in these n^* columns. It follows that the weight of each nonzero column of M' is precisely 2^{m-1}, and there are at most $n^*d^\#$ such columns. Thus the total weight of M' is at most $n^*d^\#2^{m-1}$. The lemma now follows by observing that M' has 2^m-1 rows. □

We note that Lemma 4 is just a variation of the well-known Plotkin bound [27, p. 41]. Yet, it provides exactly the kind of instrument we need for our purposes. Indeed, suppose that $d^\# \le 2l_1$ as in (13), where l_1 is defined by (6) and (7). Then Lemma 4 implies that

$$d \;\le\; 2l_1 n^* \frac{2^{m-1}}{2^m-1}. \tag{16}$$

On the other hand, suppose that $d^\# \ge 2l_1+1$ as in (14). Then, since $d \ge d^*d^\#$ by construction, we obviously have

$$d \;\ge\; d^*(2l_1+1). \tag{17}$$

In the present context, one is more interested in the reverse interpretation of the bounds in (16) and (17). Namely, given d (say, by an oracle for MINIMUM DISTANCE), we would like to distinguish between the two possibilities for $d^\#$. Fortunately, if

$$d^* > \; n^* \frac{2l_1}{2l_1+1}\,\frac{2^{m-1}}{2^m-1} \tag{18}$$

then the right-hand side of (17) is strictly greater than the right-hand side of (16). Thus our goal can be achieved, provided the minimum distance of $\mathbb{C}^*$ is sufficiently large.

We observe that $2l_1 < w^3$ in view of (10), and $w \le m-1$ as discussed in Section II-B. Thus in order to satisfy (18), it would certainly suffice to require that

$$\frac{d^*}{n^*} \;\ge\; \frac{m^3-1}{m^3}\,\frac{2^{m-1}}{2^m-1} \;=\; 0.5 - \frac{1}{m^3}\,\frac{2^m-m^3}{2(2^m-1)}. \tag{19}$$

These considerations may be translated into a specific set of conditions relating to the code $\mathbb{C}^*$ used as the inner code in our construction:

- **P1:** The length of $\mathbb{C}^*$ is bounded by a polynomial in n, and a generator matrix for $\mathbb{C}^*$ can be constructed in polynomial time.
- **P2:** The dimension of $\mathbb{C}^*$ is at least m (if $\dim \mathbb{C}^*$ is strictly greater than m, then any subcode of $\mathbb{C}^*$ will suffice for our purposes).
- **P3:** The ratio of the minimum distance of $\mathbb{C}^*$ to its length satisfies (19).

Less formally, what we need is a sequence of binary linear codes, whose relative distance approaches the Plotkin bound $d^*/n^* \le 0.5$, and whose rate tends to zero only polynomially fast as a function of their dimension. Furthermore, we should be able to construct each code in the sequence in polynomial time. This rules out codes that attain the Gilbert–Varshamov bound [27, p. 557], as well as Zyablov codes [40], since the complexity of Zyablov's construction [40] becomes exponential at low rates. Nevertheless, many other known constructions of asymptotically good families of low-rate codes suffice for our purposes: concatenated binary codes constructed in [38] from Drinfeld's modular curves, low-rate codes constructed in [31] using a variation of Justesen's concatenation, and codes constructed using expander graphs in [4] are just a few examples. As pointed out by a referee, duals of the binary BCH codes also have the required parameters, in view of the Carlitz–Uchiyama bound [27, p. 281]. In what follows, however, we shall use a simple construction, suggested by Noga Alon [3], which is concise enough to be completely described in one paragraph.

