Generalized sphere-packing bounds on the size of codes for combinatorial channels

Daniel Cullina, Student Member, IEEE and Negar Kiyavash, Senior Member, IEEE

Abstract—Many of the classic problems of coding theory are highly symmetric, which makes it easy to derive sphere-packing upper bounds on the size of codes. We discuss the generalizations of sphere-packing bounds to arbitrary error models. These generalizations become especially important when the sizes of the error spheres are nonuniform. The best possible sphere-packing bounds are solutions to linear programs. We derive a series of bounds from approximations to packing and covering problems and study the relationships and trade-offs between them. We show how to obtain upper bounds by optimizing across a family of channels that admit the same codes. We present a generalization of the local degree bound of Kulkarni and Kiyavash and use it to improve the best known upper bounds on the sizes of single deletion correcting codes and single grain error correcting codes.

I. INTRODUCTION

The classic problem of coding theory, correcting substitution errors in a vector of q-ary symbols, is highly symmetric. First, if s errors are required to change a vector x into another vector y, then s errors are also required to change y into x. Second, the number of vectors that can be produced from x by making up to s substitutions, the size of the sphere around x, does not depend on x. The sizes of these spheres play a crucial role in both upper and lower bounds on the size of the largest s-substitution-error-correcting codes. The Hamming bound is a sphere-packing upper bound and the Gilbert-Varshamov lower bound is a sphere-covering lower bound. The two symmetries that we have described make the proofs of the Hamming and Gilbert-Varshamov bounds extremely simple.

Many other interesting error models do not have this degree of symmetry. Substitution errors with a restricted set of allowed substitutions are sometimes of interest. The simplest example is the binary asymmetric errors, which can replace a one with a zero but cannot replace a zero with a one. Binary asymmetric errors have neither of the two symmetries we have described. Erasure and deletion errors differ from substitution errors in a more fundamental way: the error operation takes an input from one set and produces an output from another.

In this paper, we will discuss the generalizations of sphere-packing bounds to arbitrary error models. These generalizations become especially important when the sizes of the error spheres are nonuniform. Sphere-packing bounds are fundamentally related to linear programming and the best possible versions of the bounds are solutions to linear programs. In highly symmetric cases, including many classical error models, it is often possible to get the best possible sphere-packing bound without directly considering any linear programs. For less symmetric channels, the linear programming perspective becomes essential.

In fact, recently a new bound, explicitly derived via linear programming, was applied by Kulkarni and Kiyavash to find an upper bound on the size of deletion-correcting codes [2]. It was subsequently applied to grain errors [3], [4] and multipermutation errors [5]. We will refer to this as the local degree bound. The local degree bound constructs a dual feasible point for the sphere-packing linear program because computation of the exact solution is intractable. Deletion errors, like most interesting error models, act on an exponentially large input space. Because computation of the best possible packing and covering bounds is often intractable, simplified bounds such as the local degree bound are useful.

Sphere-packing and sphere-covering arguments have been applied in an ad hoc fashion throughout the coding theory literature. We attempt to present a unifying framework that permits such arguments in their most general form applicable to both uniform and nonuniform error sphere sizes. More precisely, we derive a series of bounds from approximations to packing and covering problems. The local degree bound of [2] is one of the bounds in the series. We associate each bound with an iterative procedure such that the original bound is the result of a single step. This characterization makes it easy to study the relationships between the bounds. We apply our generalization of the local degree bound to improve the best known upper bounds on the sizes of single deletion correcting codes and single grain error correcting codes.

We use the concept of a combinatorial channel to represent an error model in a fashion that makes the connection to linear programming natural. These bounds use varying levels of information about structure of the error model and consequently make trade-offs between performance and complexity. For example, one bound uses the distribution of the sizes of spheres in the space while another uses only the size of the smallest sphere.

In general, there are many different combinatorial channels that admit the same codes. However, each channel gives a different sphere-packing upper bound. We show that the
Hamming bound, which can be derived from a substitution error channel, the Singleton bound, which can be derived from an erasure channel, and a family of intermediate bounds provide an example of this phenomenon.

Our contributions can be summarized as follows. We provide a unified framework for describing upper bounds on code size. This allows us to make very general statements about the relative strengths of the bounds. In particular, our generalization of the local degree bound allows us to improve the best known upper bounds for a few channels. Finally, we demonstrate the power of considering families of channels with the same codes.

In Section II, we discuss the linear programs associated with sphere-packing bounds. In Section III, we present a generalization of the local degree bound that is related to an iterative procedure. We use this to improve the best known upper bounds on the sizes of single deletion correcting codes. In Section IV, we discuss sphere-packing bounds related to the degree sequence and average degree of a channel. In Section V-B, we discuss families of channels that have the same codes but give different sphere-packing bounds.

II. SPHERE-PACKING BOUNDS AND LINEAR PROGRAMS

In this section, we review the well-known connections between combinatorial channels, hypergraphs, sphere-packing bounds, and linear programming.

A. Notation

Let $X$ and $Y$ be finite sets. For a semiring $R$, let $R^X$ denote the set of $|X|$-dimensional column vectors of elements of $R$ indexed by $X$. Inequalities for vectors are element-wise. Let $R^{X \times Y}$ denote the set of $|X|$ by $|Y|$ matrices of elements of $R$ with the rows indexed by $X$ and the columns indexed by $Y$. Let $2^X$ denote the power set of $X$. Let $\mathbb{N}$ denote the set of nonnegative integers and let $[n]$ denote the set of nonnegative integers less than $n$: $\{0, 1, \ldots, n - 1\}$. Let $1$ be the column vector of all ones and let $0$ be the column vector of all zeros. For a set $S \subseteq X$, let $1_S \in [2]^X$ be the indicator column vector for the set $S$.

B. Combinatorial channels

We use the concept of a combinatorial channel to formalize a set of possible errors.

Definition 1. A combinatorial channel is a matrix $A \in [2]^{X \times Y}$, where $X$ is the set of channel inputs and $Y$ is the set of channel outputs. An output $y$ can be produced from an input $x$ by the channel if $A_{x,y} = 1$. Each row or column of $A$ must contain at least a single one, so each input can produce some output and each output can be produced from some input.

We will often think of a channel as a bipartite graph. In this case, the left vertex set is $X$, the right vertex set is $Y$, and $A$ is the bipartite adjacency matrix. We will refer to this bipartite graph as the channel graph.\(^1\)

For $x \in X$, let $N_A(x) \subseteq Y$ be the neighborhood of $x$ in the channel graph (the set of outputs that can be produced from $x$). The degree of $x$ is $|N_A(x)|$. For $y \in Y$, let $N_A(y) \subseteq X$ be the neighborhood of $y$ in the channel graph (the set of inputs that can produce $y$). In most cases, the channel involved will be evident and we will drop the subscript on $N$.

Note that $A1_{\{y\}} = 1_{N(y)}$ and $1^T_A A = 1^T_{N(\cdot)}$. Thus $A1$ is the vector of input degrees of the channel graph, $A^T1$ is the vector of output degrees, and $1^T A1$ is the number of edges.

We are interested in the problem of transmitting information through a combinatorial channel with no possibility of error. To do this, the transmitter only uses a subset of the possible channel inputs in such a way that the receiver can always determine which input was transmitted.

Definition 2. A code for a combinatorial channel $A \in [2]^{X \times Y}$ is a set $C \subseteq X$ such that for all $y \in Y$, $|N(y) \cap C| \leq 1$.

This condition ensures that decoding is always possible: if $y$ is received, the transmitted symbol must have been the unique element of $N(y) \cap C$.

C. Sphere-packing

A code is a packing of the neighborhoods of the inputs into the output space. The neighborhoods of the codewords must be disjoint and each neighborhood contains at least $\min_{x \in X} |N(x)|$ outputs. Thus the simplest sphere-packing upper bound on the size of a code $C$ is

$$|C| \leq \frac{|Y|}{\min_{x \in X} |N_A(x)|}.$$

This is the minimum degree upper bound, because $|N_A(x)|$ is the degree of $x$ in the channel graph of $A$. The sphere-packing upper bounds discussed in this paper are generalizations of and improvements on this bound.

