Minimal Vectors in Linear Codes

A. Ashikhmin and A. Barg

***Abstract*— Minimal vectors in linear codes arise in numerous applications, particularly, in constructing decoding algorithms and studying linear secret sharing schemes. However, properties and structure of minimal vectors have been largely unknown. We prove basic properties of minimal vectors in general linear codes. Then we characterize minimal vectors of a given weight and compute their number in several classes of codes, including the Hamming codes and second-order Reed–Muller codes. Further, we extend the concept of minimal vectors to codes over rings and compute them for several examples.**

Turning to applications, we introduce a general gradient-like decoding algorithm of which minimal-vectors decoding is an example. The complexity of minimal-vectors decoding for long codes is determined by the size of the set of minimal vectors. Therefore, we compute this size for long randomly chosen codes. Another example of algorithms in this class is given by zero-neighbors decoding. We discuss relations between the two decoding methods. In particular, we show that for even codes the set of zero neighbors is strictly optimal in this class of algorithms. This also implies that general asymptotic improvements of the zero-neighbors algorithm in the frame of gradient-like approach are impossible. We also discuss a link to secret-sharing schemes.

***Index Terms*— Minimal vectors, minimum distance decoding, Reed–Muller codes, secret sharing, zero neighbors.**

I. Introduction

The subject of this correspondence is minimal vectors in linear codes, i.e., vectors that do not cover other nonzero vectors except maybe proportional ones. Minimal vectors were extensively studied in combinatorics (cycles in linear matroids). In the coding context, minimal vectors were introduced in [14] where they were used to construct a minimum-distance decoding algorithm of linear codes (see Section IV). For the Euclidean space, this connection was again addressed in [1]. Recently, interest in this subject has been renewed in a series of works sparked by [17], where it was observed that minimal vectors in linear codes describe minimal access structures in linear secret sharing schemes defined by these codes.

We begin with general properties of collections of minimal vectors in linear codes. Then we consider some examples, computing minimal vectors in the Hamming, second-order Reed–Muller, and some other codes. It turns out that there exist linear codes all of whose nonzero vectors are minimal. Under the name of intersecting these codes were studied in [8]. The Carlitz–Uchiyama bound shows (see below) that codes dual to the binary Bose–Chaudhuri–Hocquengham (BCH) codes are intersecting. On the other hand, for BCH codes themselves the problem of characterizing minimal vectors seems difficult to approach. Even for two-error-correcting binary BCH codes a recent attempt [7] ended with only a partial result.

Next we show how to extend the concept of minimality to codes over Galois rings and compute minimal vectors in $\boldsymbol{Z}_4$ Kerdock codes, first-order Reed–Muller, and Hamming codes. Turning to the minimal-vectors decoding algorithm, we observe that the underlying idea is to construct a certain fixed set of code vectors used to successively improve the current decision. This idea bears similarity with methods of steepest descent in continuous spaces. This feature enables us to introduce a general gradient-like decoding algorithm of which minimal-vectors decoding and another known method, the zero-neighbors decoding [15], are examples. We show basic properties of this method, which allows us to analyze both examples in a simple and unified manner. Further, we show that under certain conditions, gradient-like algorithms must examine all zero neighbors, and therefore, the size of this set provides a *lower* bound on the complexity of algorithms in this class.

In the final section, we briefly review a link of our subject to secret-sharing schemes.

Manuscript received February 15, 1997; revised November 5, 1997.

A. Ashikhmin is with the Los Alamos National Laboratory, Mail Stop P990, Los Alamos, NM 87545 USA.

A. Barg was with the Department of Mathematics and Computing Science, Technical University of Eindhoven, Eindhoven, The Netherlands. He is now with Lucent Technologies, Bell Laboratories, Rm. 2C-375, Murray Hill, NJ 07974 USA.

Publisher Item Identifier S 0018-9448(98)05084-6.

II. Minimal Vectors in Linear Codes

A. General Properties

Let E_q^n be the n-dimensional coordinate space over the field $\boldsymbol{F}_q$. Let $C \subseteq E_q^n$ be an $[n, k, d]$ linear code. We use a shorthand notation $[n] := \{1, 2, \cdots, n\}$ for the set of code coordinates. A support of a vector $\boldsymbol{c}$ is defined as $\mathsf{supp}(\boldsymbol{c}) = \{i \in [n]: c_i \neq 0\}$. If $\mathsf{supp}(\boldsymbol{c}') \subset \mathsf{supp}(\boldsymbol{c})$ (respectively, $\subseteq$), we also write $\boldsymbol{c}' \prec \boldsymbol{c}$ (respectively, $\preceq$).

Definition: A nonzero vector $\boldsymbol{c} \in C$ is called *minimal* if $0 \neq \boldsymbol{c}' \preceq \boldsymbol{c}$ implies $\boldsymbol{c}' = a\boldsymbol{c}$, where $\boldsymbol{c}'$ is another code vector and a is a nonzero constant. The support of a minimal code vector is called *minimal* with respect to C.

Therefore, no minimal vector covers a nonzero code vector with a smaller support. Let $\mathcal{M}(C)$ be the set of minimal vectors of a given code C. If the context does not allow ambiguity, we omit C in this notation and write simply $\mathcal{M}$. For binary codes, $\mathcal{M}(C)$ can be also viewed as the set of minimal supports. In the general case, minimal supports define a set of lines in the code.

Let H be the parity-check matrix of C. By $H(U)$ we denote its restriction to columns indexed by a subset $U \subseteq [n]$. Basic properties of $\mathcal{M}$ are characterized in the following lemma.

Lemma 2.1:

1) Let $U \subset [n]$ be the support of a vector $\boldsymbol{c} \in C$. Then U is minimal if and only if $\mathsf{rk}(H(U)) = |U| - 1$.
2) (U is minimal) $\Rightarrow$ ($|U| \leq n - k + 1$).
3) Every support of size $|U| \leq d(1 + 1/(q-1)) - 1$ is minimal.
4) The linear span of $\mathcal{M}(C)$ coincides with C.
5) Let C be a binary code. Then if $\boldsymbol{c} \in C$, $\boldsymbol{c} \notin \mathcal{M}(C)$ there is a pair of nonzero code vectors $\boldsymbol{c}_1 \prec \boldsymbol{c}$ and $\boldsymbol{c}_2 \prec \boldsymbol{c}$ with disjoint supports such that $\boldsymbol{c} = \boldsymbol{c}_1 + \boldsymbol{c}_2$.

Proof: The *only if* part of Part 1) is obvious. Let us prove the converse. Let $\boldsymbol{h}_i$ be the ith column of $H(U)$. By the assumption, there exist $w = |U|$ nonzero numbers λ_i such that

$$\sum_{i=1}^{w} \lambda_i \boldsymbol{h}_i = 0$$

and some $w - 1$ of these columns, say the first, are linearly independent. Suppose there exists a code vector $\boldsymbol{c}'$, $\boldsymbol{c}' \prec \boldsymbol{c}$, i.e., there exists a vanishing linear combination of columns that does not involve at least one of the first $w - 1$ columns, for instance,

$$\sum_{i=2}^{w} \mu_i \boldsymbol{h}_i = 0$$

with $\mu_w \neq 0$. Multiply this sum by λ_w/μ_w and subtract from the first one. This gives a linear dependence between the first $w-1$ columns, a contradiction.

Part 2) is implied by Part 1).

