Multipermutation Codes in the Ulam Metric for Nonvolatile Memories

Farzad Farnoud (Hassanzadeh) and Olgica Milenkovic
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

***Abstract*—We address the problem of multipermutation code design in the Ulam metric for novel storage applications. Multipermutation codes are suitable for flash memory where cell charges may share the same rank. Changes in the charges of cells manifest themselves as errors whose effects on the retrieved signal may be measured via the Ulam distance. As part of our analysis, we study multipermutation codes in the Hamming metric, known as constant composition codes. We then present bounds on the size of multipermutation codes and their capacity, for both the Ulam and the Hamming metrics. Finally, we present constructions and accompanying decoders for multipermutation codes in the Ulam metric.**

I. Introduction

Permutations and multipermutations as information representation formats have a long history, with early applications in communication theory dating back to the work of Slepian [1], who proposed using multipermutation codes for transmission in the presence of additive white Gaussian noise. More recently, Vinck proposed using permutation codes in the Hamming metric for combatting impulse noise and permanent frequency noise in power grids [2]. Permutation codes have received renewed interest in the past few years due to their promising application in storage systems, such as flash memories [3]–[5].

Flash memories are nonvolatile storage units (i.e., storage units that remain operational when unpowered), and are usually used for archival or long-term storage. Information is organized in blocks of cells, all of which have to be processed jointly during information erasure cycles. The gist of the approach underlying permutation coding in flash memories, which uses the fact that the memories consist of specially organized cells storing charges, is that information is represented via the relative order of charge levels of cells rather than their absolute charge levels [3]. This approach, termed *rank modulation*, alleviates the problems of cell over-injection, reduces the need for block erasures, and is more robust to errors caused by charge leakage [3]. For instance, while *all* absolute values are subject to errors caused by charge leakage, the relative ordering of the quantitative data may remain largely unchanged [6]. The modeling assumption behind rank modulation is that only errors swapping adjacently ranked cell charges are likely [4], [6]. As a result, code design for flash memories was mainly performed in the domain of the Kendall τ metric, which accounts for small magnitude errors causing swaps of adjacent elements. A thorough treatment of codes in the Kendall metric may be found in [4] and references therein.

In contrast, a more general error model was proposed by the authors in [7], based on the observation that increasing the number of charge levels in order to increase capacity decreases the difference between adjacent charge levels and thus unwanted variations in the charge of a cell may cause its rank to rise above or fall below the ranks of several other cells instead of only swapping two adjacent ranks. In addition, the proposed *translocation error* model adequately accounts for more general types of error such as read-disturb and write-disturb errors. In this context, the distance measure of interest is the Ulam distance, related to the length of the longest common subsequence of two permutations and consequently, the deletion/insertion or edit distance [8]. The Ulam distance has also received independent interest in the bioinformatics and the computer science communities for the purpose of measuring the “sortedness” of data [9]. Other metrics used for permutation code construction include the Hamming distance [2], [10] and the Chebyshev distance (the ℓ_∞ metric) [11], [12].

Multipermutation codes are a generalization of permutation codes where each message is encoded as a permutation of the elements of a multiset. Multipermutation codes in the Hamming metric, known as *constant composition codes* or *frequency permutation array*s (FPAs), were studied in several papers including [13]–[16]. For nonvolatile memories, multipermutation coding was proposed by En Gad et al. [17], as well as by Shieh and Tsai [18]. These works were motivated by different considerations – the former aiming to increase the number of possible re-writes between block erasures, and the latter focusing on the advantages of multipermutation coding with respect to cell leakage, over-injection issues, and charge fluctuations. In addition, multipermutation codes were also recently reported for the Chebyshev distance in [18], [19] and for the Kendall τ distance in [20], [21].

