Correcting Charge-Constrained Errors in the Rank-Modulation Scheme

Anxiao (Andrew) Jiang, *Member, IEEE,* Moshe Schwartz, *Member, IEEE,* and Jehoshua Bruck, *Fellow, IEEE*

***Abstract*—We investigate error-correcting codes for a novel storage technology for flash memories, the rank-modulation scheme. In this scheme, a set of n cells stores information in the permutation induced by the different charge levels of the individual cells. The resulting scheme eliminates the need for discrete cell levels, overcomes overshoot errors when programming cells (a serious problem that reduces the writing speed), and mitigates the problem of asymmetric errors.**

In this paper we study the properties of error-correcting codes for charge-constrained errors in the rank-modulation scheme. In this error model the number of errors corresponds to the minimal number of adjacent transpositions required to change a given stored permutation to another erroneous one – a distance measure known as Kendall's τ-distance.

We show bounds on the size of such codes, and use metric-embedding techniques to give constructions which translate a wealth of knowledge of binary codes in the Hamming metric as well as q-ary codes in the Lee metric, to codes over permutations in Kendall's τ-metric. Specifically, the one-error-correcting codes we construct are at least half the ball-packing upper bound.

***Index Terms*—flash memory, rank modulation, error-correcting codes, permutations, metric embeddings, Kendall's τ-metric**

I. Introduction

FLASH memory is an electronic non-volatile memory (NVM) that uses floating-gate cells to store information [4]. In the standard technology, every flash cell has q 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 or removing charge into/from the cell. To increase a cell from a lower state to a higher state, charge (e.g., electrons for nFETs) is injected into the cell and is trapped there. This operation is called *cell programming*. To decrease a cell's state, charge is removed from the cell, which is called *cell erasing*. Flash memory is widely used in mobile, embedded, and mass-storage systems because of its physical robustness, high density, and good performance [4]. To expand its storage capacity, research on multi-level cells with large values of q is actively underway.

For flash memories, writing is more time- and energy-consuming than reading [4]. The main factor is the iterative cell-programming procedure designed to avoid over-programming [2] (raising the cell's charge level above its target level). In flash memories, cells are organized into blocks, where each block has a large number ($\approx 10^5$) of cells [4]. Cells can be programmed individually, but to decrease the state of a cell, the whole block has to be erased to the lowest state and then re-programmed. Since over-programming 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.

A second challenge for flash memory is data reliability. The stored data can be lost 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 [4], etc. Many of the error mechanisms have an asymmetric property: they make the numerous 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.

In a recent paper [9], a new scheme for storing data in flash memories was proposed, the *rank-modulation scheme*. It aims at eliminating the risk of cell over-programming, and reducing the effect of asymmetric errors. Given a set of n cells with distinct charge levels, the *rank* of a cell indicates the relative position of its own charge level, and the ranks of the n cells induces a permutation of $\{1, 2, \ldots, n\}$. The rank modulation scheme uses this permutation to store information. To write data into the n cells, we first program the cell with the lowest rank, then the cell with the second lowest rank, and finally the cell with the highest rank. While programming the cell with rank i ($1 < i \leqslant n$), the only requirement is to make its charge level be above that of the cell with rank $i-1$.

The rank-modulation scheme eliminates the need to use the absolute values of cell levels to store information. Instead, the relative ranks are used. 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.

Anxiao (Andrew) Jiang is with the Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, U.S.A. (e-mail: ajiang@cs.tamu.edu).

Moshe Schwartz is with the Department of Electrical and Computer Engineering, Ben-Gurion University, Beer Sheva 84105, Israel (e-mail: schwartz@ee.bgu.ac.il).

Jehoshua Bruck is with the Department of Electrical Engineering, California Institute of Technology, 1200 E. California Blvd., Mail Code 136-93, Pasadena, CA 91125, U.S.A. (e-mail: bruck@paradise.caltech.edu).

In this paper, we study error-correcting codes for rank modulation. Even though asymmetric drifts of cell levels are tolerated better by rank modulation, errors can still happen because the cell levels do not necessarily drift at the same rate. The specific error model we explore is one in which the number of errors corresponds to the minimal number of adjacent transpositions required to change a given stored permutation to another erroneous one. This distance measure between permutations is known as Kendall's τ-distance. This models errors arising from an upper-bounded charge-level change in the cells, and the codes we construct are therefore named *charge-constrained rank-modulation codes (CCRM codes)*.

We prove bounds on the size of CCRM codes. We further employ metric-embedding techniques to translate binary codes in the Hamming metric, as well as q-ary codes in the Lee metric, to CCRM codes in Kendall's τ-metric. This establishes a general method for designing CCRM codes using an abundance of well-known codes over the other metrics. Specifically, we present a single-error-correcting code whose size is at least half of the ball-packing upper bound.

The rest of the paper is organized as follows. In Section II we define the notation and introduce Kendall's τ-metric. We continue in Section III and present code constructions through metric embeddings. In Section IV we investigate bounds on CCRM codes. We conclude in Section V with a summary of the results and a description of some ad-hoc constructions and resulting bounds.

II. Preliminaries

Let n flash memory cells be denoted by $1, 2, \ldots, n$. For $1 \leqslant i \leqslant n$, let $c_i \in \mathbb{R}$ denote the charge level of cell i. The ranks of the cells' charge levels induce a permutation of $\{1, 2, \ldots, n\}$ in the following way: The induced permutation is $[a_1, a_2, \ldots, a_n]$ iff $c_{a_1} > c_{a_2} > \cdots > c_{a_n}$, i.e., the cell a_1 has the highest charge level and the cell a_n has the lowest.

The *rank-modulation scheme* (see [9]) uses the permutations induced by the cells' charge levels to store information. Let S_n denote the set of $n!$ permutations over $\{1, 2, \ldots, n\}$. Let $Q = \{0, 1, 2, \ldots, q-1\}$ denote the alphabet of the symbol stored in the n cells. In the rank-modulation scheme, a decoding function, $D : S_n \to Q$, maps permutations to symbols from the user alphabet.

Since every channel may be subject to noise, which corrupts the transmitted data, designers of systems employing a rank-modulation scheme for flash memories need to consider the possibility of a stored permutation $\alpha \in S_n$ being transformed by any of a variety of possible channel disturbances (see [4]) to $\beta \in S_n$ such that $D(\alpha) \neq D(\beta)$. To model such a channel, often a metric is chosen such that $d(\alpha, \beta)$, i.e., the distance between the original value and its noisy version, is upper bounded with a high probability. An appropriate error-correcting code may then be designed with respect to that metric. There is a wide choice of possible metrics over S_n (see the survey [6]).