Maximum input packing and its dual, minimum output covering, are naturally expressed as integer linear programs.

Definition 3. For a channel $A \in [2]^{X \times Y}$, the size of the largest input packing, or code, is

$$p(A) = \max_{w \in \mathbb{R}^X} 1^T w$$

subject to $A^T w \leq 1$.

\(^1\)An equivalent approach, taken by Kulkarni and Kiyavash [2], is to represent an error model by a hypergraph. A hypergraph consists of a vertex set and a family of hyperedges. Each hyperedge is a nonempty subset of the vertex set. A hypergraph is a nonempty subset of the vertex set and a family of hyperedges. Each hyperedge is a nonempty subset of the vertex set. A hyperedge can be produced from an input $x$ by the channel if $A_{x,h} = 1$. Each row or column of $A$ must contain at least a single one, so each input can produce some output and each output can be produced from some input.

Throughout this paper, we use the language of channels and bipartite channel graphs rather than that of hypergraphs. This allows us to refer to channel inputs and outputs using symmetric language and avoids any confusion between a hypergraph and its dual.
The size of the smallest output covering is
\[ \kappa(A) = \min_{z \in \mathbb{R}^Y} 1^T z \]
\[ \text{s. t. } A z \geq 1. \]

An output covering can be thought of as a strategy for the adversary operating the channel. Whenever the transmitter selects an input \( x \), there is at least one \( y \in N(x) \) included in the covering and the adversary can select this as the channel output. This restricts the number of distinguishable messages available to the transmitter to the size of the output covering. Note that the adversary’s choices do not require knowledge of the transmitter’s choice of code.

D. Fractional relaxations

The maximum independent set and minimum dominating set problems over general graphs are NP-Hard [6]. The approximate versions of these problems are also hard. The maximum independent set of an \( n \)-vertex graph cannot be approximated within a factor of \( n^{1-\epsilon} \) for any \( \epsilon \) in polynomial time unless any problem in NP can be solved in probabilistic polynomial time [7]. We seek efficiently computable bounds. These bounds cannot be good for all graphs, but they will perform reasonably well for many of the graphs that we are interested in.

The relaxed problem, maximum fractional set packing, provides an upper bound on the original packing problem.

**Definition 4.** Let \( A \in \{0,1\}^{X \times Y} \) be a channel. The size of the maximum fractional input packing in \( A \) is
\[ p^*(A) = \max_{w \in \mathbb{R}^X} 1^T w \]
\[ \text{s. t. } w \geq 0 \]
\[ A^T w \leq 1. \]
The size of the minimum fractional output covering is
\[ \kappa^*(A) = \min_{z \in \mathbb{R}^Y} 1^T z \]
\[ \text{s. t. } z \geq 0 \]
\[ A^T z \geq 1. \]

The fractional programs have larger feasible spaces, so \( p(A) \leq p^*(A) \) and \( \kappa^*(A) \leq \kappa(A) \). By strong linear programming duality, \( p^*(A) = \kappa^*(A) \).

Unlike the integer programs, the values of the fractional linear programs can be computed in polynomial time. However, we are usually interested in sequences of channels with exponentially large input and output spaces. In these cases, finding exact solutions to the linear programs is intractable but we would still like to know as much as possible about the behavior of the solutions. There is a vast literature devoted to iterative algorithms for solving linear programs. Because our programs are exponentially large, we cannot estimate their values by simulating these algorithms. Instead, we analyze the behavior of a few initial iterations of an algorithm. This leads us to value an unusual set of properties and to propose some very simple iterative algorithms that meet our needs.

**III. The local degree iterative algorithm**

Let \( A \in \{0,1\}^{X \times Y} \) be a channel. We can obtain an upper bound for \( p^*(A) \) (and consequently \( p(A) \)) by finding a feasible point in the program for \( \kappa^*(A) \). Given some \( t \in \mathbb{R}^Y \) such that \( t \geq 0 \) and \( (At)_x > 0 \) for all \( x \), let \( z \in \mathbb{R}^Y \) be the smallest scaling of \( t \) that is feasible for \( \kappa^*(A) \):
\[ z_y = \frac{t_y}{\min_{x \in X} (At)_x}. \]
We have the upper bound \( p^*(A) \leq 1^T z \), which we call \( \kappa_{MD}(A,t) \). The special case
\[ \kappa_{MD}(A,1) = \frac{|Y|}{\min_{x \in X} |N_A(x)|}, \]
is the minimum degree upper bound, which explains the subscript. Throughout, bounds notated by \( \kappa^* \) with a subscript come from the construction of a particular dual feasible point (i.e. feasible in the program for \( k^* \)) and bounds notated by \( p^* \) with a subscript come from the value of a relaxation of the primal program.

A. The local degree bound

For channels that are both input and output regular, computation of the sphere packing bound \( p^* \) is trivial: the minimum degree bound is exact. However, even a single low degree input will ruin the effectiveness of the minimum degree bound. To obtain a better upper bound on \( p(A) \) and \( p^*(A) \), we will construct a different feasible point in the program for \( \kappa^*(A) \) by making a small change to (1).

**Definition 5.** Let \( A \in \{0,1\}^{X \times Y} \) be a channel. For \( t \in \mathbb{R}^Y \) such that \( (At)_x > 0 \) for all \( x \in X \), define \( \varphi_A(t) \) in \( \mathbb{R}^Y \) as follows:
\[ \varphi_A(t)_y = \frac{t_y}{\min_{x \in N(y)} (At)_x}. \]
Define the local degree upper bound \( \kappa_{LD}(A,z) = 1^T \varphi_A(z) \).

**Lemma 1.** Let \( t \in \mathbb{R}^Y \) such that \( t \geq 0 \) and \( (At)_x > 0 \) for all \( x \in X \). Then \( \varphi_A(t) \) is feasible in the program for \( \kappa^*(A) \). If \( t \) is feasible for \( \kappa^*(A) \), then \( \varphi_A(t) \leq t \).

**Proof:** To demonstrate feasibility of \( z = \varphi_A(t) \), we need \( z \geq 0 \) and \( Az \geq 1 \). The first condition is trivially met. For \( x \in X \) and \( y \in N(x) \), we have
\[ z_y = \frac{t_y}{\min_{x' \in N(y)} (At)_{x'}} \geq \frac{t_y}{(At)_x}, \]
\[ (Az)_x = \sum_{y \in N(x)} z_y \geq \frac{1}{(At)_y} \sum_{y \in N(x)} t_y = 1 \]
and \( z \) is feasible.
If \( t \) is feasible, then \( At \geq 1 \). For all \( y \in Y \) we have
\[ z_y = \frac{t_y}{\min_{x \in N(y)} (At)_x} \leq t_y. \]

We can view the application of \( \varphi_A \) as a single iteration of an algorithm with the following intuitive description. Suppose that we have a vector \( t \in \mathbb{R}^Y \) that is a feasible vector in the program for \( \kappa^*(A) \). For any channel, we can take \( t = 1 \) as
an initial vector. At each input $x$, the total coverage, $(At)_x$, is at least one. The input $x$ informs each output in $N(x)$ that it can reduce its value by a factor of $(At)_x$. Each output $y$ receives such a message for each input in $N(y)$, then makes the largest reduction consistent with the messages.

An iteration fails to make progress under the following condition. From the definition $\varphi_A(t)_y = t_y$ if and only if $\min_{x \in N(y)}(At)_x = 1$. Thus $\varphi_A(t)_y = t$ if for all $y \in Y$ there is some $x \in N(y)$ such that $(At)_x = 1$. This algorithm is monotonic in each entry of the feasible vector, so it cannot make progress if its input is at the frontier of the feasible space.

Scaling the input by a positive constant does not affect the output of $\varphi_A$: for $c \in \mathbb{R}$, $c > 0$, $\varphi_A(t) = \varphi_A(ct)$. We could think of $\kappa_{\text{md}}(A, t)$ as involving an iterative procedure as well. It has the same scaling property. In contrast to the local degree iteration, the minimum degree iteration always stops after a single step because the output vector is a constant multiple of the input. The local degree iteration scales different entries in the initial vector by different amounts, so it is possible for it to make progress for multiple iterations.