Here, we continue our study of codes in the Ulam metric for nonvolatile memories by extending it to the level of multipermutation codes. Our results include bounds on the size of the largest multipermutation codes, code constructions using multipermutation codes in the Hamming metric and interleaving as well as permutation codes in the Ulam metric [7], [22]. In the process of analyzing these schemes, we establish new connections between resolvable balanced incomplete block designs (RBIBDs) [23], [24], semi-Latin squares [25], and multipermutation codes in the Ulam metric. As multipermutation codes in the Hamming metric are used in our constructions, we also provide new bounds on the size of these codes, and find their asymptotic capacity. In addition, our results include simple decoding schemes for the proposed constructions based on designs and those based on interleaving permutation codes in the Ulam metric.

The paper is organized as follows. In Section II, we present the notation used throughout the paper as well as formal definitions regarding multipermutation codes. In addition, this section includes motivating examples for our work. Section III is devoted to bounds on the size of multipermutation codes in the Ulam and Hamming metrics, as well as to the computation of the asymptotic capacity of these codes. Section IV provides constructions for codes in the Ulam metric. We conclude the paper in Section V with a summary of our results and a number of remarks.

II. Preliminaries and Notation

A. Multipermutations, ordered set partitions, and codes

For an integer k, let $[k] = \{1, \ldots, k\}$. Furthermore, let $\mathbb{S}_n$ denote the symmetric group of order $n!$, i.e., the set of permutations of n distinct elements (typically the elements of $[n]$).

A multipermutation is an arrangement of the elements of a multiset. For example, $(2, 1, 2, 3, 1, 2)$ is a multipermutation of $\{1, 1, 2, 2, 2, 3\}$. For a positive integer n and a multiplicity vector $\vec{r} = (r_1, \ldots, r_m)$, such that $n = \sum_{i=1}^{m} r_i$, we use $\mathsf{M}(n, \vec{r})$ to denote the multiset

$$\{\underbrace{1, \ldots, 1}_{r_1}, \underbrace{2, \ldots, 2}_{r_2}, \ldots, \underbrace{m, \ldots, m}_{r_m}\}.$$

A multiset that has r copies of each of its elements is termed an *r-regular multiset*. For brevity, we henceforth denote the r-regular multiset $\mathsf{M}(n, (r, \cdots, r))$ by $\mathsf{M}(n, r)$. An *r-regular multipermutation* is a permutation of an r-regular multiset. Throughout the paper, we focus on r-regular multipermutations. Many of the subsequently described results, however, can easily be extended to multipermutations of $\mathsf{M}(n, \vec{r})$ for general multiplicity vectors $\vec{r}$.

Let n also denote the number of cells in a block of a flash memory. We assume that r is a positive integer that divides n. Consider a permutation $\pi \in \mathbb{S}_n$ that lists the cells in decreasing order of charge. For example, $\pi = (3, 2, 4, 1)$ means that cell 3 has the highest charge, cell 2 has the second-highest charge, and so on. The inverse of π is the vector of the ranks of the cells, $\pi^{-1} = (4, 2, 1, 3)$; cell 1 has rank 4, cell 2 has rank 2, and so on.

To obtain an r-regular multipermutation of cell rankings, instead of assigning rank i to the element in position i in π, we assign rank i to all the elements in positions $\{(i-1)r + 1, \ldots, ir\}$. This multipermutation is denoted by $\mathfrak{m}_\pi^r$, where

$$\mathfrak{m}_\pi^r(j) = i, \text{ iff } (i-1)r + 1 \leq \pi^{-1}(j) \leq ir.$$

For instance, given $n = 4, r = 2$, and $\pi = (3, 2, 4, 1)$, we have $\mathfrak{m}_\pi^2 = (2, 1, 1, 2)$.

Observe that for $\pi \in \mathbb{S}_n$, $\mathfrak{m}_\pi^r$ is a multipermutation of $\mathsf{M}(n, r)$ and that for $r = 1$, the multipermutation $\mathfrak{m}_\pi^r$ reduces to the inverse of π, i.e., $\mathfrak{m}_\pi^1 = \pi^{-1}$.