To prove Part 3), suppose that $\boldsymbol{c} \in C$ is a nonminimal vector of weight $\mathsf{wt}(\boldsymbol{c}) \leq d(1+1/(q-1))-1$. Consider $q-1$ code vectors $\boldsymbol{c}-a\boldsymbol{c}'$, where a runs over all nonzero constants. Summing up their weights, we get $(q-1)\,\mathsf{wt}(\boldsymbol{c}) - \mathsf{wt}(\boldsymbol{c}')$. Thus their average weight is $\mathsf{wt}(\boldsymbol{c}) - (q-1)^{-1}\,\mathsf{wt}(\boldsymbol{c}')$. One of these vectors, say $\boldsymbol{c}''$ has weight at most the average. Together with our assumption this implies a contradiction

$$\begin{aligned}\mathsf{wt}(\boldsymbol{c}'') &\leq \mathsf{wt}(\boldsymbol{c}) - \frac{\mathsf{wt}(\boldsymbol{c}')}{q-1} \leq d\left(1+\frac{1}{q-1}\right) - 1 - \frac{d}{q-1} \\ &= d-1.\end{aligned}$$

Part 4) will follow from Lemma 4.3 below. Part 5) is obvious. □

Note that Part 1) of this lemma gives a straightforward way to check whether a given code vector is minimal.

This lemma enables one to give immediate characterization of minimal vectors in some codes.

Examples:

1) *Binary Golay Codes:* Let $C = \mathcal{G}_{23}$ be the binary $[23, 12, 7]$ Golay code. We have $n-k+2 = 2d-1 = 13$. Thus

$$\mathcal{M}(\mathcal{G}_{23}) = \{3335 \text{ vectors of weight } \leq 12\}$$

(this was found by a search algorithm in [1]). The same argument applies to the dual $[23, 11, 8]$ code $\mathcal{G}_{23}^{\perp}$, which gives

$$\mathcal{M}(\mathcal{G}_{23}^{\perp}) = \{1794 \text{ vectors of weights } 8 \text{ and } 12\}.$$

For the extended code $\mathcal{G}_{24}$, we have $n-k+2 = 2d-1$, and the answer is also obvious.

2) *Binary Intersecting Codes:* These codes were introduced in [8]. They are linear codes in which any pair of nonzero code vectors intersect. By Lemma 2.1, Part 5, this is equivalent to the fact that $\mathcal{M}(C) = C\backslash\{0\}$.

Let C be the binary code dual to the BCH code of length $n = 2^m - 1$ with designed distance $d = 2t+1$ and $t \leq \frac{1}{3}2^{(m/2)-1}$. Then by the Carlitz–Uchiyama bound [16, Ch. 9], the maximum weight D of C is bounded from above as $D \leq 2^{m-1} + (t-1)2^{m/2}$. By the same bound, the quantity $2d \geq 2^m - 2(t-1)2^{m/2} > D$. Thus $\mathcal{M}(C) = C\ \{0\}$ and C is intersecting [8, Proposition 9].

3) *Maximum-Distance-Separable (MDS) Codes:* In an $[n, k, d]$ MDS code C, the set of minimal vectors coincides with the set of all $(q-1)\binom{n}{d}$ codewords of weight d (by Part 2) of the lemma).

For an $[n, k, n-k]$ code C, the answer is generally not as obvious. However, there is a subclass of codes with these parameters, namely "near-MDS" codes of [9] for which it is easily given.

These codes are defined as follows. If a code C is MDS, then so is its dual $C^{\perp}$, and

$$d(C) + d(C^{\perp}) = (n-k+1) + (k+1) = n+2.$$

This is the largest possible value for this sum. If C is not MDS, then clearly $d(C) + d(C^{\perp}) \leq n$. A code is called near-MDS [9] if this holds with equality. This definition implies that any $k^{\perp}+1$ columns of the parity-check matrix of C have rank $k^{\perp}$ [9]. Thus $\mathcal{M}(C) = \{\text{vectors of weight } d \text{ and } d+1\}$.

B. Random Codes

To understand the structure of minimal vectors in long codes, let us suppose that C is a random linear code whose parity-check matrix has independent equiprobable entries. Let M_w be the number of minimal vectors in C and $\mathsf{E}M_w$ its average number of the ensemble over random linear codes.

Theorem 2.2: We have

$$\mathsf{E}M_w = \begin{cases} \binom{n}{w}\dfrac{(q-1)^w}{q^{n-k}} \displaystyle\prod_{i=0}^{w-2}(1-q^{-(n-k-i)}), & w \leq n-k+1 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Proof: Let $\pi_{n,k}(w)$ be the probability that a given support of size w is minimal. By the definition, code vectors sharing the same support are proportional, therefore, $\mathsf{E}M_w = (q-1)\binom{n}{w}\pi_{n,k}(w)$. The event considered is that some (say, first) $w-1$ columns of H among the chosen w columns are linearly independent and the remaining column is their linear combinations with $w-1$ nonzero coefficients. The number of collections of w columns that satisfy the above conditions equals

$$(q^{n-k}-1)(q^{n-k}-q)\cdots(q^{n-k}-q^{w-2})(q-1)^{w-1}$$

and the total number of choices is $q^{w(n-k)}$. The probability $\pi_{n,k}(w)$ equals the quotient of these quantities. □

Intuitive understanding of this result is acquired by asymptotic analysis. This is not only interesting in itself, but also is used below in Section IV to assess certain decoding algorithms. Let $n \to \infty$, $(n-k) \to \infty$. We shall compare the number of minimal vectors M_w with the number of all code vectors of weight w. Let N_w denote this number. The probability that a given vector satisfies a random check equation is q^{-1}; therefore, the probability that this vector is contained in a random code with $n-k$ checks equals $q^{-(n-k)}$. Thus

$$\mathsf{E}N_w = \binom{n}{w}\frac{(q-1)^w}{q^{n-k}} \tag{2}$$

a classical result of coding theory [10]. From this we see that the difference between $\mathsf{E}M_w$ and $\mathsf{E}N_w$ is in the factor

$$\prod_{i=0}^{w-2}(1-q^{-(n-k-i)}).$$

It will be seen that the asymptotic behavior of $\mathsf{E}M_w$ depends on the difference between w and $n-k+1$. Let $w = (n-k+1)-\ell$, $\ell \geq 0$. To simplify the analysis, we shall use the notation $t = n-k$, so that $\ell = t-w+1$. Using this notation, the product in question takes the form $\prod_{i=\ell+1}^{t}(1-q^{-i})$. Since we study its limit value as $n \to \infty$, we are interested in the behavior of the function

$$\gamma(q, \ell) := \prod_{i=\ell+1}^{\infty}(1-q^{-i}).$$

Its properties are given in the following lemma.

Lemma 2.3:

1) The product $\gamma(q, \ell)$ converges for any $\ell \geq 0$.
2) For $\ell \to \infty$ we have $\gamma(q, \ell) \to 1$.
3) For $\ell = \text{const}$, $1 - q^{-\ell} < \gamma(q, \ell) < 1$.
4) The function $\gamma(q, \ell)$ is monotone increasing in one argument if the other argument is fixed.

Proof: By [13, Theorem 353]

$$\begin{aligned}\gamma(q, 0) &= \sum_{i=-\infty}^{+\infty}(-1)^i q^{-(1/2)(3i^2+i)} \\ &= 1 - q^{-1} - q^{-2} + q^{-5} + q^{-7} - \cdots.\end{aligned}$$

It is known and can be easily checked that this series converges. The quantity $\gamma(q, \ell)$ for any fixed $\ell > 0$ differs from $\gamma(q, 0)$ by a

constant. This proves Part 1) for constant ℓ. Further, for any $\ell > 0$ we have

$$1 > \prod_{i=\ell+1}^{t} (1 - q^{-i}) > 1 - \sum_{i=\ell+1}^{t} q^{-i} > 1 - q^{-\ell}.$$

This proves the convergence of $\gamma(q, \ell)$ for ℓ growing and implies Parts 1)–3). Part 4) is obvious. □

Thus if w is not too close to $n-k+1$, then on the average almost all code vectors of weight w in a random code are minimal. Let us formulate this as a corollary.

Corollary 2.4: Let $n \to \infty$, $0 < w < (n-k+1) - \ell$, $\ell \to \infty$. Then $\lim_{n\to\infty} (\mathsf{E}M_w / \mathsf{E}N_w) = 1$.