B. Application to the single deletion channel

Let $A_n$ be the $n$-bit 1-deletion channel. The input to the binary single deletion channel is a string $x \in \{0, 1\}^n$ and the output is a substring of $x$, $y \in \{0, 1\}^{n-1}$. Each output vertex in $A_n$ has degree $n + 1$. Thus $\kappa^*(A_n) \geq \kappa_{\text{md}}(A_n) = \frac{2^n}{n+1}$.

Levenshtein [8] showed that

$$\kappa^*(A_n) \leq \frac{2^n}{n+1}(1 + o(1)).$$

Kulkarni and Kiyavash computed the local degree upper bound, or equivalently $\varphi_{A_n}(1)$ [2]. This shows that $\kappa^*(A_n)$ is at most

$$\frac{2^n}{n-1} = \frac{2^n}{n+1}(1 + \frac{2}{n-1}) = \frac{2^n}{n+1}(1 + O(n^{-1})).$$

Recently, Fazeli et al. found a fractional covering for $A_n$ that provides a better upper bound [9].

For the remainder of this section we abbreviate $\varphi_{A_n}$ by $\varphi$. In this section, we compute $\varphi \circ \varphi(1)$ for these channels and analyze the values of these points. We show that Fazeli’s improved covering is related to the covering $\varphi \circ \varphi(1)$, but $\varphi \circ \varphi(1)$ provides a better bound asymptotically.

More precisely, the upper bound from $\varphi \circ \varphi(1)$, given in Theorem 2, shows that $\kappa^*(A)$ is at most

$$\frac{2^n}{n-1} \left(1 - \frac{2}{n-1} + O(n^{-2})\right) = \frac{2^n}{n+1}(1 + O(n^{-2})).$$

The covering in Fazeli et al. gives an upper bound of

$$\frac{2^n}{n+1} \left(1 + \frac{1}{n-1} + O(n^{-2})\right).$$

A run in a string is a maximal set of consecutive indices that have the same symbol. Let $r, u, b \in \mathbb{N}^{[2]^*}$ be vectors such that for all $x \in [2]^*$, $r_x$ is the number of runs in $x$, $u_x$ is the number of length-one runs, or unit runs, in $x$, and $b_x$ is the number of unit runs at the start or end of $x$.

Proofs of the theorems and lemmas stated in this section can be found in Appendix A.

**Theorem 1.** Let

$$f(r, u, b) = \frac{1}{r} \left(1 + \max(2(n - b - 2, 0))\right)^{-1}.$$

Then the vector $z_y = f(r_y, u_y, b_y)$ is feasible for $\kappa^*(A_n)$, so $\kappa^*(A_n) \leq 1^Tz_y$.

**Theorem 2.** For $n \geq 2$,

$$\kappa^*(A_n) \leq \frac{2^n}{n+1} \left(1 + \frac{26}{n(n-1)}\right).$$

Now we will compare this bound to the bound corresponding to the cover of Fazeli et al. Let

$$f'(r, u, b) = \begin{cases} \frac{1}{r} (1 - \frac{u-b}{r}) & u - b \geq 2 \\ \frac{1}{3} & u - b \leq 1. \end{cases}$$

Fazeli et al. establish that $y = f'(r_y, u_y, b_y)$ is feasible for $\kappa^*(A_n)$. Compare this with the cover given by $f'$ and note that the coefficient on $u$ is $1$ in $f'$ and $2$ in $f$.

**Lemma 2.** Let $y = f'(r_y, u_y, b_y)$. Then

$$1^Tz \geq \frac{2^n - 2}{n+1} \left(1 + \frac{1}{\min\{n-1, 2n-2\}}\right).$$

This shows that the bound of Theorem 2 is asymptotically better than the bound corresponding to the cover of Fazeli et al. We could continue to iterate $\varphi$ to produce even better bounds. The fractional covers produced would depend on more statistics of the strings. For example, the value at a particular output of the cover produced by the third iteration of $\varphi$ would depend on the number of runs of length two in that output string, in addition to the total number of runs and the number of runs of length one.

The largest known single deletion correcting codes are the Varshamov-Tenengolts (VT) codes [8]. The largest length-$n$ VT code, denoted $VT_0$, contains at least $\frac{2^n}{n+1}$ codewords, so this sequence of codes is asymptotically optimal. $VT_0$ is known to be a maximum independent set for $n \leq 10$, but this question is open for larger $n$ [10]. Kulkarni and Kiyavash computed the exact value of $\kappa^*(A_n)$ for $n \leq 14$ [2]. For $7 \leq n \leq 14$, the gap between $\kappa^*(A_n)$ and the size of the VT codes was at least one, so it is unlikely that sphere-packing bounds will resolve the optimality of the VT codes for larger $n$. Despite this, it would be interesting to know whether $\kappa^*(A_n) \leq \frac{2^n}{n+1} + O(2^{cn})$ for some constant $c < 1$.

C. Application to the single grain error channel

Recently, there has been a great deal of interest in grain error channels, which are related to high-density encoding on magnetic media. A grain in a magnetic medium has a single polarization. If an encoder attempts to write two symbols to a single grain, only one of them will be retained. Because the locations grain boundaries are generally unknown to the encoder, this situation can be modeled by a channel.

Mazumdar et al. applied the degree sequence bound to non-overlapping grain error channels [11]. Sharov and Roth
applied the degree sequence bound to both non-overlapping and overlapping grain error channels [12]. We discuss the degree sequence bound and its relationship to the local degree bound in Section IV. Kashyap and Zémor applied the local degree bound to improve on Mazumdar et al. for the 1, 2, or 3 error cases [3]. They conjectured an extension for larger numbers of errors. Gabrys et al. applied the local degree bound to improve on Sharov and Roth [4].

The input and output of this channel are strings \(x, y \in [2]^n\). To produce an output from an input, select a grain pattern with at most one grain of length two and no larger grains. The grain of length two, if it exists, bridges indices \(j\) and \(j + 1\) for some \(0 \leq j \leq n - 2\). Then the channel output is

\[
y_i = \begin{cases} 
  x_i & i \neq j \\
  x_{i+1} & i = j
\end{cases}
\]

If \(x_j = x_{j+1}\) or if there is no grain of length two, then \(y = x\).

The degree of an input string is equal to the number of runs \(r\): each of the \(r - 1\) run boundaries could be bridged by a grain or there could be no error. A grain error reduces the number of runs by 0, 1, or 2. The number of runs is reduced by 1 if \(j = 0\) and \(x_0 \neq x_1\), by 2 if \(j \geq 1, x_j \neq x_{j+1},\) and \(x_{j-1} = x_{j+1}\), and by 0 otherwise. Equivalently, the number of runs is reduced by 1 if a length-1 run at index 0 is eliminated and by 2 if a length-1 run elsewhere is eliminated. In the previous section, we let \(u_x\) be the number of length-1 runs in \(x\) and \(b_x\) be the number of length-1 runs appearing at the start or end of \(x\). For the grain channel, we need to distinguish between length-1 runs at the start and at the end, so let \(b_x^L\) and \(b_x^R\) count these.

**Theorem 3.** Let \(A_n\) be the \(n\)-bit 1-grain-error channel. The vector

\[
z_y = \frac{1}{r_y} \left(1 + \frac{2n_y - 2b_y^L - b_y^R}{(r_y + y)(r_y + 1)}\right)^{-1}
\]

is feasible for \(\kappa^*(A_n)\).

By applying the techniques used in the proof of Theorem 2, it can be shown that Theorem 3 implies that \(\kappa^*(A_n) = \frac{2^{n+1}}{n+2}(1 + O(n^{-2}))\).

### IV. The Degree Sequence Upper Bound

If a channel \(A \in [2]^X\times Y\) is input regular, then the local degree bound reduces to the minimum degree bound. We have \(A1 = 1d\) and \(\kappa^*_D(A, 1) = \kappa^*_M(A, 1) = |Y|/d\). The advantage of the local degree bound is robustness to variation in the input degree distribution. The degree threshold upper bound is an alternative technique for dealing with nonuniform combinatorial channels that predates the local degree bound by many decades. Levenshtein applied this idea to obtain an upper bound on codes for the deletion channel [8]. Kulkarni and Kiyavash applied the local degree bound to the deletion channel and showed that the resulting bound improved on Levenshtein’s result [2].

