Asymmetric Lee Distance Codes for DNA-Based Storage

Ryan Gabrys*†, Han Mao Kiah‡, and Olgica Milenkovic§
* § Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, USA
†Spawar Systems Center San Diego, Code 532, USA
‡ School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
Emails: *gabrys@illinois.edu, † ryan.gabrys@navy.mil, ‡hmkiah@ntu.edu.sg, §milenkov@illinois.edu

***Abstract*—We consider a new family of asymmetric Lee codes that arise in the design and implementation of DNA-based storage systems and systems with parallel string transmission protocols. The codewords are defined over a quaternary alphabet, although the results carry over to other alphabet sizes, and have symbol distances dictated by their underlying binary representation. Our contributions are two-fold. First, we derive upper bounds on the size of the codes under the asymmetric Lee distance measure based on linear programming techniques. Second, we propose code constructions which imply lower bounds.**

Keywords. Coding for DNA-based storage, Coding theory

I. Introduction

Codes for classical channels with single-sequence inputs and single-sequence outputs have been studied extensively, leading to a diverse suite of solutions including algebraic codes [15], codes on graphs, such as LDPC codes [14], and polar codes [16]. Similar advances have been reported for parallel channels [7], with the rather common underlying assumption that the channels introduce uncorrelated errors. The alphabet size of the codes used in both scenarios is usually restricted by the system design, and often, input sequences are de-interleaved or represented as arrays over smaller alphabets in order to enable more efficient transmission. Far less is known about channels that operate on several sequences at the same time and introduce correlated symbol errors, or output ordering errors. The goal of this work is to analyze one such scenario, motivated by emerging applications in DNA-based storage systems.

To motivate our analysis, consider a transmission model in which two binary input sequences are simultaneously passed through *two channels* that introduce substitution errors with some probability $0 < p < 1/2$ (Figure 1). Simultaneous errors in both strings are less likely than individual string errors. In addition to the substitution errors, the outputs of the channels may be switched – in other words, the label of the channel from which the output symbol originated may be in error. The confusion graph of this type of channel is depicted in Figure 2. In Figure 2, the vertices are indexed by pairs of bits denoting the inputs into the two channels. The labels of the edges denote the channel confusion parameters (weights, distances).

One application of such a model arises in DNA sequencing for archival storage [6], where multiple sequences are read in parallel. The readout errors are rare and it is very uncommon to make simultaneous mistakes in both sequences; nevertheless, the identity of the strands may be confused due to string sorting issues. Another unrelated way to view this model is to assume that the binary encodings represent four symbols of the DNA alphabet $\{A, T, G, C\}$, say $00 \to G$, $11 \to C$, $01 \to A$ and $10 \to T$. In this case, the proposed metric illustrates a DNA readout channel in which the bases $\{A, T\}$ are very likely to be mutually confused during sequencing, while the basis $\{G, C\}$ are much less likely to be misinterpreted for each other. Illumina systems and some other sequencing devices have substitution errors that exhibit such "bias" phenomena, and similar effects may be expected for single base sequencing technologies of the third generation [13].

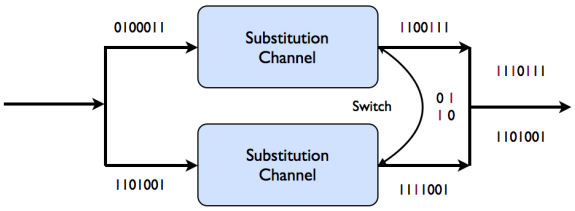

Fig. 1. A pair of channels with individual substitution errors, for which the outputs may also be switched. For the given example, the outputs of the channels at position three are switched.

The problem of interest is to design pairs of sequences – henceforth, termed codewords – such that any two codewords are at a sufficiently large "distance" from each other. We subsequently refer to the distance imposed by Figure 2 as the *asymmetric Lee distance* (ALD). The ALD resembles a weighted version of the Lee metric [2], with the exception of *two symbols* being treated differently. These two symbols capture the uncertainty about the actual ordering of the readouts. Given the connection with the Lee metric, a partial analysis of the ALD and related code construction questions may be addressed by invoking results for codes in the Lee metric [1], [2]. Nevertheless, due to the asymmetry of the distance, specialized techniques need to be developed to find bounds on the code size and to construct codes that approach these bounds. To accomplish this task, we formally define the ALD as a judiciously chosen combination of the Lee and the Hamming distance.

The paper is organized as follows. In Section II we introduce the ALD, and then proceed to derive upper and lower bounds on the size of the largest code in this metric using results on codes in the Lee metric. In Section III, we turn our attention on computing tighter upper bounds on the size of codes by applying generalized sphere packing techniques not known from the Lee coding literature. New code constructions and related questions are discussed in Section IV.

II. Problem Formulation

We start by characterizing the channel errors by introducing a new distance measure, which we refer to as the *asymmetric*

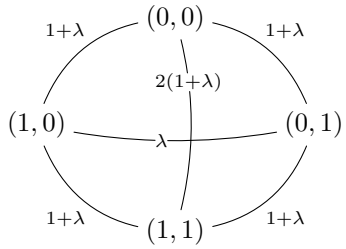

Fig. 2. Weighted confusion graph for codelength $n = 1$.

Lee distance (ALD).