Alon's Construction: Given an integer $\nu \ge 2$ and a nonnegative integer $s \le 2^\nu - 2$, consider a concatenation of the $(2^\nu, s+1, 2^\nu - s)$ Reed–Solomon code over GF(2^ν) with the $(2^\nu-1, \nu, 2^{\nu-1})$ binary simplex code [27, p. 30]. The result is a binary linear code $\mathbb{C}^*(\nu, s)$ with the following parameters:

$$n^* = 2^\nu(2^\nu-1) \tag{20}$$

$$k^* = \nu(s+1) \tag{21}$$

$$d^* = 2^{\nu-1}(2^\nu - s). \tag{22}$$

Alon [3] notes that a generator matrix G^* for $\mathbb{C}^*(\nu, s)$ may be specified directly as follows. The columns of this matrix are indexed by pairs (x, y), where $x, y \in$ GF(2^ν) and $y \ne 0$, while its rows are indexed by integer pairs (i, j), where $i = 0, 1, \cdots, s$ and $j = 1, 2, \cdots, \nu$. Let $\alpha_1, \alpha_2, \cdots, \alpha_\nu$ be a basis for GF(2^ν) over GF(2). Then the entry in row (i, j) and column (x, y) is defined as $\langle \alpha_j x^i, y\rangle$, where $\alpha_j x^i$ is computed in GF(2^ν), and $\langle \cdot, \cdot \rangle$ denotes the inner product of $\alpha_j x^i$ and y as binary ν-tuples with respect to the basis $\alpha_1, \alpha_2, \cdots, \alpha_\nu$.

We take $s = m$ and $\nu = \lceil 5 \log_2 m \rceil$ in the foregoing construction. Then $\mathbb{C}^* = \mathbb{C}^*(\nu, s)$ trivially satisfies property **P2**, since $k^* = \nu(m+1) \ge m$. Furthermore,

$$n^* = \; 2^\nu(2^\nu - 1) \;\le\; 2^{2(5\log_2 m + 1)} = 4m^{10} = O(n^{10})$$

so that $\mathbb{C}^*$ also satisfies property **P1**. Thus the length $n^*n^\#$ of the concatenated code $\mathbb{C}$ is at most

$$\tfrac{1}{2}\, m^{10}(m-1)^4(n+1) = O(n^{15}).$$

Now, for our choice of s and ν, we have

$$\begin{aligned}\frac{d^*}{n^*} \;&=\; \frac{2^{\nu-1}(2^\nu - s)}{2^\nu(2^\nu-1)} \;\ge\; 0.5 - \frac{s}{2^{\nu+1}} \\ &\ge\; 0.5 - \frac{1}{2m^4} \ge\; 0.5 - \frac{1}{m^3}\,\frac{2^m - m^3}{2(2^m-1)}\end{aligned}$$

where the last inequality holds for all $m > 10$ (and follows straightforwardly from the fact that $2^m \ge m^3 + m^2 + m + 1$ for such m). Thus $\mathbb{C}^*$ also satisfies property **P3**. With both $\mathbb{C}^\#$ and $\mathbb{C}^*$ at hand, we are finally ready to prove our main result.

Theorem 5: MINIMUM DISTANCE is NP-complete.

Proof: Clearly, MINIMUM DISTANCE is in NP. A polynomial transformation from FINITE-FIELD SUBSET SUM to MINIMUM DISTANCE can be described as follows. Given the input $\alpha_1, \alpha_2, \cdots, \alpha_n, \beta \in$ GF(2^m) and w to FINITE-FIELD SUBSET SUM, we answer the question of FINITE-FIELD SUBSET SUM by exhaustive search if $m \le 10$. Otherwise, we construct

in polynomial time a parity-check matrix for the concatenated code $\mathbb{C}$ as described above. We then query an oracle for MINIMUM DISTANCE for the existence of a codeword of weight at most

$$2l_1 n^* \frac{2^{m-1}}{2^m - 1} \; = \; l_1 2^\nu (2^\nu - 1) \frac{2^m}{2^m - 1}$$

where l_1 is defined by (6) and (7), and $\nu = \lceil 5 \log_2 m \rceil$. By the foregoing discussion, the oracle for MINIMUM DISTANCE will return "Yes" if and only if the answer to the question of FINITE-FIELD SUBSET SUM is "Yes." □

This concludes the proof of the conjecture of Berlekamp, McEliece, and van Tilborg [9]. In the next section, we discuss certain extensions and consequences of this result.

