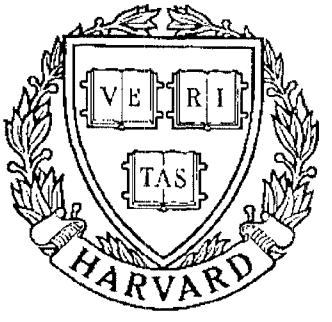
TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

A Generalized Gilbert-Varshamov Bound Derived via Analysis of a Code-Search Algorithm

by J. Gu and T. Fuja

TR 92-87

A Generalized Gilbert-Varshamov Bound Derived via Analysis of a Code-Search Algorithm

Jian Gu and Tom Fuja*
Department of Electrical Engineering
Systems Research Center
University of Maryland
College Park, MD 20742

Abstract

This correspondence derives a generalization of the Gilbert-Varshamov bound that is applicable to block codes whose codewords must be drawn from irregular sets; the bound improves by a factor of four a similar result recently published by Kolesnik and Krachkovsky. It is derived by analyzing a code search algorithm we refer to as the "Altruistic Algorithm". This algorithm iteratively deletes potential codewords so that at each iteration the "worst" candidate is removed; the bound is derived by demonstrating that, as the algorithm proceeds, the average volume of a sphere of a given radius approaches the maximum such volume and so a bound previously expressed in terms of the maximum volume can in fact be expressed in terms of the average volume. Examples of applications where the bound is relevant include error-correcting (d, k)-constrained codes and binary codes for code division multiple access.

Submitted January 1992 to *IEEE Transactions on Information Theory.*
Revised August 1992.

* Supported in part by National Science Foundation grant NCR-8957623; also by the NSF Engineering Research Centers Program, CDR-8803012.

Index Terms: Coding bounds, search algorithms, irregular spaces, runlength constraints, code division multiple access.

Captions

Table 1: Parameters relevant to bounding the cardinality of a single-error correcting block code satisfying a (1,3) runlength constraint.

Figure 1: Lower bounds on the achievable rates of single-error correcting block codes satisfying a (1,3) runlength constraint.

1. Introduction

Let $\mathcal{X}$ be a finite set and let $\rho : \mathcal{X} \times \mathcal{X} \rightarrow \{0, 1, \ldots\}$ be an integer-valued metric defined on $\mathcal{X}$. Suppose our goal is to construct a subset of $\mathcal{X}$ – call the subset $\mathcal{C}$ – such that $\rho(c_1, c_2) \geq d$ for all $c_1, c_2 \in \mathcal{C}$.

This very general description covers many different types of block coding schemes. For instance:

- $\mathcal{X} = \{0,1\}^n$ and ρ is the Hamming metric. To construct $\mathcal{C}$ is thus to construct a binary error correcting code with minimum distance d.
- $\mathcal{X} = \{\mathbf{x} : \mathbf{x} \in \{0,1\}^n, \mathrm{wt}(\mathbf{x}) = w\}$ and ρ is again the Hamming metric. This describes the construction of a constant weight error correcting code.
- Let $\mathcal{X}$ denote a collection of binary n-tuples with the property that any concatenation of elements of $\mathcal{X}$ satisfies a runlength constraint – i.e., has the property that between every pair of 1's there is at least some minimum number of 0's and at most some maximum number of 0's. Then letting ρ denote the Hamming metric we can construct a code with both runlength *and* error correcting properties.
- Let $\mathcal{X}$ denote the set of weight-w binary n-tuples with cyclic auto-correlation at most λ – i.e., $\mathcal{X} = \{\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}] : wt(\mathbf{x}) = w, \sum_{i=0}^{n-1} x_i x_{i\oplus j} \leq \lambda \text{ for } j = 1, 2, \ldots, n-1\}$. (Here, "$\oplus$" denotes modulo-$n$ addition.) Furthermore, let ρ be defined by $\rho(\mathbf{x}, \mathbf{y}) = \min\{d_H(\mathbf{x}, D^i\mathbf{y}) : i = 0, 1, \ldots, n-1\}$, where d_H denotes the Hamming metric and $D^i\mathbf{y}$ is the binary n-tuple obtained by performing i right-cyclic shifts on the binary n-tuple $\mathbf{y}$. Then this procedure describes the construction of an $(n, w, \lambda, \lceil d/2 \rceil - w)$ optical orthogonal code for multiple access, as defined by Chung, Salehi, and Wei [2].

