Correcting Limited-Magnitude Errors in the Rank-Modulation Scheme

Itzhak Tamo and Moshe Schwartz, *Member, IEEE*

***Abstract*—We study error-correcting codes for permutations under the infinity norm, motivated by a novel storage scheme for flash memories called *rank modulation*. In this scheme, a set of n flash cells are combined to create a single virtual multi-level cell. Information is stored in the permutation induced by the cell charge levels. Spike errors, which are characterized by a limited-magnitude change in cell charge levels, correspond to a low-distance change under the infinity norm. We define codes protecting against spike errors, called limited-magnitude rank-modulation codes (LMRM codes), and present several constructions for these codes, some resulting in optimal codes. These codes admit simple recursive, and sometimes direct, encoding and decoding procedures. We also provide lower and upper bounds on the maximal size of LMRM codes both in the general case, and in the case where the codes form a subgroup of the symmetric group. In the asymptotic analysis, the codes we construct outperform the Gilbert–Varshamov-like bound estimate.**

***Index Terms*—Asymmetric channel, flash memory, infinity norm, permutation arrays, rank modulation, subgroup codes.**

I. Introduction

In the race to dominate nonvolatile information-storage devices, flash memory is a prominent contender. Flash memory is an electronic nonvolatile memory that uses floating-gate cells to store information [7]. While initially, flash memory cells used to contains a single bit of information, in the standard multilevel flash-cell technology of today, every cell has $q > 2$ discrete states, $\{0, 1, \ldots, q-1\}$, and, therefore, can store $\log_2 q$ bits. The flash memory changes the state of a cell by injecting (*cell programming*) or removing (*cell erasing*) charge into/from the cell.

Flash memories possess an inherent asymmetry: writing is more time- and energy-consuming than reading [7]. The main reason behind this asymmetry is the iterative cell-programming procedure designed to avoid over-programming [2] (raising the cell's charge level above its target level). While cells can be programmed individually, only whole blocks (today, containing approximately 10^5 cells, see [7]) can be erased to the lowest state and then reprogrammed. Since overprogramming can only be corrected by the block erasure, in practice a conservative procedure is used for programming a cell, where charge is injected into the cell over quite a few rounds [2]. After every round, the charge level of the cell is measured and the next-round injection is configured. The charge level of the cell is made to gradually approach the target state until it achieves the desired accuracy. The iterative-programming approach is costly in time and energy.

Another major concern for flash memory is data reliability. The stored data can be corrupted due to charge leakage, a long-term factor that causes the data retention problem. The data can also be affected by other mechanisms, including read disturbance, write disturbance [7], etc. Many of the error mechanisms have an asymmetric property: they make the cells' charge levels drift in one direction. (For example, charge leakage makes the cell levels drift down.) Such a drift of cell charge levels causes errors in aging devices. The problem of data corruption is further aggravated as the number of levels in multilevel cells increases, since this reduces the safety margins for correct reading and writing.

To address these issues, the *rank-modulation scheme* has been recently suggested [17]. By removing the need to measure absolute cell-charge levels, the new scheme eliminates the risk of cell over-programming, and reduces the effect of asymmetric errors. In this scheme, a virtual cell that is composed of n cells with distinct charge levels, induces a permutation which is used to represent the stored information. Each cell has a *rank* which indicates its relative position when ordering the cells according to descending charge-level. The ranks of the n cells induce a permutation of $\{1, 2, \ldots, n\}$.

When writing or reading the cell charge levels, we only need to compare the charge levels *between cells*. Thus, the rank-modulation scheme eliminates the need to use the absolute values of cell levels to store information. Since there is no risk of over-programming and the cell charge levels can take continuous values, a substantially less conservative cell programming method can be used and the writing speed can be improved. In addition, asymmetric errors become less serious, because when cell levels drift in the same direction, their ranks are not affected as much as their absolute values. This way both the writing speed and the data reliability can be improved.

While the rank-modulation scheme alleviates some of the problems associated with current flash technology, the flash-memory channel remains noisy and an error-control mechanism is required. In this work we consider an error model which corresponds to spike errors. Such errors are characterized by a limited-magnitude change in the charge level of cells, and readily translates into a limited-magnitude change in the rank of, possibly, *all* cells in the stored permutation. This corresponds to a

Manuscript received July 30, 2009; revised December 25, 2009. Current version published May 19, 2010. This work was supported in part by the GIF by Grant 2179-1785.10/2007. The material in this paper was presented in part at the Information Theory and Applications Workshop (ITA 2010), San Diego, CA January 2010.

The authors are with the Department of Electrical and Computer Engineering, Ben-Gurion University, Beer Sheva 84105, Israel (e-mail: tamo@ee.bgu.ac.il; schwartz@ee.bgu.ac.il).

Communicated by H.-A. Loeliger, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2010.2046241

0018-9448/$26.00 © 2010 IEEE

bounded-distance change under the ℓ_∞-metric. We call codes protecting against such errors limited-magnitude rank-modulation codes (LMRM).

A similar error model for flash memory was considered not in the context of rank modulation in [8], while a different error-model (charge-constrained errors for rank modulation) was studied in [18]. Codes over permutations are also referred to as *permutation arrays* and have been studied in the past under different metrics [3], [4], [9], [10], [14], [15], [30]. Specifically, permutation arrays under the ℓ_∞-metric were considered in [24].

The main contribution of this paper is a set of constructions and bounds for such codes. The constructions presented are applicable for a wide range of parameters, and admit simple decoding and encoding procedures. We also present bounds on code parameters both for the general case, as well as for the more restricted case of subgroup codes. Most notably, we present an asymptotically good family of codes, with nonvanishing normalized distance and rate, which exceed the Gilbert–Varshamov-like lower bound estimate.

It is important to note that, independently and concurrently, Kløve, Lin, Tsai, and Tzeng [20] describe Construction 1 and its immediate generalization, Construction 2. As the overall overlap is small, and since the two constructions lead to our Construction 3, which we show to produce an optimal code, we bring these first two here for the sake of completeness.

The rest of the paper is organized as follows. In Section II we define the notation, and introduce the error-model as well as the associated ℓ_∞-metric. We proceed in Section III and present the code constructions and encoding/decoding algorithms. In Section IV we investigate general bounds on LMRM codes, code-anticode bounds, and asymptotic-form bounds. We conclude in Section V with a summary of the results and a short concluding remarks.

II. Definitions and Notations

For any $m, n \in \mathbb{N}$, $m \leqslant n$, let $[m, n]$ denote the set $\{m, m+1, \ldots, n\}$, where we also denote by $[n]$ the set $[1, n]$. Given any set A of cardinality n, we denote by S_A the set of all permutations over the set A. By convention, we use S_n to denote the set $S_{[n]}$.

We will use both the vector notation for permutations $f \in S_n$, where $f = [f_1, f_2, \ldots, f_n]$ denotes the permutation mapping $i \mapsto f_i = f(i)$ for all $i \in [n]$, and the cycle notation, where $f = (f_1, f_2, \ldots, f_k)$ denotes the permutation mapping $f_i \mapsto f_{i+1}$ for $i \in [k-1]$ as well as $f_k \mapsto f_1$. Given two permutations $f, g \in S_n$, the product fg is a permutation mapping $i \mapsto f(g(i))$ for all $i \in [n]$.