V. FURTHER RESULTS AND CONCLUDING REMARKS

We note here that our proof of Theorem 5 can be immediately extended to codes over an arbitrary, fixed, finite field GF(q). This is based on the observation (cf. [7]) that the transformation from THREE-DIMENSIONAL MATCHING to MAXIMUM-LIKELIHOOD DECODING in [9] holds without change if the input to MAXIMUM-LIKELIHOOD DECODING is an $m \times n$ matrix H over GF(q), rather than a binary matrix. Given the NP-completeness of MAXIMUM-LIKELIHOOD DECODING over GF(q), one can essentially go through the proof in Sections II–IV, replacing each instance of 2 by q. There are a few intricate points along the way, that require some explanation.

First, in rephrasing MAXIMUM-LIKELIHOOD DECODING as FINITE-FIELD SUBSET SUM, one should leave the expression $\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_\delta} = \beta$ in the question of FINITE-FIELD SUBSET SUM as is, rather than ask whether β is a linear combination of $\alpha_{i_1}, \alpha_{i_2}, \cdots, \alpha_{i_\delta}$. This is certainly not the question that one would be concerned with for decoding purposes, but it is legitimate in an NP-completeness proof given the specific transformation from THREE-DIMENSIONAL MATCHING to MAXIMUM-LIKELIHOOD DECODING in [9]. (It is easy to see that a vector $x \in \text{GF}(q)^n$ of weight $\leq m/3$ satisfies $Hx^t = (11 \cdots 1)^t$ for the $m \times n$ incidence matrix H constructed in [9] only if all the $m/3$ nonzero positions in x are equal to 1.) Secondly, the bound in Lemma 4 becomes

$$d \; \leq \; n^* d^\# \frac{q^{m-1}}{q^{m-1} + q^{m-2} \cdots + q + 1}$$

and one has to modify (19) accordingly. Fortunately, Alon's construction [3] works in this case as well. Here, the columns of G^* would be indexed by $x, y \in \text{GF}(q^\nu)$, so that (21) remains without change, (20) becomes $n^* = q^\nu (q^\nu - 1)$, and (22) becomes

$$d^* \geq (q-1) q^{\nu-1} (q^\nu - s). \tag{23}$$

The key observation in the proof of (23) is as follows: if $\xi, y \in \text{GF}(q^\nu)$ and $\xi \neq 0$, then as y ranges over all the elements of GF(q^ν), the inner product $\langle \xi, y \rangle$ takes each value in GF(q) exactly $q^{\nu-1}$ times. (Alternatively, this can be viewed as a concatenation of the $(q^\nu, s+1, q^\nu - s)$ Reed–Solomon code over GF(q^ν) with the $(q^\nu - 1, \nu, (q-1)q^{\nu-1})$ first-order generalized Reed–Muller code over GF(q), see [8, p. 362].) To complete the proof, one can again take $s = m$ and $\nu = \lceil 5 \log_q m \rceil$ in this construction.

The complexity of approximation algorithms for NP-hard problems has been a subject of much research recently (see [6] and references therein), and it is natural to ask whether approximating the minimum distance of a linear code is still hard. Since our proof of the NP-completeness of MINIMUM DISTANCE is based on a transformation from MAXIMUM-LIKELIHOOD DECODING and it is known [5], [33] that MAXIMUM-LIKELIHOOD DECODING remains NP-complete under approximation within a constant factor, it is plausible that the same should be true for MINIMUM DISTANCE. We leave a more rigorous investigation of this question as an open problem.

Another immediate consequence of our proof is that certain useful computational tasks in coding theory are NP-hard, as there is an easy transformation from MINIMUM DISTANCE to each of these tasks. There is a large number of computational problems of this kind; we will give just three examples here.

First, we observe that determining whether a given linear code is MDS is NP-complete. Formally, let p be a fixed prime, and consider the following decision problem:

Problem: MDS CODE
Instance: Positive integers r, n, m, and an $r \times n$ matrix H over GF(p^m).
Question: Is there a nonzero vector x of length n over GF(p^m), such that $Hx^t = \mathbf{0}$ and $\text{wt}(x) \leq r$?