Given such a set $\mathcal{X}$ from which we can draw codewords and such a metric ρ, an issue of obvious interest is the size of the largest code that can be thus constructed with a specified "minimum distance" – i.e., what is

$$A(d) \stackrel{\Delta}{=} \max\{|\mathcal{C}| : \mathcal{C} \subseteq \mathcal{X}, \rho(c_1, c_2) \geq d \text{ for all } c_1, c_2 \in \mathcal{C}, c_1 \neq c_2\}.$$

Notation: Let $x \in \mathcal{X}$ and define $V_r(x)$ to be the volume of a sphere in $\mathcal{X}$ centered on x – i.e.,

$$V_r(x) = |\{y \in \mathcal{X} : \rho(x,y) \leq r\}|.$$

If $V_r(x)$ is constant over all $x \in \mathcal{X}$, then there exist several well-known techniques to bound $A(d)$; the best known lower bound is the Gilbert-Varshamov bound, which states that $A(d) \geq |\mathcal{X}|/V_{d-1}$, where V_{d-1} is the constant volume of a sphere of radius $d-1$ centered at an element of $\mathcal{X}$. However, if $V_r(x)$ is *not* constant, then the Gilbert-Varshamov bound is not applicable. We call the space $\mathcal{X}$ *irregular* with respect to ρ if $V_r(x)$ varies with x. Of the four examples of interest listed above, the last two represent irregular spaces; clearly, the analysis of codes drawn from irregular spaces is of practical interest.

This correspondence presents a new lower bound on $A(d)$ that is applicable to codes drawn from irregular sets. This is accomplished by introducing and analyzing a code search algorithm that recursively deletes potential codewords from a pool of candidates.

2. Background

The simplest bound on $A(d)$ is the straight-forward generalization of the Gilbert-Varshamov bound in which the constant volume of a sphere of radius $d-1$ is replaced by the *maximum* volume. This result is presented in Theorem 1.

Theorem 1: Let $A(d)$ be defined as above. Then

$$A(d) \geq \frac{|\mathcal{X}|}{V_{d-1,\max}}, \tag{2}$$

where

$$V_{d-1,\max} \stackrel{\Delta}{=} \max\{V_{d-1}(x) : x \in \mathcal{X}\}.$$

The proof of Theorem 1, like the proof of the Gilbert-Varshamov bound, is based on the greedy algorithm. To construct a code using the greedy algorithm, one simply selects codewords iteratively from a pool of potential codewords; the pool initially contains every element of $\mathcal{X}$, but every time a codeword is selected every element lying within a distance $d-1$ of the newly selected codeword is removed from the pool. Thus at every step of

the algorithm each potential codeword in the pool lies at a distance at least d from all previously chosen codewords. Theorem 1 is derived by noting that, at each step of the algorithm, at most $V_{d-1,\max}$ elements are deleted from the pool, and so the process can be repeated at least $|\mathcal{X}|/V_{d-1,\max}$ times.

A recent paper by Kolesnik and Krachkovsky [1] derived another bound on $A(d)$, given in Theorem 2 below.

Theorem 2: The following inequality holds.

$$A(d) \geq \frac{|\mathcal{X}|}{4 \cdot V_{d-1,\text{avg}}}, \tag{1}$$

where

$$V_{d-1,\text{avg}} \triangleq \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} V_{d-1}(x).$$

The proof of Theorem 2 in [1] is based on the observation that no more than half of the spheres of radius $d-1$ in $\mathcal{X}$ can have a volume that exceeds twice the average value; thus there exists a set of at least $|\mathcal{X}|/2$ elements of $\mathcal{X}$ with maximum volume no more than $2V_{d-1,\text{avg}}$, and so by applying Theorem 1 we get the desired result. Kolesnik and Krachkovsky use this result along with the generating function of runlength-constrained sequences to arrive at an asymptotically achievable rate for error-correcting runlength-constrained block codes.

It's worth noting that neither of the two bounds given above is strictly better than the other – i.e., whether $V_{d-1,\max}$ is larger or smaller than $4 \cdot V_{d-1,\text{avg}}$ is determined by the choice of $\mathcal{X}$ and ρ.

3. New Results

Our main result is that the "4" in the denominator of (1) can be removed – i.e., we have shown that that $A(d) \geq |\mathcal{X}|/V_{d-1,\text{avg}}$. This bound is strictly better than (1) and it is always at least as good as the bound in (2). To get to the desired result will require the analysis of a code-searching algorithm – much like the bound in Theorem 1 required a (pessimistic) analysis of the greedy algorithm.

Consider the following approach to searching for a code from among the elements of $\mathcal{X}$.

The Altruistic Algorithm:

(1) Set $i = 0$ and $\mathcal{X}^{(0)} = \mathcal{X}$.

(2) Compute $V_{d-1,\max}^{(i)}$, defined as the maximum volume of a sphere of radius $d-1$ in $\mathcal{X}^{(i)}$ – i.e.,

$$V_{d-1,\max}^{(i)} \triangleq \max\{V_{d-1}^{(i)}(x) : x \in \mathcal{X}^{(i)}\},$$

where

$$V_r^{(i)}(x) \triangleq |\{y \in \mathcal{X}^{(i)} : \rho(x,y) \leq r\}|.$$

(3) If $V_{d-1,\max}^{(i)} = 1$ then stop. If $V_{d-1,\max}^{(i)} > 1$ then create $\mathcal{X}^{(i+1)}$ by deleting one element from $\mathcal{X}^{(i)}$; choose the element that is deleted by picking one lying at the center of a sphere of maximum volume in $\mathcal{X}^{(i)}$ – i.e., delete x^*, where $|\{y \in \mathcal{X}^{(i)} : \rho(x^*,y) \leq d-1\}| = V_{d-1,\max}^{(i)}$.