In a plausible realization of the rank-modulation scheme, given the precision constraints of the charge-placement mechanism, a minimal amount of charge is required to be inserted or removed to change a given induced permutation, and that will result in an *adjacent transposition*. Given a permutation, an adjacent transposition is the local exchange of two adjacent elements in the permutation: $[a_1, \ldots, a_{i-1}, a_i, a_{i+1}, a_{i+2}, \ldots, a_n]$ is changed to $[a_1, \ldots, a_{i-1}, a_{i+1}, a_i, a_{i+2}, \ldots, a_n]$. In this error model, a noisy version of an original permutation is said to contain t errors if the minimal number of adjacent transpositions required to transform the original permutation into the noisy one is t. For example, for $n = 4$, if errors change the permutation from $[2, 1, 3, 4]$ to $[2, 3, 4, 1]$, the number of errors is $t = 2$, because at least two adjacent transpositions are needed to change one into the other:

$$[2, 1, 3, 4] \to [2, 3, 1, 4] \to [2, 3, 4, 1].$$

A. Kendall's τ-metric

Throughout the paper we will use the vector notation for permutations: $\alpha = [a_1, a_2, \ldots, a_n] \in S_n$ denotes the permutation $\alpha(i) = a_i$ for all $1 \leqslant i \leqslant n$. Given some element $j \in \{1, 2, \ldots, n\}$, assume $\alpha(i) = j$. *Deleting* the element j from α results in the vector $[a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n]$ which we denote as $\alpha_{\downarrow j}$. Conversely, given some $j \in \{n+1, n+2, \ldots\}$ and an index $i \in \{1, 2, \ldots, n+1\}$, we can *insert* the element j in the i-th position resulting in the vector $[a_1, \ldots, a_{i-1}, j, a_i, \ldots, a_n]$ which we denote as $\alpha_{i\uparrow j}$.

For two permutations $\alpha, \beta \in S_n$, define their distance, $d_K(\alpha, \beta)$, as the minimal number of adjacent transpositions needed to change α into β. This distance measure is called Kendall's τ in statistics [10] or the bubble-sort distance, and it induces a metric over S_n. Where it is clear from the context that we use Kendall's τ-distance measure we will omit the subscript K.

The resulting metric is *graphic*: Let $\mathcal{K}_n = (V_n, E_n)$ be an undirected graph defined over the vertex set $V_n = S_n$, where we define $E_n = \{(\alpha, \beta) \mid d(\alpha, \beta) = 1\}$. Then it is well-known that for *any* $\alpha, \beta \in S_n$ the length of the shortest path connecting α and β in $\mathcal{K}_n$ equals $d(\alpha, \beta)$ [1]. The resulting graph, $\mathcal{K}_n$ is called the *adjacency graph* of the metric.

If $d(\alpha, \beta) = 1$, α and β are called *adjacent*. Any two permutations of S_n are at distance at most $\binom{n}{2}$ from each other. Two permutations of maximum distance are a reverse of each other.

The distance between two permutations can be computed by the algorithm hinted at by the following theorem (which appeared without proof in [10], Section 1.13).

Theorem 1. *Let $\alpha = [a_1, a_2, \ldots, a_n]$ and $\beta = [b_1, b_2, \ldots, b_n]$ be two permutations of length n. Suppose that $a_p = b_n$ for some $1 \leqslant p \leqslant n$. Then,*

$$d(\alpha, \beta) = d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n}) + n - p.$$

Proof: Let T be a sequence of $d(\alpha, \beta)$ adjacent transpositions that change α into β. Let us partition T into two sub-sequences T_1 and T_2, such that T_1 contains those adjacent transpositions that involve a_p, and T_2 contains those adjacent

[1] Not all metrics over S_n are graphic, such as the ℓ_∞-metric, for example.

transpositions that do not involve a_p. Let $|T|$, $|T_1|$ and $|T_2|$ denote the number of adjacent transpositions in T, T_1 and T_2, respectively. Clearly, $|T| = |T_1| + |T_2| = d(\alpha, \beta)$.

It is not hard to see that T_2 can also change $\alpha_{\downarrow a_p}$ into $\beta_{\downarrow b_n}$. That is because any adjacent transposition in T_1 does not change the relative positions of the elements $\{a_i\}_{i\neq p}$ in α and in $\alpha_{\downarrow a_p}$. Meanwhile, any adjacent transposition in T_2 changes the relative positions of $\{a_i\}_{i\neq p}$ in the same way for α and $\alpha_{\downarrow a_p}$. Therefore, $|T_2| \geqslant d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n})$. It can also be seen that $|T_1| \geqslant n - p$, because every adjacent transposition moves a_p forward in the permutation by one position, and from α to β, the element a_p has to be moved at by at least $n - p$ positions. Thus, $d(\alpha, \beta) = |T| = |T_1| + |T_2| \geqslant d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n}) + n - p$.

We now show that $d(\alpha, \beta) \leqslant d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n}) + n - p$. Consider a sequence of $d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n}) + n - p$ adjacent transpositions which is defined as follows: the first $n - p$ transpositions change $\alpha = [a_1, \ldots, a_{p-1}, a_p, a_{p+1}, \ldots, a_n]$ into $[a_1, \ldots, a_{p-1}, a_{p+1}, \ldots, a_n, a_p] = [\alpha_{\downarrow a_p}, a_p]$, while the remaining $d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n})$ steps change $[\alpha_{\downarrow a_p}, a_p]$ into $[\beta_{\downarrow b_n}, a_p] = \beta$. It follows that $d(\alpha, \beta) \leqslant d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n}) + n - p$, and therefore $d(\alpha, \beta) = d(\alpha_{\downarrow a_p}, \beta_{\downarrow b_n}) + n - p$. ■

The process of moving the appropriate element of α to its position as the last element of β may be now recursively repeated for transforming $\alpha_{\downarrow a_p}$ into $\beta_{\downarrow b_n}$. When β is the identity permutation, ι, the resulting algorithm is none other than the bubble-sort algorithm.

The adjacency graph of permutations under Kendall's τ-metric, $\mathcal{K}_n$, described in the previous section, is not distance regular in general and so the nice properties of such graphs (see [3]) cannot be used. In particular, the powerful code-anticode method of Delsarte [5], which was used in [1], [13], [14] does not apply here immediately. We will, however, provide a sphere-packing-like bound (which is actually a ball-packing bound) in a later section.