If w differs from $n - k + 1$ by a constant, then the quotient $\mathsf{E}M_w/\mathsf{E}N_w$ tends to a constant between 0 and 1. In particular, from the series expansion for $\gamma(q, 0)$ we compute $\gamma(2, 0) = 0.288\cdots$, which is a familiar fraction of nonsingular square matrices over $\boldsymbol{F}_2$. Otherwise, $\gamma(q, \ell)$ is always greater than $1/2$. This is shown by computing $\gamma(3, 0) = 0.560\cdots$ and applying Lemma 2.3, Part 4). This shows that for all $q \geq 2$ and all $w \leq n - k + 1$ except for the case $q = 2$, $w = n - k + 1$ on the average more than half of code vectors of weight w are minimal.

The total average number of minimal vectors in a random code is given in the following corollary.

Corollary 2.5: Let $n \to \infty$, $k = Rn$, $0 < R < 1$. Then

$$\frac{1}{n} \log_q \mathsf{E}|\mathcal{M}| = \begin{cases} H_q(1-R) - (1-R), & 0 < 1 - R < \dfrac{q-1}{q} \\ R, & \dfrac{q-1}{q} \leq 1 - R < 1. \end{cases}$$

Here $H_q(\cdot)$ is the entropy function.

Proof: As long as $1 - R < (q-1)/q$, asymptotically the sum

$$\mathsf{E}|\mathcal{M}| = \sum_{w=0}^{n-k+1} \mathsf{E}M_w$$

is dominated by the term $\mathsf{E}M_{n-k+1}$. We have just shown that

$$\mathsf{E}M_{n-k+1} = \gamma(q, 0)\mathsf{E}N_{n-k+1}.$$

Conclude by using (2). □

In Section IV we use the variance of the number of minimal vectors in C. This has been estimated in [3]. We quote this result only for the binary case.

Theorem 2.6 [3]: Let C be a random binary linear code with distance d. Then

$$\operatorname{Var} M_w \leq \mathsf{E}M_w(1 + 2^{-d/2}\mathsf{E}M_w).$$

C. Hamming Codes

Let C be the q-ary Hamming code of length $n = (q^m - 1)/(q-1)$.

For the binary case, the required set of vectors forms a configuration defined by J. Steiner, from which later the modern notion of Steiner systems has been coined. Formula (3) is quoted in [12] with a reference to [20]. Its proof for any q is given below. Steiner's original definition is cited in the Appendix.

Theorem 2.7: The set $\mathcal{M}(C)$ is formed by M_w vectors of every weight w, $3 \leq w \leq m + 1$, where

$$M_w = \frac{1}{w!} \prod_{i=0}^{w-2} (q^m - q^i). \tag{3}$$

Proof: Consider $s = w - 1$ linearly independent columns in the parity-check matrix H of the code C. The total number of linear combinations of these columns with nonzero coefficients equals $(q-1)^s$; the $1/(q-1)$th fraction of them appear as columns in H distinct from the chosen columns (since they are linearly independent). Every choice of w linearly dependent columns of which $s = w - 1$ are linearly independent, defines a minimal code vector. Thus one has to count the number of distinct choices of s linearly independent columns in H. This number equals

$$\frac{1}{s!} n(n-1)\left(n - \frac{q^2-1}{q-1}\right) \cdots \left(n - \frac{q^{s-1}-1}{q-1}\right).$$

Taking into account that all the $\binom{w}{w-1}$ choices of $w - 1$ linearly independent columns within a given support of size w yield one and the same code vector, we find that the number of minimal vectors of weight w in the code equals

$$M_w = \frac{1}{(w-1)!w} n(n-1)\left(n - \frac{q^2-1}{q-1}\right) \cdots \left(n - \frac{q^{s-1}-1}{q-1}\right)(q-1)^s.$$

The substitution of the value of n gives the desired result. □

A similar argument in the binary case yields the following fact.

Theorem 2.8: In the extended Hamming code of length 2^m, the number of minimal codewords of even weight w, $4 \leq w \leq m + 2$, equals

$$M_w^{ex} = \frac{1}{w!} 2^m \prod_{i=0}^{w-3} (2^m - 2^i).$$

Proof: As above, we have to count the number of choices of w linearly independent columns in the parity-check matrix, of which $w - 1$ are linearly dependent. Since only half of the total of 2^{m+1} columns of length $m + 1$ are present in H, every $t - 1$ linearly independent columns forbid 2^{t-2} columns in H. Therefore, we can choose $w - 1$ linearly independent columns in

$$\frac{n}{(w-1)!} \prod_{i=0}^{w-3} (n - 2^i)$$

different ways. As above, this has to be divided by $\binom{w}{w-1}$. □

D. Second-Order Reed–Muller Codes

Let $C = \mathrm{RM}(2, m)$ be the second-order binary Reed–Muller code [16, Ch. 15]. Its parameters are $[n = 2^m, k = 1 + m + \binom{m}{2}, d = 2^{m-2}]$. Let A_w the number of vectors of weight w in C. Then $A_w = 0$ except for

$$w = 2^{m-1},\ w = 2^{m-1} \pm 2^{m-1-h}, \qquad 0 \leq h \leq \lfloor m/2 \rfloor \tag{4}$$

(see [16, ch. 15]). In particular, it is known that

$$A_d = (4/3)(2^m - 1)(2^{m-1} - 1).$$

Let M_w be the number of minimal vectors of weight $w > 0$ in C.

Theorem 2.9: For $w = 2^{m-1} + 2^{m-1-h}$, $h = 0, 1, 2,$ there are no minimal code vectors ($M_w = 0$). Otherwise, $M_w = A_w$, except for the case $w = 2^{m-1}$, when the number of nonminimal vectors equals

$$A_{2^{m-1}} - M_{2^{m-1}} = 2^{m+1} - 2 + A_d(2^{m-1} - 2). \tag{5}$$

Thus the only weights when there exist nonminimal codewords are $(3/4)n$, $(5/8)n$, n (all codewords) and $(1/2)n$ (part of them).

Proof: Let $\boldsymbol{c} \in C$ be nonminimal. Then by Lemma 2.1, Part 5) there are $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C \setminus \{0\}$ such that $\boldsymbol{c}_1 + \boldsymbol{c}_2 = \boldsymbol{c}$. Let w_1, w_2, w be the weights of these vectors. We have

$$w = w_1 + w_2 \geq 2d = 2^{m-1} = n/2. \tag{6}$$

First suppose that $w > n/2$. Then there are two possibilities, namely, either one of the weights w_1, w_2 equals $n/2$ or not. In the former case, (4) and (6) imply the following equality:

$$2^{m-1} + 2^{m-1-h} = 2^{m-1} + 2^{m-1} - 2^{m-1-h_1}$$

where h, h_1 are some integers between 1 and $\lfloor m/2 \rfloor$. This is possible only if $h = h_1 = 1$. Thus if either w_1 or w_2 equals $n/2$, we have the following subcase:

i) $(w, w_1, w_2) = (\frac{3}{4}n, \frac{1}{2}n, \frac{1}{4}n)$.

If $w > n/2$ and both w_1 and w_2 differ from $n/2$, then (6) yields the equation

$$2^{m-1} + 2^{m-1-h} = 2^{m-1} - 2^{m-1-h_1} + 2^{m-1} \pm 2^{m-1-h_2}$$

or

$$2^{-h} = 1 - 2^{-h_1} \pm 2^{-h_2}, \qquad h, h_1, h_2 \neq 0.$$

Obviously, this equality cannot be satisfied with the "+" sign whereas for the "−" the only possibilities for (h, h_1, h_2) are $(1, 2, 2)$ and $(2, 2, 1)$. This gives rise to two subcases:

ii) $(w, w_1, w_2) = (\frac{3}{4}n, \frac{3}{8}n, \frac{3}{8}n)$;
iii) $(w, w_1, w_2) = (\frac{5}{8}n, \frac{3}{8}n, \frac{1}{4}n)$.