Let us consider n flash memory cells which we name $1, 2, \ldots, n$. The charge level of each cell is denoted by $c_i \in \mathbb{R}$ for all $i \in [n]$. In the *rank-modulation scheme* defined in [17], the charge levels of the cells induce a permutation in the following way: The induced permutation (in vector notation) is $[f_1, f_2, \ldots, f_n]$ iff $c_{f_1} > c_{f_2} > \cdots > c_{f_n}$.

The rank-modulation scheme is defined by two functions: an encoding function $E : Q \to S_n$, which takes a symbol from the input alphabet $a \in Q$ and maps it to a permutation $f = E(a) \in S_n$, and a decoding function $D : S_n \to Q$. Since no channel is devoid of noise, a stored permutation $f = E(a)$ may be corrupted by any of a variety of possible disturbance found in flash memory (see [7]). Assuming the changed version of f, denoted f', is not too corrupted, we would like the decoding function to restore the original information symbol, i.e., $D(f') = a$.

For a measure of the corruption of a stored permutation we may use any of a variety of metrics over S_n (see [12]). Given a metric over S_n, defined by a distance function $d : S_n \times S_n \to \mathbb{N} \cup \{0\}$, an *error-correcting code* is a subset of S_n with lower-bounded distance between distinct members.

In [18], the Kendall-τ metric was used, where the distance between two permutations is the number of adjacent transpositions required to transform one into the other. This metric corresponds to a situation in which we can bound the total difference in charge levels, and the error-correcting codes are therefore named charge-constrained rank-modulation codes.

In this paper, we consider a different type of common error—a limited-magnitude spike error. Suppose a permutation $f \in S_n$ was stored by setting the charge levels of n flash memory cells to $c_1, c_2, \ldots, c_n$. We say a single spike error of limited-magnitude L has occurred in the ith cell if the corrupted charge level, c'_i, obeys $|c_i - c'_i| \leqslant L$. In general, we say spike errors of limited-magnitude L have occurred if the corrupted charge levels of all the cells $c'_1, c'_2, \ldots, c'_n$ obey

$$\max_{i \in [n]} |c_i - c'_i| \leqslant L.$$

Let us denote by f' the permutation induced by the cell charge levels $c'_1, c'_2, \ldots, c'_n$ under the rank-modulation scheme. Under the plausible assumption that distinct charge levels are not arbitrarily close (due to resolution constraints and quantization at the reading mechanism), i.e., $|c_i - c_j| > \ell$ for some positive constant $\ell \in \mathbb{R}$ for all $i \neq j$, a single spike error of limited-magnitude L implies a constant $d = \lfloor 2L/\ell \rfloor + 1$ such that

$$\max_{i \in [n]} \left| f^{-1}(i) - f'^{-1}(i) \right| < d. \tag{1}$$

Loosely speaking, an error of limited magnitude cannot change the *rank* of the cell i (which is simply $f^{-1}(i)$) by d or more positions. Intuitively, the reason for this is that even at the worst case, where all the cell levels are separated by ℓ, a single cell reducing its charge by L and all the rest increasing their charge by L will cause a maximal change in rank bounded by d as above.

If the bound d of (1) satisfies $d \geqslant n-1$, then any permutation may change into any other permutation by a single spike error. However, it is important to note that both L and ℓ are independent of n, and so for any bound d we can choose n large enough to be able to create meaningful error-correcting schemes as described in the rest of the paper.

The limited-magnitude-error model has been studied in the past in the context of the generalized n-cube [1], [8], and more related to our context, over permutations [20], [24].

We therefore find it suitable to use the ℓ_∞-metric over S_n defined by the distance function

$$d_\infty(f, g) = \max_{i \in [n]} |f(i) - g(i)|$$

for all $f, g \in S_n$. Since this will be the distance measure used throughout the paper, we will usually omit the ∞ subscript.

Definition 1: An LMRM-code with parameters (n, M, d), is a subset $C \subseteq S_n$ of cardinality M, such that $d_\infty(f, g) \geqslant d$ for all $f, g \in C$, $f \neq g$. (We will sometimes omit the parameter M.)

We note that unlike the charge-constrained rank-modulation codes of [18], in which the codeword is stored in the permutation induced by the charge levels of the cells, here the codeword is stored in the *inverse* of the permutation.

It may be the case that the code C forms a subgroup of the symmetric group S_n, which we will denote by $C \leqslant S_n$. We shall call such a code a *subgroup code*. Since groups offer a rich structure, we will occasionally constrain ourselves to discuss subgroup codes.

III. Code Constructions

In this section we describe three constructions for LMRM subgroup codes. The first two were discovered independently and concurrently by [20]. We begin our constructions with the following, which bears a resemblance to the unidirectional limited-magnitude codes described in [1]. This construction will turn out to be a simple case of a more general construction given later.

Construction 1: Given $n, d \in \mathbb{N}$ we construct

$$C = \{f \in S_n \mid f(i) \equiv i \pmod d\}.$$

Alternatively, for every $i \in [d]$ let

$$A_i = (d\mathbb{Z} + i) \cap [n] = \{j \in [n] \mid j \equiv i \pmod d\}$$

and define C to be the direct product of the symmetric groups over the A_i's

$$C = S_{A_1} \times S_{A_2} \times \cdots \times S_{A_d}.$$

Theorem 2: The code C from Construction 1 is an (n, M, d)-LMRM code with

$$M = (\lceil n/d \rceil !)^{n \bmod d} (\lfloor n/d \rfloor !)^{d-(n \bmod d)}.$$

Proof: The length and size of the code are easily seen to be as claimed. All we have to do now is show that the minimal distance of the code is indeed d. Let $f, g \in C$ be two distinct codewords, and let $i \in [n]$ be such that $f(i) \neq g(i)$. Since $f(i) \equiv g(i) \pmod d$ it follows that $|f(i) - g(i)| \geqslant d$, and so $d(f, g) \geqslant d$. ∎

This construction allows a simple encoding procedure. To simplify the presentation let us assume that d divides n. The encoder takes as input an integer $m \in [0, M-1]$ (where M is the size of the code), e.g., by translating from a string of $\lfloor \log_2 M \rfloor$ binary input symbols. The number M can then be written in base $(n/d)!$, that is

$$M = \sum_{i=0}^{d-1} m_i \left((n/d)!\right)^i$$

where $0 \leqslant m_i \leqslant (n/d)! - 1$. Finally, for every i we map the ith digit, m_i, to $S_{A_{i+1}}$ using some function

$$\mathcal{F}_i : \{0, 1, \ldots, (n/d)! - 1\} \to S_{A_{i+1}}.$$

There are numerous efficiently computable functions to satisfy $\mathcal{F}_i$, such as the factoradic representation (see [22], [23], and [27]), as well others (see [21] and references therein). Then, by using $\{\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{d-1}\}$ the resulting encoding becomes

$$m \mapsto \mathcal{F}_0(m_0) \times \mathcal{F}_1(m_1) \times \cdots \times \mathcal{F}_{d-1}(m_{d-1}).$$

A straightforward decoding procedure is also obtainable. Let us assume that $f \in C$ was stored, where C is an (n, M, d)-LMRM code from Construction 1, while the retrieved permutation was $f' \in S_n$. We further assume that the maximum magnitude of errors introduced by the channel is $\lfloor (d-1)/2 \rfloor$, i.e., $|f(i) - f'(i)| \leqslant \lfloor (d-1)/2 \rfloor$ for all $i \in [n]$.