We refer to an error that causes a paired-symbol transition from $(1, 0)$ to $(0, 1)$ as a Class 1 (switch) error; similarly, we refer to an error that causes a single substitution in one of the input strings as a Class 2 error. An error that causes a paired-symbol transition from $(0, 0)$ to $(1, 1)$ is referred to as a Class 3 (simultaneous substitution) error. Note that based on Figure 2, an edge in the confusion graph corresponding to a Class 1 error has weight λ, an edge for a Class 2 error has weight $1+\lambda$, while an edge for a Class 3 error has weight $2(1 + \lambda)$.

Let $m \geqslant 2$ be an integer. For $a_1, a_2, \ldots, a_m \in \mathbb{Z}_2$, the indicator function χ is such that $\chi(a_1, a_2, \ldots, a_m) = 1$, if $a_1 = a_2 = \cdots = a_m$ and $\chi(a_1, a_2, \ldots, a_m) = 0$ otherwise. Consider next four sequences $\boldsymbol{a} = (a_1, \ldots, a_n), \boldsymbol{b} = (b_1, \ldots, b_n), \boldsymbol{c} = (c_1, \ldots, c_n), \boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{Z}_2^n$, paired as $(\boldsymbol{a}, \boldsymbol{b})$ and $(\boldsymbol{c}, \boldsymbol{d})$, and define two pseudo-metrics:

$$d_S((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d})) := \sum_{i=1}^{n} \chi(a_i, b_i) + \chi(c_i, d_i) - 2\chi(a_i, b_i, c_i, d_i),$$

$$d_D((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d})) := \sum_{i=1}^{n} 2(\chi(a_i, b_i) + \chi(c_i, d_i)) + \chi(a_i, \bar{b}_i, \bar{c}_i, d_i) - 4\chi(a_i, b_i, c_i, d_i),$$

where $\bar{x} = 1 - x$, and $x \in \mathbb{Z}_2$. Note that the pseudo-metrics $d_S((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d}))$ and $d_D((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d}))$ both depend on how much the sequences within the pairing $(a_i, b_i), (c_i, d_i)$, $i \in [n]$, disagree from each other, as well as how much the pairs differ themselves. Furthermore, observe that d_S is invariant under the change of order in the pairing, i.e., $d_S((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d})) = d_S((\boldsymbol{b}, \boldsymbol{a}); (\boldsymbol{c}, \boldsymbol{d}))$, and that it takes the maximum value $2n$ if and only if the sequences $\boldsymbol{a}, \boldsymbol{b}$ are identical and complements of $\boldsymbol{c}, \boldsymbol{d}$. Hence, this pseudo-metric does not penalize reorderings of symbols in the same pair of strings. On the other hand, it is easy to see that d_D is actually a metric which assigns non-zero distances to pairs in which the bits are switched. Consequently, combining d_S and d_D in the coding process should allow one to control both the effects of symbol switching events between the channels and the substitution errors in the channels.

For a positive real number λ, define the ALD $d_\lambda((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d}))$ as a combination of d_S and d_D,

$$\begin{aligned} &\lambda d_D((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d})) + (1 - \lambda) d_S((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d})) \qquad (1) \\ =&\sum_{i=1}^{n} (1 + \lambda)\left(\chi(a_i, b_i) + \chi(c_i, d_i)\right) + \lambda\chi(a_i, \bar{b}_i, \bar{c}_i, d_i) \\ &-2(1 + \lambda)\chi(a_i, b_i, c_i, d_i). \end{aligned}$$

It can be shown that $d_\lambda((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d}))$ is a metric. Hence, for any $\lambda > 0$, d_λ is non-negative, symmetric and satisfies the triangle inequality. We henceforth focus our attention on integer λ.

From (1), we observe that the paired symbol distance is asymmetric, in so far that complementary pairs are treated differently than non-complementary pairs. Furthermore, when pairs are complementary, the distance depends on the binary weight of the pairs. The choice of the parameter λ governs the degree of the asymmetry in the distance.

In order to highlight the relationship between $d_\lambda((\boldsymbol{a}, \boldsymbol{b}); (\boldsymbol{c}, \boldsymbol{d}))$ and the Lee distance, define the mapping $\mathcal{Z} : \mathbb{Z}_2^2 \to \mathbb{Z}_4$ so that $00 \to 1$, $10 \to 0$, $01 \to 2$ and $11 \to 3$. Then, by changing the weight between (10) and (01) – from λ to $2(1 + \lambda)$, we arrive at a scaled version of the Lee distance $d_L(x, y)$ between symbols $x, y \in \mathbb{Z}_4$, which reads as

$$(1 + \lambda) \cdot \min\{4 - |x - y|, |x - y|\} = (1 + \lambda) \cdot d_L(x, y). \qquad (2)$$

More precisely, for two sequences $\boldsymbol{x}, \boldsymbol{y}$ over $\mathbb{Z}_4$, we have

$$\begin{aligned} d_\lambda(\boldsymbol{x}, \boldsymbol{y}) =&(1 + \lambda) \cdot d_L(\boldsymbol{x}, \boldsymbol{y}) - \qquad (3) \\ &(\lambda + 2) \sum_{\{x_i, y_i\} = \{0,2\}} \chi(x_i, y_i) \leqslant (1 + \lambda) \cdot d_L(\boldsymbol{x}, \boldsymbol{y}). \end{aligned}$$

The main difference between the ALD and the Lee metric is that the distance between pairs of symbols, say $(s, p), (q, r) \in \mathbb{Z}_2^2$ is not completely characterized by the weight of the vector $(s, p) + (q, r)$, and that in particular, it depends on the exact values of (s, p) in a manner analogous to asymmetric error correcting codes. This connection between the ALD, Lee metric, and asymmetric error correcting codes will be used in our subsequent derivations.