The degree threshold bound is a simple generalization of the minimum degree bound and the basic idea behind the bound does not require linear programming. Pick a degree threshold \(d\) and let \(S = \{x \in X : |N(x)| < d\}\), the set of low degree inputs. Each member of \(S\) appears at most once in the maximum packing. Applying the minimum degree bound to \(X \setminus S\) gives

\[
p(A) \leq \frac{|X| - |S|}{d} + |S|.
\]

This approach is effective when the degree distribution concentrates around its mean but still has a few vertices with much lower degree.

This idea can be taken a bit further. For any code \(C \subseteq X, \sum_{x \in C} |N(x)| \leq |Y|\). The size of the largest input set \(C\) satisfying this inequality is an upper bound on the size of the largest code. This set can be found greedily by repeatedly adding the minimum degree remaining input vertex. Call this the degree sequence upper bound.

The degree sequence upper bound is monotonic and decreasing in the degree of each vertex. The degree threshold bound corresponds to a simplified degree sequence containing only degrees 1 and \(d\).

Although its definition does not require a linear program, the degree sequence bound still has a nice linear programming interpretation. Taking this perspective, we compare the performance of the degree sequence bound to the local degree bound for arbitrary channels and show that the local degree bound is always better.
A. Linear programs for the degree sequence bound

While the local degree upper bound is naturally expressed as a feasible point in the program for \( \kappa^* \), the easiest linear programming interpretation of the degree sequence bound works differently. The degree sequence upper bound is the value of a further relaxation of the program for \( p^* \). It turns out that the minimum degree upper bound is easily expressed as both a dual feasible point and a primal relaxation. Section III included the former interpretation and the latter is given here.

Let \( A \in [2]^{|X \times Y|} \) and a vector \( t \in \mathbb{R}^{|Y|} \), define
\[
p^*_{\text{MD}}(A,t) = \max_{w \in \mathbb{R}^{|X|}} 1^T w \quad \text{s. t. } w \geq 0 \quad t^T A^T w \leq t^T 1.
\]

The solution to this program puts all of the weight on the minimum degree input \( \arg \min_x (At)_{x} \), so \( p^*_{\text{MD}}(A,t) = \kappa^*_{\text{MD}}(A,t) \).

Recall that \( \mathcal{A}_1 \) is the vector of input degrees of the channel graph of \( A \). Thus the main constraint of the program for \( \kappa^*_{\text{MD}}(A,1) \) is \( \sum_{x \in X} |N(x)|w_x \leq |Y| \). In a code, each vertex can only be included once. We can capture this fact and improve the upper bound by adding the additional constraint \( w \leq 1 \) to the program.

**Definition 6.** For a channel \( A \in [2]^{|X \times Y|} \) and a vector \( t \in \mathbb{R}^{|Y|} \) such that \((At)_x > 0 \) for all \( x \in X \), define the degree sequence bound
\[
p^*_{\text{DS}}(A,t) = \max_{w \in \mathbb{R}^{|X|}} 1^T w \quad \text{s. t. } 0 \leq w \leq 1 \quad t^T A^T w \leq t^T 1.
\]

The degree sequence upper bound is tight: for a given input degree distribution and output space size, there is some channel where the neighborhoods of the small degree inputs are disjoint. For this channel, the degree sequence upper bound is tight. The bound cannot be improved with incorporating more information about the structure of the channel.

The local degree upper bound, which incorporates information about the channel beyond the degree sequence, is always at least as good as the degree sequence bound. To see this, we associate the degree sequence bound with a particular feasible point in the program for \( \kappa^*(A) \).

**Lemma 3.** Let \( A \in [2]^{|X \times Y|} \) be a channel and let \( t \in \mathbb{R}^{|Y|} \) such that \((At)_x > 0 \) for all \( x \in X \). Then there is a vector \( \psi_A(t) \in \mathbb{R}^{|X|} \) such that
\[1^T \psi_A(t) = p^*_{\text{DS}}(A,t), \quad \varphi_A(t) \leq \psi_A(t) \quad \psi_A(t) \text{ is feasible in the program for } \kappa^*(A)\]

**Proof:** The vector \( At \) contains the weighted degree of each input under the output weighting \( t \). For \( d \in \mathbb{R} \), define the following sets of inputs:
\[
S(d) = \{x \in X : (At)_x < d\}, \\
S'(d) = \{x \in X : (At)_x \leq d\}.
\]

Define \( f(d) = 1^T_{S(d)} At \), the sum of the weighted degrees of all of the inputs with weighted degree less than \( d \), and \( f'(d) = 1^T_{S'(d)} At \). Both \( f(d) \) and \( f'(d) \) are nondecreasing functions of \( d \) and \( f(d) \leq f'(d) \). Because \( f(0) = 0 \leq 1^T t \leq 1^T At = f'(1^T t) \), there is some \( d \) such that \( f(d) \leq 1^T t \leq f'(d) \). Then there is some \( \lambda \) satisfying \( 1^T t = \lambda f'(d) + (1 - \lambda) f(d) \).

First we establish that \( p^*_{\text{DS}}(A,t) = (1 - \lambda) |S(d)| + \lambda |S'(d)| \) by constructing a primal feasible point and a dual feasible point with this value.

The point \( w = (1 - \lambda) 1_{S(d)} + \lambda 1_{S'(d)} \) is feasible for the primal program for \( p^*_{\text{DS}}(A,t) \). This puts a weight of one, the maximum possible weight, on each of the inputs with degree below the threshold and fractional weight of \( \lambda \) on inputs with degree equal to the threshold. To see that the nontrivial feasibility condition is satisfied, observe that \( 1^T A^T w = (1 - \lambda) f(d) + \lambda f'(d) = 1^T t \).

The dual program for \( p^*_{\text{DS}}(A,t) \) is
\[
\min_{c,z \in \mathbb{R}^{|X|}, z \in \mathbb{R}} 1^T tc + 1^T z \quad \text{s. t. } c \geq 0 \quad z \geq 0 \quad Atc + z \geq 1.
\]

The point \( c = \frac{1}{t} \), \( z_x = \max(0, 1 - \frac{(At)_x}{d}) \) is feasible in the dual program. Note that \( z_x > 0 \) exactly for those \( x \in S(d) \).

The value of this point is
\[
\frac{1^T t}{d} + \sum_{x \in S(d)} \frac{d - (At)_x}{d} = |S(d)| + \frac{1^T t - 1^T_{S(d)} At}{d} \quad \text{s. t. } c \geq 0 \quad z \geq 0 \quad Atc + z \geq 1.
\]

The final equality follows from the fact that each vertex in \( S'(d) \setminus S(d) \) has weighted degree \( d \).

Next, we construct \( \psi_A(t) \) from \( c, z \):
\[
\psi_A(t)_y = t_y \left( c + \sum_x \frac{z_x A_{x,y}}{(At)_x} \right).
\]

To establish the first claim, we compute
\[
1^T \psi_A(t) = 1^T tc + \sum_y t_y \sum_x \frac{z_x A_{x,y}}{(At)_x} \quad 1^T tc + \sum_x \frac{z_x (At)_x}{d} \sum_y A_{x,y} t_y
\]
\[
= 1^T tc + 1^T z.
\]

Substituting the values \( c = \frac{1}{t} \) and \( z_x = \max(\frac{d - (At)_x}{d}, 0), \)
we obtain
\[ \frac{\psi_A(t)}{\tau} = \frac{1}{d} + \sum_{x \in N(y)} \frac{1}{(At)_x} \max \left( \frac{d - (At)_x}{d}, 0 \right) \]
\[ = \frac{1}{d} + \sum_{x \in N(y)} \max \left( \frac{1}{(At)_x} - \frac{1}{d}, 0 \right) \]
\[ \geq \frac{1}{d} + \max_{x \in N(y)} \max \left( \frac{1}{(At)_x} - \frac{1}{d}, 0 \right) \]
\[ = \max_{x \in N(y)} \frac{1}{(At)_x} \frac{1}{d} \]
\[ \geq \max_{x \in N(y)} \frac{1}{(At)_x} \]
\[ = \frac{1}{\tau} \varphi_A(t). \]

This establishes the second claim.