Since C is a code of minimum distance d, there is a unique codeword f^* at distance at most $\lfloor (d-1)/2 \rfloor$ from f'. Recovering this codeword is simple and may be done independently for each of the coordinates: For every coordinate $i \in [n]$, there is a unique $f_i^* \in [n]$ such that $|f_i^* - f'(i)| \leqslant \frac{d-1}{2}$ and $f_i^* \equiv i \pmod d$. The recovered permutation $f^* \in S_n$ is given by $f^*(i) = f_i^*$. By definition, $f^* \in C$, and by the algorithm presented we also have $d(f^*, f') \leqslant \lfloor (d-1)/2 \rfloor$, hence $f^* = f$ which is the original permutation which was stored.

Finding the original input message may be accomplished by decomposing $f \in C$ into a product of permutations from S_{A_i} and applying $\mathcal{F}_i^{-1}$ appropriately.

We now extend the direct-product approach and generalize the previous construction. First we introduce a new notation. Given $f \in S_n$, and a set $A \subseteq \mathbb{N}$ of size n, we denote by f_A the same permutation but over A. More formally, assuming $A = \{a_1, a_2, \ldots, a_n\}$, with $a_1 < a_2 < \cdots < a_n$, we set

$$f_A = [a_{f(1)}, a_{f(2)}, \ldots, a_{f(n)}].$$

Furthermore, given a set $C \subseteq S_n$, we define

$$C_A = \{f_A \mid f \in C\}.$$

Construction 2: Let $n, k \in N$, and define the sets

$$A_i = (k\mathbb{Z} + i) \cap [n]$$

for all $i \in [k]$. Furthermore, for all $i \in [k]$ let C^i be an (n_i, M_i, d_i)-LMRM code, with $n_i = |A_i|$. We construct the code $C \subseteq S_n$

$$C = C^1_{A_1} \times C^2_{A_2} \times \cdots \times C^k_{A_k}.$$

Theorem 3: The code C from Construction III is an (n, M, d)-LMRM with $M = \prod_{i=1}^{k} M_i$, and $d = \min_{i\in[k]} kd_i$. (By convention, the distance of a code with one codeword is defined as infinity.)

Proof: Again, the length and size of the code are easily verified. In addition, given $f, g \in C$, $f \neq g$, it is easy to see that $f(i) - g(i)$ is a multiple of k, for any $i \in [n]$, and so the distance of each of the constituent codes is scaled by k, giving the desired result. ∎

Before describing the next construction we briefly observe some properties which may be thought of as analogs to the case of linear subspace codes. The metric defined by d_∞ over S_n is a right invariant metric (see [12]), i.e., for any $f, g, h \in S_n$

$$d_\infty(f, g) = d_\infty(fh, gh).$$

We can then define the *weight* of a permutation $f \in S_n$ as

$$\mathrm{wt}(f) = d_\infty(f, \iota)$$

where ι denotes the identity permutation. Thus, for any $C \leqslant S_n$, an (n, d)-LMRM subgroup code, it follows that

$$d = \min_{f\in C, f\neq\iota} \mathrm{wt}(f).$$

For convenience, given a set $H \subseteq S_n$, we denote

$$d(H) = \min_{f,g\in H, f\neq g} d(f, g)$$
$$\overline{d}(H) = \max_{f,g\in H, f\neq g} d(f, g).$$

Finally, we recall the following notation: For $H, K \subseteq S_n$ we denote

$$H^K = \left\{h^k = khk^{-1} \mid h \in H, k \in K\right\}.$$

Construction 3: Let H and K be subgroups of S_n such that $H^K = H$ and $H \cap K = \{\iota\}$. We construct the code C from the following semidirect group product

$$C = H \rtimes K \cong HK = \{hk \mid h \in H, k \in K\}.$$

Theorem 4: The code from Construction 3 is an (n, M, d)-LMRM subgroup code with $M = |H||K|$ and

$$d \geqslant \min\left\{d(H), d(K), \max\left\{d(H) - \overline{d}(K), d(K) - \overline{d}(H)\right\}\right\}.$$

Proof: It is well known (see, for example, [16]) that if $H^K = H$ and $H \cap K = \{\iota\}$ then $HK = KH \leqslant S_n$ and $|HK| = |H||K|$. Given $h \in H$ and $k \in K$, where $h, k \neq \iota$, then from the triangle inequality

$$\begin{aligned} d(\iota, hk) &\geqslant d(\iota, k) - d(hk, k) = \mathrm{wt}(k) - \mathrm{wt}(h) \\ &\geqslant d(K) - \overline{d}(H). \end{aligned}$$

Interchanging h and k gives the other lower bound, and so when $h, k \neq \iota$

$$\mathrm{wt}(hk) \geqslant \max\left\{d(H) - \overline{d}(K), d(K) - \overline{d}(H)\right\}.$$

To finish the proof, if $hk \neq \iota$ and either $h = \iota$ or $k = \iota$, we have $hk \in H \cup K$ and so $\mathrm{wt}(hk) \geqslant \min\{d(H), d(K)\}$. ∎

The lower bound on the distance given in Theorem 4 , which we shall call the *design distance*, is sometimes not tight as is shown in the following example.

Example 5: Let us construct an LMRM code of length $n = 6$ and distance $d = 3$. According to construction III , the code $S_2 \times S_2 \times S_2$ is a $(6, 8, 3)$-LMRM code.

We can improve this by looking at the code $C_3 \leqslant S_3$ defined by

$$C_3 = \{[1, 2, 3], [2, 3, 1], [3, 1, 2]\}$$

i.e., the cyclic group of size 3, which is a $(3, 3, 2)$-LMRM code. By Construction III , the code $C_3 \times C_3$ is a $(6, 9, 4)$-LMRM code, providing us a larger code than the previous one, with a larger distance.

Finally, let us define $K \leqslant S_6$, a $(6, 2, 5)$-LMRM code, as

$$K = \{[1, 2, 3, 4, 5, 6], [6, 5, 4, 3, 2, 1]\}.$$

It may be verified that $H = C_3 \times C_3$ and K can be used with Construction III , resulting in a $(6, 18, 3)$-LMRM code. We note that while the design distance guaranteed by Theorem 4 is just 1, the resulting distance of the code is actually 3. □

One might think that the bound of Theorem 4 can produce only weak lower bounds. However, the following example shows the bound not only produces a high lower bound, but is also tight in this case.

Example 6: Let $n = 4t$, and consider H' to be the group generated by $(1, 2, \ldots, 2t)$ (in cycle notation) and H'' generated by $(2t + 1, 2t + 2, \ldots, 4t)$. Set $H = H' \times H''$. Also, let K be the group generated by $(1, 4t)(2, 4t - 1)\ldots(2t, 2t + 1)$.