This exhausts the possibilities for $w > n/2$. Let us examine them. All code vectors of one and the same weight $w \neq n/2$ are affinely equivalent, i.e., if there exists one nonminimal vector of weight w, then applying a suitable automorphism, one concludes that all code vectors of weight w are nonminimal. Suppose $(x_1, \cdots, x_m)$ are the affine coordinates on $F^m = \mathrm{AG}(m, 2)$. Then the code vector given by the incidence vector of the equation $x_1 x_2 = 0$ has weight $3n/4$ and covers the incidence vector (of weight $n/2$) of the hyperplane $x_1 = 0$. This shows that every code vector of weight $3n/4$ is nonminimal and is formed by a disjoint union of a vector of weight $n/2$ and a vector of weight $n/4$, while subcase ii) is never realized. Likewise, in case iii), the incidence vector of $x_1 x_2 + x_3 x_4 = 0$ has weight $5n/8$ and contains the vector given by $(x_1 + x_2)(x_3 + x_4) = 1$.

What is left is the case of $w = n/2$. This case is more difficult. Fortunately, the structure of nonminimal code vectors of weight $n/2$ is known. Let $\boldsymbol{c}$ be such a vector. Then $\boldsymbol{c}$ is a sum of two nonzero code vectors of minimal weight. By [16, Theorem 13.5], any vector of minimal weight in C corresponds to an $(m-2)$-dimensional flat in F^m. Hence the subset X of F^m corresponding to $\boldsymbol{c}$ is a disjoint union of two $(m-2)$-dimensional flats in F^m, say A_1 and A_2. Let V_1 and V_2 be the $(m-2)$-dimensional linear spaces parallel to A_1 and A_2, respectively. The disjointness of A_1 and A_2 implies that $\dim(V_1 + V_2) < m$. Hence either $V_1 = V_2$ and X is an $(m-1)$-flat or $W = V_1 \bigcap V_2$ has dimension $m-3$. The number N_1 of nonminimal vectors of weight $n/2$ of the first type equals the number of $(m-1)$-flats in F^m

$$N_1 = 2(2^m - 1).$$

In the second case, the image of X in the (three-dimensional) quotient space F^m/W is a set of four points that do not constitute an affine plane. Hence the total number of vectors X of this type equals

$$N_2 = \begin{bmatrix} m \\ m-3 \end{bmatrix} \left(\begin{pmatrix} 8 \\ 4 \end{pmatrix} - 2 \begin{bmatrix} 3 \\ 2 \end{bmatrix} \right).$$

Thus the number of nonminimal vectors $A_{2^{m-1}} - M_{2^{m-1}} = N_1 + N_2$, which gives the claimed number if one recalls the expression for A_d given before the theorem. □

Remark: The number of minimal vectors of weight $n/2$ in $\mathrm{RM}(2, m)$ equals

$$B_{n/2} = \sum_{h=2}^{\lfloor m/2 \rfloor} A_{2^{m-1}-2^{m-1-h}}(2^{m-2h+1} - 2). \tag{7}$$

III. Minimum Vectors in Codes Over Rings

Codes over Galois rings have been a subject of considerable attention lately. In this section we extend the definition of minimal vectors to this case and give some examples.

Let S be a finite commutative ring S with identity e, whose set of zero divisors has the form pS for a certain prime p, also known as a Galois ring. It is known [19] that $|S| = q^m$, $m \geq 1$, where $q = p^s$ for some $s \geq 1$, and the characteristic of S (the order of e in the group $(S, +)$) equals p^m. Since fixing the numbers p^m and q^m identifies S up to isomorphism, it may be also denoted as $\mathrm{GR}(q^m, p^m)$. All ideals of S form the following chain:

$$N_0 = S \supset N_1 = pS \supset N_2 = p^2 S \supset \cdots \supset N_{m-1} = p^{m-1} S \supset N_m = p^m S = 0 \tag{8}$$

and $|N_i| = q^{m-i}$. Consider a "linear" code C over S, i.e., a set of strings of n elements of S such that if $\boldsymbol{c}_1, \boldsymbol{c}_2 \in C$ then also $a_1 \boldsymbol{c}_1 + a_2 \boldsymbol{c}_2 \in C$ for any $a_1, a_2 \in S$, i.e., an S-module.

The original definition in Section II is not applicable in this case because of zero divisors in the ring. Namely, it is often possible to multiply a nonzero codeword by a nonzero constant so that it becomes all-zero. Therefore, in this section we find it more convenient to speak of supports than of codewords. Another reason is that S is not a vector space.

The number

$$T(\boldsymbol{c}) = \min_{i \in \mathrm{supp}(c)} \{u \colon c_i \in N_u\}$$

will be called the *type* of the word $\boldsymbol{c}$. Let us call the number $T(I) = \min_{\mathrm{supp}(c) = I} T(\boldsymbol{c})$ the type of a subset $I \subseteq [n]$. If there is no word with support I, the type of I is undefined.

Definition: A subset $I \subseteq [n]$ of type t is called minimal if there does not exist a codeword $\boldsymbol{c}$ with $T(\boldsymbol{c}) \leq t$ and $\mathrm{supp}\,\boldsymbol{c} \subset I$.

This yields a hierarchy of minimal subsets of types $0 \leq t \leq m-1$. The collection of type t minimal subsets will be denoted by $\mathcal{M}_t(C)$.

Examples:

4) Consider the first-order Reed–Muller code $\mathrm{ZRM}(1, v)$ of length $n = 2^v$ over $\boldsymbol{Z}_4$ [11]. Then there are two types of minimal words, namely, those of types 0 and 1. It can be easily seen that $\mathcal{M}_0$ consists of a single set $\widetilde{I} = [n]$ and $\mathcal{M}_1$ consists of $2^{v+1} - 2$ subsets (supports of words) of size $n/2$.

5) Let C be the $\boldsymbol{Z}_4$ Kerdock code of length $n = 2^v$, where v is an odd number, $v \geq 5$, [11], [18]. Then $\mathcal{M}_0$ is formed by the type 0 minimal subsets of sizes $2^{v-1} + 2^{v-2} \pm 2^{(v-3)/2}$ (the number of subsets of either size is $2^{v+1}(2^v - 1)$) and $\mathcal{M}_1$ consists of $2^{v+1} - 2$ subsets of size $n/2$. Therefore, all supports except the one of size n are minimal.

6) Let C be the $\boldsymbol{Z}_4$ "Hamming" code with the parity-check matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \end{bmatrix}$$

whose columns are formed by all the $n = 2^v$ possible vectors of zeros and twos, each preceded by a 1. This code is orthogonal over $\boldsymbol{Z}_4$ to the $\mathrm{ZRM}(1, v)$ code of Example 1. The binary image of this

code under the mapping $(0 \to 00,\ 1 \to 10,\ 2 \to 11,\ 3 \to 01)$ is a nonlinear $(2^{v+1},\ 2^{2^{v+1}-(v+1)-1},\ 4)$ code. Let $\mathcal{M} = \mathcal{M}_0 \cup \mathcal{M}_1$ be the set of minimal supports with respect to $\mathcal{H}_v$. We refer to [3] for the proof of the following theorem.

Theorem 3.1 [3]: The number of minimal supports of type 0 and size w in $\mathcal{H}_v$ equals

$$M_w^{(0)} = \frac{1}{w!}2^v \prod_{i=0}^{w-3}(2^v - 2^i), \qquad 4 \le w \le v+1,\ \ w \text{ even}. \tag{9}$$

Every pair of coordinates forms a minimal support of type 1, thus

$$M_2^{(1)} = \binom{n}{2}. \tag{10}$$

IV. Minimum Distance Decoding

In this and the next section we outline two applications of minimal vectors mentioned in the Introduction. We begin with minimum distance decoding algorithms. In this section we deal with binary codes only. We introduce a general gradient-like decoding algorithm and study its properties. One of the first works devoted to minimal vectors was paper [14], where they were used to construct such a decoding algorithm. This algorithm bears similarity to the steepest descent methods for computing optima in continuous spaces. Another example of algorithms of this type, the *zero-neighbors* decoding, was provided in [15]. Our results provide a framework for the study of algorithms of this type and show their limits.