Definition 2. *The* sphere $\mathcal{S}_r(\alpha)$ *centered at* α *and of radius* r *is the set*

$$\mathcal{S}_r(\alpha) = \{\beta \in S_n \mid d(\alpha, \beta) = r\},$$

while the ball $\mathcal{B}_r(\alpha)$ *centered at* α *and of radius* r *is defined as the set*

$$\mathcal{B}_r(\alpha) = \{\beta \in S_n \mid d(\alpha, \beta) \leqslant r\}.$$

Even though $\mathcal{K}_n$ is not distance regular, fortunately, Kendall's τ-metric is right invariant [6], i.e., for any three permutations $\alpha, \beta, \gamma \in S_n$, we have $d(\alpha\gamma, \beta\gamma) = d(\alpha, \beta)$. Thus, the sizes of spheres and balls in this metric depend on their radius only, and not on the choice of center. We can therefore denote the size of a sphere (respectively, a ball) of radius r as $|\mathcal{S}_r|$ (respectively, $|\mathcal{B}_r|$).

Definition 3. *The* weight *of a permutation* $\alpha \in S_n$ *is defined as* $w(\alpha) = d(\alpha, \iota)$, *where* ι *is the identity permutation.*

By the previous observation, for any two permutations $\alpha, \beta \in S_n$, we have $d(\alpha, \beta) = w(\alpha\beta^{-1})$. We can also observe that $\mathcal{S}_r(\iota)$ is the set of all permutations of weight r in S_n.

If we define an *inversion* as a pair $(\alpha(i), \alpha(j))$ such that $\alpha(i) > \alpha(j)$ and $i < j$, then it is well-known (see Knuth, [11]) that the weight of a permutation is simply the number of inversions it contains, i.e.,

$$w(\alpha) = |\{(\alpha(i), \alpha(j)) \mid i < j \wedge \alpha(i) > \alpha(j)\}|.$$

We can extend this to get the expression

$$d(\alpha, \beta) = |\{(i, j) \mid \alpha(i) < \alpha(j) \wedge \beta(i) > \beta(j)\}|. \quad (1)$$

III. Codes from Metric Embeddings

We first define the object of interest in this study – codes for the rank-modulation scheme correcting charge-constrained errors.

Definition 4. *A* charge-constrained-error-correcting code for the rank-modulation scheme *of length* n*, size* M*, and minimal distance* d *(an* (n, M, d)*-CCRM code) is a subset* $C \subseteq S_n$ *of size* M *such that* $d_K(\alpha, \beta) \geqslant d$ *for all* $\alpha, \beta \in C$, $\alpha \neq \beta$.

In this section we explore ways of embedding the graph $\mathcal{K}_n$ which encapsulates Kendall's τ-metric over S_n, into two different graphs: $\mathbb{Z}_2^{\binom{n}{2}}$ with the Hamming metric, and $\mathbb{Z}_n!$ with the ℓ_1-metric. By doing so, a wealth of knowledge of coding techniques and constructions can be translated back to $\mathcal{K}_n$. One such result is a family of $(n, 3)$-CCRM codes, capable of correcting one adjacent-transposition error, which in a later section, will be shown to be of size at least half the ball-packing upper bound.

A. Embedding $\mathcal{K}_n$ into $\mathbb{Z}_2^{\binom{n}{2}}$

Let us consider the space $\mathbb{Z}_2^{\binom{n}{2}}$ endowed with the Hamming distance function: For all $v_1, v_2 \in \mathbb{Z}_2^{\binom{n}{2}}$, the Hamming distance between v_1 and v_2 is the number of positions in which they disagree. By abuse of notation we shall also refer to $\mathbb{Z}_2^{\binom{n}{2}}$ as the graph with vertices which are binary vectors of length $\binom{n}{2}$, and edges connecting vertices at Hamming distance 1.

We index the $\binom{n}{2}$ positions in every vector of $\mathbb{Z}_2^{\binom{n}{2}}$ by the set of ordered pairs $\{(i, j) \mid 1 \leqslant i < j \leqslant n\}$. Let us define the following mapping $\phi : S_n \to \mathbb{Z}_2^{\binom{n}{2}}$ in the following way: For all $\alpha \in S_n$, we set $\phi(\alpha)$ to be the binary vector whose position (i, j) is 0 if $\alpha^{-1}(i) < \alpha^{-1}(j)$ and 1 otherwise. In other words, position (i, j) is set to 1 iff (j, i) is an inversion of α.

Example 5. *Consider the permutation* $\alpha = [3, 4, 1, 2] \in S_4$. *We then have*

$$\begin{aligned}\phi(\alpha) &= (v_{(1,2)}, v_{(1,3)}, v_{(2,3)}, v_{(1,4)}, v_{(2,4)}, v_{(3,4)}) \\ &= (0, 1, 1, 1, 1, 0),\end{aligned}$$

since α *contains the inversions* $(3, 1)$, $(3, 2)$, $(4, 1)$, *and* $(4, 2)$.

Lemma 6. *The mapping* ϕ *is injective.*

Proof: Let v be a vector in the image of ϕ, i.e., $v \in \phi(S_n)$. We will show there exists exactly one permutation $\alpha \in S_n$ such that $\phi(\alpha) = v$.

First, the expression $p_n = \sum_{i=1}^{n-1} v_{(i,n)}$ counts the number of elements *smaller* than n which appear to the right of n in α. It follows that in vector notation n must appear in the p_n-th position from the right.

Next, we examine $p_{n-1} = \sum_{i=1}^{n-2} v_{(i,n-1)}$ which counts the number of elements smaller than $n-1$ which appear to its right. Thus, in the remaining $n-1$ as-yet unset positions in α, the element $n-1$ must appear in the p_{n-1}-th position from the right. Repeating the process, there is exactly one resulting permutation α for which $\phi(\alpha) = v$. ∎

Lemma 7. *For any two permutations $\alpha, \beta \in S_n$, if $d_K(\alpha,\beta) = 1$ then $d_H(\phi(\alpha),\phi(\beta)) = 1$.*

Proof: Let $\alpha, \beta \in S_n$ be two permutations such that $d_K(\alpha,\beta) = 1$. Since a single adjacent transposition is responsible for the change between the two permutations, denote the positions of the change as i and $i+1$, and let $\alpha(i) = a_i$ and $\alpha(i+1) = a_{i+1}$, while $\beta(i) = a_{i+1}$ and $\beta(i+1) = a_i$.

We now examine the case $a_i < a_{i+1}$ (the case $a_{i+1} > a_i$ is symmetric). It follows that the only difference between $\phi(\alpha)$ and $\phi(\beta)$ is the element in position (a_i, a_{i+1}) which switches from 0 to 1 since an inversion (a_{i+1}, a_i) was formed in β. Thus, $d_H(\phi(\alpha),\phi(\beta)) = 1$. ∎

The last lemma leads directly to the following conclusion.