One can easily verify that H and K satisfy the conditions of Theorem 4, and also that $d(H) = t$, $\overline{d}(H) = 2t - 1$, $d(K) = \overline{d}(K) = 4t - 1$. It follows that the code HK constructed by Theorem 4 has twice the size of H, and $d(HK) = t = n/4$, thus attaining the design distance with equality. □

IV. Bounds

A. General Bounds

The first two bounds we present are the obvious analogues of the Gilbert–Varshamov bound, and the ball-packing bound (see, for example, [25]). Their proofs are standard and are omitted. We first define the *ball* of radius r and centered about $f \in S_n$ as the set

$$B_{r,n}(f) = \{g \in S_n \mid d(f, g) \leqslant r\}.$$

As aforementioned, the ℓ_∞ metric over S_n is right invariant, and so the size of a ball depends only on r and n, and not on the choice of center. We will therefore denote by $|B_{r,n}|$ the size of a ball of radius r in S_n.

Proposition 7: Let n, M, and d, be positive integers such that $|B_{d-1,n}|\, M \leqslant n!$. Then there exists an (n, M, d)-LMRM code.

Proposition 8: Let C be an (n, M, d)-LMRM code. Then

$$\left|B_{\lfloor (d-1)/2 \rfloor, n}\right| M \leqslant n!.$$

We now proceed to present two upper bounds which are stronger, in general, than the ball-packing bound of Proposition 8. The first pertains to subgroup codes, while the second is more general. Before starting, we recall some well-known results from group theory (see [16]).

Let G be a subgroup of S_n. For any $i \in [n]$, the orbit of i under the action of G is defined as the set

$$i^G = \{g(i) \mid g \in G\}.$$

The stabilizer of i under the action of G is defined as

$$G_i = \{g \in G \mid g(i) = i\}$$

and is a subgroup of G. Furthermore

$$|G| = |i^G| \cdot |G_i|. \tag{2}$$

Theorem 9: If C is an (n, M, d)-LMRM subgroup code, then

$$M \leqslant \frac{n!}{(d!)^{\lfloor n/d \rfloor}(n \bmod d)!}.$$

Proof: For convenience, let us denote $r = n \bmod d$, and $k = \lfloor n/d \rfloor$. Let us now consider C as it acts on the d-subsets of $[n]$. By (2) we get

$$M = |C| = \left|[1,d]^C\right| \cdot \left|C_{[1,d]}\right| \leqslant \binom{n}{d} \left|C_{[1,d]}\right|$$

where the last inequality follows from the fact that the orbit of $[1, d]$ under C contains at most all the d-subsets of $[n]$. We can take another similar step and get

$$\begin{aligned} M &\leqslant \binom{n}{d} \left|C_{[1,d]}\right| = \binom{n}{d} \left|[d+1,2d]^{C_{[1,d]}}\right| \left|C_{[1,d],[d+1,2d]}\right| \\ &\leqslant \binom{n}{d}\binom{n-d}{d} \left|C_{[1,d],[d+1,2d]}\right| \end{aligned}$$

where $[d+1, 2d]^{C_{[1,d]}}$ denotes the orbit of $[d+1, 2d]$ under the action of $C_{[1,d]}$, i.e., the stabilizer of $[1, d]$ under C, while $C_{[1,d],[d+1,2d]}$ denotes the subgroup of C stabilizing both $[1, d]$ and $[d+1, 2d]$.

Reiterating the argument above we reach

$$M \leqslant \prod_{i=0}^{k-1} \binom{n-di}{d} \left|C_{[1,d][d+1,2d],\ldots,[(k-1)d+1,kd]}\right|.$$

It is now easy to see that

$$C_{[1,d],[d+1,2d],\ldots,[(k-1)d+1,kd]} = \{\iota\}$$

or else the minimum distance d of C would be violated. Thus

$$M \leqslant \prod_{i=0}^{k-1} \binom{n-di}{d} = \frac{n!}{(d!)^k r!}.$$

■

We can strengthen the upper bound of Theorem 9 by showing that codes attaining it with equality must also satisfy certain divisibility conditions.

A group $G \leqslant S_n$ is said to be *transitive* if for any $i, j \in [n]$ there is a permutation $f \in G$ such that $f(i) = j$. By (2), the size of such a group G must be divisible by n, since the orbit of i under the action of G is $[n]$.

Extending this definition, we say a group $G \leqslant S_n$ is *k-homogeneous* if for any two k-sets $A, B \subseteq [n]$, there exists a permutation $f \in G$ such that $f(A) = B$, where $f(A) = \{f(a) \mid a \in A\}$. It then follows from (2), that the size of such a group G must be divisible by $\binom{n}{k}$.

The following theorem was given in [6].

Theorem 10: Let $G \leqslant S_n$ be a k-homogeneous finite group, where $2k \leqslant n+1$. Then G is also $(k-1)$-homogeneous.

Hence, for a k-homogeneous group $G \leqslant S_n$, $2k \leqslant n+1$, the size of the group G is divisible by

$$K_{n,k} = \operatorname{lcm}\left\{\binom{n}{k}, \binom{n}{k-1}, \ldots, \binom{n}{1}\right\}.$$

Theorem 11: Let $C \leqslant S_n$ be an (n, M, d)-LMRM subgroup code attaining the upper bound of Theorem 9 with equality, i.e.

$$M = \frac{n!}{(d!)^{\lfloor n/d \rfloor}(n \bmod d)!}.$$

Then

$$\operatorname{lcm}\left\{K_{n-id,d} \;\middle|\; 0 \leqslant i \leqslant \frac{n-2d+1}{d}\right\} \;\middle|\; M.$$

Proof: If we examine the proof of Theorem 9, for C to attain the upper bound we must have $\left|[1,d]^C\right| = \binom{n}{d}$. Thus, for any d-subset $A \subseteq [n]$, there exists a permutation $f_A \in C$ such that $f_A([1, d]) = A$. It now follows, that for any two d-subsets $A, B \subseteq [n]$, we have that $f_B f_A^{-1}(A) = B$, and $f_B f_A^{-1} \in C$ since C forms a subgroup. Hence, C is d-homogeneous. If $2d \leqslant n+1$ then by Theorem 10 we have $K_{n,d} \mid M$.

Continuing in the same manner, the group $C_{[1,d]}$ may be viewed as a permutation group over $[n-d]$ by deleting the elements of $[d]$ and relabeling the rest. Again, we must have $\left|[d+1,2d]^{C_{[1,d]}}\right| = \binom{n-d}{d}$ which means that $C_{[1,d]}$ is also d-homogeneous. Again, if $2d \leqslant n-d+1$ then $K_{n-d,d}$ divides $\left|C_{[1,d]}\right|$, but $|C_{1,d}|$ divides $|C|$ since $C_{[1,d]} \leqslant C$. Reiterating the above arguments proves the claim. ■

It is also important to notice that if an (n, M, d)-LMRM subgroup code C exists, then $M \mid n!$ since $C \leqslant S_n$.

Example 12: Continuing Example 5 we would like to find an upper bound to LMRM subgroup codes of length $n = 6$ and minimum distance $d = 3$.

We first substitute n and d in the ball-packing bound of Proposition 8. We get an upper bound (not only for subgroup codes)

of $\lfloor 6!/13 \rfloor = 55$ since the size of a ball of radius 1 in S_6 equals 13.

Setting $n = 6$ and $d = 3$ in Theorem 9 we get an upper bound of size $6!/(3!)^2 = 20$. If a $(6, 20, 3)$-LMRM subgroup code exists, then by Theorem 11 its size must be divisible by its length (since it must be 1-homogeneous). However, 6 does not divide 20, and the next candidate for an upper bound, 19, does not divide $6! = 720$. Thus, the resulting upper bound is 18. This makes the $(6, 18, 3)$-LMRM subgroup code from Example 5 optimal. □

B. Codes and Anticodes

We turn to describe another powerful bounding technique. The resulting bounds bear a striking resemblance to the code-anticode method of Delsarte [11] and the set-antiset method of Deza [13]. However, both methods are not directly applicable to the case at hand.