The minimum distance decoding problem that we consider is formulated as follows. We are given a linear code $C \subseteq E_2^n$. The problem is to implement the mapping $f\colon E_2^n \to C$ such that

$$\forall_{\boldsymbol{x} \in E_2^n}\ \mathsf{dist}\,(\boldsymbol{x},\ f(\boldsymbol{x})) = \mathsf{dist}\,(\boldsymbol{x},\ C).$$

If for a certain $\boldsymbol{x}$, this is satisfied for many code vectors, the value of $f(\boldsymbol{x})$ is chosen arbitrarily from them. This function gives rise to the concept of Voronoi regions of code vectors in E_2^n. Let $\boldsymbol{c} \in C$, then the *Voronoi region* $D(\boldsymbol{c})$ is defined as follows:

$$D(\boldsymbol{c}) := \{\boldsymbol{x} \in E_2^n \mid \mathsf{dist}\,(\boldsymbol{x},\ \boldsymbol{c}) \le \mathsf{dist}\ \ (\boldsymbol{x},\ \boldsymbol{c}'),\ \boldsymbol{c}' \in C\}.$$

Any point of E_2^n is contained in at least one Voronoi region; some points fall into many regions. Note that geometrically Voronoi regions of different code vectors in a linear code C all have the same shape. Namely, the following property follows directly from the definition.

Lemma 4.1: Let $\boldsymbol{c},\ \boldsymbol{c}' \in C$ and let $\boldsymbol{x} \in D(\boldsymbol{c})$. Then $\boldsymbol{x} + \boldsymbol{c}' \in D(\boldsymbol{c} + \boldsymbol{c}')$.

Let us define the general gradient-like decoding method. A general principle of the decoding is to construct a set $\mathcal{T}$ of codewords in such a way that every vector $\boldsymbol{y}$ either lies in $D(0)$ or there exists a $\boldsymbol{z} \in \mathcal{T}$ such that

$$\mathsf{wt}\,(\boldsymbol{y} + \boldsymbol{z}) < \mathsf{wt}\,(\boldsymbol{y}). \tag{11}$$

Any set $\mathcal{T} \subseteq C$ satisfying this property will be called a *test set*. This suggests that the decoding can be accomplished by recursively inspecting the test set for the existence of such a vector $\boldsymbol{z}$ and subtracting it from the current vector. Let $\boldsymbol{y}$ be the received vector. Let us formulate the algorithm.

Gradient-like decoding:

1) Set $\boldsymbol{c} = 0$.
2) Find $\boldsymbol{z} \in \mathcal{T}$ such that $\mathsf{wt}\,(\boldsymbol{y} + \boldsymbol{z}) < \mathsf{wt}\,(\boldsymbol{y})$. Let $\boldsymbol{c} \leftarrow \boldsymbol{c} + \boldsymbol{z}$, $\boldsymbol{y} \leftarrow \boldsymbol{y} + \boldsymbol{z}$.
3) Repeat until no such $\boldsymbol{z}$ is found. Output $\boldsymbol{c}$.

Let us prove that this algorithm always converges to the nearest code vector.

Theorem 4.2: For any set of code vectors satisfying (11) the gradient-like algorithm performs a complete minimum-distance decoding. The time complexity of this algorithm is $\mathcal{O}\,(n^2|\mathcal{T}|)$. The space complexity is $\mathcal{O}\,(n|\mathcal{T}|)$.

Proof: Let $\boldsymbol{y} \notin D(0)$. The algorithm expands $\boldsymbol{y}$ into a sum of test vectors. Suppose that after m steps no further test vectors satisfying (11) are found. This means that we managed to bring $\boldsymbol{y}$ "down" to $D(0)$

$$\boldsymbol{e} = \boldsymbol{y} + \sum_{u=1}^{m} \boldsymbol{z}_u \in D(0).$$

By Lemma 4.1 this means that $\boldsymbol{y} \in D\left(\sum_{u=1}^{m} \boldsymbol{z}_u\right)$. □

Submitting a code vector $\boldsymbol{c} \neq 0$ to this algorithm, we observe that it constructs a decomposition of zero in the form

$$0 = \boldsymbol{c} + \sum_u \boldsymbol{z}_u.$$

In addition, we can observe that in each step the algorithm produces a vector of a strictly smaller weight. Let us formulate this as a lemma.

Lemma 4.3: Let $\mathcal{T} \subseteq C$ be a test set. Then any code vector $\boldsymbol{c} \neq 0$ can be decomposed into a sum

$$\boldsymbol{c} = \sum_{u=1}^{m} \boldsymbol{z}_u, \qquad \boldsymbol{z}_u \in \mathcal{T},\ m \ge 1$$

where

$$\mathsf{wt}\,(\boldsymbol{c}) > \mathsf{wt}\,(\boldsymbol{c} + \boldsymbol{z}_1) > \mathsf{wt}\,(\boldsymbol{c} + (\boldsymbol{z}_1 + \boldsymbol{z}_2)) > \cdots \ge 0.$$

Thus the linear span of $\mathcal{T}$ equals the entire code C.

The set $\mathcal{M}$ of minimal vectors of a binary code forms a test set.

Lemma 4.4: Minimal vectors in a binary linear code form a test set.

Proof: Let $\boldsymbol{y} \notin D(0)$. Then there is a code vector $\boldsymbol{c}$ such that $\mathsf{wt}\,(\boldsymbol{y} + \boldsymbol{c}) < \mathsf{wt}\,(\boldsymbol{y})$. If $\boldsymbol{c}$ is not minimal, then it can be decomposed into a sum $\boldsymbol{c} = \sum_u \boldsymbol{m}_u$ of minimal vectors with disjoint supports. Clearly, for at least one of these vectors, say $\boldsymbol{m}_1$, we must have $\mathsf{wt}\,(\boldsymbol{y} + \boldsymbol{m}_1) < \mathsf{wt}\,(\boldsymbol{y})$. □

Note that Lemma 2.1, Part 5) left without proof earlier now follows from the last two lemmas.

Therefore, minimal vectors can be used for decoding. To estimate the complexity of this decoding for long random codes, we use Corollaries 2.4, 2.5, and Theorem 2.6. First, Corollary 2.5 implies that the *average* decoding complexity for rates $0 < R < (q-1)/q$ behaves exponentially in the same way as that of the exhaustive search. To estimate the worst case complexity, we use the expression for the variance in Theorem 2.6. This amounts in standard calculations using Stirling approximation (see [3]) that we omit. The conclusion is that, at least for low code rates, the worst case complexity of minimal-vectors decoding has the same order of magnitude as the average-case complexity. Note that in examples the number of minimal code vectors can be much smaller than the total size of the code. This is the case for all codes whose distance is close to $n-k+1$ since then many vectors have weight greater than $n-k+1$ and cannot be minimal. An extreme example is MDS codes (Example 3 in the previous section). Another example is Hamming codes. Namely, using (3) we see that as $n \to \infty$, the number of minimal vectors is of exponential order at most $q^{m^2} = q^{\log_q^2 n(1+o(1))}$. The total number of code vectors is $q^{n-\mathcal{O}(\log_q n)}$.

Another example of decoding algorithms in this class was given in [15]. Let $A \subset E_2^n$ and let $\mathcal{X}(A)$ be formed by all the points of E_2^n at a distance 1 from A

$$\mathcal{X}(A) = \{\boldsymbol{x} \,|\, \mathsf{dist}\,(\boldsymbol{x},\, A) = 1\}.$$

Define the *boundary* of A as follows:

$$\partial A = \mathcal{X}(A) \cup \mathcal{X}(\overline{A}).$$

Definition: Two code vectors $\boldsymbol{c}_1, \boldsymbol{c}_2$ are called *neighbors* if their Voronoi regions share a common boundary, i.e., if $\partial D(\boldsymbol{c}_1) \cap \partial D(\boldsymbol{c}_2) \neq \emptyset$. A neighbor of the zero vector is called a *zero neighbor*.

Note that here we deviate slightly from [15]. This enables us to give the definition of zero neighbors in symmetric form.