Corollary 8. *For any two permutations $\alpha, \beta \in S_n$ we have*

$$d_H(\phi(\alpha),\phi(\beta)) \leqslant d_K(\alpha,\beta).$$

Proof: Consider a path of length $d_K(\alpha,\beta)$ connecting α and β in $\mathcal{K}_n$:

$$\alpha = \gamma_1 \to \gamma_2 \to \cdots \to \gamma_{d_K(\alpha,\beta)} = \beta.$$

By Lemma 7, the following is a path of length $d_K(\alpha,\beta)$ which connects $\phi(\alpha)$ and $\phi(\beta)$ in $\mathbb{Z}_2^{\binom{n}{2}}$:

$$\phi(\alpha) = \phi(\gamma_1) \to \phi(\gamma_2) \to \cdots \to \phi(\gamma_{d_K(\alpha,\beta)}) = \phi(\beta).$$

This may not be the shortest path connecting $\phi(\alpha)$ and $\phi(\beta)$ and so $d_H(\phi(\alpha),\phi(\beta)) \leqslant d_K(\alpha,\beta)$. ∎

The fact that distances contract under the mapping ϕ allows us to take constructions over $\mathbb{Z}_2^{\binom{n}{2}}$ and translate them to $\mathcal{K}_n$.

Theorem 9. *Let C_H be a binary $[\binom{n}{2}, k, d]$ linear code. Then there exists an (n, M, d)-CCRM code of size $M \geqslant n!/2^{\binom{n}{2}-k}$.*

Proof: Let C_H be a code as above. We define the following code over S_n:

$$C_K = \{\alpha \in S_n \mid \phi(\alpha) \in C_H\}.$$

By Corollary 8 the minimal distance between codewords of C_K is at least d. To prove the lower bound on the size C_K we note that C_H has $2^{\binom{n}{2}-k}$ cosets which partition $\mathbb{Z}_2^{\binom{n}{2}}$, each forming a binary $(\binom{n}{2}, 2^k, d)$ code. It follows that at least one of the cosets intersects $\phi(S_n)$, the image of ϕ, in at least $n!/2^{\binom{n}{2}-k}$ words. ∎

We note that the *design distance* d of the code C_H is not necessarily the actual distance of the resulting code C_K.

Theorem 9 also suggests a decoding algorithm for the code C_K provided one exists for C_H. The permutation $\alpha' \in S_n$ received from the channel is converted to the Hamming space by applying ϕ. If $\alpha \in S_n$ was the transmitted permutation, and no more than $\lfloor (d-1)/2 \rfloor$ errors occurred (where d is the design distance), then

$$d_H(\phi(\alpha'),\phi(\alpha)) \leqslant d_K(\alpha',\alpha) \leqslant \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Since ϕ is injective we are also guaranteed that $\phi(\alpha') = \phi(\alpha)$ iff no errors occurred, i.e., $\alpha' = \alpha$. We can now apply the decoding algorithm for C_H, correctly decoding to $\phi(\alpha)$, and then translating the resulting vector back to get α.

Example 10. *If we examine the codes resulting from this embedding construction we first note that the binary MDS codes in the Hamming metric (the whole space, the repetition code, and the parity code) construct MDS codes (see Section* IV*, Theorem* 19*) in Kendall's τ-metric (respectively, the whole space, a permutation and its reverse, and the even permutation code).*

Example 11. *Continuing to another example, if we set C_H to be the $[\binom{n}{2}, \binom{n}{2} - \lceil \log_2 \binom{n}{2} \rceil, 3]$ appropriately-shortened binary Hamming code, the constructed code C_K, is an $(\binom{n}{2}, 3)$-CCRM code of size at least $n!/2^{\lceil \log_2 \binom{n}{2} \rceil}$. This is comparable with the Gilbert-Varshamov-like lower bound (see Section* IV*, Theorem* 18*) and, for some values of n, even better: While the bound of Theorem* 18 *guarantees the existence of an $(n,3)$-CCRM code of size at least $n!/(\binom{n}{2} + n - 1)$, taking for example n which is a power of* 2*, Theorem* 9 *constructs a code of size at least $n!/(\binom{n}{2} - \frac{n}{2})$. The same may be said when using t-error-correcting binary BCH codes as the original code C_H.*

B. Embedding $\mathcal{K}_n$ into $\mathbb{Z}_n!$

We now turn to consider a different metric embedding. Let us define

$$\mathbb{Z}_n! = \mathbb{Z}_2 \times \mathbb{Z}_3 \times \cdots \times \mathbb{Z}_{n-1} \times \mathbb{Z}_n.$$

We further endow this space with the ℓ_1-metric. Let $v, u \in \mathbb{Z}_n!$, $v = (v_2, v_3, \dots, v_n)$, $u = (u_2, u_3, \dots, u_n)$, their ℓ_1-distance is defined as

$$d_1(v,u) = \sum_{i=2}^{n} |v_i - u_i|.$$

Again, by abuse of notation we shall also refer to $\mathbb{Z}_n!$ as the graph whose vertices are the elements of $\mathbb{Z}_n!$ and edges connect vertices at ℓ_1-distance 1.

We define the mapping $\psi : S_n \to \mathbb{Z}_n!$ in the following way: We map every $\alpha \in S_n$ to the vector $v \in \mathbb{Z}_n!$, $v = (v_2, \dots, v_n)$, such that v_j equals the number of inversions in α of the form (j,i), $1 \leqslant i \leqslant j-1$.

Lemma 12. *The mapping ψ is bijective.*

Proof: Let $\alpha \in S_n$ be some permutation, and let us examine the two mappings $v = \phi(\alpha)$ and $u = \psi(\alpha)$. It is easily seen that $u_j = \sum_{i=1}^{j-1} v_{(i,j)}$. A careful examination of the proof of Lemma 6 reveals that only these sums, i.e., the elements of $\psi(\alpha)$, are used to show that ϕ is injective. Hence, ψ is injective as well. To complete the proof we note that $|S_n| = |\mathbb{Z}_n!| = n!$ and so ψ is bijective. ∎