Let d be a real positive integer. We say that two pairs of sequences $(\boldsymbol{a}, \boldsymbol{b}), (\boldsymbol{c}, \boldsymbol{d}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$ are *(d,λ)-distinguishable* if their ALD d_λ is at least d; similarly, we say that two pairs of sequences are *(d,λ)-indistinguishable* if their ALD is less than d. We use $A_\lambda(n, d)$ to denote the largest number of (d, λ)-distinguishable sequences of length n. In the next section, we derive a number of upper bounds on $A_\lambda(n, d)$.

III. Upper Bounds

We start by computing $A_\lambda(n, d)$ using bounds on the size of codes in the Lee metric. Let $\mathcal{Z} : \mathbb{Z}_2^2 \to \mathbb{Z}_4$ denote the mapping introduced in the previous section. Furthermore, let $A_L(n, d, q)$ denote the size of the largest code of length n with minimum Lee distance d over $\mathbb{Z}_q$. Since $\mathcal{Z}$ is invertible, we have the following result which easily follows from (3).

Proposition 1. *For positive integers n and d, and positive integer λ, $A_\lambda(n, d) \leqslant A_L(n, \lfloor \frac{d}{1+\lambda} \rfloor, 4)$.*

A recent Singleton-type bound for codes in the Lee metric and over an even-sized alphabet [1] asserts that

$$A_\lambda(n, d) \leqslant A_L(n, \lfloor \frac{d}{1 + \lambda} \rfloor, 4) \leqslant 2^{2n - \lfloor \frac{d}{1+\lambda} \rfloor + 1}. \qquad (4)$$

Note that the bound is general, as it applies to codes that are not necessarily linear.

A. Sphere Packing Bounds on $A_\lambda(n,d)$ for large n

In what follows, we first derive an expression for the size of a ball of radius r in the ALD. Using this expression, we proceed to find asymptotic upper bounds on $A_\lambda(n,2r+1)$.

Given a vector $\boldsymbol{x} \in \mathbb{Z}_2^n$, let $wt(\boldsymbol{x})$ denote the Hamming weight of $\boldsymbol{x}$. For a pair of sequences $(\boldsymbol{a},\boldsymbol{b}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$, let

$$\mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b}) := \{(\boldsymbol{c},\boldsymbol{d}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n : d_\lambda((\boldsymbol{a},\boldsymbol{b});(\boldsymbol{c},\boldsymbol{d})) \leqslant r\}$$

denote the set of pairs of sequences that are $(r+1,\lambda)$-indistinguishable from $(\boldsymbol{a},\boldsymbol{b})$. Note that from (1), the quantity $|\mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b})|$ is a function of the Hamming weight of $\boldsymbol{a}-\boldsymbol{b}$, which we for simplicity denote by $w = w(\boldsymbol{a},\boldsymbol{b})$. Let

$$S_\lambda(n,w,\delta) := |\{(\boldsymbol{c},\boldsymbol{d}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n : d_\lambda((\boldsymbol{a},\boldsymbol{b});(\boldsymbol{c},\boldsymbol{d})) = \delta\}|,$$

and

$$V_\lambda(n,w,r) := |\mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b})|,$$

where we omitted the arguments $\boldsymbol{a},\boldsymbol{b}$ on the left-hand side of the equations for simplicity of notation. Clearly, $V_\lambda(n,w,r) = \sum_{j=0}^{r} S_\lambda(n,w,j)$.

Using the same techniques as described in [2], we compute the generating function of $S_\lambda(n,w,\delta)$,

$$\sum_{w=0}^{\infty}\sum_{r=0}^{\infty} S_\lambda(n,w,r)x^w z^r = x^w\left(z^\lambda + (1+2z^{1+\lambda})\right)^w \cdot \left(z^{2(1+\lambda)} + (1+2z^{1+\lambda})\right)^{n-w}. \quad (5)$$

From (5), it follows that $S_\lambda(n,w,r)$ equals the coefficient of $x^w z^r$. Since $V_\lambda(n,w,r) = \sum_{i=0}^{r} S_\lambda(n,w,i)$, the next lemma follows after some straightforward algebraic manipulations. For simplicity of notation, we use $k*\ell*m \leqslant (r,\lambda)$ to denote $(2k+\ell)(1+\lambda)+\lambda m \leqslant r$.

Lemma 2. *For integers n,w,r, $V_\lambda(n,w,r)$ may be written as*

$$\sum_{k*\ell*m\leqslant(r,\lambda)} \binom{w}{m}\binom{n-w}{k}\binom{n-k-m}{\ell} 2^\ell.$$

Corollary 3. *For positive integers n,w, and r, it holds that $V_\lambda(n,w,r) \geqslant V_\lambda(n,w-1,r)$.*

Using Lemma 2, one can derive the following asymptotic upper bound on the size of an ALD, following similar arguments as those used in [11, Theorem 3].

Lemma 4. *There exists an N such that for $n \geqslant N$, we have*

$$A_\lambda(n,2r+1) \leqslant \frac{4^n}{V_\lambda(n,\lfloor n/2 - \sqrt{5nr\log_2(n)}\rfloor,r)} \cdot (1+o(1)).$$

From Lemma 4, one recovers the following two special upper bounds.