A multipermutation $\mathfrak{m}$ of $\mathsf{M}(n, r)$ can also be represented as an ordered set partition $\mathfrak{o}$, where the ith part of $\mathfrak{o}$ is the set

$$\mathfrak{o}(i) = \{j : \mathfrak{m}(j) = i\}.$$

This definition can naturally be extended to multipermutations of other multisets. For $\pi \in \mathbb{S}_n$, let $\mathfrak{o}_\pi^r$ be an ordered set partition where

$$\mathfrak{o}_\pi^r(i) = \{j : \mathfrak{m}_\pi^r(j) = i\}.$$

For the aforementioned example $\pi = (3, 2, 4, 1)$, we have $\mathfrak{o}_\pi^2 = (\{2, 3\}, \{1, 4\})$.

For $\pi, \sigma \in \mathbb{S}_n$, we write $\pi \equiv_r \sigma$ if $\mathfrak{m}_\pi^r = \mathfrak{m}_\sigma^r$, and $\pi \not\equiv_r \sigma$ otherwise. It is easy to show that $\equiv_r$ is an equivalence relation. The equivalence class of permutations including π is denoted by $\mathsf{R}_r(\pi)$, i.e.,

$$\mathsf{R}_r(\pi) = \{\sigma : \sigma \equiv_r \pi\}.$$

As an illustration, the equivalence class of $(3, 2, 4, 1)$ under $\equiv_2$ equals

$$\begin{aligned} &\mathsf{R}_2((3, 2, 4, 1)) = \\ &\qquad \{(3, 2, 4, 1), (2, 3, 4, 1), (3, 2, 1, 4), (2, 3, 1, 4)\}. \end{aligned} \tag{1}$$

In this case, the set $\mathsf{R}_2((3, 2, 4, 1))$ is isomorphic to the subgroup $\mathbb{S}_2 \times \mathbb{S}_2$ of $\mathbb{S}_4$.

Let S be a set of size n. An *r-regular multipermutation code* $\mathsf{MPC}(n, r)$ over S is a code C whose codewords are permutations of S with the property that for any $\pi \in C$, $\mathsf{R}_r(\pi) \subseteq C$. For example,

$$\begin{aligned} &\{(2, 3, 1, 4), (3, 2, 1, 4), (2, 3, 4, 1), (3, 2, 4, 1), \\ &\qquad (1, 3, 2, 4), (3, 1, 2, 4), (1, 3, 4, 2), (3, 1, 4, 2))\} \end{aligned} \tag{2}$$

is an $\mathsf{MPC}(4, 2)$ code. We typically assume that $S = [n]$, but the results hold for any set S of size n.

Each permutation in C represents an ordering of cell charges. For example $\pi = (1, 3, 2, 4)$ indicates that the cell 1 has the highest charge, followed by cell 3, and so on. In multipermutation coding, each r cells are assigned the same rank and all permutations corresponding to the same multipermutation encode the same information. In the previous example, the multipermutation $(2, 1, 1, 2)$ may be represented by any of the permutations on the right side of (1).

As a result, it is clear that an $\mathsf{MPC}(n, r)$ code C can be represented as a set of multipermutations $\mathfrak{M}_r(C)$, where

$$\mathfrak{M}_r(C) = \{\mathfrak{m}_\pi^r : \pi \in C\},$$

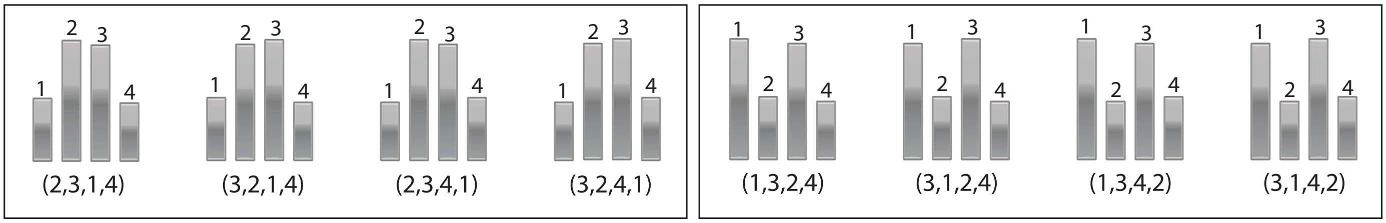
or as a set of ordered set partitions $\mathfrak{O}_r(C)$, where