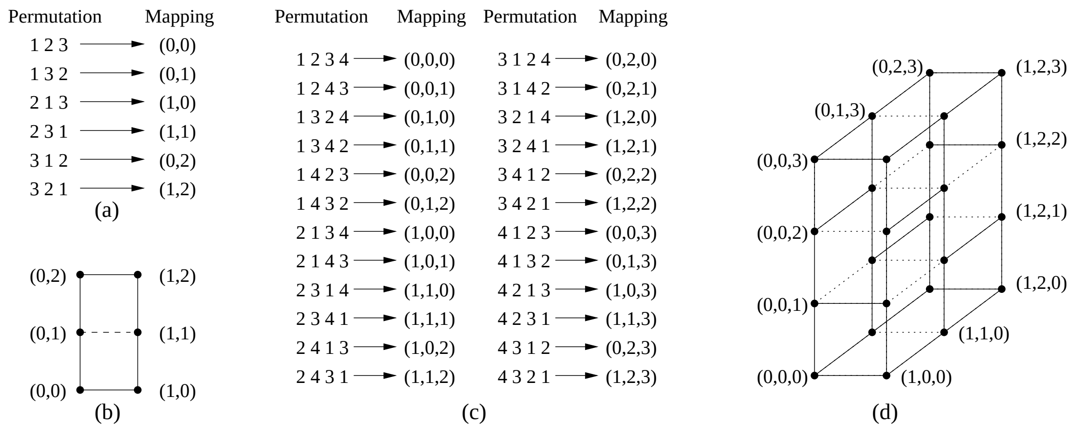

Figure 1. $\mathcal{K}_n$ and its embedding into $\mathbb{Z}_n!$. In the two arrays, the solid lines are the edges in both $\mathcal{K}_n$ and $\mathbb{Z}_n!$, and the dotted lines are the edges only in $\mathbb{Z}_n!$. (a) Mapping S_3 to $\mathbb{Z}_3!$ (b) Embedding $\mathcal{K}_3$ into $\mathbb{Z}_3!$ (c) Mapping S_4 to $\mathbb{Z}_4!$ (d) Embedding $\mathcal{K}_4$ into $\mathbb{Z}_4!$

Lemma 13. *For any two permutations $\alpha, \beta \in S_n$, if $d_K(\alpha, \beta) = 1$ then $d_1(\psi(\alpha), \psi(\beta)) = 1$.*

Proof: We again exploit the connection between ϕ and ψ. Let $\alpha, \beta \in S_n$ be two permutations such that $d_K(\alpha, \beta) = 1$. By Lemma 7, $d_H(\phi(\alpha), \phi(\beta)) = 1$. Since $\psi(\alpha)$ and $\psi(\beta)$ are just summations of the elements of $\phi(\alpha)$ and $\phi(\beta)$ according to some partition of the $\binom{n}{2}$ positions, it follows that $d_1(\psi(\alpha), \psi(\beta)) = 1$. ■

Corollary 14. *For any two permutations $\alpha, \beta \in S_n$ we have*

$$d_1(\psi(\alpha), \psi(\beta)) \leqslant d_K(\alpha, \beta).$$

Proof: The proof is essentially the same as that of Corollary 8. ■

Some examples of the embedding ψ are shown in Fig. 1. It can be seen that while each permutation has exactly $n-1$ adjacent permutations in $\mathcal{K}_n$, a vertex in $\mathbb{Z}_n!$ can have a higher degree, i.e., some edges of $\mathbb{Z}_n!$ do not exist in $\mathcal{K}_n$.

Since codes over a grid graph endowed with the Lee metric are much more common than codes over the ℓ_1-metric, we need one final trivial mapping. Let $\mathbb{Z}_q^m$ be the set of vectors of length m over the alphabet $\mathbb{Z}_q$ and let u and v be two such vectors. The Lee distance between them is defined as

$$d_L = \sum_{i=1}^{m} \min\left\{|v_i - u_i|, q - |v_i - u_i|\right\}.$$

By abuse of notation we again use $\mathbb{Z}_q^m$ to denote the graph whose vertices are the elements of $\mathbb{Z}_q^m$ and two vertices are connected by an edge iff their Lee distance is 1.

It is easily verifiable that $\mathbb{Z}_n!$ is a subgraph of $\mathbb{Z}_q^{n-1}$ when $q \geqslant n$. We note that endowing $\mathbb{Z}_q^{n-1}$ with the Lee metric, compared with endowing $\mathbb{Z}_q^{n-1}$ with the ℓ_1-metric, is expressed by several additional edges, which at the worst case, contract distances even further. We can now state the main construction.

Theorem 15. *Let C_L be an $(n-1, d)$ Lee-metric error-correcting over the alphabet $\mathbb{Z}_q$, $q \geqslant n$. Then there exists an (n, M, d)-CCRM code of size $M = |C_L \cap \mathbb{Z}_n!|$.*

Proof: Let C_L be a code as above. We define the following code:

$$C_K = \{\alpha \in S_n \mid \psi(\alpha) \in C_L\}.$$

Since $\psi(S_n) = \mathbb{Z}_n! \subseteq \mathbb{Z}_q^{n-1}$, and by Corollary 14, we have that the minimal distance of C_K is at least d. Furthermore, since ψ is bijective by Lemma 12, the size of the code C_K is exactly $|C_L \cap \mathbb{Z}_n!|$. ■

We now present an explicit construction for a family of CCRM codes that can correct one adjacent-transposition error. The code is based on a perfect code in the Lee-metric space by Golomb and Welch [7].

Construction 1. *Let C_L be the perfect 1-error-correcting code in the Lee metric of length $n-1$ over the alphabet $\mathbb{Z}_{2n-1}$ defined by (see [7]):*

$$C_L = \left\{ v \in \mathbb{Z}_{2n-1}^{n-1} \;\middle|\; \sum_{i=1}^{n-1} i \cdot v_i \equiv 0 \pmod{2n-1} \right\}.$$

The code C_L forms a linear subspace over $\mathbb{Z}_{2n-1}^{n-1}$ and since it is perfect, its $2n-1$ cosets (where $2n-1$ is the index of C_L in $\mathbb{Z}_{2n-1}^{n-1}$) partition the space.

The code C_K is constructed as in Theorem 15 from the coset of C_L that has the largest intersection with $\mathbb{Z}_n!$. The resulting code C_K is an $(n, M, 3)$-CCRM with size $M \geqslant \frac{n!}{2n-1}$.

We observe that the code resulting from Construction 1 is at least half the size of the upper bound of the ball-packing bound (see Section IV, Theorem 17). This is because a ball of radius 1 in $\mathcal{K}_n$ is of size n, and so the upper bound on the size of any $(n, M, 3)$-CCRM code is $M \leqslant \frac{n!}{n}$.

Checking which of the $2n-1$ cosets of C_L from Construction 1 has the largest intersection with $\mathbb{Z}_n!$ may be a difficult task. We can reduce the number codes to check at the cost of a lower size guarantee, as is shown in the next construction.

Construction 2. *Let C_L be defined as in Construction 1 and define also*

$$C_L' = \left\{ v \in \mathbb{Z}_{2n-1}^{n-1} \;\middle|\; v_{n-1} + \sum_{i=1}^{n-1} i \cdot v_i \equiv 0 \pmod{2n-1} \right\}.$$

Construct the code C_K as in Theorem 15 from either C_L or C'_L (whichever has the larger intersection with $\mathbb{Z}_n!$).