Corollary 5. *For $n \geqslant N$ and $L(n) = \lfloor n/2 - \sqrt{10n\log_2(n)}\rfloor$,*

$$A_1(n,5) \leqslant \frac{4^n}{1+2n+\sum_{j=1}^{2}\binom{L(n)}{j}}.$$

Corollary 6. *For $n \geqslant N$ and $L(n) = \lfloor n/2 - \sqrt{15n\log_2(n)}\rfloor$,*

$$A_1(n,7) \leqslant \frac{4^n}{1+\sum_{j=1}^{3}\binom{L(n)}{j}+2L(n)(n-1)+2n}.$$

B. Non-Asymptotic Upper Bounds

We start by introducing our notation, following [4]. Fix r, λ, $\boldsymbol{x}$ and recall the definition of $\mathcal{B}_{(r,\lambda)}(\boldsymbol{x})$. Define a directed graph $\mathcal{G}_{(r,\lambda)}$ on the vertex set $\mathbb{Z}_2^n \times \mathbb{Z}_2^n$ such that there exists an arc from $\boldsymbol{x}$ to $\boldsymbol{y}$ if $\boldsymbol{y} \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{x})$. Let $\mathbf{A}_{(r,\lambda)}$ be an adjacency matrix of dimension $4^n \times 4^n$, indexed by the elements in $\mathbb{Z}_2^n \times \mathbb{Z}_2^n$. Then $\mathbf{A}_{(r,\lambda)}(i,j) = 1$ if $\boldsymbol{x}_i \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{x}_j)$, and zero otherwise.

Let $\mathbb{R}_+$ denote the set of non-negative reals, and define

$$\tau^*\left(\mathbf{A}_{(r,\lambda)}\right) = \min\left\{\sum_{i=1}^{4^n} w_i : \boldsymbol{w} \in \mathbb{R}_+^{4^n},\ \mathbf{A}_{(r,\lambda)}^T \cdot \boldsymbol{w} \geqslant \mathbf{1}\right\}. \quad (6)$$

The results of [10] show that one may use $\tau^*\left(\mathbf{A}_{(r,\lambda)}\right)$ as an upper bound for $A_\lambda(n,2r+1)$. However, (6) is a linear program involving 4^n equations and we may significantly reduce this number by observing certain symmetries of $\mathcal{G}_{(r,\lambda)}$.

An automorphism of $\mathcal{G}_{(r,\lambda)}$ is a permutation of its vertices that preserves adjacency. Let $\mathbb{S}_n$ be the symmetric group on n symbols. The set of all automorphisms of $\mathcal{G}_{(r,\lambda)}$ is defined as $\mathrm{Aut}(\mathcal{G}_{(r,\lambda)}) = \{\pi \in \mathbb{S}_n | \pi \text{ is an automorphism of } G\}$. Given a subgroup H of $\mathrm{Aut}(\mathcal{G}_{(r,\lambda)})$, let H partition the vertex set into n_H equivalence classes $\{X_1,\ldots,X_{n_H}\}$. Let $\mathbf{A}_{H,(r,\lambda)}$ be an $n_H \times n_H$ adjacency matrix corresponding to the subgroup H, such that for $1 \leqslant i,j \leqslant n_H$,

$$\mathbf{A}_{H,(r,\lambda)}(i,j) = \frac{|\{(\boldsymbol{x},\boldsymbol{y}) : \boldsymbol{x} \in X_j, \boldsymbol{y} \in X_i, \boldsymbol{y} \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{x})\}|}{|X_i|}.$$

The authors of [4] demonstrated that solving (6) is equivalent to solving a linear program involving only $\mathbf{A}_{H,(r,\lambda)}$. Combining this result with those of [10], we have the following.

Theorem 7. *(c.f. [4], [10]) Let H be a subgroup of $\mathrm{Aut}(\mathcal{G}_{(r,\lambda)})$ and define $\mathbf{A}_{H,(r,\lambda)}$ as above. Then,*

$$\tau^*(\mathbf{A}_{(r,\lambda)}) = \min\left\{\sum_{i=1}^{n_H} |X_i|\cdot w_i : \boldsymbol{w} \in \mathbb{R}_+^{n_H}, \mathbf{A}_{H,(r,\lambda)}^T \cdot \boldsymbol{w} \geqslant 1\right\}.$$

and $A_\lambda(n,2r+1) \leqslant \tau^*(\mathbf{A}_{(r,\lambda)})$.