(4) Set $i \leftarrow i+1$ and go to (2).

This algorithm is "smarter" than the greedy algorithm because it makes a more logical choice at every iteration; rather than selecting an arbitrary element to be a codeword, it deletes a "bad" element from among the potential codewords – bad in the sense that it lies within distance $d-1$ of a maximal number of other potential codewords. When the algorithm is finished – when every potential codeword is alone within a sphere of radius $d-1$ – the resulting set is a code with minimum distance d.

(We refer to this algorithm as "altruistic" because at every iteration the least promising potential codeword selflessly removes itself from further consideration. After submission of this correspondence the authors were made aware that this algorithm was developed independently in a more general context by Ytrehus [4-5].)

3.1. An Easy Bound

Before proceeding with the proof of our ultimate result, we show how an analysis of this algorithm quickly leads to a bound that is strictly better than the one in Theorem 2.

Theorem 3: Let $\mathcal{X}$ be a set with integer-valued metric ρ, and let $A(d) = \max\{|\mathcal{C}| : \mathcal{C} \subseteq \mathcal{X}, \rho(c_1, c_2) \geq d \text{ for all } c_1, c_2 \in \mathcal{C}\}$. Then the following inequality holds:

$$A(d) \geq \frac{|\mathcal{X}|}{1.5 \cdot V_{d-1,\text{avg}}},$$

where

$$V_{d-1,\text{avg}} \triangleq \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} V_{d-1}(x).$$

Proof of Theorem 3: Consider the following "abbreviated" form of the Altruistic Algorithm. It is abbreviated in that we may stop deleting elements *before* a code is obtained; we stop when the maximum volume of a sphere of radius $d-1$ exceeds the average volume of all such spheres by no more than 0.5.

The Abbreviated Altruistic Algorithm:

(1) Set $i = 0$ and $\mathcal{X}^{(0)} = \mathcal{X}$.

(2) Compute $V^{(i)}_{d-1,\text{max}}$ and $V^{(i)}_{d-1,\text{avg}}$, defined as the maximum and average volumes of a sphere of radius $d-1$ in $\mathcal{X}^{(i)}$ – i.e.,

$$V^{(i)}_{d-1,\text{avg}} \triangleq \frac{1}{|\mathcal{X}^{(i)}|} \sum_{x \in \mathcal{X}^{(i)}} V^{(i)}_{d-1}(x)$$

and

$$V^{(i)}_{d-1,\text{max}} \triangleq \max\{V^{(i)}_{d-1}(x) : x \in \mathcal{X}^{(i)}\},$$

where

$$V^{(i)}_{r}(x) \triangleq |\{y \in \mathcal{X}^{(i)} : \rho(x, y) \leq r\}|.$$

(3) If $V^{(i)}_{d-1,\text{max}} \leq V^{(i)}_{d-1,\text{avg}} + 0.5$, then stop. If $V^{(i)}_{d-1,\text{max}} > V^{(i)}_{d-1,\text{avg}} + 0.5$, then create $\mathcal{X}^{(i+1)}$ by deleting one element from $\mathcal{X}^{(i)}$; choose the element that is deleted by picking one lying at the center of a sphere of maximum volume in $\mathcal{X}^{(i)}$ – i.e., delete x^*, where $|\{y \in \mathcal{X}^{(i)} : \rho(x^*, y) \leq d-1\}| = V^{(i)}_{d-1,\text{max}}$.

(4) Set $i \leftarrow i + 1$ and go to (2).

Let i^* denote the value of i that stops the algorithm. Note that $0 \leq i^* \leq |\mathcal{X}| - 2$ because when there are only two elements in a set they are either within distance $d-1$

of each other (in which case $V_{d-1,\text{avg}} = V_{d-1,\text{max}} = 2$) or they are not (in which case $V_{d-1,\text{avg}} = V_{d-1,\text{max}} = 1$).

Obviously, any lower bound on the number of codewords that can be drawn from $\mathcal{X}^{(i^*)}$ is a lower bound on the number of codewords that can be drawn from $\mathcal{X}$. Thus by applying the bound from Theorem 1 to the space $\mathcal{X}^{(i^*)}$, we obtain

$$\begin{aligned} A(d) &\geq \frac{|\mathcal{X}^{(i^*)}|}{V_{d-1,\text{max}}^{(i^*)}} \\ &= \frac{|\mathcal{X}| - i^*}{V_{d-1,\text{max}}^{(i^*)}} \\ &\geq \frac{|\mathcal{X}| - i^*}{V_{d-1,\text{avg}}^{(i^*)} + 0.5} \\ &\geq \frac{|\mathcal{X}| - i^*}{1.5 \cdot V_{d-1,\text{avg}}^{(i^*)}}, \end{aligned} \tag{3}$$

where the last inequality follows because $V_{d-1,\text{avg}}^{(i^*)} \geq 1$.