Because \( \varphi_A(t) \) is feasible in \( \kappa^*(A) \), \( \psi_A(t) \) is as well, which establishes the third claim.

**Theorem 4.** Let \( A \in [2]^{X \times Y} \) be a channel and let \( t \in \mathbb{R}^Y \) such that \((At)_x > 0\) for all \( x \in X \). Then \( \kappa_{LD}(A, t) \leq p_{GS}(A, t) \).

**Proof:** This follows immediately from the first and second claims of Lemma 3.

This interpretation of the degree sequence bound allows us to iteratively construct dual feasible points, but these points are dominated by those produced by the local degree algorithm. However, this interpretation does give some intuition about the source of the superior performance of the local degree bound. When multiple low-degree inputs have a common output, the local degree bound takes advantage of this fact. This information is not contained in the degree distribution.

**V. Confusability Graphs and Families of Channels**

Sphere packing upper bounds are obtained from combinatorial channels. However, it is well-known that for any channel there is a simpler object that also characterizes the set of codes: the confusability graph. Furthermore, any particular confusability graph arises from many combinatorial channels. To obtain upper bounds on the size of codes for one channel it can be useful to consider the sphere packing bounds that arise from some other equivalent channel. At the end of this section, we show how the Hamming and Singleton bounds are an example of this phenomenon.

**A. Confusability graphs and independent sets**

First we give a few standard definitions, generally following [15], among others.

**Definition 7.** For a channel \( A \in [2]^{X \times Y} \), the confusability graph of \( A \) has vertex set \( X \). Distinct vertices \( u \) and \( v \) are adjacent in the confusability graph of \( A \) if and only if \( N(u) \) and \( N(v) \) intersect.

**Definition 8.** Let \( G \) be an undirected simple graph with vertex set \( X \). A set \( S \subseteq X \) is independent in \( G \) if and only if for all \( u, v \in S \), \( u \) and \( v \) are not adjacent. The maximum size of an independent set in \( G \) is denoted by \( \alpha(G) \).

Now we have a second important characterization of codes.

**Lemma 4.** Let \( G \) be the confusability graph for a channel \( A \in [2]^{X \times Y} \). Then a set \( C \subseteq X \) is code for \( A \) if and only if it is an independent set in \( G \). Thus \( \alpha(G) = p(A) \).

**Proof:** A set \( C \) is not a code if and only if there is some \( y \) such that \( N(y) \) contains distinct codewords \( u \) and \( v \). Equivalently \( N(u) \) intersects \( N(v) \) and \( u \) and \( v \) are adjacent in \( G \).

The confusability graph does not contain enough information to recover the original channel graph, but it contains enough information to determine whether a set is a code for the original channel.

**B. Families of channels with the same codes**

There are many different channels that have \( G \) as a confusability graph. A **clique** in a graph \( G \) is a set of vertices \( S \) such that for all distinct \( u, v \in S \), \( \{u, v\} \in E(G) \). If \( G \) is the confusability graph for a channel \( A \in [2]^{X \times Y} \), then for each \( y \in Y \), \( N(y) \) is a clique in \( G \). Let \( \Omega \subseteq 2^X \) be a family of cliques that covers every edge in \( G \). This means that for all \( \{u, v\} \in E(G) \), there is some \( S \in \Omega \) such that \( u, v \in S \). Let \( H \in [2]^{X \times \Omega} \) be the vertex-clique incidence matrix: \( H_{x,s} = 1 \) is \( x \in S \) and \( H_{x,s} = 0 \) otherwise. Then \( \alpha(G) = p(H) \).

Thus each family of cliques that covers every edge gives us an integer linear program that expresses the maximum independent set problem for \( G \). These programs all contain the same integer points, the indicators of the independent sets of \( G \). However, their polytopes are significantly different so the fractional relaxations of these programs give widely varying upper bounds on \( \alpha(G) \).

Each edge in \( G \) is a clique, so \( E(G) \) is one natural choice for \( \Omega \). Then \( \alpha(G) = p(H_E) \), where \( H_E \in [2]^{X \times E(G)} \) is the vertex edge incidence matrix for \( G \). However, relaxing the integrality constraint for this program gives a useless upper bound. The vector \( w = \frac{1}{2}1 \) is feasible, so \( p^*(H_E) \geq \frac{|X|}{2} \) regardless of the structure of \( G \).

**Definition 9.** Let \( \Omega \) be the set of maximal cliques in \( G \) and let \( H_\Omega \in [2]^{X \times \Omega} \) be the vertex-clique incidence matrix. Define the minimum clique cover of \( G \), \( \theta(G) = \kappa(H_\Omega) \) and the minimum fraction clique cover \( \theta^*(G) = \kappa^*(H_\Omega) \).

Every edge is contained in at least one maximal clique, so \( \alpha(G) = p(H_\Omega) \). Unlike the program derived from the edge set, \( \theta^*(G) \) gives a nontrivial upper bound on \( \alpha(G) \). In fact, \( \theta^*(G) \) is the best sphere packing bound for any channel that has \( G \) as its confusability graph.

**Lemma 5.** Let \( G \) be a graph with vertex set \( X \) and let \( \Omega_1, \Omega_2 \subseteq 2^X \) be families of cliques that cover every edge in \( G \). Let \( H_1, H_2 \) be the vertex-clique incidence matrices for \( \Omega_1 \) and \( \Omega_2 \) respectively. If for each \( R \in \Omega_1 \) there is some \( S \in \Omega_2 \) such that \( R \subseteq S \), then \( p^*(H_2) \leq p^*(H_1) \).

**Proof:** A clique \( S \) gives the constraint \( \sum_{x \in S} w_x \leq 1 \) in \( p \). If \( R \in \Omega_1 \), \( S \in \Omega_2 \), and \( R \subseteq S \), then the constraint from \( R \) is implied by the constraint for \( S \). Any additional cliques in \( \Omega_2 \) can only reduce the feasible space for \( p(H_2) \). Thus the
feasible space for \( p(H_2) \) is contained in the feasible space for \( p(H_1) \).

**Corollary 1.** Let \( A \in \mathbb{Z}^{X \times Y} \) be a channel and let \( G \) be the confusability graph for \( A \). Then \( \theta^*(G) \leq \kappa^*(A) \).

**Proof:** For each output \( y \in Y \), \( N(y) \) is a clique in \( G \) and these cliques cover every edge of \( G \). Each clique in \( G \) is contained in a maximal clique, so the claim follows immediately from Lemma 5.

Corollary 1 suggests that we should ignore the structure of our original channel \( A \) and try to compute \( \theta^*(G) \) instead of \( \kappa^*(A) \). However, there is no guarantee that we can efficiently construct the linear program for \( \theta^*(G) \) by starting with \( G \) and searching for all of the maximal cliques. We are often interested in graphs with an exponential number of vertices. Even worse, the number of maximal cliques in \( G \) can grow exponentially in the number of vertices. To demonstrate this, consider a complete \( k \)-partite graph with 2 vertices in each part. If we select one vertex from each part, we obtain a maximal (and also maximum) clique. The graph has \( 2k \) vertices, but there are \( 2^k \) maximal cliques.