The fact that MDS CODE is NP-hard, even for $p = 2$, follows directly from Lemma 1. The NP-completeness of MDS CODE then follows from the observation that the phrase "of weight $\leq w$" in the question of MAXIMUM-LIKELIHOOD DECODING can be changed to the phrase "of weight exactly w," as discussed in Section III.

As another example, consider the problem of determining the trellis complexity of a linear code. More precisely, the computational task is to find a coordinate permutation that minimizes (the logarithm of) the number of vertices s_i at a given time i in the minimal trellis for a binary linear code. The corresponding decision problem [21] can be posed as:

Problem: PARTITION RANK
Instance: A binary $k \times n$ matrix H, and positive integers i and w.
Question: Is there a column permutation that takes H into a matrix $H' = [A_i | B_{n-i}]$, such that A_i is a $k \times i$ matrix and $\text{rank}(A_i) + \text{rank}(B_{n-i}) \leq w$?

This problem is important in the theory of block-code trellises (for more details on this, see [36]). Horn and Kschischang [21] recently proved that this problem is NP-complete, using an ingenious and elaborate transformation from SIMPLE MAX CUT [20, p. 210] which spans over five pages. On the other hand, given the NP-completeness of MINIMUM DISTANCE, this result can be established in a few lines as follows. First, observe that the least integer i for which

$$\text{rank}(A_i) + \text{rank}(B_{n-i}) \; < \text{rank}(H) + i$$

is equal to $\min\{d, d^\perp\}$ where $d, d^\perp$ denote, respectively, the distance and the dual distance of the code defined by H. Notice that it does not matter whether H is viewed as a parity-check or as a generator matrix in this problem. Now, suppose that $\mathbb{C}$ is an (n, k, d) binary linear code whose minimum distance we would like to determine, and let $d^\perp$ denote the dual distance of $\mathbb{C}$. Given $\mathbb{C}$, we first construct a binary linear Reed–Muller code $\mathbb{C}'$ of length 2^m and order r, where $m = 2\lceil \log_2 n \rceil + 1$ and $r = \lceil \log_2 n \rceil$. Then $\mathbb{C}'$ is an (n', k', d') self-dual code, where

$$\begin{aligned} n' &= 2^{2\lceil \log_2 n \rceil + 1} \leq 8n^2 \\ k' &= n'/2 \leq 4n^2 \\ d' &= 2^{m-r} = 2^{\lceil \log_2 n \rceil + 1} \geq 2n. \end{aligned}$$

We then use the well-known Kronecker product construction [27, p. 568] to obtain a generator matrix for the product code $\mathbb{C}^* = \mathbb{C}^\perp \otimes \mathbb{C}'$, where $\mathbb{C}^\perp$ is the dual code of $\mathbb{C}$. Evidently, the length of $\mathbb{C}^*$ is $n^* = nn' \leq 8n^3$, and its minimum distance is

$$d^* = d^\perp d' \geq 2nd^\perp \geq n > d.$$

On the other hand, it is easy to see that the dual distance of $\mathbb{C}^*$ is the minimum of the dual distances of $\mathbb{C}^\perp$ and $\mathbb{C}'$, namely, $\min\{d, d'\} = d$. Hence, running a polynomial-time algorithm for PARTITION RANK with the input H being a generator matrix for $\mathbb{C}^*$, we can determine d in polynomial time. The foregoing Turing reduction from MINIMUM DISTANCE shows that, given a linear code $\mathbb{C}$, computing either the minimum distance d or the minimum dual distance $d^\perp$ is NP-hard. This furthermore proves that PARTITION RANK remains NP-hard, even if the input is restricted to $w = \mathrm{rank}\,(H) + i - 1$. In other words, even if all we want to know is whether $s_i \neq i$ for some permutation, the computational task of determining this is still NP-hard. This is a somewhat stronger result than the one reported by Horn and Kschischang in [21].

Moreover, we believe that the techniques developed in the proof of NP-completeness of MINIMUM DISTANCE can be now used to show that determining the *maximum* trellis state-complexity of a code, namely $s_{\max} = \max_i s_i$, is also NP-complete. Indeed, Jain, Măndoiu, and Vazirani [22] have recently employed the results of Section III of this paper to prove that computing $s_{\max}$ is NP-hard for linear codes of characteristic 2, namely codes over $\mathrm{GF}(2^m)$ where m is variable. This result is similar in spirit to our Theorem 3, and the argument used by Jain, Măndoiu, and Vazirani [22] is essentially a variation of Lemma 1. We point out, however, that the problem is still open for binary codes.