Given a metric space with integer distances, we can construct a graph whose vertices are the points in the space, and an edge connects two vertices if and only if they are at distance 1 from each other. We call this the *induced graph* of the metric. If the metric distance between any two points in the space equals the length of the shortest path between the corresponding vertices in the induced graph (i.e., the distance in the graph), we say the metric space is *graphic*.

The code-anticode method of Delsarte requires a graphic metric space which forms a distance-regular graph. In our case, the ℓ_∞-metric over S_n is not even graphic, and hence the code-anticode method does not apply. The set-antiset method requires a metric over S_n which is both right and left invariant. Again, the ℓ_∞ metric-fails to meet the method's requirements since it is not left invariant.

Given a set $A \subseteq S_n$, we denote

$$\mathcal{D}(A) = \{d(f, g) \mid f, g \in A\}.$$

We also denote the *inverse* of A as

$$A^{-1} = \left\{f^{-1} \mid f \in A\right\}.$$

Definition 13: Two sets, $A, B \subseteq S_n$ are said to be a set and an antiset if

$$\mathcal{D}(A) \cap \mathcal{D}(B) = \{0\}.$$

The following is the set-antiset bound for right-invariant metrics over S_n.

Theorem 14: Let $d : S_n \times S_n \to \mathbb{N} \cup \{0\}$ be a distance measure inducing a right-invariant metric. Let $A, B \subseteq S_n$ be a set and an antiset. Then

$$|A| \cdot |B| \leqslant |S_n| = n!.$$

Proof: It is obvious that

$$A^{-1}B = \left\{f^{-1}g \mid f \in A, g \in B\right\} \subseteq S_n.$$

We contend that $\left|A^{-1}B\right| = \left|A^{-1}\right| \cdot |B| = |A| \cdot |B|$. Let us assume the contrary, i.e., that there exist $f_1, f_2 \in A$ and $g_1, g_2 \in B$ such that $f_1^{-1}g_1 = f_2^{-1}g_2$ but not both $f_1 = f_2$ and $g_1 = g_2$.

In that case, it follows that $g_1 g_2^{-1} = f_1 f_2^{-1}$. We now have

$$d(g_1, g_2) = d(g_1 g_2^{-1}, \iota) = d(f_1 f_2^{-1}, \iota) = d(f_1, f_2).$$

But then

$$d(g_1, g_2) = d(f_1, f_2) \in \mathcal{D}(A) \cap \mathcal{D}(B) = \{0\}$$

implying that $g_1 = g_2$ and $f_1 = f_2$, a contradiction. ■

To apply the set-antiset method to LMRM codes we need the following definition.

Definition 15: An LMRM anticode with parameters (n, M, d), is a subset $A \subseteq S_n$ of cardinality M, such that $d_\infty(f, g) \leqslant d$ for all $f, g \in A$.

Corollary 16: Let C be an (n, M_C, d)-LMRM code, and let A be an $(n, M_A, d-1)$-LMRM anticode. Then $M_A M_C \leqslant n!$.

Proof: By the definition of a code and an anticode it is easily seen that $\mathcal{D}(A) \cap \mathcal{D}(C) = \{0\}$. The claim is then a direct consequence of Theorem 14. ■

Corollary 16 generalizes previous results. It may be easily verified that a ball of radius $\lfloor (d-1)/2 \rfloor$ centered about the identity permutation ι is an $(n, d-1)$-LMRM anticode. Thus, the ball-packing bound of Proposition 8 is a special case of Corollary 16.

The following is a generalization of Theorem 9 to LMRM codes which are not necessarily subgroups.

Theorem 17: If C is an (n, M, d)-LMRM code, then

$$M \leqslant \frac{n!}{(d!)^{\lfloor n/d \rfloor}(n \bmod d)!}.$$

Proof: We construct the following $(n, M', d-1)$-LMRM anticode A: Let us denote

$$A_i = ([1, d] + (i-1)d) \cap [n].$$

We now define the anticode A as

$$A = S_{A_1} \times S_{A_2} \times \cdots \times S_{A_{\lceil n/d \rceil}}.$$

It is easy to verify that A is indeed an anticode of maximum distance $d-1$, and that its size is

$$M' = (d!)^{\lfloor n/d \rfloor}(n \bmod d)!.$$

By Corollary 16, $M \cdot M' \leqslant n!$, and the claim on the maximal size of an LMRM code follows. ■

It should be noted that Theorem 17 does not make Theorem 9 redundant, since through the proof of the latter we were able to provide stricter necessary conditions for potential subgroup codes attaining the bound with equality, as seen in Theorem 11.

The next obvious question is: What is the size of the maximal size of an $(n, d-1)$-LMRM anticode?

Theorem 18: Let A be an $(n, M, d-1)$-LMRM anticode. Then $M \leqslant (d!)^{n/d}$.

Proof: For all $1 \leqslant i \leqslant n$ let $i^A = \{f(i) \mid f \in A\}$. It is easy to see that $\left|i^A\right| \leqslant d$, otherwise there would exist $f, g \in A$ such that $|f(i) - g(i)| \geqslant d$ which contradicts that maximal distance of A.

Let P be the following $n \times n$ binary matrix, where $P_{i,j} = 1$ iff there exists $f \in A$ such that $f(i) = j$, otherwise $P_{i,j} = 0$. It is well known (see for example [28]) that

$$|A| \leqslant \operatorname{per}(P) = \sum_{f \in S_n} \prod_{i=1}^{n} P_{i,f(i)}$$

since all summands are either 0 or 1, and every permutation in A corresponds to a nonvanishing summand.

According to Brégman's Theorem (see [5]), for any $n \times n$ binary matrix P with r_i 1's in the ith row

$$\operatorname{per}(P) \leqslant \prod_{i=1}^{n} (r_i!)^{\frac{1}{r_i}}.$$

In our case, every row of P contains at most d 1's. We can certainly change some 0's into 1's in P so that every row contains *exactly* d 1's, and by doing so, only increase the value of $\operatorname{per}(P)$. It now follows that

$$M = |A| \leqslant \operatorname{per}(P) \leqslant (d!)^{n/d}.$$

■

Thus, for the case of $d|n$ we have an optimal anticode:

Corollary 19: The anticode constructed as part of Theorem 17 is optimal when $d|n$.

When d does not divide n the anticodes constructed in the proof of Theorem 17 are not necessarily optimal. The following theorem shows we can build larger anticodes.