Let $\mathcal{Z}$ be the set of zero neighbors. The definition has the following simple consequence:

$$(\mathcal{X}(D(0)) \cap D(\boldsymbol{z}) \neq \emptyset) \quad \Rightarrow \quad \boldsymbol{z} \in \mathcal{Z}. \tag{12}$$

Indeed, $\boldsymbol{x} \in \mathcal{X}(D(0)) \cap D(\boldsymbol{z})$ implies that there is a $\boldsymbol{y} \in D(0)$ at a distance 1 from $\boldsymbol{x}$. Hence $\boldsymbol{y} \in \partial D(0) \cap \partial D(\boldsymbol{z})$.

Decoding with zero neighbors proceeds in the same way as with minimal supports except that now we choose the test set $\mathcal{T}$ in Algorithm 2.1 equal to $\mathcal{Z}$. This version of the algorithm is called *zero-neighbors decoding*, first introduced in [15].

The zero-neighbors decoding always converges to the closest code vector. To justify this we again verify that $\mathcal{Z}$ is a test set.

Theorem 4.5 [15]: The zero-neighbors algorithm performs a complete minimum distance decoding.

Proof: Let $\boldsymbol{y} \notin D(0)$. Consider a chain of inclusions

$$0 \prec \ldots \prec \boldsymbol{y}_2 \prec \boldsymbol{y}_1 \prec \boldsymbol{y}_0 = \boldsymbol{y}$$

where $\mathsf{wt}\,(\boldsymbol{y}_i) = \mathsf{wt}\,(\boldsymbol{y}_{i-1}) - 1$. Clearly, there exists a number i such that $\boldsymbol{y}_{i+1} \in D(0)$ and $\boldsymbol{y}_i \in \partial D(0) \setminus D(0)$. Then $\boldsymbol{y}_i \in D(\boldsymbol{z})$ for some $\boldsymbol{z} \in \mathcal{Z}$. We have

$$\begin{aligned}\mathsf{wt}\,(\boldsymbol{y} - \boldsymbol{z}) = \mathsf{dist}\,(\boldsymbol{y},\, \boldsymbol{z}) &\leq \mathsf{dist}\,(\boldsymbol{y},\, \boldsymbol{y}_i) + \mathsf{dist}\,(\boldsymbol{y}_i,\, \boldsymbol{z}) \\ &< \mathsf{dist}\,(\boldsymbol{y},\, \boldsymbol{y}_i) + \mathsf{dist}\,(\boldsymbol{y}_i,\, 0) = \mathsf{wt}\,(\boldsymbol{y}).\end{aligned}$$

Hence $\mathcal{Z}$ is a test set and the theorem follows. □

The complexity of zero-neighbors decoding was estimated in [15] as follows.

Theorem 4.6 [15]: For almost all codes, both time and space complexity of zero-neighbors decoding behaves as $2^{\alpha(R)n(1+o(1))}$, where

$$\alpha(R) = \begin{cases} R, & 0 \leq R \leq 1 - H_2(1/4) \\ (H_2(2\delta_0) - (1 - R)), & 1 - H_2(1/4) < R \leq 1 \end{cases}$$

where δ_0 is the smallest positive root of $R = 1 - H_2(\delta)$.

The memory used by the algorithm is spent on storing zero neighbors. Therefore, $\alpha(R)$ also gives an estimate of the exponent of the size of $\mathcal{Z}$ for most long codes. This size grows slower than the total size of the code for $R > 1 - H_2(1/4) \approx 0.189$.

We conclude that the complexity of this decoding for almost all codes and for $R > 0.189$ is exponentially smaller than that of minimal-vectors decoding.

Two last results of this section deal with characterization theorems for zero neighbors and minimal vectors in linear codes. Let us first take a closer look at the set of zero neighbors. The only property of the set $\mathcal{Z}$ that is essential for the successful decoding is formulated in (12)

$$\mathcal{X}(D(0)) \subset \bigcup_{\boldsymbol{z} \in \mathcal{Z}} D(\boldsymbol{z}). \tag{13}$$

Thus we may further restrict the test set of vectors by choosing a *smallest* subset of $\mathcal{Z}$ with this property. Denote this subset by $\mathcal{Z}_{\min}$. (This is how zero neighbors were originally defined in [15].) Note that though the set $\mathcal{Z}_{\min}$ may not be unique, its size is well-defined. Therefore, let $Z_{\min} = |\mathcal{Z}_{\min}|$.

First, we prove that for codes with only even weights of codewords zero neighbors in the set $\mathcal{Z}_{\min}$ form a test set of the *smallest* possible size.

Theorem 4.7: Let C be a binary linear code all of whose codewords have even weight and let $\mathcal{T} \subseteq C$ be a test set. Then $|\mathcal{T}| \geq Z_{\min}$.

Proof: Let $y \in \mathcal{X}(D(0))$ and let $\boldsymbol{z} \in \mathcal{T}$ be such a vector that $\mathsf{wt}\ (\boldsymbol{y} - \boldsymbol{z}) < \mathsf{wt}\,(\boldsymbol{y})$. Since $\mathsf{dist}\,(\boldsymbol{y},\, D(0)) = 1$, we can choose a vector $\boldsymbol{x} \in D(0)$ with $\mathsf{dist}\,(\boldsymbol{x},\, \boldsymbol{y}) = 1$ and $\boldsymbol{x} \prec \boldsymbol{y}$. We have

$$\begin{aligned}\mathsf{dist}\,(\boldsymbol{z},\, \boldsymbol{y}) &< \mathsf{dist}\,(0,\, \boldsymbol{y}) = \mathsf{dist}\,(0,\, \boldsymbol{x}) + 1 \\ &\leq \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{x}) + 1, \qquad \forall\, \boldsymbol{c} \in C.\end{aligned} \tag{14}$$

Clearly, for any $\boldsymbol{c} \in C$ we have $\mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{x}) = \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{y}) \pm 1$.

a) Consider the subset $C' \subseteq C$ for which

$$\mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{x}) = \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{y}) - 1.$$

Then (14) implies

$$\mathsf{dist}\,(\boldsymbol{z},\, \boldsymbol{y}) < \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{y}), \qquad c \in C'. \tag{15}$$

b) Let $C'' \subseteq C$ be the subset of codewords for which

$$\mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{x}) = \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{y}) + 1.$$

Definition (11) implies $\mathsf{dist}\,(0,\, \boldsymbol{y}) - \mathsf{dist}\,(\boldsymbol{z},\, \boldsymbol{y}) \geq 1$. Suppose that this holds with equality. Let $\mathsf{wt}\,(\boldsymbol{y}) = w$, then

$$\mathsf{dist}\,(\boldsymbol{z},\, \boldsymbol{y}) = \mathsf{wt}\,(\boldsymbol{z}) + \mathsf{wt}\,(\boldsymbol{y}) - 2\mathsf{wt}\,(\boldsymbol{z} \cap \boldsymbol{y})$$

or

$$2\mathsf{wt}\,(\boldsymbol{z} \cap \boldsymbol{y}) = \mathsf{wt}\,(\boldsymbol{z}) + 1.$$

This contradicts our assumption that C has only even weights. Therefore,

$$\mathsf{dist}\,(0,\, \boldsymbol{y}) - \mathsf{dist}\,(\boldsymbol{z},\, \boldsymbol{y}) \geq 2.$$

Then (14) implies

$$\begin{aligned}\mathsf{dist}\,(\boldsymbol{z},\, \boldsymbol{y}) &\leq \mathsf{dist}\,(0,\, \boldsymbol{y}) - 2 \leq \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{x}) - 1 \\ &= \mathsf{dist}\,(\boldsymbol{c},\, \boldsymbol{y}), \qquad \forall c \in C''.\end{aligned} \tag{16}$$

Inequalities (15) and (16) together imply that

$$\boldsymbol{y} \in D(z).$$

Running over all $\boldsymbol{y} \in \mathcal{X}(D(0))$, we collect a subset $\mathcal{T}' \subset C$ with

$$\mathcal{X}(D(0)) \subset \bigcup_{\boldsymbol{z} \in \mathcal{T}'} D(\boldsymbol{z}).$$

Then $|\mathcal{T}| \geq |\mathcal{T}'| \geq Z_{\min}$. □

Since $\mathcal{M}$ is a test set, this theorem implies that for C an even binary linear code, $|\mathcal{M}| \geq Z_{\min}$. However, it is possible to prove a stronger fact, namely, that in any even binary linear code there is a set $\mathcal{Z}_{\min}$ all of whose elements are minimal codewords.