The problem now is to bound $V_{d-1,\text{avg}}^{(i^*)}$ in terms of $V_{d-1,\text{avg}}$. We begin by making two observations:

$$V_{d-1,\text{max}}^{(j)} \geq V_{d-1,\text{avg}}^{(j)} + 0.5 \quad \text{for } 0 \leq j \leq i^* - 1, \tag{4}$$

and

$$V_{d-1,\text{avg}}^{(k)} = \frac{(|\mathcal{X}| - k + 1)V_{d-1,\text{avg}}^{(k-1)} - 2V_{d-1,\text{max}}^{(k-1)} + 1}{|\mathcal{X}| - k} \quad \text{for } 1 \leq k \leq i^*. \tag{5}$$

Inequality (4) follows from the definition of i^*, and equation (5) follows from the observation that, in deleting an element lying at a center of a sphere of volume $V_{d-1,\text{max}}^{(k-1)}$ we are reducing the sum of all such volumes by $2V_{d-1,\text{max}}^{(k-1)} - 1$. Using inequality (4) in (5), we obtain

$$V_{d-1,\text{avg}}^{(k)} \leq \frac{|\mathcal{X}| - k - 1}{|\mathcal{X}| - k} V_{d-1,\text{avg}}^{(k-1)}. \tag{6}$$

Repeated applications of (6) yield the identity

$$V_{d-1,\text{avg}}^{(i^*)} \leq \frac{|\mathcal{X}| - i^* - 1}{|\mathcal{X}| - 1} V_{d-1,\text{avg}}. \tag{7}$$

If we now substitute (7) into (3) we obtain

$$A(d) \geq \frac{(|\mathcal{X}| - i^*)(|\mathcal{X}| - 1)}{1.5 \cdot (|\mathcal{X}| - i^* - 1)V_{d-1,\text{avg}}}. \tag{8}$$

But equation (8) is minimized over all $i^* = 0, 1, \ldots, |\mathcal{X}| - 2$ by $i^* = 0$; thus $A(d) \geq |\mathcal{X}|/1.5 \cdot V_{d-1,\text{avg}}$. QED

3.2. A Better (But More Difficult) Bound

We now prove the promised result.

Theorem 4: Let $\mathcal{X}$ be a finite set with integer-valued metric ρ, and let $A(d) = \max\{|\mathcal{C}| : \mathcal{C} \subseteq \mathcal{X}, \rho(c_1, c_2) \geq d \text{ for all } c_1, c_2 \in \mathcal{C}\}$. Then the following inequality holds:

$$A(d) \geq \frac{|\mathcal{X}|}{V_{d-1,\text{avg}}},$$

where

$$V_{d-1,\text{avg}} \triangleq \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} V_{d-1}(x).$$

The remainder of this section is a proof of Theorem 4. We shall prove Theorem 4 by induction on $\lceil V_{d-1,\text{avg}} \rceil$. (Recall that $\lceil x \rceil$ is the smallest integer no smaller than x.) It is obviously true for $\lceil V_{d-1,\text{avg}} \rceil = 1$; if $\lceil V_{d-1,\text{avg}} \rceil = 1$ then $V_{d-1,\text{avg}} = 1$ and so every pair of elements in $\mathcal{X}$ lie at least a distance d apart – meaning $\mathcal{X}$ itself is the optimal code. In this case $A(d) = |\mathcal{X}|$ and so Theorem 4 holds.

Now assume that Theorem 4 holds for $\lceil V_{d-1,\text{avg}} \rceil \leq M - 1$; we wish to show that it holds for $\lceil V_{d-1,\text{avg}} \rceil = M$.

Let $\mathcal{X}$ be a set with $\lceil V_{d-1,\text{avg}} \rceil = M$. Apply to this set the abbreviated Altruistic Algorithm from Section 3.1, and let i^* denote the value of i that stops the algorithm. (Throughout this section we will use the same notation as in Section 3.1.) When the algorithm stops we will have a set $\mathcal{X}^{(i^*)}$ such that $|\mathcal{X}^{(i^*)}| = |\mathcal{X}| - i^* \geq 2$ and the following relation holds:

$$V_{d-1,\max}^{(i^*)} - 0.5 \leq V_{d-1,\text{avg}}^{(i^*)} \leq V_{d-1,\max}^{(i^*)}. \tag{9}$$

Suppose $V_{d-1,\text{avg}}^{(i^*)} = V_{d-1,\max}^{(i^*)}$ – i.e., after the algorithm stops every sphere of radius

$d-1$ contains the same volume. Then from Theorem 1 we have

$$\begin{aligned} A(d) &\geq \frac{|\mathcal{X}^{(i^*)}|}{V_{d-1,\max}^{(i^*)}} \\ &= \frac{|\mathcal{X}| - i^*}{V_{d-1,\text{avg}}^{(i^*)}} \\ &\geq \frac{(|\mathcal{X}| - i^*)(|\mathcal{X}| - 1)}{(|\mathcal{X}| - i^* - 1)V_{d-1,\text{avg}}} \\ &\geq \frac{|\mathcal{X}|}{V_{d-1,\text{avg}}}. \end{aligned}$$

The second inequality follows from identity (7), and the last inequality follows from the fact that $(|\mathcal{X}| - i^*)(|\mathcal{X}| - 1)/(\mathcal{X} - i^* - 1)V_{d-1,\text{avg}}$ is minimized over all $i^* = 0, 1, \ldots, |\mathcal{X}| - 2$ by $i^* = 0$. Thus we have proven Theorem 4 for the case $V_{d-1,\max}^{(i^*)} = V_{d-1,\text{avg}}^{(i^*)}$.