$$\mathfrak{O}_r(C) = \{\mathfrak{o}_\pi^r : \pi \in C\}.$$

For example, if C is the code given in (2), we have

$$\begin{aligned} \mathfrak{M}_2(C) &= \{(2, 1, 1, 2), (1, 2, 1, 2))\}, \\ \mathfrak{O}_2(C) &= \{(\{2, 3\}, \{1, 4\}), (\{1, 3\}, \{2, 4\})\}. \end{aligned}$$

With slight abuse of notation, for an $\mathsf{MPC}(n, r)$ C, we may use C to mean $\mathfrak{M}_r(C)$ or $\mathfrak{O}_r(C)$ if doing so does not lead to ambiguity. Similarly, we may consider C to be a set of



$\mathfrak{m} = (2, 1, 1, 2)$, $\mathfrak{o} = (\{2, 3\}, \{1, 4\})$

$\mathfrak{m} = (1, 2, 1, 2)$, $\mathfrak{o} = (\{1, 3\}, \{2, 4\})$

Figure 1: The two equivalence classes of the code given in (2). The numbers on the top of the bars indicate cell indices, while the heights of the bars represent the charge levels of the indicated cells. The equivalence classes on the left and right correspond to the multipermutations $\mathfrak{m} = (2, 1, 1, 2)$ and $\mathfrak{m} = (1, 2, 1, 2)$, respectively. Note that each multipermutation can be programmed into the memory as four different permutations, each representing a complete ranking of cell charges without ties.

multipermutations or a set of ordered set partitions instead of a set of permutations.

The *cardinality* or the *size* of C, denoted by $|C|$, equals the number of multipermutations in $\mathfrak{M}_r(C)$, or equivalently, the number of equivalence class of C under the relation $\equiv_r$.

In what follows, we describe why multipermutation formats are suitable for flash memory coding applications. We start with the readback process. To be able to read the information stored in a flash memory, cells with different ranks must have charge levels that differ by at least a certain amount Δ, since if the difference between charge levels of two cells is too small, it cannot be reliably decided which one had the higher charge level. Hence, in permutation coding, to store a permutation of length n, the range of possible charge values must be at least $n\Delta$ to allow for n different charge levels corresponding to n different ranks. In contrast, an r-regular multipermutation of length n has only n/r ranks and thus it can be stored in a flash memory whose range of possible charge level values is $n\Delta/r$. Specifically, the relative order of charge levels of cells of the same rank of a multipermutation is irrelevant as all possibilities correspond to the same multipermutation, i.e., the same information message.

Note that in order to store information represented by r-regular multipermutations, charges are injected to achieve a desired multipermutation ranking. As it is neither necessary nor possible for cells of the same rank to have precisely the same charge levels, the actual representation of such a multipermutation $\mathfrak{m}$ is some permutation π, such that $\mathfrak{m}_\pi^r = \mathfrak{m}$. The multipermutation is available to the user retrieving information in the form of the cell charge ordering π. As an illustration, consider Figure 1 for the code given in (2). For instance, to store the multipermutation $(2, 1, 1, 2)$, any of the permutations given in (1) may be programmed into the memory. To retrieve the information, the user reads the permutation, or possibly an erroneous copy of it, and performs error correction to identify the multipermutation corresponding to the stored permutation.

Next, we show how multipermutations can achieve a higher information rate compared to permutations. Consider a flash memory that can accommodate m sufficiently spread charge levels. In a group of m cells of such a device, one can store a permutation of length m. Suppose r is a positive integer. It follows that in mr cells, the number of possible messages that can be stored is $(m!)^r$.