The fractional clique cover number has been considered in the coding theory literature in connection with the Shannon capacity of a graph, \( \Theta(G) \). The capacity of a combinatorial channel \( A \) is \( \lim_{n \to \infty} p(A^n)^\frac{1}{n} \), the number of possible messages per channel use when the channel can be used many times. Like \( p(A) \), the capacity of the channel depends only on its confusability graph. Thus the Shannon capacity of a graph \( G \) can be defined as the capacity of a channel with confusability graph \( G \). The Shannon capacity of a graph is at least as large as the maximum independent set and is not known to even be computable. Shannon used something equivalent to a clique cover as an upper bound for Shannon capacity [13]. Rosenfeld showed the connection between Shannon’s bound and linear programming [14]. Shannon also showed that the feedback capacity of a combinatorial channel \( A \) is \( p^*(A) \). Lovasz introduced the Lovasz theta function of a graph, \( \vartheta(G) \), and showed that it was always between the Shannon capacity and the fractional clique cover number [15]. All together, we have

\[
\alpha(G) \leq \Theta(G) \leq \vartheta(G) \leq \theta^*(G).
\]

The Lovasz theta function is derived via semidefinite programming and consequently is not a sphere-packing bound.

There are also several connections between these concepts and communication over probabilistic channels. For a combinatorial channel \( A \), the minimum capacity over the probabilistic channels with support \( A \) is \( p^*(A) \). Recently Dalai has proven upper bounds on the reliability function of a probabilistic channel that are finite for all rates above the (logarithmic) Shannon capacity of the underlying confusion graph, in contrast to previous bounds that were finite for rates above \( \log p^*(A) \) [16]. The idea of multiple channels with the same confusion graph plays an important role here.

### C. Hamming and Singleton Bounds

Sometimes a channel has some special structure that allows us to find an easily described family of channels with the same codes. Then we can optimize over the family by computing the bound for each channel and using the best. This technique has been successfully applied to deletion-insertion channels by Cullina and Kiyavash [1]. Any code capable of correcting \( s \) deletions can also correct any combination of \( s \) total insertions and deletions. Two input strings can appear in an \( s \)-deletion-correcting code if and only if the deletion distance between them is more than \( s \). In the asymptotic regime with \( n \) going to infinity and \( s \) fixed, each channel in the family becomes approximately regular. Thus the degree threshold bound gives a good approximation to the exact sphere-packing bound for these channels. The best bound comes from a channel that performs approximately \( \frac{q^2}{q+1} \) deletions and \( \frac{s}{q+1} \) insertions, where \( q \) is the alphabet size.

In this section we present a very simple application of this technique. Consider the channel that takes a \( q \)-ary vector of length \( n \) as its input, erases \( a \) symbols, and substitutes up to \( b \) symbols. Thus there are \( q^a \) channel inputs, \( \binom{n}{a} q^{n-a} \) outputs, and each input can produce \( \binom{n}{a} q^{n-a} \) outputs, and each output can produce \( \frac{q^b}{q+1} \) possible outputs. Two inputs share a common output if and only if their Hamming distance is at most \( s = a + 2b \). For each choice of \( n \) and \( s \), we have a family of channels with identical confusability graphs. Call the \( q \)-ary \( n \)-symbol \( a \)-erasures \( b \)-substitution channel \( A_{q,n,a,b} \). These channels are all input and output regular, so

\[
\kappa^*(A_{q,n,a,b}) = \frac{\binom{n}{a} q^{n-a}}{\binom{n}{a} \sum_{i=0}^{b} \binom{n-a}{i} (q-1)^i} = \frac{\sum_{i=0}^{b} \binom{n-a}{i} (q-1)^i}{\sum_{i=0}^{b} \binom{n-a}{i} (q-1)^i}.
\]

Two special cases give familiar bounds. For even \( s \), setting \( a = 0 \) and \( b = s/2 \) produces the Hamming bound:

\[
\kappa^*(A_{q,n,0,s/2}) = \frac{q^n}{\sum_{i=0}^{s/2} \binom{s}{i} (q-1)^i}.
\]

Setting \( a = s \) and \( b = 0 \) produces the Singleton bound:

\[
\kappa^*(A_{q,n,s,0}) = q^{n-s}.
\]

For \( q = 2 \), the Hamming bound is always the best bound in this family. When \( q \) is at least 3, each bound in the family is the best for some region of the parameter space.

**Lemma 6.** \( \kappa^*(A_{q,n,a,b}) \leq \kappa^*(A_{q,n,a+2,b-1}) \) when \( a + qb \leq n-1 \).

**Proof:** We can rewrite the initial inequality as

\[
\sum_{i=0}^{b-1} \binom{n-a-2}{i} (q-1)^i \geq \sum_{i=0}^{b-1} \binom{n-a}{i} (q-1)^i \geq q^2 \sum_{i=0}^{b-1} \binom{n-a-2}{i} (q-1)^i.
\]
To simplify (2), we use the following identity:
\[
\sum_{i=0}^{b} \binom{n-c+1}{i} (q-1)^i = \sum_{i=0}^{b} \left(\binom{n-c}{i-2} + 2 \binom{n-c}{i-1} + \binom{n-c}{i}\right) (q-1)^i = \sum_{i=0}^{b} \binom{n-c}{i} (q-1)^i + 2 \sum_{i=0}^{b-1} \binom{n-c}{i} (q-1)^{i+1} + \binom{n-c}{b} (q-1)^b + \binom{n-c}{b-1} (q-1)^b (\frac{n-c-b+1}{b} - q + 1) + q^2 \sum_{i=0}^{b-1} \binom{n-c}{i} (q-1)^i.
\]

By setting \(c = a + 2\), we can use this to rewrite the left side of (2). Eliminating the common term from both sides of the inequality gives
\[
\left(\frac{n-a-2}{b-1}\right) (q-1)^b \left(\frac{n-a-b-1}{b} - q + 1\right) \geq 0
\]
\[
\frac{n-a-b-1}{b} - q + 1 \geq 0
\]
\[
\frac{n-a-1}{b} - qb \geq 0
\]

which proves the claim.

**Theorem 5.** Let \(q, n, s \in \mathbb{N}\) such that \(q \geq 3\), \(0 \leq s \leq n-1\), and \(s\) even. Then
\[
\arg\min_{0 \leq b \leq s/2} \kappa^*(A_{q,n,s-2b,b}) = \begin{cases} \frac{s}{2} & s \leq \frac{n-1}{q} \\ \left\lfloor \frac{n-1-s}{q-2} \right\rfloor & s \geq \frac{n}{q}(n-1) \end{cases}
\]

For fixed \(\delta, \frac{q}{2} \leq \delta \leq 1\), and \(s = \delta n\)
\[
\lim_{n \to \infty} \frac{1}{n} \log \min_{\frac{q}{2} \leq b \leq s/2} \kappa^*(A_{q,n,s-2b,b}) = (1 - \delta) \log(q-1).
\]

**Proof:***

Let \(a + 2b = s\), so \(a + qb = s + (q-2)b\).

Lemma 6 allows us to determine the value of \(b\) minimizing \(\kappa^*(A_{q,n,s-2b,b})\). When \(s + (q-2) \frac{q}{2} \leq n-1\), or equivalently \(s \leq \frac{n}{q}(n-1), \kappa^*(A_{q,n,0,s/2})\) is the smallest in the family. For \(b \geq 1\)
\[
\kappa^*(A_{q,n,a+2b,b-1}) \geq \kappa^*(A_{q,n,a,b}) \leq \kappa^*(A_{q,n,a-2b+1})
\]
if and only if
\[
b \leq \frac{n-1-s}{q-2} \leq b + 1.
\]

Let \(b^*\) be the optimal choice of \(b\). Then
\[
\lim_{n \to \infty} \frac{b^*}{n} = 1 - \frac{\delta}{q-2},
\]
\[
\lim_{n \to \infty} \frac{n-s-2b^*}{n} = 1 - \delta + 2 \frac{1 - \delta}{q-2} = \frac{q(1 - \delta)}{q-2},
\]
\[
\lim_{n \to \infty} \frac{b^*}{n} = \frac{1}{q}.
\]

Finally,
\[
\lim_{n \to \infty} \frac{1}{n} \log \min_{\frac{q}{2} \leq b \leq s/2} \kappa^*(A_{q,n,s-2b,b}) = (1 - \delta) \log(q-1).
\]

**Figure 2.** Sphere-packing bounds for channel performing substitution errors and erasures. The curved line is the Hamming bound, which is \(\lim_{n \to \infty} \frac{1}{n} \log \kappa^*(A_{4,n,0,s/2})\). The upper straight line is the Singleton bound, which is \(\lim_{n \to \infty} \frac{1}{n} \log \kappa^*(A_{4,n,s,0})\). The straight line running from \((\frac{1}{2}, \frac{1}{2} \log 3)\) to \((1, 0)\) is the optimized sphere-packing bound, \(\lim_{n \to \infty} \frac{1}{n} \log \min_{0 \leq b \leq s/2} \kappa^*(A_{4,n,s-2b,b})\).