It remains to prove Theorem 4 for the case when the abbreviated Altruistic Algorithm stops with a set containing spheres of differing volume – i.e., for the case $V_{d-1,\max}^{(i^*)} > V_{d-1,\text{avg}}^{(i^*)}$. To proceed further, we require a fairly technical lemma.

Lemma 1: Let $\mathcal{Y}$ be a finite set with integer-valued metric ρ; assume $|\mathcal{Y}| \geq 2$ and define the average and maximum volumes of a sphere of radius $d-1$ in $\mathcal{Y}$ in the "usual" way – i.e.,

$$V_{d-1,\text{avg}} \stackrel{\Delta}{=} \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} V_{d-1}(y)$$

and

$$V_{d-1,\max} \stackrel{\Delta}{=} \max\{V_{d-1}(y) : y \in \mathcal{Y}\},$$

where $V_r(y) = |\{x \in \mathcal{Y} : \rho(x, y) \leq r\}|$. Define $\delta \stackrel{\Delta}{=} V_{d-1,\max} - V_{d-1,\text{avg}}$ and assume $0 < \delta < 1$. (Note that this means that $\lceil V_{d-1,\text{avg}} \rceil = V_{d-1,\max}$.) Then there exists a subset of $\mathcal{Y}$ - call this subset $\mathcal{Y}^{(k^*)}$ and let $V_{d-1,\text{avg}}^{(k^*)}$ denote the average volume of a sphere of radius $d-1$ in $\mathcal{Y}^{(k^*)}$ – such that both of the following two properties hold:

(1) $|\mathcal{Y}^{(k^*)}| = |\mathcal{Y}| - k^*$, where $k^* = \lceil (1-\delta) \cdot |\mathcal{Y}|/V_{d-1,\max} \rceil$;

(2) $V_{d-1,\text{avg}}^{(k^*)} = [|\mathcal{Y}| \cdot V_{d-1,\text{avg}} - k^*(2V_{d-1,\max} - 1)]/(|\mathcal{Y}| - k^*) \leq V_{d-1,\max} - 1$;

Proof of Lemma 1: See Appendix.

We now proceed by applying Lemma 1 to the set $\mathcal{X}^{(i^*)}$. Let $\delta = V_{d-1,\max}^{(i^*)} - V_{d-1,\text{avg}}^{(i^*)}$; then from (9) we know that $0 < \delta \leq 0.5$, and Lemma 1 states that we can obtain a new subset of $\mathcal{X}^{(i^*)}$ – call the subset $\mathcal{X}^{(i^*+k^*)}$ – such that

$$\begin{aligned} V_{d-1,\text{avg}}^{(i^*+k^*)} &= \frac{|\mathcal{X}^{(i^*)}| \cdot V_{d-1,\text{avg}}^{(i^*)} - k^*(2V_{d-1,\max}^{(i^*)} - 1)}{|\mathcal{X}^{(i^*)}| - k^*} \\ &= \frac{(|\mathcal{X}| - i^*) \cdot (V_{d-1,\max}^{(i^*)} - \delta) - k^*(2V_{d-1,\max}^{(i^*)} - 1)}{|\mathcal{X}| - i^* - k^*} \\ &\leq V_{d-1,\max}^{(i^*)} - 1. \\ &= \lceil V_{d-1,\text{avg}}^{(i^*)} \rceil - 1 \\ &\leq M - 1. \end{aligned} \tag{10}$$

So we can apply the induction hypothesis to the new set $\mathcal{X}^{(i^*+k^*)}$ – i.e.,

$$A(d) \geq \frac{|\mathcal{X}^{(i^*+k^*)}|}{V_{d-1,\text{avg}}^{(i^*+k^*)}} = \frac{|\mathcal{X}| - i^* - k^*}{V_{d-1,\text{avg}}^{(i^*+k^*)}}. \tag{11}$$

Furthermore, using equation (10) to substitute for $V_{d-1,\text{avg}}^{(i^*+k^*)}$ in (11), we obtain

$$A(d) \geq \frac{(|\mathcal{X}| - i^* - k^*)^2}{(|\mathcal{X}| - i^*) \cdot (V_{d-1,\max}^{(i^*)} - \delta) - k^*(2V_{d-1,\max}^{(i^*)} - 1)}.$$

Lemma 2: Let L, N, and δ be constants; L and N are positive integers and $0 < \delta < 1$. Define the function $g : \mathcal{R} \to \mathcal{R}$ by

$$g(x) \triangleq \frac{(L-x)^2}{L(N-\delta) - x(2N-1)}.$$

Then $g(x)$ is a monotone non-decreasing function of x for $(1-2\delta)L/(2N-1) \leq x \leq L$.