On the same device, one can store an r-regular multipermutation of length mr in mr cells. In this case, the number of possible messages equals to the number of possible multipermutations, i.e., $\frac{(mr)!}{(r!)^m}$. It is clear that for $r \geq 2$, we have

$$\frac{(mr)!}{(r!)^m} > (m!)^r .$$

Hence, in this setting, more information messages can be stored if one uses multipermutations instead of permutations.

As an illustration, suppose that $m = 2$ and $r = 10$. Using multipermutations, we can store

$$\lg \frac{(mr)!}{(r!)^m} \approx 17.5 \text{ bits}$$

while using permutations, we can store

$$\lg (m!)^r = 10 \text{ bits}$$

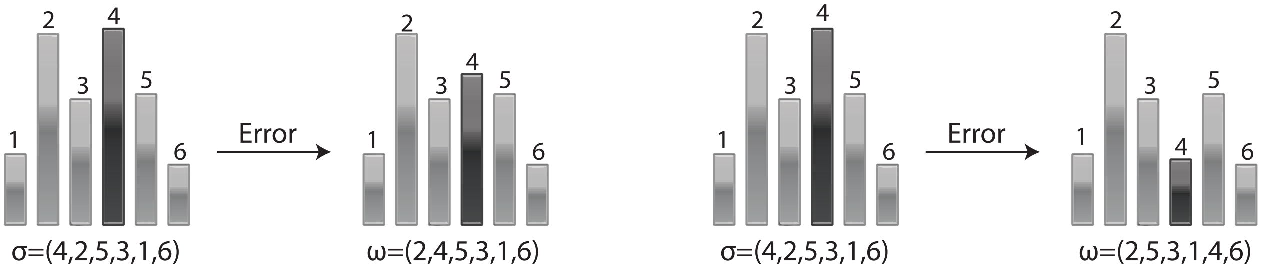
in 20 cells. Note that here we considered the uncoded regime.

The saving in the number of possible charge levels can also be used to increase the number of possible re-writes before a block erasure becomes necessary [17]. As an example, suppose that 5 charge levels are available. If one uses permutations of length 5, it is only possible to write once before an erasure becomes necessary, and if one uses permutations of length 3, it is possible to write twice before an erasure. Encoding with r-regular multipermutations of length $3r$ also provides the ability to write twice before an erasure. While both methods, permutation coding and multipermutation coding, allow for writing twice before an erasure, using multipermutations leads to a higher information storage rate. For further details on multipermutation re-write codes, we refer the reader to [17].

Before proceeding with an analytical treatment of multipermutation codes in the Ulam metric, we remark that throughout the paper, we use $\mathbb{Z}^+$ to denote the set of positive integers. Whenever it is clear from the context, we use the well-known result that $\ln x! = x \ln x + O(x)$, for any nonnegative real value x. By convention, we adopt $0 \ln 0 = 0$.

B. The Multipermutation Hamming distance

For an integer r, the *r-regular Hamming distance* (or simply the Hamming distance) d_H^r between two permutations $\pi, \sigma \in$



(a) $\mathfrak{m}_{\sigma}^{2} = \mathfrak{m}_{\omega}^{2} = (3,1,2,1,2,3)$.

(b) $\mathfrak{m}_{\sigma}^{2} = (3,1,2,1,2,3)$, $\mathfrak{m}_{\omega}^{2} = (2,1,2,3,1,3)$.

Figure 2: Examples of errors: A small-magnitude charge drop error (a) manifests itself as a swap of adjacent ranks in the permutation while a large-magnitude error (b) manifests itself as a translocation in the permutation. In multipermutation coding, charge fluctuations may or may not lead to erroneous multipermutations. Assuming $r = 2$, in (a) the multipermutation does not change, while in (b) it does.