Theorem 4.8: Let C be a binary linear code with only even weights of codewords. Then the set $\mathcal{Z}_{\min}$ can be chosen so that $\mathcal{Z}_{\min} \subseteq \mathcal{M}$.

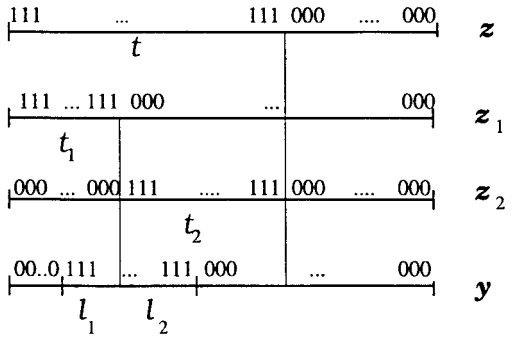

Fig. 1.

Proof: Let us assume that there is a codeword $\boldsymbol{z} \in \mathcal{Z}$, $\boldsymbol{z} \notin \mathcal{M}$ and let $\boldsymbol{y} \in E_2^n$ be a vector such that $\boldsymbol{y} \in \mathcal{X}(D(0))$, $\boldsymbol{y} \in D(\boldsymbol{z})$. Since $\boldsymbol{z}$ is not minimal, there are nonzero vectors $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ with disjoint supports such that $\boldsymbol{z} = \boldsymbol{z}_1 + \boldsymbol{z}_2$. Let

$$\mathsf{wt}(\boldsymbol{z}) = t \qquad \mathsf{wt}(\boldsymbol{z}_1) = t_1 \qquad \mathsf{wt}(\boldsymbol{z}_2) = t_2.$$

We want to show that if one of the vectors $\boldsymbol{z}_1$, $\boldsymbol{z}_2$ is farther from $\boldsymbol{y}$ than $\boldsymbol{z}$, then the other one is at most as far as $\boldsymbol{z}$.

By our assumptions,

$$\ell_1 + \ell_2 = \mathsf{dist}(0, \boldsymbol{y}) = \frac{t}{2} + 1 \qquad \mathsf{dist}(\boldsymbol{z}, \boldsymbol{y}) = \frac{t}{2} - 1.$$

Let $\mathsf{dist}(\boldsymbol{z}_2, \boldsymbol{y}) > \mathsf{dist}(\boldsymbol{z}, \boldsymbol{y})$. We then plug in our notation and perform straightforward computations using the Fig. 1 to find that $\mathsf{dist}(\boldsymbol{z}_1, \boldsymbol{y}) \leq t/2 - 1$.

Thus $\boldsymbol{y} \in D(\boldsymbol{z})$ and $\boldsymbol{y} \in D(\boldsymbol{z}_1)$, i.e., $\boldsymbol{z}$ and $\boldsymbol{z}_1$ cannot both be in the set $\mathcal{Z}_{\min}$ at the same time. Moreover, given a nonminimal code vector ($\boldsymbol{z}$ in our case) and a vector $\boldsymbol{y} \in \mathcal{X}(D(0))$, $\boldsymbol{y} \in D(\boldsymbol{z})$, we can always cast it away so that the remaining subset of zero neighbors still satisfies condition (13). Therefore, $\mathcal{Z}_{\min}$ can be chosen to be a subset of $\mathcal{M}$. □

For more details and a general overview we refer to [4].

Remarks:

i) Generally, not all zero neighbors are minimal. Indeed, consider the code $\{0000, 1100, 0011, 1111\}$. Then vector 0110 lies equally far from all the code vectors which proves that all nonzero code vectors are zero neighbors. However, the all-one vector is not minimal. Looking at smallest sets of zero neighbors defined by (13) we easily see that $\boldsymbol{z} \in \mathcal{Z}_{\min}$ implies $\mathsf{wt}(\boldsymbol{z}) \leq 2\,(\text{covering radius of } C) - 1$. Let C be a binary linear code such that its covering radius equals at most its minimum distance. For instance, long BCH codes are known to satisfy this. By Lemma 2.1, Part 3), in such codes *any* set $\mathcal{Z}_{\min}$ is formed by minimal code vectors.

ii) In view of Theorem 4.7, the set $\mathcal{Z}_{\min}$ is in the general case unavoidable in gradient-like decoding methods. For this reason it is no surprise that in the case of arbitrary q the zero-neighbors algorithm is also applicable and leads to similar results [4]. Interestingly, minimal vectors do not always form a test set in q-ary linear codes.

V. Secret Sharing

A general introduction to secret sharing schemes can be found for instance in Stinson's survey article [21]. Some familiarity with this concept is helpful in reading this section. The relation to linear codes was observed in [17] and analyzed in [6]. In the context of secret-sharing schemes one coordinate of the code is associated with values of the secret information and the remaining $n-1$ coordinates are associated with users of a system of restricted access to the secret. Let $H = \|h_{ij}\|$, $1 \leq i \leq r$, $1 \leq j \leq n$, be a matrix with entries from $\boldsymbol{F}_q$. Define a linear transformation ϕ: $E_q^m \to E_q^n$ by $\phi(\boldsymbol{e}) = \boldsymbol{e}H$, $\boldsymbol{e} \in E_q^m$. Suppose the first coordinate of $\phi(\boldsymbol{e})$ carries the value of the secret. The remaining coordinates contain shares of information given to the $n-1$ users. It can be shown [6] that users corresponding to nonzero entries in $(\phi_2(\boldsymbol{e}), \phi_3(\boldsymbol{e}), \cdots, \phi_{n-1}(\boldsymbol{e}))$, putting their shares together, can uniquely reconstruct the secret. Each such group of users is called an authorized coalition. Any group of users that does not form an authorized coalition is called unauthorized. When $\boldsymbol{e}$ runs over E_q^n, we obtain the entire set of authorized coalitions, called the *access structure* of the scheme. If no unauthorized coalition can obtain any *a posteriori* information of the secret value, the scheme is called *perfect*. A *minimal authorized coalition* is an authorized coalition that becomes unauthorized upon deletion of any of the users. The set of minimal authorized coalitions provides a complete description of a perfect secret-sharing scheme.

Viewing H as a parity-check matrix of a linear code C, one can establish a one-to-one correspondence between minimal authorized coalitions and a subset of minimal supports in C.

Theorem 5.1 [6], [17]: Let $\mathcal{C}$ be a linear secret-sharing scheme defined by a q-ary $r \times n$ matrix H and let $C = \ker H$ be an $[n, n-r]$ q-ary linear code. Then the set of minimal supports in C intersecting the first coordinate equals the set of minimal authorized coalitions in $\mathcal{C}$. Moreover, the scheme is perfect.

For some of the above examples it is easy to find minimal supports intersecting the first (or any other fixed) coordinate.

Examples 1–3 (Continued): In the extended Golay code $\mathcal{G}_{24}$ a code vector is minimal if and only if its weight is 8 or 12. Since puncturing $\mathcal{G}_{24}$ in any coordinate we get $\mathcal{G}_{23}$, the number of minimal vectors with a one in any fixed coordinate is the same.

The same holds for binary intersecting codes, namely, the number of minimal vectors with a one in any fixed coordinate is $|C|/2$.