Theorem 16. *The code C_K from Construction 2 is an $(n, M, 3)$-CCRM of size $M \geqslant \frac{n!}{2n}$.*

Proof: We first note that C_L has minimal distance 3. We further note that

$$v_{n-1} + \sum_{i=1}^{n-1} i \cdot v_i \equiv \sum_{i=1}^{n-2} i \cdot v_i - (n-1)v_{n-1} \pmod{2n-1}$$

and so the code C'_L is simply a mirror image of C_L along the last dimension. Thus, C'_L also has minimal distance 3, and therefore, by Theorem 15, the constructed code C_K is an $(n, M, 3)$-CCRM code.

To show the lower bound on the size of the code M we note the following: $n-1$ and $2n-1$ are co-prime, and so, for every choice of $0 \leqslant v_i \leqslant i$, $1 \leqslant i \leqslant n-2$, the equations

$$\sum_{i=1}^{n-2} i \cdot v_i + (n-1)v_{n-1} \equiv 0 \pmod{2n-1}$$
$$\sum_{i=1}^{n-2} i \cdot v_i - (n-1)v_{n-1} \equiv 0 \pmod{2n-1}$$

have a unique solution for v_{n-1}. Every solution in which $0 \leqslant v_{n-1} \leqslant n-1$ results in a vector $(v_1, \dots, v_{n-1}) \in C_L \cap \mathbb{Z}_n!$, while every solution in which $0 \leqslant -v_{n-1} \leqslant n-1$ results in a vector $(v_1, \dots, v_{n-1}) \in C'_L \cap \mathbb{Z}_n!$. Since the total number of choices of $0 \leqslant v_i \leqslant i$, $1 \leqslant i \leqslant n-2$ is $(n-1)!$, we have $C_L \cap \mathbb{Z}_n!$ or $C'_L \cap \mathbb{Z}_n!$ at least of size $\frac{n!}{2n}$. ■

IV. Bounds on Code Parameters

In this section we present some bounds on the parameters of CCRM codes. Some of the bounds are direct, while others employ a recursion.

A. Direct Bounds

Following the notation of [11], the number of permutations over n elements with r inversions is denoted by $I_n(r)$, which equals $|\mathcal{S}_r|$, the size of the sphere of radius r (where the parameter n is implicit). An expression for $I_n(r)$ was given in [11]:

$$|\mathcal{S}_r| = I_n(r) = \binom{n+r-1}{r} + \\ + \sum_{j\geqslant 1} (-1)^j \left(\binom{n+r-u_j-1}{r-u_j} + \binom{n+r-u_j-j-1}{r-u_j-j} \right),$$

where $u_j = (3j^2 - j)/2$ is a pentagonal number. Muir [12] has also shown $I_n(r)$ to be the coefficient of x^r in $\prod_{j=1}^{n} \frac{1-x^j}{1-x}$. By our definition, a ball is a union of spheres, i.e., $\mathcal{B}_r(\alpha) = \cup_{i=0}^{r} \mathcal{S}_i(\alpha)$, and since the spheres in the union are certainly disjoint we have

$$|\mathcal{B}_r| = \sum_{i=0}^{r} |\mathcal{S}_i| .$$

We have the following simple ball-packing bound (usually misnamed as a sphere-packing bound):

Theorem 17. *Let C be an (n, M, d)-CCRM code, then*

$$M \leqslant \frac{n!}{\left|\mathcal{B}_{\lfloor (d-1)/2 \rfloor}\right|}.$$

Proof: The space S_n with Kendall's τ-distance is a metric space. Since C is an (n, M, d)-CCRM code, balls of radius $\lfloor (d-1)/2 \rfloor$ centered at the codewords are disjoint, and the claim follows. ■

A similar Gilbert-Varshamov-like bound is the following.

Theorem 18. *Let n, M, and d, be positive integers such that*

$$M \leqslant \frac{n!}{|\mathcal{B}_{d-1}|}$$

then there exists an (n, M, d)-CCRM code.

Proof: Start with the space S_n and arbitrarily choose a codeword. Remove the codeword from the space along with the ball of radius $d-1$ centered about it. Repeat the process with the remaining space as long as it is non-empty. The resulting set of codewords are easily seen to form an (n, M, d)-CCRM code, where M is the number of iterations. In addition, we can see that the number of guaranteed iterations is as given in the claim. ■

We also introduce a Singleton-like bound in the following theorem.

Theorem 19. *Let C be an (n, M, d)-CCRM code.*

1) *Let t be the largest integer such that $M > \frac{n!}{(n-t)!}$. If $0 \leqslant t \leqslant n-2$, then $d \leqslant \binom{n-t}{2}$.*
2) *If $M = \frac{n!}{(n-t)!}$ for some integer $2 \leqslant t \leqslant t-2$, then $d \leqslant \binom{n-t}{2} + 1$.*

Proof: Let us write the M codewords of C in an $M \times n$ array, each codeword forming a single row. We now examine the first t columns of the array, which contain the t-prefixes of the permutations. We note that there are at most $t! \cdot \binom{n}{t} = \frac{n!}{(n-t)!}$ possible distinct prefixes.

For the proof of the first claim, since $M > \frac{n!}{(n-t)!}$ there must exist two rows in the array with the same t-prefix. Thus, the distance between the two codewords is generated by the $(n-t)$-suffixes of the codewords, hence, $d \leqslant \binom{n-t}{2}$.

For the proof of the second claim, if $M = \frac{n!}{(n-t)!}$, then either we have two t-prefixes agreeing and $d \leqslant \binom{n-t}{2}$ as in the previous claim, or every possible t-prefix appears exactly once in the first t columns of the array. In that case, we can find two t-prefixes at distance 1 from each other and then $d \leqslant \binom{n-t}{2} + 1$. ■

Codes attaining the bound of Theorem 19 with equality are called *maximum distance separable (MDS)*. A few MDS codes are, for example: The whole space S_n is an MDS $(n, n!, 1)$-CCRM code, and a permutation and its reverse $\{[1, 2, \dots, n], [n, n-1, \dots, 1]\}$ form an MDS $(n, 2, \binom{n}{2})$-CCRM code (see Example 10).

The more interesting example of an MDS code is the analogue of the binary-parity code. Though this code was also given in Example 10, we give it here again using different arguments which allow an extension of the results. It is well known (for example, see [8]) that every permutation can be described as a product of transpositions (not necessarily adjacent ones), and that the parity of the number of transpositions is invariant. Permutations $\pi \in S_n$ with an even (respectively, odd) number of transpositions in their descriptions are called *even permutations* (respectively, *odd permutations*) and their permutation sign is set to $\mathrm{sgn}(\pi) = 1$ (respectively, $\mathrm{sgn}(\pi) = -1$). For any two permutations, $\alpha, \beta \in S_n$, we have $\mathrm{sgn}(\alpha\beta) = \mathrm{sgn}(\alpha)\,\mathrm{sgn}(\beta)$. We also have $\mathrm{sgn}(\iota) = 1$, and therefore, $\mathrm{sgn}(\alpha) = \mathrm{sgn}(\alpha^{-1})$ for all $\alpha \in S_n$.