Theorem 20: Let us denote $r = n \bmod d$. Then there exists an $(n, M', d-1)$-LMRM anticode of size

$$M' = \frac{(d!)^{\left\lfloor \frac{n}{d} \right\rfloor - 1} (d-r)! \left\lfloor \frac{d+r}{2} \right\rfloor ! \left\lceil \frac{d+r}{2} \right\rceil !}{\left(\left\lfloor \frac{d+r}{2} \right\rfloor - r\right)! \left(\left\lceil \frac{d+r}{2} \right\rceil - r\right)!}.$$

Proof: Consider the following $(d+r) \times (d+r)$ binary matrix P:

$$P = \left(\begin{array}{c|c} 1_{\lceil (d+r)/2 \rceil \times d} & 0_{\lceil (d+r)/2 \rceil \times r} \\ \hline 0_{\lfloor (d+r)/2 \rfloor \times r} & 1_{\lfloor (d+r)/2 \rfloor \times d} \end{array} \right)$$

where $1_{i \times j}$ (respectively, $0_{i \times j}$) denotes the all 1's (respectively, all 0's) matrix of size $i \times j$. It may now be verified that

$$\operatorname{per}(P) = (d-r)! \prod_{i=0}^{r-1} \left(\left\lfloor \frac{d+r}{2} \right\rfloor - i \right) \left(\left\lceil \frac{d+r}{2} \right\rceil - i \right).$$

We now construct the following $n \times n$ binary matrix Q:

$$Q = \left(\begin{array}{ccccc} 1_{d \times d} & & & & \\ & 1_{d \times d} & & 0 & \\ & & \ddots & & \\ & 0 & & 1_{d \times d} & \\ & & & & P \end{array} \right)$$

where along the diagonal we have $\left\lfloor \frac{n}{d} \right\rfloor - 1$ blocks of $1_{d \times d}$.

All the rows contain a contiguous block of 1's of size d, and thus, all the permutations contributing to $\operatorname{per}(Q)$ form an anticode of maximum distance $d-1$. It can be easily seen that

$$M' = \operatorname{per}(Q) = (d!)^{\left\lfloor \frac{n}{d} \right\rfloor - 1} \operatorname{per}(P)$$

as claimed. ■

With these anticodes we get the following two theorems.

Corollary 21: If C is an (n, M, d)-LMRM code, then

$$M \leqslant \frac{n! \left(\left\lfloor \frac{d+r}{2} \right\rfloor - r\right)! \left(\left\lceil \frac{d+r}{2} \right\rceil - r\right)!}{(d!)^{\left\lfloor \frac{n}{d} \right\rfloor - 1} (d-r)! \left\lfloor \frac{d+r}{2} \right\rfloor ! \left\lceil \frac{d+r}{2} \right\rceil !}$$

where $r = n \bmod d$.

Proof: Simply use the size of the anticodes of Theorem 20 with Corollary 16. ■

Corollary 22: The optimal $(n, n-1)$-LMRM code, $n \geqslant 3$, has size 3.

Proof: By Corollary 21 we have the following upper bound on the size of $(n, n-1)$-LMRM codes:

$$\frac{n!}{(n-2)! \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil} = \begin{cases} \frac{n-1}{n} \cdot 4 & n \text{ even} \\ \frac{n}{n+1} \cdot 4 & n \text{ odd} \end{cases}$$

and since the size must be an integer, it cannot exceed 3. Such a code can be easily constructed for any $n \geqslant 3$ and is simply the cyclic group of order 3 on the coordinates $\{1, 2, n\}$:

$$C = \{\iota, (1,2,n), (1,n,2)\}$$

given in cycle notation. ■

On a side note, Corollary 22 was also shown in [20] using different arguments. Whether other infinite families can be shown to be optimal using these anticodes is still unresolved.

C. Asymptotic Bounds

Some of the constructions and bounds presented in previous sections take on a simple asymptotic form, which we explore below. We will compare the resulting asymptotic bounds with those implied by the previous constructions of [24].

Definition 23: Given an (n, M, d)-LMRM code, we say it has *rate* $R = \frac{\log_2 M}{n}$ and normalized distance $\delta = \frac{d}{n}$.

A slight peculiarity arises here: One might expect the rate of a code to be defined as $\frac{\log_2 M}{\log_2 n!}$ and not $\frac{\log_2 M}{\log_2 2^n} = \frac{\log_2 M}{n}$ since the ambient space S_n is of size $n!$. However, doing so results in asymptotic bounds equal to 0. Furthermore, the rates reported in the following sequence of theorem, are no longer bounded to the interval $[0, 1]$. This is the case in applications in which the alphabets at the input and output of the encoder are different, binary tuples as input and permutations as output in our case (for example, see [26] and [29]). The intuition, though, remains the same: Rn is the number of input bits we can encode into n flash cells using an LMRM-code of rate R.

We begin with the asymptotic form of Corollary 21, and remind that the binary entropy function $H_2 : [0,1] \to [0,1]$ is defined as

$$H_2(p) = -p \log_2 p - (1-p) \log_2 (1-p).$$

Theorem 24: For any (n, M, d)-LMRM code

$$R \leqslant \left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta\right) \log_2 \left(\left\lfloor \frac{1}{\delta} \right\rfloor - 1\right) + H_2\left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta\right) + 2 - 2\delta \left\lfloor \frac{1}{\delta} \right\rfloor + o(1).$$

Proof: According to Corollary 21

$$M \leqslant \frac{n! \left(\left\lfloor \frac{d+r}{2} \right\rfloor - r\right)! \left(\left\lceil \frac{d+r}{2} \right\rceil - r\right)!}{(d!)^{\left\lfloor \frac{n}{d} \right\rfloor - 1} (d-r)! \left\lfloor \frac{d+r}{2} \right\rfloor ! \left\lceil \frac{d+r}{2} \right\rceil !}$$

where $r = n \bmod d$. Moving to the R and δ notation and slightly simplifying the expression we get

$$2^{Rn} \leqslant \frac{n!}{((\delta n)!)^{\left\lfloor \frac{1}{\delta} \right\rfloor - 1} \left(\left\lfloor \frac{n\left(\delta + 1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right)}{2} \right\rfloor !\right)^2} \cdot \frac{\left(\left\lfloor \frac{n\left(\delta - 1 + \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right)}{2} \right\rfloor !\right)^2}{\left(n\left(\delta - 1 + \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right)\right)!} \cdot 2^{o(n)}.$$

At this point we use the well-known Stirling's approximation, $m! = \sqrt{2\pi m}(m/e)^m (1 + O(1/m))$. After rearranging we get

$$2^{Rn} \leqslant \frac{2^{\left(2 - 2\delta \left\lfloor \frac{1}{\delta} \right\rfloor\right) n}}{\delta^{\left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta\right) n} \left(\delta + 1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right)^{\left(\delta + 1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right) n}} \cdot 2^{o(n)}.$$

We take $\log_2$ of both sides, divide by n, and do some rearranging to reach

$$\begin{aligned} R &\leqslant 2 - 2\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta\right) \log_2 \delta \\ &\quad - \left(\delta + 1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right) \log_2 \left(\delta + 1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right) + o(1) \\ &= \left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta\right) \log_2 \left(\left\lfloor \frac{1}{\delta} \right\rfloor - 1\right) + H_2\left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta\right) \\ &\quad + 2 - 2\delta \left\lfloor \frac{1}{\delta} \right\rfloor + o(1) \end{aligned}$$

as claimed. ∎

For the next two asymptotic forms we need an estimate on the size of a ball in the ℓ_∞-norm. While for any *fixed* radius r, tight asymptotic bounds on $|B_{r,n}|$ are given in [28], we require an estimate for $r = \Theta(n)$. The best estimate, to our knowledge, for $0 \leqslant r \leqslant \frac{n-1}{2}$, was given in [19]:

$$|B_{r,n}| \geqslant \frac{\sqrt{2\pi n}}{2^{2r}} \left(\frac{2r+1}{e}\right)^n \tag{3}$$

$$|B_{r,n}| \leqslant ((2r+1)!)^{\frac{n-2r}{2r+1}} \prod_{i=r+1}^{2r} (i!)^{\frac{2}{i}}. \tag{4}$$

For our purposes, however, we do require an upper bound on $|B_{r,n}|$ for the entire range $0 \leqslant r \leqslant n-1$. Therefore, we present an augmentation of (4) in the following lemma.