As a third example, we mention the problem of finding the largest subcode with a prescribed contraction index [37]. Namely, given a $k \times n$ generator matrix for a binary linear code $\mathbb{C}$ and a positive integer λ, we wish to find the largest subcode $\mathbb{C}' \subseteq \mathbb{C}$ which has a generator matrix with at most $\lambda + \dim \mathbb{C}'$ distinct columns. This problem is of importance in soft-decision and majority-logic decoding (see [37] for an extensive treatment), and it is possible to show that it is NP-hard using a transformation from MINIMUM DISTANCE. The proof of this is a bit tedious, and we omit the details.

Finally, we would like to mention two important problems in coding theory, for which we do not have a polynomial transformation from MINIMUM DISTANCE, but believe that it should be possible to find one.

The first problem is that of bounded-distance decoding of binary linear codes. While the intractability of maximum-likelihood decoding has been thoroughly studied [5], [7], [9], [12], and [33], most of the decoders used in practice are bounded-distance decoders. It is still not known whether bounded-distance decoding is NP-hard for the general class of binary linear codes. For bounded-distance decoding up to the error-correction radius of a code, the corresponding decision problem can be formulated as follows:

Problem: BOUNDED-DISTANCE DECODING
Instance: An integer d, a binary $m \times n$ matrix H, such that every $d - 1$ columns of H are linearly independent, a vector $s \in \mathbb{F}_2^m$, and an integer $w \leq \lfloor (d-1)/2 \rfloor$.
Question: Is there a vector $x \in \mathbb{F}_2^n$ of weight $\leq w$, such that $Hx^t = s$?

Notice that BOUNDED-DISTANCE DECODING is not likely to be in NP, since in view of our main result in this paper, verifying that every $d - 1$ columns of H are linearly independent is NP-hard. Hence, this is an example of a *promise* problem (cf. [18]). Nevertheless, we could ask whether BOUNDED-DISTANCE DECODING is NP-hard. We concur with the remark of Barg [7], and conjecture that this is so. Moreover, we believe that the NP-completeness of MINIMUM DISTANCE should be instrumental in proving this conjecture.

We point out that a hardness result for bounded-distance decoding of binary linear codes in a somewhat different context was recently established in [16]. Downey, Fellows, Vardy, and Whittle [16] show that MAXIMUM-LIKELIHOOD DECODING is hard for the parametrized complexity class $W[1]$. Namely, it is unlikely that there exists an algorithm which solves MAXIMUM-LIKELIHOOD DECODING in time $f(w)n^c$, where c is a constant independent of w and $f(\cdot)$ is an arbitrary function. Many NP-complete problems are fixed-parameter-tractable. For example, VERTEX COVER, a well-known NP-complete problem [20, p. 53] which asks whether a graph G on n vertices has a vertex cover of size at most k, can be solved in time $O(kn + (4/3)^k k^2)$. Loosely speaking, the parametrized complexity hierarchy

$$\mathrm{FPT} = W[0] \subseteq W[1] \subseteq W[2] \subseteq \cdots$$

introduced by Downey and Fellows [14], [15] distinguishes between those problems that are fixed-parameter-tractable and those that are not. The result of [16] implies that bounded-distance decoding of linear codes is hard in the following sense: if a polynomial-time algorithm for this purpose exists then the parametrized complexity hierarchy collapses with $W[1] = \mathrm{FPT}$. Nevertheless, the question whether the BOUNDED-DISTANCE DECODING problem, as defined above, is NP-hard is still open.