Proof of Lemma 2: Simple calculus.

Using the notation of Lemma 2, if we let $L = |\mathcal{X}^{(i^*)}| = |\mathcal{X}| - i^*$ and $N = V_{d-1,\max}^{(i^*)}$, we have shown that $A(d) \geq g(k^*)$, where $k^* = \lceil (1-\delta)L/N \rceil$; furthermore, since

$$L \geq \lceil \frac{(1-\delta)L}{N} \rceil \geq \frac{(1-\delta)L}{N} \geq \frac{(1-2\delta)L}{2N-1},$$

Lemma 2 indicates that

$$A(d) \geq g(k^*) = g(\lceil (1-\delta)L/N \rceil) \geq g((1-\delta)L/N).$$

Furthermore,

$$\begin{aligned} g((1-\delta)L/N) &= \frac{(L-(1-\delta)L/N)^2}{L(N-\delta)-(1-\delta)L(2N-1)/N} \\ &= \frac{L(N-1+\delta)}{N(N-1)} \\ &= \frac{(N-1+\delta)(N-\delta)}{N(N-1)} \cdot \frac{L}{N-\delta} \\ &\geq \frac{L}{N-\delta}. \end{aligned}$$

The last inequality follows from the fact that $h(x) \triangleq (N-1+x)(N-x)$ is minimized over $0 \leq x \leq 0.5$ by $x = 0$. Thus we've shown that $A(d) \geq L/(N-\delta)$. Recalling that $L = |\mathcal{X}^{(i^*)}| = |\mathcal{X}| - i^*$ and $N - \delta = V_{d-1,\max}^{(i^*)} - \delta = V_{d-1,\text{avg}}^{(i^*)}$, we obtain $A(d) \geq |\mathcal{X}^{(i^*)}|/V_{d-1,\text{avg}}^{(i^*)}$. If we (once again) apply equation (7) we obtain

$$\begin{aligned} A(d) &\geq \frac{|\mathcal{X}| - i^*}{V_{d-1,\text{avg}}^{(i^*)}} \\ &\geq \frac{(|\mathcal{X}| - i^*)(|\mathcal{X}| - 1)}{(|\mathcal{X}| - i^* - 1)V_{d-1,\text{avg}}} \\ &\geq \frac{|\mathcal{X}|}{V_{d-1,\text{avg}}}. \end{aligned}$$

Here the final inequality is due to the fact that $(|\mathcal{X}| - i^*)(|\mathcal{X}| - 1)/(\mathcal{X} - i^* - 1)V_{d-1,\text{avg}}$ is minimized over all $i^* = 0, 1, \ldots, |\mathcal{X}| - 2$ by $i^* = 0$. This proves Theorem 4.

4. An Example

Suppose we wish to bound the rate that can be attained by a single-error correcting block code satisfying a $(d = 1, k = 3)$ runlength constraint. (Recall that a (d, k) constraint means that between every pair of binary ones there must be one, two, or three binary zeroes.) In [3] Beenker and Schouhamer Immink described a technique ("Construction 2") for constructing (d, k)-constrained block codes without minimum distance requirements. We

begin our example by applying their technique to the construction of a $(1,3)$-constrained code.

Let $C^{n-1}(1,3)$ consist of all binary $(n-1)$-tuples that satisfy the $(1,3)$ constraint in isolation – i.e., without worrying about concatenation. Now let $C^{n-1}(1,3;2,2)$ consist of all the elements of $C^{n-1}(1,3)$ with at most two leading zeroes and at most two trailing zeroes. Then every element of $C^{n-1}(1,3;2,2)$ can be used as a codeword with a "merging rule"; an extra bit is inserted at the beginning of each codeword so as to satisfy the $(1,3)$ constraint. This is accomplished as follows:

- If the element of $C^{n-1}(1,3;2,2)$ on which the current codeword is based begins with a "1", append a "0" to the beginning.
- If the element of $C^{n-1}(1,3;2,2)$ on which the current codeword is based begins with a "0", then look at the codeword that *precedes* the current codeword.
  - If it ends with a "0", append a "1" to the beginning of the current codeword.
  - If it ends with a "1", append a "0" to the beginning of the current codeword.

Using this technique we can construct a set of $|C^{n-1}(1,3;2,2)|$ potential codewords – i.e., $\mathcal{X} = C^{n-1}(1,3;2,2)$, with the understanding that in actually implementing this code an extra bit will be appended to the beginning of each codeword. This extra bit is used only to satisfy the (d,k) constraint at the codeword boundaries; it will be used in determining the code rate but will not be used in determining distances between codewords.