Lemma 25: For all $0 \leqslant r \leqslant n-1$

$$|B_{r,n}| \leqslant \begin{cases} ((2r+1)!)^{\frac{n-2r}{2r+1}} \prod_{i=r+1}^{2r} (i!)^{\frac{2}{i}} & 0 \leqslant r \leqslant \frac{n-1}{2}, \\ (n!)^{\frac{2r+2-n}{n}} \prod_{i=r+1}^{n-1} (i!)^{\frac{2}{i}} & \frac{n-1}{2} \leqslant r \leqslant n-1. \end{cases}$$

Proof: It is easily seen that $B_{r,n}(\iota)$ is the set of all permutations corresponding to nonvanishing terms in $\operatorname{per}(A)$ where A is the binary banded Toeplitz matrix defined by $A_{i,j} = 1$ iff $|i - j| \leqslant r$. This observation has been used both in [28] and in [19].

The upper bound is immediately derived by using Brégman's Theorem. For example, for $\frac{n-1}{2} \leqslant r \leqslant n-1$, the matrix A has $2r + 2 - n$ rows with n 1's, and two rows with i 1's for each $r + 1 \leqslant i \leqslant n - 1$. ∎

We now state the asymptotic form of the Gilbert–Varshamov-like bound of Proposition 7.

Theorem 26: For any constant $0 < \delta \leqslant 1$ there exists an infinite sequence of (n, M, d)-LMRM codes with $\frac{d}{n} \geqslant \delta$ and rate $R = \frac{\log_2 M}{n}$ satisfying $R \geqslant f_{\mathrm{GV}}(\delta) + o(1)$, where

$$f_{\mathrm{GV}}(\delta) = \begin{cases} \log_2 \frac{1}{\delta} + 2\delta(\log_2 e - 1) - 1 & 0 < \delta \leqslant \frac{1}{2} \\ -2\delta \log_2 \frac{1}{\delta} + 2(1-\delta) \log_2 e & \frac{1}{2} \leqslant \delta \leqslant 1. \end{cases}$$

Proof: By Proposition 7 we are guaranteed the existence of an (n, M, d)-LMRM code of size $M \geqslant n! / |B_{d-1,n}|$. We can now use Lemma 25 and replace $|B_{d-1,n}|$ with an appropriate upper bound.

Suppose $\frac{n-1}{2} \leqslant d - 1 \leqslant n - 1$ (the proof for the other case is similar). Then by Lemma 25

$$\begin{aligned} |B_{\delta n - 1, n}| &\leqslant (n!)^{2\delta - 1} \prod_{i=\delta n}^{n-1} (i!)^{\frac{2}{i}} \\ &= \left(\frac{n}{e}\right)^{(2\delta-1)n} \prod_{i=\delta n}^{n-1} \left(\frac{i}{e}\right)^2 \cdot 2^{o(n)} \\ &= \frac{n^{(2\delta-1)n}}{e^n} \left(\frac{(n-1)!}{(\delta n - 1)!}\right)^2 \cdot 2^{o(n)} \\ &= \frac{n^n}{e^{(3-2\delta)n} \delta^{2\delta n}} \cdot 2^{o(n)}. \end{aligned}$$

We now have

$$2^{Rn} \geqslant \frac{n!}{|B_{d-1,n}|} \geqslant e^{(2-2\delta)n} \delta^{2\delta n} \cdot 2^{o(n)}.$$

Taking $\log_2$ of both sides and dividing by n completes the proof. ∎

The ball-packing bound of Proposition 8 has the following asymptotic equivalent.

Theorem 27: For any (n, M, d)-LMRM code

$$R \leqslant \delta + \log_2 \frac{1}{\delta} + o(1).$$

Proof: The bound of Proposition 8 together with the lower bound of (3) becomes

$$M \leqslant \frac{n!}{\left|B_{\lfloor (d-1)/2 \rfloor, n}\right|} \leqslant \frac{n! 2^{d'}}{\sqrt{2\pi n}} \left(\frac{e}{d'+1}\right)^n$$

where $d' = d - 2 + (d \bmod 2)$. Changing to the R and δ notation, using Stirling's approximation, and then taking $\log_2$ and dividing by n gives as

$$R \leqslant \delta + \log_2 \frac{1}{\delta} + o(1)$$

as desired. ∎

Finally, we analyze the asymptotics of the codes produced by Construction 1.

Theorem 28: For any constant $0 < \delta \leqslant 1$, Construction 1 produces codes of rate

$$R = \left(1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right) \log_2 \left(\left\lceil \frac{1}{\delta} \right\rceil !\right) + \left(\delta + \delta \left\lfloor \frac{1}{\delta} \right\rfloor - 1\right) \log_2 \left(\left\lfloor \frac{1}{\delta} \right\rfloor !\right).$$

Proof: For any (n, M, d)-LMRM code produced by Construction 1 we know that

$$M = (\lceil n/d \rfloor !)^{n \bmod d} (\lfloor n/d \rfloor !)^{d - (n \bmod d)}.$$

Just like before, we change to the δ and R notation:

$$2^{Rn} = \left(\left\lceil \frac{1}{\delta} \right\rceil !\right)^{n\left(1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor\right)} \left(\left\lfloor \frac{1}{\delta} \right\rfloor !\right)^{n\left(\delta + \delta \left\lfloor \frac{1}{\delta} \right\rfloor - 1\right)}.$$

We then take $\log_2$ of both sides, and divide by n to reach the claimed result. ∎

All the asymptotic bounds are shown in Fig. 1. Several interesting observations can be made. First, the ball-packing bound of Theorem 27 is weaker than the code-anticode bound of Theorem 24. This, however, may be due to a poor lower bound on the size of a ball from (3). It was conjectured in [19] that this lower bound might be improved substantially. We also note that Construction 1 produces codes which asymptotically out-perform the Gilbert–Varshamov-like bound of Theorem 26 for a wide range of δ (with crossover at $\delta \approx 0.34904$), and appear to be quite close to the bound otherwise. Again, this might be a result of a weak upper bound on the size of a ball. Finally, the codes presented by [24] are severely restricted since they are derived from binary codes in the n-cube, and as such, are bounded by the n-cube versions of the Gilbert–Varshamov bound and the MRRW bound (see, for example, [25]).

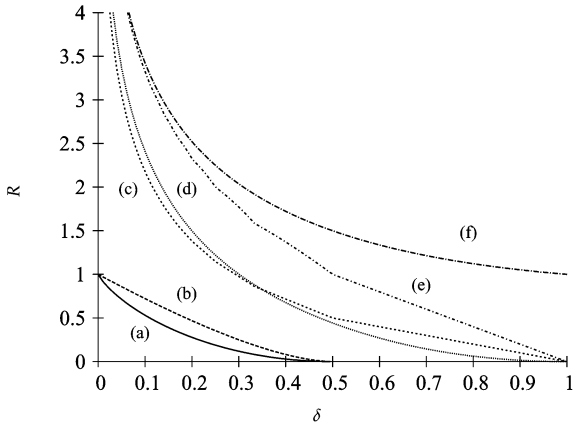

Fig. 1. (a) Gilbert–Varshamov bound in the n-cube. (b) The MRRW bound in the n-cube. (c) The rate of the code from Construction 1. (d) The Gilbert-Varshamov-like bound of Theorem 26. (e) The code-anticode bound of Theorem 24. (f) The ball-packing bound of Theorem 27.