Next, we define a set of automorphisms on $\mathcal{G}_{(r,\lambda)}$ as follows. We first introduce a mapping denoted $\pi_{\sigma,\boldsymbol{x}}$. For every permutation σ in the symmetric group $\mathbb{S}_n$ and any $\boldsymbol{x} \in \mathbb{Z}_2^n$, let $\pi_{\sigma,\boldsymbol{x}} : \mathbb{Z}_2^n \times \mathbb{Z}_2^n \to \mathbb{Z}_2^n \times \mathbb{Z}_2^n$ be a mapping such that for all $(\boldsymbol{a},\boldsymbol{b}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$, $(\pi_{\sigma,\boldsymbol{x}}(\boldsymbol{a},\boldsymbol{b}))_i = (\boldsymbol{a}_{\sigma(i)},\boldsymbol{b}_{\sigma(i)})$ if $x_i = 0$ and $(\pi_{\sigma,\boldsymbol{x}}(\boldsymbol{a},\boldsymbol{b}))_i = (\boldsymbol{b}_{\sigma(i)},\boldsymbol{a}_{\sigma(i)})$ otherwise. For instance, if $(\boldsymbol{a},\boldsymbol{b}) = ((0,0,1),(1,0,1))$, $\boldsymbol{x} = (0,1,0)$, and $\sigma = (3,1,2)$, then $\pi_{\sigma,\boldsymbol{x}}(\boldsymbol{a},\boldsymbol{b}) = ((1,1,0),(1,0,0))$. It can be shown that the set $H = \{\pi_{\sigma,\boldsymbol{x}} : \sigma \in \mathbb{S}_n, \boldsymbol{x} \in \mathbb{Z}_2^n\}$ is a subgroup of $\mathrm{Aut}(\mathcal{G}_{(r,\lambda)})$. As a result of the previous discussion, a bound on $A_\lambda(n,2r+1)$ may be obtained by considering the quantity

$$\tau^*(\mathbf{A}_{(r,\lambda)}) = \min\left\{2^n \cdot \sum_{\ell=0}^{n}\binom{n}{\ell}\cdot w_\ell : \boldsymbol{w} \in \mathbb{R}_+^{n+1}, \mathbf{A}_{H,(r,\lambda)}^T \cdot \boldsymbol{w} \geqslant 1\right\}. \quad (7)$$

Note that using the automorphism group, the dimension of the weight vector $\boldsymbol{w}$ has been reduced from $O(4^n)$ (as in (6)) to

$n+1$ in (7). The next theorem provides a feasible weight vector for the optimization problem in (7).

Theorem 8. *Let $V_\lambda(n,w,r) = V_\lambda(n,0,r)$ for $w < 0$. Suppose that $n,r,\lambda > 0$ and that $\mu = \lfloor \frac{r}{1+\lambda} \rfloor$. Let $\boldsymbol{w} = (w_0, w_1, \ldots, w_n) \in \mathbb{R}_+^{n+1}$ be defined as $w_i = \frac{1}{V_\lambda(n,i-\mu,r)}$. Then,*

$$A_\lambda(n,2r+1) \leqslant 2^n \cdot \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} \cdot \frac{1}{V_\lambda(n,i-\mu,r)}.$$

Proof: The result follows directly from Theorem 7 and (7), provided that we can show that $\mathbf{A}^T_{H,(r,\lambda)} \cdot \boldsymbol{w} \geqslant 1$. For the given weight assignment, and for an arbitrary choice of $(\boldsymbol{a},\boldsymbol{b}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$, we have

$$\begin{aligned}\sum_{(\boldsymbol{c},\boldsymbol{d}) \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b})} w_{w(\boldsymbol{c},\boldsymbol{d})} &= \sum_{(\boldsymbol{c},\boldsymbol{d}) \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b})} \frac{1}{V_\lambda(n, w(\boldsymbol{c},\boldsymbol{d}) - \mu, r)} \\ &\geqslant \sum_{(\boldsymbol{c},\boldsymbol{d}) \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b})} \frac{1}{V_\lambda(n, w(\boldsymbol{a},\boldsymbol{b}), r)} \\ &= \frac{V_\lambda(n, w(\boldsymbol{a},\boldsymbol{b}), r)}{V_\lambda(n, w(\boldsymbol{a},\boldsymbol{b}), r)} = 1.\end{aligned}$$

Since $(\boldsymbol{c},\boldsymbol{d}) \in \mathcal{B}_{(r,\lambda)}(\boldsymbol{a},\boldsymbol{b})$, $w(\boldsymbol{c},\boldsymbol{d}) \leqslant w(\boldsymbol{a},\boldsymbol{b}) + \mu$ and now the inequality follows from Corollary 3. ∎

Throughout the remainder of this section, we consider more sophisticated weight assignments that in many cases improve the upper bound of Theorem 8. Notice first that if $r = \lambda$, then the choice of weights $\boldsymbol{w}$ from the previous theorem produces the best possible upper bound on $A_\lambda(n,3)$ achievable via Theorem 7. This can be seen by noting when $r = \lambda$, one may write

$$\begin{aligned}\tau^*(\mathbf{A}_{(\lambda,\lambda)}) = \min \Bigg\{ & 2^n \cdot \sum_{\ell=0}^{n} \begin{pmatrix} n \\ \ell \end{pmatrix} \cdot w_\ell : \\ & w_\ell \cdot (\ell+1) \geqslant 1, 0 \leqslant \ell \leqslant n, w_\ell \geqslant 0 \Bigg\}.\end{aligned}$$

Proposition 9 *Suppose that $r = \lambda$. Then*

$$\tau^*(\mathbf{A}_{(r,\lambda)}) = 2^n \cdot \sum_{\ell=0}^{n} \frac{\binom{n}{\ell}}{\ell+1} = \frac{2^n(2^{n+1}-1)}{n+1}.$$

Hence, $A_\lambda(n,2\lambda+1) \leqslant 2^n(2^{n+1}-1)/(n+1)$.