There are several open questions regarding families of channels with the same confusability graphs. Under what conditions can we find these families? What is the relationship between these families and distance metrics? When we have a family of channels that are not input or output regular, what should we do to get the best bounds?
VI. CONCLUSION

We have discussed two aspects of the problem of finding upper bounds on the size of codes for combinatorial channels: fractional coverings for a particular channel and families of channels with the same codes. In both cases, there is a well-defined optimal version of the bound: for a particular channel there is the minimum weight fractional covering, and for a family there is the minimum fractional clique cover of the confusion graph. In both cases, finding these optimal bounds can be intractable.

When the channel is input-regular, the minimum degree, degree sequence, and local degree upper bounds here are equivalent, but not necessarily equal to the fractional covering number. The local degree bound is always at least as good as the degree sequence bound but uses more information about the structure of the channel. The local degree bound can be iterated to obtain stronger bounds. The best sphere packing bound for a given channel can be much weaker than the best sphere packing bound for some other channel that admits the same codes. Consequently, finding a family of channels equivalent to the channel of interest can be very powerful.

APPENDIX A

Proofs

Theorem 1. Let

\[
f(r, u, b) = \frac{1}{r} \left( 1 + \frac{\max(2u - b - 2, 0)}{r + 2(r + 1)} \right)^{-1}.
\]

Then the vector \( z_y = f(r_y, u_y, b_y) \) is feasible for \( \kappa^*(A_n) \), so \( \kappa^*(A_n) \leq 1^T z \).

Proof: By Lemma 1, \( \varphi \circ \varphi(1) \) is feasible for \( \kappa^*(A_n) \). From the definition of \( \varphi \),

\[
\varphi(x) = \min_{y \in N(y)} (A_n x)_y
\]

Each \( x \in [2]^n \) has \( r_x \) total substrings, so \( (A_n z')_x = r_x \),

\[
\frac{1}{\varphi(1)_y} = \min_{x \in N(y)} (A_n 1)_x = \min_{x \in N(y)} r_x = r_y,
\]

and \( \varphi(1)_y = 1/r_y \).

Of the substrings of \( x \), \( u_x \) have \( r_x - 2 \) runs, \( b_x \) have \( r_x - 1 \) runs, and \( r_x - u_x \) have \( r_x \) runs, so

\[
(A_n \varphi(1))_x = \sum_{y \in N(x)} \frac{1}{y^r} = \frac{u_x - b_x}{r_x - 2} + \frac{b_x}{r_x - 1} + \frac{r_x - u_x}{r_x}
\]

\[
= 1 + u_x \left( \frac{1}{r_x - 2} - \frac{1}{r_x} \right) + b_x \left( \frac{1}{r_x - 1} - \frac{1}{r_x} \right)
\]

\[
= 1 + \frac{2u_x (r_x - 1) - b_x r_x}{r_x (r_x - 1)}
\]

\[
= 1 + \frac{2u_x - b_x}{r_x (r_x - 1)}.
\]

The inequality follows from \( u_x - b_x \geq 0 \).

Let \( y \in [2]^{n-1} \) be a string and let \( x \in [2]^n \) be a superstring of \( y \). It is possible to create a superstring by extending an existing run, adding a new run at the end of the string, or by splitting an existing run into three new runs, so \( r_x \leq r_y + 2 \) The only way to destroy a unit run in \( y \) is to extend it into a run of length two, so \( u_x \geq u_y - 1 \). Similarly, \( u_x - b_x \geq u_y - b_y - 1 \), so \( 2u_x - b_x \geq 2u_y - b_y - 2 \). Applying these inequalities to \( (A_n \varphi(1))_y \), we conclude that

\[
\frac{\varphi(1)_y}{(\varphi \circ \varphi(1))_y} = \min_{x \in N(y)} (A_n \varphi(1))_x
\]

\[
\geq 1 + \frac{\max(2u - b - 2, 0)}{(r_y + 2)(r_y + 1)},
\]

\[
(\varphi \circ \varphi(1))_y \leq \frac{1}{r_y} \left( 1 + \frac{\max(2u - b - 2, 0)}{(r_y + 2)(r_y + 1)} \right)^{-1}.
\]

Lemma 7. The number of strings in \([2]^n\) with \( r \) runs is \( 2^{n-1 \choose r-1} \).

The number of strings in \([2]^n\) with \( r \) runs and \( u \) unit runs is \( 2^{n-1 \choose r-1} \binom{r-2}{b} \).

Proof: For \( k \geq 1 \), there are \( \binom{n+k-1}{n} \) ways to partition \( n \) identical items into \( k \) distinguished groups. Thus there are \( \binom{n+k-1}{n} \binom{n-k}{u} \) ways to partition \( n \) items into \( k \) groups such that each group contains at least \( u \) items.

A binary string is uniquely specified by its first symbol and its run length sequence. We have \( n \) symbols to distribute among \( r \) runs such that each run contains at least one symbol, so there are \( {n-1 \choose r-1} \) arrangements. This proves the first claim. We can also specify the run sequence of a string by giving the locations of the unit runs and the lengths of the longer runs. The \( r - u \) runs of length at least two can appear in \( r \) positions so there are \( {r \choose r-u} \) arrangements. We have \( n - u \) symbols to distribute among \( r - u \) runs such that each run contains at least 2 symbols, so there are \( \binom{n-u-1}{r-u-2} \) arrangements, which proves the second claim. As long as \( r \geq 2 \), the internal unit runs can appear in \( r - 2 \) positions and the external unit runs can appear in 2 positions, so there are \( {r-2 \choose b} \) possible arrangements, which proves the third claim.

Note that Lemma 7 uses the polynomial definition of binomial coefficients, which can be nonzero even when the top entry is negative. For example, the number of strings of length \( n \) with \( n \) runs, \( n \) unit runs, and \( 2 \) external unit runs is \( \binom{n-2}{2} \).

For compactness, let

\[
E_r[f(r)] = \frac{1}{2^{n-1}} \sum_{r \geq 1} \binom{n-1}{n-r} f(r).
\]

and let \( \mathbb{E}_{u,b} [f(r, u, b)] \) equal

\[
\frac{1}{\binom{n-1}{n-r}} \sum_{u=0}^{n-r} \sum_{b=0}^{n-u} \binom{n-r-1}{n-2r+u} \binom{n-u-b}{b} f(r, u, b)
\]
for \( r > 1 \) and let \( \mathbb{E}_{u,b}[f(1,u,b)] = f(1,0,0) \).

If \( z_x = f(r_x,u_x,b_x) \), then

\[
1^T z = \sum_{x \in [2]^n} f(r_x,u_x,b_x)
= 2f(1,0,0) + \sum_{r=2}^{n} \sum_{u=0}^{r} \sum_{b=0}^{u} \binom{n-r-1}{n-2r+u} \binom{r-2}{u-b} (2) f(r,u,b)
= 2^n \mathbb{E}_r \mathbb{E}_{u,b}[f(r,u,b)]
\]

(3)  

Analysis of the feasible point constructed in Theorem 1 relies on the following identities. For \( k \geq 0 \),

\[
\mathbb{E}_r \left[ \frac{1}{r^{k+1}} \right] = \sum_{r \geq 1} \frac{(n+k-1)}{2(n-r)\binom{n+k-1}{n-1}}
\leq \frac{2^k}{(n+k-1)\binom{n+k-1}{n-1}}
\]

(4)

\[
\mathbb{E}_{u,b} \left[ \binom{u}{u-k} \right] = \frac{\binom{r}{r-k} \binom{r-1}{r-k-1}}{\binom{n-1}{n-k-1}}
\]

(5)

\[
\mathbb{E}_{u,b}[b] = \frac{2(r-1)}{n-1}.
\]

(6)

Each of these can be easily derived from the binomial theorem and Vandermonde’s identity.