The only minimal supports in an $[n, k, d]$ MDS code are $\binom{n}{d}$ supports of size d. Of them $\binom{n-1}{d-1}$ intersect the first (or any fixed) coordinate. □

For codes over Galois rings the situation is more complicated in the sense that some of the minimal supports characterize groups of users that can recover only a part of the secret. More specifically, let C be a "linear" code over $\mathrm{GR}(q^m, p^m)$ as discussed in Section III, and suppose we construct a linear secret-sharing scheme as above using the parity-check matrix of C to generate distribution rules. Suppose again that the first coordinate corresponds to the secret. Minimal authorized coalition in this case can reconstruct either a part of the secret or the secret in full, depending on the type of the corresponding minimal support. More precisely, the following is true.

Theorem 5.2 [2], [3]: Let $\widetilde{I} = \{1\} \cup I \subset [n]$ be a minimal support of type t in C such that there is a codeword $\boldsymbol{c} \in C$ with $\mathsf{supp}(\boldsymbol{c}) = \widetilde{I}$ and $c_1 \in N_t$. Then the users in I, taking their shares of information together, can reconstruct exactly $m - t$ q-ary symbols of the secret.

For instance, if C is a $\mathrm{ZRM}(1, 3)$ first-order Reed–Muller code, then $\mathcal{M}_0$ consists of a single set $\widetilde{I} = [n]$ and $\mathcal{M}_1$ is formed by 14 sets of size 8 (see Example 4). A half of them contain coordinate 1; therefore, there are seven groups of users that can reconstruct one of the two bits of the secret.

Note that since the binary image of the $\mathrm{ZRM}(1, v)$ code is $\boldsymbol{Z}_2$-linear, this scheme can be realized by two linear schemes over $\boldsymbol{Z}_2$, one corresponding to the $[8, 1, 8]$ repetition code and the other to

the $[8, 4, 4]$ binary RM code. In both schemes, the number of bits in the secret (one) equals the number of bits in the information share of each participant. Such schemes are called *ideal*. One of the reviewers suggested that any scheme over $\boldsymbol{Z}_4$ can be realized by two ideal (not necessarily linear) binary schemes, one responsible for sharing the first (say, less significant) bit of the secret and the other one the second bit. We conclude by showing that this is not true.

The counterexample is furnished by the Nordstrom–Robinson code C of length 8 over $\boldsymbol{Z}_4$ [11]. Suppose its first coordinate corresponds to the secret. Puncturing C in this coordinate, we get a cyclic code of length 7, whose type 0 supports are given by the vectors $1223233, 1013102, 1100123, 1033320$ and their cyclic shifts. Minimal supports of type 0 are defined by the last three vectors. Thus minimal coalitions authorized to recover both bits of the secret correspond to supports of vectors $1013102, 1100123, 1033320$ and those of their cyclic shifts that have 1 or 3 on the first coordinate. We shall show that this access structure cannot be realized by a binary ideal scheme. It is known [5], [19] that every binary ideal scheme is either linear or affine, i.e., corresponds to a binary linear code or to a binary affine code (a binary code is affine if the sum of any three code vectors is a code vector).

Suppose that the minimal coalitions in this scheme correspond to minimal vectors (with a 1 in the first coordinate) of some binary linear or affine code, say A. In either case, the sum of any three code vectors should be again a code vector. On the other hand, it is immediate to observe that there are three vectors in A that sum up to a vector of weight 3. Since the size of all minimal authorized coalitions in the original system is 4, this proves that code A does not realize our access structure.

We leave as an open problem to prove that every scheme corresponding to a $\boldsymbol{Z}_4$-linear code whose binary image is not $\boldsymbol{Z}_2$-linear cannot be represented by two binary ideal schemes.

APPENDIX

Steiner's Original Problem [20]. Given two numbers k and v, $k \leq v$, construct a pair $(X, \mathcal{B})$, where X is a finite set and $\mathcal{B}$ a collection of its subsets, which satisfies the following conditions:

i) $|X| = v$;
ii) $\mathcal{B} = \bigcup_{n=3}^{k} \mathcal{B}(n)$ and $|B_i| = n$ for every $B_i \in \mathcal{B}(n)$;
iii) every pair $(x, y) \subset X$ is contained in exactly one block of $\mathcal{B}(3)$;
iv) every i-subset of X, $3 \leq i \leq k-1$, which does not contain a block of $\bigcup_{j=3}^{i} \mathcal{B}(j)$, is contained in exactly one block of $\mathcal{B}(i+1)$; no block of $\mathcal{B}(i+1)$ contains as subsets blocks of $\bigcup_{j=3}^{i} \mathcal{B}(j)$.

ACKNOWLEDGMENT

The short and nice geometric proof of Theorem 2.9 that now replaces our original (much longer) one with coordinate approach was suggested by Juriaan Simonis.

REFERENCES

[1] E. Agrell, "Voronoi regions for binary linear block codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 310–316, 1996.
[2] A. Ashikhmin and A. Barg, "Minimal vectors in linear codes and sharing of secrets," Universität Bielefeld, SFB 343 Diskrete Strukturen in der Mathematik, preprint 94-113, 1994, available online at www.mathematik.uni-bielefeld.de/sfb343/preprints.
[3] A. Ashikhmin, A. Barg, G. Cohen, and L. Huguet, "Variations on minimal codewords in linear codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-11)* (Lecture Notes in Computer Science, vol. 948), G. Cohen, M. Giusti, and T. Mora, Eds. Berlin: Springer-Verlag, 1995, pp. 96–105.
[4] A. Barg, "Complexity issues in coding theory," in *Handbook of Coding Theory*, V. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, to be published.
[5] A. Beimel and B. Chor, "Universally ideal secret-sharing schemes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 786–794, 1994.
[6] G. R. Blakley and G. A. Kabatianskii, "Linear algebra approach to secret sharing schemes," in *Error Control, Cryptology, and Speech Compression, Selected Papers from Int. Workshop Information Protection* (Lecture Notes in Computer Science, vol. 829). Berlin, Germany: Springer-Verlag, 1994, pp. 33–40.
[7] Y. Borissov and N. Manev, "On the minimal words of the primitive BCH codes," in *Proc. Int. Workshop Algebraic and Combinatorial Coding Theory (ACCT-5)* (Sozopol, Bulgaria, June 1996), pp. 59–65.
[8] G. D. Cohen and A. Lempel, "Linear intersecting codes," *Discr. Math.*, vol. 56, pp. 35–43, 1984.
[9] S. Dodunekov and I. Landgev, "On near-MDS codes," *J. Geom.*, vol. 54, no. 1–2, pp. 30–43, 1995.
[10] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.
[11] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The $\boldsymbol{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 301–319, Mar. 1994.
[12] H. Hanani, "On the original Steiner systems," *Discr. Math.*, vol. 51, pp. 309–310, 1984.
[13] G. H. Hardy and E. M. Wright, *Introduction to the Theory of Numbers.* Oxford, U.K.: Oxford Univ. Press, 1960.
[14] T.-Y. Hwang, "Decoding linear block codes for minimizing word error rate," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 733–737, Nov. 1979.
[15] L. Levitin and C. R. P. Hartmann, "A new approach to the general minimum distance decoding problem: The zero-neighbors algorithm," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 378–384, May 1985.
[16] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam, The Netherlands: North-Holland, 1977.
[17] J. Massey, "Minimal codewords and secret sharing," in *Proc. 6th Joint Swedish–Russian Workshop on Information Theory* (Mölle, Sweden, 1993), pp. 246–249.
[18] A. A. Nechaev, "The Kerdock code in a cyclic form," *Diskr. Mat.*, vol. 1, no. 4, pp. 123–139, 1989. English translation in *Discr. Math. Appl.*, vol. 1, pp. 365–384, 1991.
[19] J. Simonis and A. Ashikhmin, "Almost affine codes," *Des., Codes Cryptogr.*, vol. 14, pp. 179–197, 1998.
[20] J. Steiner, "Combinatorische Aufgabe," *J. Reine Angew. Math.*, vol. 45, pp. 181–182, 1853.
[21] D. R. Stinson, "An explication of secret sharing schemes," *Des., Codes Cryptogr.*, vol. 2, no. 4, pp. 357–390, 1992.