Proposition 10 *For integers r,λ, where $\lambda | r$,*

$$A_\lambda(n,2r+1) \leqslant 2^n \cdot \sum_{\ell=0}^{n} \frac{\binom{n}{\ell}}{\sum_{j=0}^{r/\lambda} \begin{pmatrix} \ell \\ j \end{pmatrix}}.$$

We now consider the case where $r \geqslant 2$ and $\lambda = 1$. To ease the notation, we introduce the function

$$V(n,w,r,\ell) = \sum_{4k+m \leqslant r-2\ell} \begin{pmatrix} n-w \\ k \end{pmatrix} \begin{pmatrix} w \\ m \end{pmatrix} 2^\ell.$$

We also assume that $V(n,w,r,\ell) = 0$ if $r - 2\ell < 0$. Given this setup, we may write $\mathbf{A}^T_{H,(r,1)} = (a_{i,j})_{i=1,j=1}^{n+1}$, where $a_{i,j} = V(n, j-1, r, |i-j|)$. We produce a weight assignment for (7) by considering another matrix $\hat{\mathbf{A}}(n,r)$ related to $\mathbf{A}^T_{H,(r,1)}$. The weights are given as $\boldsymbol{w} = \hat{\mathbf{A}}(n,r)^{-1} \cdot \mathbf{1}$. Theorem 11 states that, indeed, $\boldsymbol{w} = \hat{\mathbf{A}}(n,r)^{-1} \cdot \mathbf{1}$ is a feasible weight assignment for (7).

We introduce the matrix $\hat{\mathbf{A}}(n,r)$ as follows: Let $\hat{\mathbf{A}}(n,r) = (\hat{a}_{i,j})_{i=1,j=1}^{n+1}$ be defined so that for $j \neq i$, $\hat{a}_{i,j} = \min\Big(V(n,j-1,r,|i-j|), \frac{1}{r}(V(n,j-1,r,0)-1)\Big)$ and $\hat{a}_{j,j} = V(n,j-1,r,0)$.

Theorem 11. *For integers n,r,*

$$A_1(n,2r+1) \leqslant 2^n \cdot \sum_{\ell=0}^{n} \begin{pmatrix} n \\ \ell \end{pmatrix} \cdot \hat{w}_i,$$

where $\hat{\boldsymbol{w}} = (\hat{w}_0, \ldots, \hat{w}_{n+1}) = \hat{\boldsymbol{w}} = \hat{\mathbf{A}}(n,r)^{-1} \cdot \mathbf{1}$.

The results of our bounds on $A_1(n,5)$, when $n \leqslant 20$, are listed in Table I. As can be seen from the table, for every code length considered, Theorem 11 provides a tighter bound than Theorem 8.

TABLE I
COMPARISON OF UPPER BOUNDS FOR $A_1(n,5)$

Length	Bound from (7)	Theorem 8	Proposition 10	Theorem 11
5	65	716	254	197
6	209	2348	793	589
7	681	7545	2508	1771
8	2285	23959	8048	5396
9	7723	75688	26190	16719
10	27137	239112	86393	52906
11	95480	758457	288649	170584
12	340889	2422954	975954	562157
13	1233644	7812585	3336118	1885717
14	4471386	25462344	11518362	6425947
15	16320256	83943512	40130869	22271529
16	59909131	279998120	140971957	78091743
17	220589555	944741909	498899141	276648820
18	815168373	3222862985	1777507455	991500693
19	3022921187	11108080869	6371682078	3578006784
20	11241799535	38650901357	22966595378	12983261249

IV. CODE CONSTRUCTIONS

In what follows, we present constructions of linear codes under the ALD for the case where $\lambda = 1$. We first address the case where the minimum ALD is equal to three.

For a positive integer v, let $H_3' \in \mathbb{F}_2^{v \times (2^v-2)}$ be a matrix which has as its columns all non-zero vectors from $\mathbb{F}_2^v$, except for the all-ones vector. Write $H_3' = (\boldsymbol{h}_1', \boldsymbol{h}_2', \ldots, \boldsymbol{h}_{2^v-2}')$, where for $i \in \{1, \ldots, 2^v-2\}$, $\boldsymbol{h}_i'$ represents the i-th column of H_3'. Furthermore, let $\mathbf{1}_v \in \mathbb{F}_2^{v \times 1}$ be the all-ones vector.

In this setting, let $\mathcal{C}(2^v-2,3) \subseteq \mathbb{Z}_2^{2^v-2} \times \mathbb{Z}_2^{2^v-2}$ be equal to

$$\begin{aligned}\mathcal{C}(2^v-2,3) := \Big\{ & (\boldsymbol{a},\boldsymbol{b}) \in \mathbb{F}_2^{2^v-2} \times \mathbb{F}_2^{2^v-2} : \\ & \sum_{i=1}^{2^v-2} a_i \cdot \boldsymbol{h}_i' + \sum_{i=1}^{2^v-2} b_i \cdot \mathbf{1}_v = \mathbf{0} \}.\end{aligned}$$

Note that a code with minimum ALD distance three (for $\lambda = 1$) can either:

1) Correct a single Class 1 error, or
2) Detect a single Class 2 error.

In the following lemma, we show that the code $\mathcal{C}(2^v-2,3)$ can perform either 1) or 2).