**Theorem 2.** For \( n \geq 2 \),

\[
\kappa^*(A_n) \leq \frac{2^n}{n+1} \left( 1 + \frac{26}{n(n-1)} \right).
\]

**Proof:** For \( n \leq 13 \), this follows from the bound of Kulkarni and Kiyavash [2]. This proof covers \( n \geq 10 \).

From Theorem 1 and (3), we have \( \kappa^*(A_{n+1}) \leq 2^n \mathbb{E}_r \mathbb{E}_{u,b}[f(r,u,b)] \) where

\[
f(r,u,b) = \frac{1}{r} \left( 1 + \frac{\max(2u-b-2,0)}{(r+2)(r+1)} \right)^{-1}.
\]

For \( x > 0 \), \((1+x)^{-1} \leq 1 - x + x^2\), so \( f(r,u,b) \) is at most

\[
\frac{1}{r} \left( 1 - \frac{\max(2u-b-2,0)}{(r+2)(r+1)} + \frac{(2u-b-2)^2}{(r+2)^2(r+1)^2} \right)
\leq \frac{1}{r} \left( 1 - \frac{2u-b-2}{(r+2)(r+1)r} + \frac{2u(2u-2)}{(r+2)^2(r+1)^2r} \right)
\]

We will bound this term by term using (4), (5), and (6). First

\[
\mathbb{E}_r \left[ \mathbb{E}_{u,b} \left[ \frac{1}{r} - \frac{2u-b-2}{(r+2)(r+1)r} \right] \right]
= \mathbb{E}_r \left[ \frac{1}{r} \right] - \frac{1}{(r+2)(r+1)r} \left( \frac{2(r-1)}{n-1} - \frac{2(r-1)}{n-1} - \frac{2}{2} \right)
\]

\[
= \mathbb{E}_r \left[ \frac{1}{r} - \frac{2}{(r-1)^2 - n + 1} \right]
\]

\[
= 2 \mathbb{E}_{u,b} \left[ \frac{n-1}{2r} - \frac{1}{r} + \frac{5}{(r+1)r} + \frac{2(r-1)(r+1)}{2r} \right]
\]

\[
\leq 2\mathbb{E}_{u,b} \left[ \frac{n-3}{2r} + \frac{20}{(r+1)r} + \frac{8n-80}{(n+1)(n+1)n} \right]
\]

\[
= 2\mathbb{E}_{u,b} \left[ \frac{n^2 - n - 6}{(n+1)n} + \frac{28n-40}{(n+2)(n+1)n} \right]
\]

\[
= 2\mathbb{E}_{u,b} \left[ \frac{n^2 - n - 6}{(n+1)n(n-1)} + \frac{28n-40}{(n+1)n(n-1)} \right]
\]

\[
\leq 2\mathbb{E}_{u,b} \left[ \frac{1}{(n+1)n(n-1)} + \frac{22}{(n+1)n} \right]
\]

Second,

\[
\mathbb{E}_r \left[ \mathbb{E}_{u,b} \left[ \frac{4u(u-1)}{(r+2)^2(r+1)^2r} \right] \right]
\]

\[
= \mathbb{E}_r \left[ \frac{4}{(r+2)^2(r+1)^2r} \right] \left( \frac{r(r-1)^2(r-2)}{(n-1)(n+1)r} \right)
\]

\[
\leq \mathbb{E}_r \left[ \frac{4}{(r+2)(r+1)^2} \right] \left( \frac{1}{(n-1)(n+1)r} \right)
\]

\[
\leq \mathbb{E}_r \left[ \frac{4}{(r+2)(r+1)^2} \right] \left( \frac{1}{(n+2)(n+1)n} \right)
\]

\[
\leq \mathbb{E}_r \left[ \frac{1}{8(n+2)(n+1)n} \right]
\]

Combining these two terms, we get

\[
\kappa^*(A_{n+1}) \leq 2^n \frac{2}{n+2} \left( 1 + \frac{22}{(n+1)n} + \frac{4}{(n+1)n} \right).
\]

Lemma 2. Let \( z_y = f'(r_y,u_y,b_y) \). Then

\[
1^T z \geq \frac{n}{n+1} \left( 1 + \frac{1}{n-1} - \frac{3}{(n-1)(n+2)} \right)
\]

**Proof:**

\[
f'(r,u,b) \geq \frac{1}{r} \left( 1 - \frac{u-b}{r^2} \right)
\]

\[
\geq \frac{1}{r} \left( 1 - \frac{u-b}{(r-1)(r+2)} \right)
\]
Theorem 3. Let $A_n$ be the $n$-bit 1-grain-error channel. The vector

$$z_y = \frac{1}{r_y} \left( 1 + \frac{2u_y - 2b^L_y - b^R_y - 2}{(r_y + 2)(r_y + 1)} \right)^{-1}$$

is feasible for $\kappa^*(A_n)$.

Proof: By Lemma 1, $\varphi \circ \varphi(1)$ is feasible for $\kappa^*(A)$. From the definition of $\varphi$,

$$\frac{z_y}{\varphi(z)_y} = \min_{x \in \mathcal{N}(y)} (A_n z)_x$$

Each $x \in [2^n]$ has $r_x$ total neighbors, so $(A_n z)_x = r_x$,

$$\frac{1}{\varphi(1)_y} = \min_{x \in \mathcal{N}(y)} (A1)_x = \min_{x \in \mathcal{N}(y)} r_x = r_y,$$

and $\varphi(1)_y = 1/r_y$.

Of the neighbors of $x$, $u_x - b^L_x - b^R_x$ have $r_x - 2$ runs, $b^L_x$ have $r_x - 1$ runs, and $r_x - u_x + b^R_x$ have $r_x$ runs, so $(A_n \varphi(1))_x$ equals

$$\sum_{y \in \mathcal{N}(x)} \frac{1}{r_y} = \frac{u_x - b^L_x - b^R_x}{r_x - 2} + \frac{b^L_x}{r_x - 1} + \frac{r_x - u_x + b^R_x}{r_x}$$

$$= 1 + \frac{2(u_x - b^L_x)(r_x - 1) - b^L_x r_x}{r_x(r_x - 1)(r_x - 2)}$$

$$\geq 1 + \frac{2u_x - 2b^L_x - b^R_x}{r_x(r_x - 1)}.$$ 

Let $x \in [2^n]$ be an input and let $y \in \mathcal{N}(x)$. A grain error can leave the number of runs unchanged, destroy a unit run at the start of $x$, or destroy a unit run in the middle of $x$, merging the adjacent runs. Thus $r_y \geq r_x - 2$ The only way to produce a unit run in $y$ is shorten a run of length two in $x$, so $u_x \geq u_y - 1$. Similarly, $2u_x - 2b^L_x - b^R_x \geq 2u_y - 2b^L_y - b^R_y - 2$. Applying these inequalities to $(A \varphi(1))_x$, we conclude that

$$\frac{\varphi(1)_y}{(\varphi \circ \varphi(1))_y} = \min_{x \in \mathcal{N}(y)} (A \varphi(1))_x + 1 \geq 1 + 2u_y - 2b^L_y - b^R_y - 2 \geq 2u_y - 2b^L_y - b^R_y - 2.$$ 

REFERENCES


Negar Kiyavash (SM’13) is a Willett Faculty Scholar and an Associate Professor, jointly, in the Departments of Industrial and Enterprise Systems Engineering and Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. She received the B.S. degree in electrical and computer engineering from the Sharif University of Technology, Tehran, in 1999, and the M.S. and Ph.D. degrees, also in electrical and computer engineering, both from the University of Illinois at Urbana-Champaign in 2003 and 2006, respectively. Her research interests include statistical signal processing, information theory, coding theory, operations research, and their application to learning and inference in complex networks. She is a recipient of the University of Illinois College of Engineering’s Grainger Award in Emerging Technologies, AFOSR Young Investigator Award, NSF Career Award, and Deans Award for Excellence in Research at the University of Illinois at Urbana-Champaign.