Given such a set $\mathcal{X}$ one can compute numerically the quantities $|\mathcal{X}|$, $V_{2,\text{avg}}$, and $V_{2,\max}$ and so arrive at lower bounds on the cardinality of an optimal (1,3)-constrained code with minimum distance three; some of these figures are given in Table 1. In addition we've listed the number of codewords we found when the Altruistic Algorithm was permitted to continue its search to completion; just as the Gilbert-Varshamov bound tends to underestimate the performance of the greedy algorithm, we see that Theorem 4 tends to underestimate the performance of the Altruistic Algorithm.

The same information is presented in terms of code *rate* in Figure 1 – i.e.,

- $R_1 = \log_2(|\mathcal{C}|)/n$, where $\mathcal{C}$ is a code found by the Altruistic Algorithm.
- $R_2 = \log_2(\lceil |\mathcal{X}|/V_{2,\text{avg}} \rceil)/n$.

- $R_3 = \log_2(\lceil |\mathcal{X}|/V_{2,\max} \rceil)/n$.

If we compare the search results – i.e., the values of R_1 – with those in [4] we find that rates in Figure 1 exceed the rates found in [4]. However, this comparison is not fair; the codes found in [4] were "codeword concatenatable" – i.e., they consisted of collections of binary n-tuples that could be freely concatenated without violating the runlength constraint; by comparison, the values of R_1 in Figure 1 are derived from "Construction 2", which uses a state-dependent encoder and is inherently more efficient. To obtain a fair comparison one would have to apply the Altruistic Algorithm to the set $\mathcal{X}$ where $\mathcal{X}$ is a maximal collection of binary n-tuples that can be freely concatenated. In all instances where we have done this the codes found by the Altruistic Algorithm have been as good as or slightly inferior to the codes in [4]. This is not surprising, since as mentioned in Section 3 the Altruistic Algorithm is a simplified version of the algorithm used by Ytrehus to generate the results in [4].

5. Conclusion

In this paper we derived a very general result that lower bounds the efficiency that can be attained with a variety of block coding schemes. The resulting lower bound is a true generalization of the Gilbert-Varshamov bound that closes the "two-bit gap" between what the GV bound promises for regular codes and what the Kolesnick/Krachkovsky result promises for irregular codes. Furthermore, we have shown how the simplicity of the code search technique we call the Altruistic Algorithm lends itself to surprisingly detailed analysis.

Appendix

Proof of Lemma 1:

We're given $\mathcal{Y}$ such that $V_{d-1,\max} - 1 < V_{d-1,\text{avg}} < V_{d-1,\max}$. Suppose we apply the Altruistic Algorithm to $\mathcal{Y}$ until the average volume drops at least as low as $V_{d-1,\max} - 1$; that is, we iteratively delete elements of $\mathcal{Y}$ until we obtain a subset with an average volume

that is at most $V_{d-1,\max} - 1$. Let k^* denote the number of elements deleted at this point, and let $\mathcal{Y}^{(k^*)}$ denote the resulting subset.

As long as the average volume of a sphere is strictly greater than $V_{d-1,\max} - 1$ there must be *at least* one sphere with volume $V_{d-1,\max}$; this means that in deleting k^* elements of $\mathcal{Y}$ to arrive at $\mathcal{Y}^{(k^*)}$ we have each time deleted an element lying at the center of a sphere of radius $V_{d-1,\max}$. Thus the average volume of a sphere of radius $d-1$ lying in $\mathcal{Y}^{(k^*)}$ is given by

$$\begin{aligned} V_{d-1,\text{avg}}^{(k^*)} &= \frac{|\mathcal{Y}| \cdot V_{d-1,\text{avg}} - k^*(2V_{d-1,\max} - 1)}{|\mathcal{Y}| - k^*} \\ &\leq V_{d-1,\max} - 1. \end{aligned} \tag{12}$$

Solving (12) for k^* yields $k^* \geq (1-\delta) \cdot |\mathcal{Y}|/V_{d-1,\max}$. Since k^* is the smallest integer for which this relation holds, we obtain $k^* = \lceil (1-\delta) \cdot |\mathcal{Y}|/V_{d-1,\max} \rceil$. QED.

References

[1] V. D. Kolesnik and V. Y. Krachkovsky,"Generating Functions and Lower Bounds on Rates for Limited Error-Correcting Codes", *IEEE Trans. on Information Theory,* Vol.IT-37, no.3, pp.778-788, May 1991.

[2] F. Chung, J. Salehi, and V. Wei, "Optical Orthogonal Codes: Design, Analysis, and Applications, *IEEE Trans. on Information Theory,* Vol.IT-35, no.3, pp.595-604, May 1989.

[3] G. Beenker and K. Schouhamer Immink, "A Generalized Method for Encoding and Decoding Run-Length-Limited Binary Sequences," *IEEE Transactions on Information Theory,* Vol. IT-29, no. 5, pp.751-754, September 1983.

[4] Ø. Ytrehus, *Codes for Error Control,* Dr. Sc. thesis, University of Bergen, Norway, June 1989.