Lemma 12. *For any positive integer n, $d_1(\mathcal{C}(2^v-2,3)) \geqslant 3$.*

Proof: Let $n = 2^v - 2$. Suppose that $(\boldsymbol{a},\boldsymbol{b}) \in \mathcal{C}(n,3)$ was transmitted and that the vector $(\boldsymbol{c},\boldsymbol{d}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ was received, where $(\boldsymbol{c},\boldsymbol{d}) = ((c_1,d_1),(c_2,d_2),\ldots,(c_n,d_n))$ is the result of at most a single Class 2 error occurring in $(\boldsymbol{a},\boldsymbol{b})$. For the remainder of this proof, let $\boldsymbol{s} = \sum_{i=1}^n c_i \cdot \boldsymbol{h}_i' + \sum_{i=1}^n d_i \cdot \mathbf{1}_v$. If a single Class 2 error has occurred in position $j \in \{1,\ldots,n\}$ of $\boldsymbol{a}$, then $\boldsymbol{s} = \boldsymbol{h}_j' \neq \mathbf{0}$. Otherwise, if the Class 2 error occurred in position j of $\boldsymbol{b}$, then $\boldsymbol{s} = \mathbf{1}_v \neq \mathbf{0}$ holds as well. Clearly, if no Class 2 errors occured, we have $\boldsymbol{s} = \mathbf{0}$. Thus, $\mathcal{C}(n,3)$ can detect whether a single Class 2 error has occurred by checking if $\boldsymbol{s}$ is non-zero.

Suppose that $(\boldsymbol{a},\boldsymbol{b}) \in \mathcal{C}(n,3)$ was transmitted and that the vector $(\boldsymbol{c},\boldsymbol{d}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ was received, where $(\boldsymbol{c},\boldsymbol{d}) = ((c_1,d_1),(c_2,d_2),\ldots,(c_n,d_n))$ is the result of at most one Class 1 error in $(\boldsymbol{a},\boldsymbol{b})$. We describe next how to recover $(\boldsymbol{a},\boldsymbol{b})$ from $(\boldsymbol{c},\boldsymbol{d})$. Let $\boldsymbol{s}_2 = \boldsymbol{s} + \mathbf{1}_v$. Note that if $(\boldsymbol{c},\boldsymbol{d})$ is the result of a single Class 1 error in $(\boldsymbol{a},\boldsymbol{b})$, then

$$\boldsymbol{s}_2 = \left(\sum_{i=1}^n c_i \cdot \boldsymbol{h}_i' + \sum_{i=1}^n d_i \cdot \mathbf{1}_v\right) + \mathbf{1}_v = \boldsymbol{h}_j' + \mathbf{1}_v + \mathbf{1}_v = \boldsymbol{h}_j',$$

where $j \in \{1,\ldots,n\}$ is the position of the error in $(\boldsymbol{a},\boldsymbol{b})$. Otherwise, if no error occured, $\boldsymbol{s}_2 = \boldsymbol{s} + \mathbf{1}_v = \mathbf{1}_v$. From the above discussion (recall $\boldsymbol{h}_j' \neq \mathbf{1}_v$ by design) it is clear that a $\mathcal{C}(n,3)$ decoder can recover $(\boldsymbol{a},\boldsymbol{b})$ from $(\boldsymbol{c},\boldsymbol{d})$ from $\boldsymbol{s}_2$ as follows. If $\boldsymbol{s}_2 = \mathbf{1}_v$, then the decoder concludes that no errors have occurred. Otherwise, if $\boldsymbol{s}_2 = \boldsymbol{h}_j'$ for some $j \in \{1,\ldots,n\}$, then the decoder corrects a Class 1 error at position j. ∎

As a consequence of Lemma 12 and Proposition 9, we have

$$\frac{4^n}{n+2} \leqslant A_1(n,3) \leqslant \frac{2^n(2^{n+1}-1)}{n+1}.$$

We now turn our attention to the problem of constructing codes with minimum ALD equal to d. We first describe the code construction, and then proceed to provide a proof of its correctness. Let $H_d' \in \mathbb{F}_2^{s\times 2n}$ be a parity check matrix for a code $\mathcal{C}$ with Hamming distance d. Write $H_d' = (\boldsymbol{h}_1',\ldots,\boldsymbol{h}_{2n}')$, where, as before, $\boldsymbol{h}_i'$ denotes the i-th column of H_d'. Let $\mathcal{C}(n,d) \subseteq \mathbb{Z}_2^n \times \mathbb{Z}_2^n$ be defined as

$$\mathcal{C}(n,d) := \Big\{(\boldsymbol{a},\boldsymbol{b}) = ((a_1,b_1),\ldots,(a_n,b_n)) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : \sum_{i=1}^n a_i \cdot \boldsymbol{h}_i' + \sum_{i=1}^n b_i \cdot (\boldsymbol{h}_i' + \boldsymbol{h}_{n+i}') = \mathbf{0}\Big\}.$$

Lemma 13. *For a positive integer n, $d_1(\mathcal{C}(n,d)) \geqslant d$.*

Proof: Since the code $\mathcal{C}(n,d)$ is linear, we only need to show that for any $(\boldsymbol{c},\boldsymbol{d}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, we have $\boldsymbol{s} = \sum_{i=1}^n c_i \cdot \boldsymbol{h}_i' + \sum_{i=1}^n d_i \cdot (\boldsymbol{h}_i' + \boldsymbol{h}_{n+i}') \neq 0$ when $d_1((\mathbf{0},\mathbf{0});(\boldsymbol{c},\boldsymbol{d})) < d$. Suppose, in particular, that $(\boldsymbol{c},\boldsymbol{d})$ is the result of m Class 1 errors, ℓ Class 2 errors, and k Class 3 errors, where $m + 2\ell + 4k < d$. Then, $\boldsymbol{s}$ is the sum of at most $m + 2\ell + k$ columns of H_d'. Since $m + 2\ell + 4k < d$, then $m + 2\ell + k < d$, and since H_d' is a parity-check matrix of a code with Hamming distance d, we conclude that $\boldsymbol{s} \neq \mathbf{0}$. ∎