V. Conclusion

We have studied codes for the rank modulation scheme which protect against limited-magnitude errors. We presented several code constructions (one explicit and two recursive) which, in some cases, produce optimal codes. The codes constructed can also be encoded and decoded recursively, while the code of Construction 1 may be encoded/decoded directly using a simple procedure with small loss in rate. We note that all the constructions we presented create codes which are subgroups of S_n.

We also explored bounds on the parameters of these codes. The strongest upper bound appears to be the code-anticode bound of Theorem 17. In the asymptotic study of these bounds, the simple code from Construction 1 shows a better rate than the one guaranteed by the Gilbert–Varshamov-like bound of Theorem 26, and the ball-packing upper bound of Theorem 27 is always weaker than that of the code-anticode bound of Theorem 24. Both, however, may be a result of a loose bound on the size of a ball in the ℓ_∞-metric.

Acknowledgment

The authors would like to thank the anonymous reviewers, whose comments helped improve the presentation of this paper.

References

[1] R. Ahlswede, H. Aydinian, L. Khachatrian, and L. M. G. M. Tolhuizen, "On q-ary codes correcting all unidirectional errors of a limited magnitude," presented at the Int. Workshop Algebr. Combinator. Coding Theory (ACCT), Kranevo, Bulgaria, 2004.

[2] A. Bandyopadhyay, G. Serrano, and P. Hasler, "Programming analog computational memory elements to 0.2% accuracy over 3.5 decades using a predictive method," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 2148–2151.

[3] I. F. Blake, "Permutation codes for discrete channels," *IEEE Trans. Inf. Theory*, vol. 20, pp. 138–140, 1974.

[4] I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Inf. Control*, vol. 43, pp. 1–19, 1979.

[5] L. M. Brégman, "Some properties of nonnegative matrices and their permanents," *Soviet Math. Dokl.*, vol. 14, pp. 945–949, 1973.

[6] P. J. Cameron, "Transitivity of permutation groups on unordered sets," *Math. Z.*, vol. 48, pp. 127–139, 1976.
[7] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash Memories*. Boston, MA: Kluwer, 1999.
[8] Y. M. S. Cassuto, V. Bohossian, and J. Bruck, "Codes for multilevel flash memories: Correcting asymmetric limited-magnitude errors," in *Proc. 2007 IEEE Int. Symp. Inf. Theory (ISIT2007)*, Nice, France, Jun. 2007, pp. 1176–1180.
[9] H. D. Chadwick and L. Kurz, "Rank permutation group codes based on Kendall's correlation statistic," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 2, pp. 306–315, Mar. 1969.
[10] C. J. Colbourn, T. Kløve, and A. C. H. Ling, "Permutation arrays for powerline communication and mutually orthogonal latin squares," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1289–1291, Jun. 2004.
[11] P. Delsarte, "An algebraic approach to association schemes of coding theory," *Philips J. Res.*, vol. 10, pp. 1–97, 1973.
[12] M. Deza and H. Huang, "Metrics on permutations, a survey," *J. Combin. Inf. Syst. Sci.*, vol. 23, pp. 173–185, 1998.
[13] M. Deza and P. Frankl, "On maximal numbers of permutations with given maximal or minimal distance," *J. Combin. Theory Ser. A*, vol. 22, 1977.
[14] C. Ding, F.-W. Fu, T. Kløve, and V. K. Wei, "Construction of permutation arrays," *IEEE Trans. Inf. Theory*, vol. 48, no. 4, pp. 977–980, Apr. 2002.
[15] F.-W. Fu and T. Kløve, "Two constructions of permutation arrays," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 881–883, May 2004.
[16] M. Hall, Jr, *Theory of Groups*. New York: American Mathematical Society, 1999.
[17] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2659–2673, Jun. 2009.
[18] A. Jiang, M. Schwartz, and J. Bruck, "Error-correcting codes for rank modulation," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1028–1037, Mar. 2010.
[19] T. Kløve, Spheres of Permutations Under the Infinity Norm—Permutations With Limited Displacement Univ. Bergen, Bergen, Norway, 2008, Tech. Rep. 376.
[20] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, Permutation Arrays Under the Chebyshev Distance 2009 [Online]. Available: http://arxiv.org/abs/0907.2682v1
[21] D. E. Knuth, *The Art of Computer Programming Volume 3: Sorting and Searching*, 2nd ed. Reading, MA: Addison-Wesley, 1998.
[22] C. A. Laisant, "Sur la numération factorielle, application aux permutations," *Bull. Société Math. France*, vol. 16, pp. 176–183.
[23] D. H. Lehmer, "Teaching combinatorial tricks to a computer," in *Proc. Symp. Appl. Math. Combin. Anal.*, 1960, vol. 10, pp. 179–193.
[24] T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Efficient encoding and decoding with permutation arrays," in *Proc. 2008 IEEE Int. Symp. Inf. Theory (ISIT2008)*, Toronto, Canada, 2008, pp. 211–214.
[25] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1978.
[26] B. H. Marcus, R. M. Roth, and P. H. Siegel, *Constrained Systems and Coding for Recording Channels*, V. S. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998.
[27] M. Mares and M. Straka, "Linear-time ranking of permutations," *Algorithms-ESA*, pp. 187–193, 2007.
[28] M. Schwartz, "Efficiently computing the permanent and Hafnian of some banded Toeplitz matrices," *Linear Algebr. Appl.*, vol. 430, no. 4, pp. 1364–1374, Feb. 2009.
[29] M. Schwartz and J. Bruck, "On the capacity of precision-resolution systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2112–2120, May 2010.
[30] H. Vinck, J. Haering, and T. Wadayama, "Coded M-FSK for power line communications," in *Proc. 2000 IEEE Int. Symp. Inf. Theory (ISIT2000)*, Sorrento, Italy, 2000, pp. 137–137.

Itzhak Tamo was born in Israel in 1981. He received the B.A. and B.Sc. degrees in 2008 from the Mathematics Department and the Electrical and Computer Engineering Department, respectively, Ben-Gurion University, Israel.

He is now a doctoral student with the Department of Electrical and Computer Engineering, Ben-Gurion University. His research interests include algebraic coding, combinatorial structures, and finite group theory.

Moshe Schwartz (M'03) was born in Israel in 1975. He received the B.A., M.Sc., and Ph.D. degrees from the Computer Science Department, Technion—Israel Institute of Technology, Haifa, in 1997, 1998, and 2004, respectively.

He was a Fulbright Postdoctoral Researcher in the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, and a Postdoctoral Researcher in the Department of Electrical Engineering, California Institute of Technology, Pasadena. He now holds a position with the Department of Electrical and Computer Engineering, Ben-Gurion University, Israel. His research interests include algebraic coding, combinatorial structures, and digital sequences.