We now define the code as,

$$C_n^{\mathrm{even}} = \{\alpha \in S_n \mid \mathrm{sgn}(\alpha) = 1\} = A_n,$$

i.e., the code is the alternating group of order n.

Theorem 20. *The code C_n^{even} is an MDS $(n, \frac{n!}{2}, 2)$-CCRM code.*

Proof: The size of the alternating group is known to be $\frac{n!}{2}$. To show that the distance of the code is 2, assume to the contrary that there exist $\alpha, \beta \in C_n^{\mathrm{even}}$ such that $d(\alpha, \beta)$ is odd. Hence, there exists a sequence of $2t + 1$ adjacent transpositions (for some integer t), $\tau_1, \tau_2, \dots, \tau_{2t+1}$, such that

$$\alpha = \tau_1 \tau_2 \cdots \tau_{2t+1} \beta.$$

But then

$$1 = \mathrm{sgn}(\alpha\beta^{-1}) = \mathrm{sgn}(\tau_1\tau_2\cdots\tau_{2t+1}) = -1,$$

a contradiction. Therefore, the distance between any two permutations in C_n^{even} is even, and it is easy to find two permutations at distance exactly 2. ■

The parity of permutations will be used to further generalize this result in Theorem 25.

B. Recursive Bounds and Constructions

Let us denote by $P(n, d)$ the largest integer M such that there exists an (n, M, d)-CCRM code. The next theorem establishes basic monotonicity.

Theorem 21. *For all $n, d \geqslant 1$ we have*

$$P(n+1, d) \geqslant P(n, d),$$
$$P(n, d) \geqslant P(n, d+1).$$

Proof: The first claim is simple since given an (n, M, d)-CCRM code C, we can construct C' in the following way:

$$C' = \left\{\alpha_{(n+1)\uparrow(n+1)} \mid \alpha \in C\right\}.$$

Obviously C' is an $(n+1, M, d)$-CCRM code. The second claim is also trivial since by definition an $(n, M, d+1)$-CCRM code is also an (n, M, d)-CCRM code. ■

Theorem 22. ***(Code Shortening)*** *For all $n, d \geqslant 1$ we have*

$$P(n+1, d) \leqslant (n+1) \cdot P(n, d).$$

Proof: Let C be an $(n+1, d)$-CCRM code of maximal size $P(n+1, d)$. If we look at the last coordinates in the codewords of C, one of the elements from $\{1, \dots, n+1\}$ appears at least $P(n+1, d)/(n+1)$ times. Let us denote this element as j. We construct C' in the following way:

$$C' = \left\{\alpha_{\downarrow j} \mid \alpha \in C \wedge \alpha(n+1) = j\right\}.$$

After a suitable relabeling of the elements to the alphabet $\{1, \dots, n\}$, the resulting permutations are from S_n and the distance between them is certainly at least d. Thus, C' is an (n, d)-CCRM code whose size is obviously upper bounded by $P(n, d)$, and the claim follows. ■

Theorem 23. ***(Code Puncturing)*** *For all $n, d \geqslant 1$ we have*

$$P(n+1, d+n) \leqslant \left\lceil \frac{n+1}{d+n} \right\rceil P(n, d).$$

Proof: Let C be an $(n+1, d+n)$-CCRM code of maximal size $P(n+1, d+n)$. Arbitrarily choose an element $j \in \{1, 2, \dots, n+1\}$ and construct the code:

$$C' = \left\{\alpha_{\downarrow j} \mid \alpha \in C\right\}.$$

After a proper relabeling we can assume $C' \subseteq S_n$.

We first note that given $\alpha, \beta \in C$, $\alpha \neq \beta$, we may still get $\alpha_{\downarrow j} = \beta_{\downarrow j}$. This happens if α and β agree on the relative ordering of all the elements except j. Since $d(\alpha, \beta) = d + n$, the position of j in α and β differ by at least $d + n$. Therefore, $|C| \leqslant \lceil (n+1)/(d+n) \rceil |C'|$.

Finally, we claim deleting element j from all the permutations results in the minimal distance dropping by no more than n. This is easily seen by noting that (1) implies a single element can cause at most n inversions. ■

Theorem 24. ***(Code Lengthening)*** *For all $n, d \geqslant 1$ we have*

$$P(n+1, d) \geqslant \left\lceil \frac{n+1}{d} \right\rceil P(n, d).$$

Proof: Let C be an (n, d)-CCRM code of size $P(n, d)$. We construct the following code:

$$C' = \left\{\alpha_{i\uparrow(n+1)} \mid \alpha \in C \wedge i \equiv 1 \pmod{d}\right\}.$$

The size of C' is easily seen to be $\lceil (n+1)/d \rceil P(n, d)$.

We now claim that C' is an $(n+1, d)$-CCRM code. To prove this claim we examine two cases. In the first case, for any $\alpha \in C$, and $i_1 \neq i_2$, but $i_1 \equiv i_2 \pmod{d}$, it is obvious that $d(\alpha_{i_1\uparrow(n+1)}, \alpha_{i_2\uparrow(n+1)}) \geqslant d$ since to get from one to the other we need to move the element $n+1$ by at least d positions. In the second case, if $\alpha, \beta \in C$, $\alpha \neq \beta$, we have by definition $d(\alpha, \beta) \geqslant d$ and then also $d(\alpha_{i\uparrow(n+1)}, \beta_{j\uparrow(n+1)}) \geqslant d$ since the relative positions of the elements $\{1, 2, \dots, n\}$ do not change when inserting the element $n+1$ and so the number of inversions remains at least d between the two new permutations. ■

Theorem 25. ***(Code Extending)*** *For all $n, \delta \geqslant 1$ we have*

$$P(n+1, 2\delta) \geqslant \left\lceil \frac{n}{2\delta} \right\rceil P(n, 2\delta - 1).$$

Furthermore, if there exists an $(n, 2\delta - 1)$-CCRM code of size $P(n, 2\delta - 1)$ with M_{e} even codewords and M_{o} odd codewords then

$$P(n+1, 2\delta) \geqslant \left\lceil \frac{n+1}{2\delta} \right\rceil M_{\mathrm{e}} + \left\lceil \frac{n}{2\delta} \right\rceil M_{\mathrm{o}}.$$

Proof: The first claim is a weaker form of the second claim by assuming that $M_o = P(n, 2\delta - 1)$. We will therefore prove just the second claim. Let C be an $(n, 2\delta - 1)$-CCRM code of size $P(n, 2\delta - 1)$ with M_e even codewords which we denote C_e, and M_o odd codewords which we denote C_o.