The second problem we would like to mention is that of finding the shortest vector (in the Euclidean norm) in a sublattice of $\mathbb{Z}^n$. The overall status of computational problems for lattices is remarkably similar to the situation with linear codes. Peter van Emde Boas [35] proved in 1980 that finding

the nearest vector (which is equivalent to maximum-likelihood decoding) in a sublattice of $\mathbb{Z}^n$ is NP-hard, and conjectured that finding the shortest vector should be hard as well. Formally, van Emde Boas conjectured that the following problem:

Problem: SHORTEST VECTOR
Instance: A basis $v_1, v_2, \cdots, v_n \in \mathbb{Z}^n$ for a lattice Λ, and an integer $w > 0$,
Question: Is there a nonzero vector x in Λ, such that $||x||^2 \leq w$?

is NP-complete. Despite a considerable amount of work, the proof of this conjecture remains elusive. Arora, Babai, Stern, and Sweedyk [5] classify this as a "major open problem." Moreover, this conjecture becomes particularly significant in view of the celebrated result of Ajtai [1], who showed how to efficiently generate *hard instances* of certain computational problems related to integer lattices. Moreover, Ajtai [2] has recently proved that the SHORTEST VECTOR problem is hard for NP under randomized reductions. This comes very close to proving the conjecture of [35].

Intuitively, finding the shortest vector in a lattice should be at least as "difficult" as finding the minimum-weight vector in a binary linear code. Thus it is reasonable to suggest that there should be a polynomial transformation from MINIMUM DISTANCE to the SHORTEST VECTOR. Specifically, we pose the following problem: given a binary linear code $\mathbb{C}$ construct, in polynomial time, a lattice $\Lambda \subseteq \mathbb{Z}^n$ so that the minimum distance of $\mathbb{C}$ can be determined from the minimum norm of Λ. In view of our main result, solving this problem would amount to proving that SHORTEST VECTOR is NP-complete.

ACKNOWLEDGMENT

The author wishes to acknowledge helpful discussions with N. Alon, A. Barg, Y. Bresler, J. Bruck, I. Dumer, H. Edelsbrunner, M. R. Fellows, M. Naor, R. M. Roth, D. V. Sarwate, L. Schulman, and V. V. Vazirani. The author is especially indebted to N. Alon for referring him to the construction used in Section IV. Finally, the author would like to thank H. M. Itzkowitz for her invaluable help.