We now comment on the size of the code $\mathcal{C}(n,d)$ for the case when $r = 2$ or $d = 5$. We compare the ALD code $\mathcal{C}(n,5)$ with codes constructed directly for the Hamming distance, assumed to be large enough. In particular, we consider (a) binary codes of length $2n$; and (b) the binary image of quaternary codes of length n.

For (a), since a code with minimum ALD equal to five must be able to correct four errors (resulting from two Class 1 errors), we require a binary code $\mathcal{C}_2$ that has minimum Hamming distance at least nine. By definition, $|\mathcal{C}(n,5)|$ is strictly larger than $|\mathcal{C}_2|$, and the direct method offers worse code rates than our construction. As an example, for $2n = 2^v - 2$, where $v \geqslant 2$ is a positive integer, $|\mathcal{C}(n,5)| \geqslant \frac{4^n}{(2n+2)^2}$, if shortened binary BCH codes are used as defining codes. Proposition 10 for the case $\lambda = 1$ and $r = 2$, and the lower bound on $|\mathcal{C}(n,5)|$, imply that for $2n = 2^v - 2$, and integer $v > 2$,

$$\frac{4^n}{(2n+2)^2} \leqslant A_1(n,5) \leqslant \frac{3\cdot 2^n}{(n+1)(n+2)}\left(2^{n+2} - n - 3\right).$$

Next, consider the case (b). For a quaternary code $\mathcal{C}_4$ to have minimum ALD distance five, the code should have Hamming distance at least five. Writing the sphere packing bound for quaternary codes with Hamming distance five, we have that $|\mathcal{C}_4| \leqslant \frac{4^n}{\sum_{j=0}^2 \binom{n}{j} 3^j}$. This value is strictly smaller than $\frac{4^n}{(2n+2)^2}$, which is a lower bound for the value of $|\mathcal{C}(n,5)|$, i.e.,

$$|\mathcal{C}_4| \leqslant \frac{4^n}{\sum_{j=0}^2 \binom{n}{j} 3^j} < \frac{4^n}{(2n+2)^2} \leqslant |\mathcal{C}(n,5)|,$$

where $v \geqslant 5$. Hence, our construction outperformes the direct approach for the case of a quaternary alphabet as well.

Acknowledgment: This work was funded by the Strategic Research Initiative (SRI) program at the University of Illinois, a CIA Postdoctoral Fellowship, by the NSF STC on Science of Information, and by the NISE program at SSC Pacific. This work was completed when H. M. Kiah was at University of Illinois, Urbana-Champaign.

References

[1] T. Alderson and S. Huntemann. "On maximum Lee distance codes," *J. of Discrete Mathematics*, 2013.
[2] J. Astola, "The theory of Lee-codes," *Lappeenranta University of Technology, Department of Physics and Mathematics, Research Report*, Jan. 1982.
[3] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, pp. 1628-1628, Sep. 2012.
[4] A. Fazeli, A. Vardy, E. Yaakobi, "Generalized Sphere Packing Bound," available at http://arxiv.org/abs/1401.6496, 2014.
[5] R. Feynman, "There's plenty of room at the bottom," Caltech, Pasadena. 29 Dec. 1959. Lecture.
[6] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, Jan. 2013.
[7] E. Hof et al., "Capacity-achieving polar codes for arbitrarily permuted parallel channels," *IEEE Trans. on Info. Theory*, vol. 59, no. 3, pp. 1505-1516, March 2013.
[8] K.A.S. Immink, *Codes for mass data storage systems*, Shannon Foundation Publisher, 2004.
[9] M. Kaykobad, "Positive solutions of positive linear systems," *Lin. Alg. and its App.*, vol. 64, pp. 133-140, Jan. 1985.
[10] A.A. Kulkarni, and N. Kiyavash, "Nonasymptotic upper bounds for deletion correcting codes," *IEEE Trans. on Info. Theory*, vol. 59, no. 8, pp. 5115-5130, April 2013.
[11] A. Mazumdar, A. Barg, and N. Kashyap, "Coding for high-density recording on a 1-D granular magnetic medium," *IEEE Trans. on Info. Theory*, vol. 57, no. 11, pp. 7403-7417, June 2011.
[12] O. Milenkovic and N. Kashyap, "On the design of codes for DNA computing," *Coding and Cryptography*, Springer Berlin Heidelberg, pp. 100-119, 2006.
[13] K. Nakamura, et al., "Sequence-specific error profile of Illumina sequencers," *Nucleic acids research*, voll. 39, no. 13, 2011.
[14] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Info. Theory*, vol. 47, no. 2, pp. 599-618, August 2002.
[15] Roth, R., *Introduction to coding theory*, Cambridge University Press, 2006.
[16] Tal, I., Vardy, A., "How to construct polar codes," *IEEE Trans. on Info. Theory*, vol. 59, no. 10, pp. 6562 - 6582, September 2013.