[5] Ø. Trehus, "On Error-Controlling (d,k) Constrained Block Codes," *Proceedings of the Fourth Joint Swedish-Soviet International Workshop on Information Theory,* Gotland, Sweden, August 1989.

n	$\lvert\mathcal{X}\rvert$	$V_{2,\max}$	$V_{2,\text{avg}}$	Lower Bounds on $A(d)$		
				$\lceil\lvert\mathcal{X}\rvert/V_{2,\max}\rceil$	$\lceil\lvert\mathcal{X}\rvert/V_{2,\text{avg}}\rceil$	Search
5	5	3	3	2	2	2
10	36	12	7.28	3	5	12
15	241	18	10.99	14	22	51
20	1632	47	15.75	35	104	286
25	11032	58	21.00	191	526	1605

Table 1: Parameters relevant to bounding the cardinality of a single-error correcting block code satisfying a (1,3) runlength constraint.

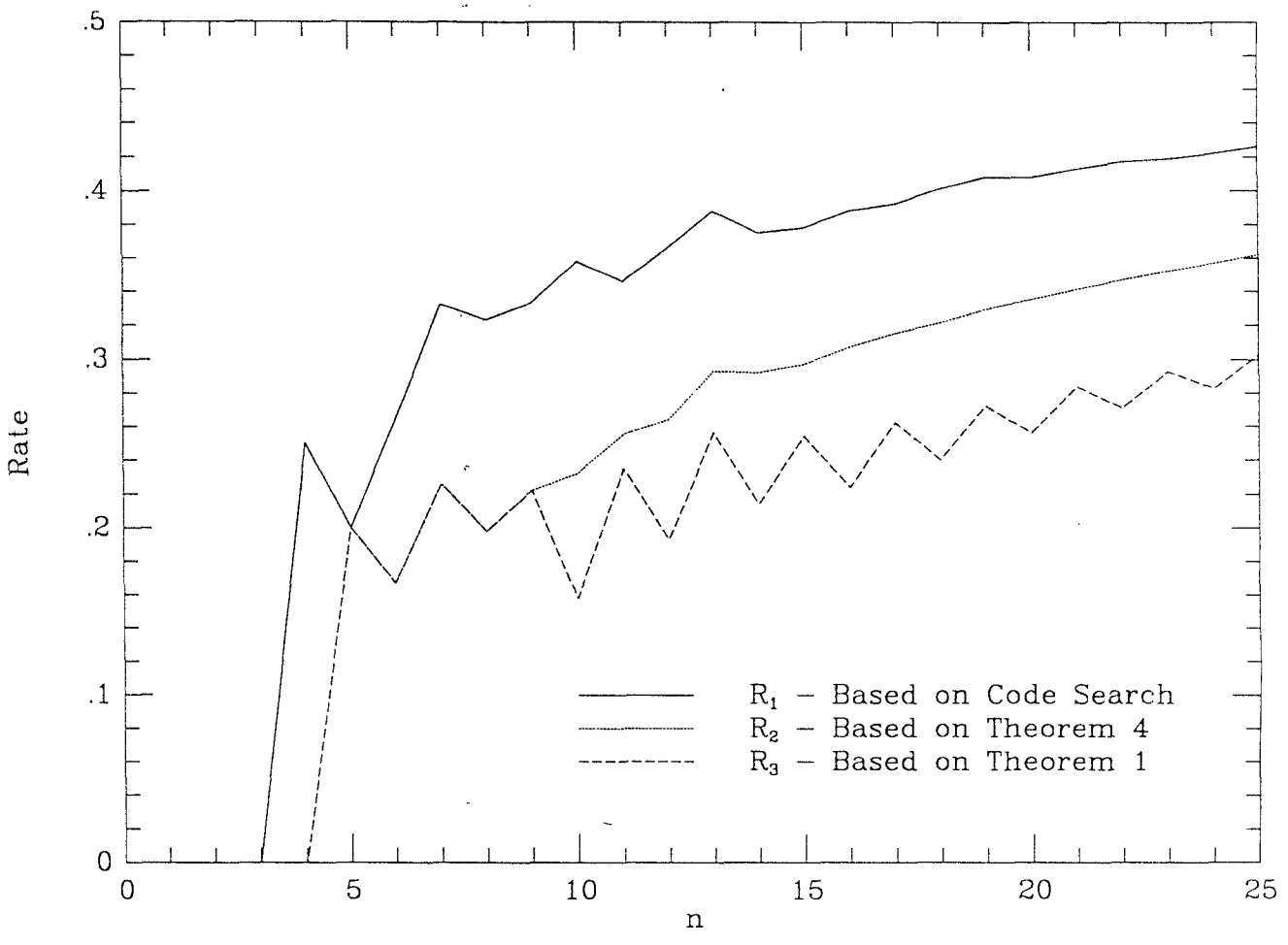


Figure 1: Lower bounds on the achievable rates of single-error correcting block codes satisfying a (1,3) runlength constraint.