REFERENCES

[1] M. Ajtai, "Generating hard instances of lattice problems," in *Proc. 28th Annu. ACM Symp. on Theory of Computing* (Philadelphia, PA, May 1996), pp. 99–108.
[2] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," personal communication, May 1997.
[3] N. Alon, "Packings with large minimum kissing numbers," personal communication, Oct. 1996.
[4] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth, "Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs," *IEEE Trans. Inform. Theory*, vol. 38, pp. 509–516, 1992.
[5] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, "The hardness of approximate optima in lattices, codes, and systems of linear equations," in *Proc. 34th Annu. Symp. on the Foundation of Computer Science* (Palo Alto, CA, 1993), pp. 724–733.
[6] S. Arora and C. Lund, "Hardness of approximations," in *Approximation Algorithms for NP-Hard Problems*, D. S. Hochbaum, Ed. Boston, MA: PWS, 1997, pp. 399–446.
[7] A. Barg, "Some new NP-complete coding problems," *Probl. Pered. Inform.*, vol. 30, pp. 23–28, 1994 (in Russian).
[8] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[9] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 384–386, 1978.
[10] M. Blaum, J. Bruck, and A. Vardy, "On MDS codes and alternants over certain rings," *Abstracts Amer. Math. Soc.*, vol. 16, p. 454, Mar. 1995.
[11] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inform. Theory*, vol. 42, pp. 529–542, 1996.
[12] J. Bruck and M. Naor, "The hardness of decoding linear codes with preprocessing," *IEEE Trans. Inform. Theory*, vol. 36, pp. 381–385, 1990.
[13] P. Diaconis and R. L. Graham, "The Radon transform on $\mathbb{Z}_2^k$," *Pacific J. Math.*, vol. 118, pp. 176–185, 1985.
[14] R. G. Downey and M. R. Fellows, "Fixed parameter tractability and completeness: Basic theory," *SIAM J. Comput.*, vol. 24, pp. 873–921, 1995.
[15] ——, "Fixed parameter tractability and completeness: Completeness for $W[1]$," *Theoret. Comput. Sci. A*, vol. 141, pp. 109–131, 1995.
[16] R. G. Downey, M. R. Fellows, A. Vardy, and G. Whittle, "On the parametrized complexity of certain fundamental problems for linear codes and integer lattices," preprint, 1997.
[17] I. I. Dumer, "Concatenated codes and their generalizations," to be published in *Handbook of Coding Theory*, V. Pless, W. C. Huffman, and R. A. Brualdi, Eds. Amsterdam, The Netherlands: Elsevier.
[18] S. Even and Y. Yacobi, "Cryptography and NP-completeness," in *Lecture Notes on Computer Science*, vol. 85. Berlin, Germany: Springer-Verlag, 1982, pp. 195–207.
[19] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.
[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
[21] G. B. Horn and F. R. Kschischang, "On the intractability of permuting a block code to minimize trellis complexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 2042–2048, 1996.
[22] K. Jain, I. Măndoiu, and V. V. Vazirani, "The 'art of trellis decoding' is computationally hard—for large fields," *IEEE Trans. Inform. Theory*, to be published.
[23] D. S. Johnson, "The NP-completeness column: An ongoing guide," *J. Algorithms*, vol. 3, pp. 182–195, 1982.
[24] ——, "The NP-completeness column: An ongoing guide," *J. Algorithms*, vol. 7, pp. 584–601, 1986.
[25] L. Khachiyan, "On the complexity of approximating extremal determinants in matrices," *J. Complexity*, vol. 11, pp. 138–153, 1995.
[26] A. Lobstein and G. D. Cohen, "Sur la complexité d'un probléme de codage," *Theor. Informatics Appl.*, vol. 21, pp. 25–32, 1987.
[27] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.
[28] T. Muir, *Treatise on the Theory of Determinants*. New York: Dover, 1960.
[29] S. C. Ntafos and S. L. Hakimi, "On the complexity of some coding problems," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 794–796, 1981.
[30] R. M. Roth and A. Lempel, "A construction of non-Reed-Solomon type MDS codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 655–657, 1989.
[31] B.-Z. Shen, "A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate," *IEEE Trans. Inform. Theory*, vol. 39, pp. 239–242, 1993.
[32] V. Shoup, "New algorithms for finding irreducible polynomials over finite fields," *Math. Comput.*, vol. 54, pp. 435–447, 1990.
[33] J. Stern, "Approximating the number of error locations within a constant ratio is NP-complete," in *Lecture Notes on Computer Science*, vol. 673, Berlin, Germany: Springer-Verlag, 1993, pp. 325–331.
[34] M. A. Tsfasman and S. G. Vlăduţ, *Algebraic Geometry Codes*. Dodrecht, The Netherlands: Kluwer, 1991.
[35] P. van Emde Boas, "Another NP-complete partition problem and the complexity of computing short vectors in a lattice," Tech. Rep. 81–04, Dept. Math., Univ. of Amsterdam, Amsterdam, The Netherlands, 1980.
[36] A. Vardy, "Trellis structure of codes," to be published in *Handbook of Coding Theory*, V. Pless, W. C. Huffman, and R. A. Brualdi, Eds. Amsterdam, The Netherlands: Elsevier.
[37] A. Vardy, J. Snyders, and Y. Be'ery, "Bounds on the dimension of codes and subcodes with prescribed contraction index," *Linear Algebra Appl.*, vol. 142, pp. 237–261, 1990.
[38] S. G. Vlăduţ, G. L. Katsman, and M. A. Tsfasman, "Modular curves and codes with polynomial complexity of construction," *Probl. Pered. Inform.*, vol. 20, pp. 47–55, 1984 (in Russian).
[39] D. J. A. Welsh, "Combinatorial problems in matroid theory," in *Combinatorial Mathematics and its Applications*, D. J. A. Welsh, Ed. London, U.K.: Academic, 1971, pp. 291–307.
[40] V. V. Zyablov, "An estimate of the complexity of constructing binary linear concatenated codes," *Probl. Pered. Inform.*, vol. 7, pp. 5–13, 1971 (in Russian).