We now construct the following code:

$$C' = \left\{ \alpha_{i\uparrow(n+1)} \mid \alpha \in C_e \wedge i \equiv n+1 \pmod{2\delta} \right\} \cup \left\{ \alpha_{i\uparrow(n+1)} \mid \alpha \in C_o \wedge i \equiv n \pmod{2\delta} \right\}.$$

The size of C' is easily seen to agree with the claim. The same line of reasoning as in the proof of Theorem 24 guarantees that the minimal distance between codewords of C' is at least $2\delta - 1$. It now suffices to show that all the codewords of C' are even permutations for then, like in the proof of Theorem 20, the distance between codewords of C' is also even, forcing it to be at least 2δ.

For all $\alpha \in S_n$ we must have $\operatorname{sgn}(\alpha_{(n+1)\uparrow(n+1)}) = \operatorname{sgn}(\alpha)$. Therefore, for all $\alpha \in C_e$ we have $\operatorname{sgn}(\alpha_{(n+1)\uparrow(n+1)}) = 1$ and then also $\operatorname{sgn}(\alpha_{i\uparrow(n+1)}) = 1$ for all $i \equiv n+1 \pmod{2\delta}$ since these are an even number of transpositions away from the even permutation $\alpha_{(n+1)\uparrow(n+1)}$. Similarly, for all $\alpha \in C_o$ we have $\operatorname{sgn}(\alpha_{n\uparrow(n+1)}) = 1$ since this is a single transposition away from an odd permutation $\alpha_{(n+1)\uparrow(n+1)}$. In addition, $\operatorname{sgn}(\alpha_{i\uparrow(n+1)}) = 1$ for all $i \equiv n \pmod{2\delta}$, which completes the proof. ■

We note that extending the MDS $(n, n!, 1)$-CCRM code S_n, results in the MDS $(n+1, \frac{(n+1)!}{2}, 2)$-CCRM code C_{n+1}^{even}.

Theorem 26. *For all $n, \delta \geqslant 1$ we have*

$$P(n, 2\delta) \geqslant \frac{1}{2} P(n, 2\delta - 1).$$

Proof: Let C be an $(n, 2\delta - 1)$-CCRM code of size $P(n, 2\delta - 1)$, and let C_o (respectively, C_e) denote the set of odd (respectively, even) codewords. Either C_o or C_e contain at least half the codewords of C. Assume w.l.o.g. that it is C_o. Since all the codewords in C_o have the same parity the distance between any two of them must be even, just like in the proof of Theorem 20. Thus, C_o is an $(n, 2\delta)$-CCRM code of size at least $\frac{1}{2} P(n, 2\delta - 1)$. ■

Again, we note that using the MDS $(n, n!, 1)$-CCRM code S_n with Theorem 26, results in the MDS $(n, \frac{n!}{2}, 2)$-CCRM code C_n^{even}.

V. Conclusion

In this paper, we explored error-correcting for charge-constrained errors (CCRM codes) in the rank-modulation scheme proposed in [9]. The rank-modulation scheme uses a new tool – the permutation induced by cell charge level – to represent data. Consequently, new error-correcting techniques suitable for permutations are needed. We have presented both bounds on the size of CCRM codes, and constructions, mainly by using metric-embedding techniques. This enables us to use well-known binary codes with the Hamming metric, as well as q-ary codes with the Lee metric, to produce CCRM codes. Specifically, we presented a family of one-error-correcting codes whose size is within half of the best upper bound.

We conclude with the following results regarding ad-hoc CCRM code constructions. It is easily seen that $P(3,3) = 2$ with the code $\{[1,2,3],[3,2,1]\}$ which is also constructed by Construction 1. We can also prove that $P(4,3) = 5$ with, for example, the code

$$\{[1,2,3,4],[4,1,3,2],[4,2,3,1],[3,1,4,2],[3,2,4,1]\},$$

which is not constructed through Construction 1. Furthermore, using ad-hoc constructions we can show that

$$\begin{array}{lll} P(5,3) \geqslant 18 & P(6,3) \geqslant 90 & P(7,3) \geqslant 526 \\ P(5,5) \geqslant 6 & P(6,5) \geqslant 23 & P(7,5) \geqslant 110 \\ P(5,7) \geqslant 2 & P(6,7) \geqslant 10 & P(7,7) \geqslant 34 \\ P(5,9) \geqslant 2 & P(6,9) \geqslant 4 & P(7,9) \geqslant 14 \end{array}$$

It is interesting to note that these codes are at least half the ball-packing upper bound of Theorem 17.

References

[1] R. Ahlswede, H. K. Aydinian, and L. H. Khachatrian, "On perfect codes and related concepts," *Designs, Codes and Cryptography*, vol. 22, no. 3, pp. 221–237, Jan. 2001.

[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 *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2005, pp. 2148–2151.

[3] A. E. Brouwer, A. M. Cohen, and A. Neumaier, *Distance-Regular Graphs*. Springer-Verlag, New York, 1989.

[4] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash memories*. Kluwer Academic Publishers, 1999.

[5] P. Delsarte, "An algebraic approach to association schemes of coding theory," *Philips J. Res.*, vol. 10, pp. 1–97, 1973.

[6] M. Deza and H. Huang, "Metrics on permutations, a survey," *J. Comb. Inf. Sys. Sci.*, vol. 23, pp. 173–185, 1998.

[7] S. W. Golomb and L. R. Welch, "Perfect codes in the Lee metric and the packing of polyominoes," *SIAM J. Appl. Math.*, vol. 18, no. 2, pp. 302–317, Jan. 1970.

[8] M. Hall, Jr., *Theory of Groups*. American Mathematical Society, 1999.

[9] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. on Inform. Theory*, Jun. 2009.

[10] M. Kendall and J. D. Gibbons, *Rank correlation methods*. Oxford University Press, NY, 1990.

[11] D. E. Knuth, *The Art of Computer Programming Volume 3: Sorting and Searching*. Addison Wesley, Reading, MA, 2nd ed., 1998.

[12] T. Muir, "On a simple term of a determinant," *Proc. Royal Society Edinburg*, vol. 21, pp. 441–477, 1898.

[13] J. Quistorff, "A survey on packing and covering problems in the Hamming permutations space," *Elec. J. of Comb.*, vol. 13, 2006.

[14] M. Schwartz and T. Etzion, "Codes and anticodes in the Grassman graph," *J. Combin. Theory Ser. A*, vol. 97, no. 1, pp. 27–42, Jan. 2002.