$\mathbb{S}_n$ is defined as

$$\mathsf{d}_H^r(\pi,\sigma) = \left|\{i : \mathfrak{m}_{\pi}^r(i) \neq \mathfrak{m}_{\sigma}^r(i)\}\right| .$$

In words, the permutations are first converted into multipermutations, which are subsequently compared coordinate-wise. The distance $\mathsf{d}_H^r(\pi,\sigma)$ is equivalent to the ordinary Hamming distance between π and σ. Thus, instead of d_H^1, we write d_H.

We observe that

$$\begin{aligned} \mathsf{d}_H^r(\pi,\sigma) &= \sum_{i=1}^{n/r} \left(r - \left|\mathfrak{o}_{\pi}^r(i) \cap \mathfrak{o}_{\sigma}^r(i)\right|\right) \\ &= \sum_{i=1}^{n/r} \left(\left|\mathfrak{o}_{\pi}^r(i)\right| - \left|\mathfrak{o}_{\pi}^r(i) \cap \mathfrak{o}_{\sigma}^r(i)\right|\right) \\ &= \sum_{i=1}^{n/r} \left|\mathfrak{o}_{\pi}^r(i) \backslash \mathfrak{o}_{\sigma}^r(i)\right| . \end{aligned}$$

Furthermore,

$$\mathsf{d}_H^r(\pi,\sigma) = \min_{\pi' \in R(\pi)} \min_{\sigma' \in R(\sigma)} \mathsf{d}_H(\pi',\sigma').$$

Let C be an $\mathsf{MPC}(n,r)$ code. The code C has minimum Hamming distance d if for all $\pi,\sigma \in C$ with $\pi \not\equiv_r \sigma$, we have $\mathsf{d}_H^r(\pi,\sigma) \geq d$. Equivalently, since for all $\pi \in C$, $\mathsf{R}_r(\pi)$ is contained in C, the code C has minimum Hamming distance d if for all $\pi,\sigma \in C$ with $\pi \not\equiv_r \sigma$, it holds that $\mathsf{d}_H(\pi,\sigma) \geq d$. An $\mathsf{MPC}(n,r)$ code with minimum Hamming distance d is said to be an $\mathsf{MPC}_H(n,r,d)$ code. The code given in (2) is an $\mathsf{MPC}_H(4,2,2)$ code.

C. Translocation errors and the Ulam distance

Figure 2 illustrates examples of errors in flash memories that may occur due to charge leakage, read-disturb, and write-disturb [26]. While errors with small magnitude represent swaps of adjacent ranks, errors with large magnitude represent *translocations* [7].

A translocation $\phi(i,j)$ is a permutation that is obtained from the identity permutation e by moving element i to the position of j and shifting all elements between i and j, including j, by one [7]. For example, for $i < j$,

$$\phi(i,j) = (1,\ldots,i-1,i+1,i+2,\ldots,j,i,j+1,\ldots,n).$$

As a convention, we assume that $\phi(i,i) = e$. A *translocation error* is an error that changes a stored permutation π to $\pi\phi(i,j)$, with $i \neq j$.

A *subsequence* of a vector $x = (x(1), x(2), \ldots, x(n))$ is a sequence $(x(i_1), x(i_2), \ldots, x(i_k))$, where $i_1 < i_2 < \cdots < i_k$ and $k \leq n$. A *common subsequence* of two vectors x and y is a sequence that is a subsequence of both x and y. Let the length of the longest common subsequence of two permutations π and σ be denoted by $\mathsf{LCS}(\pi,\sigma)$. The *Ulam distance* $\mathsf{d}_\circ(\pi,\sigma)$ between two permutations π and σ of length n is defined as $n - \mathsf{LCS}(\pi,\sigma)$. It is straightforward to see that the Ulam distance between π and σ equals the minimum number of translocations required to take π to σ [7]. It is also well known that the Ulam distance represents the *edit distance* between two permutations, i.e., the smallest number of insertion/deletion pairs needed to transform one permutation into another.

For $\pi,\sigma \in \mathbb{S}_n$, define the *(r-regular) Ulam distance* $\mathsf{d}_\circ^r$ on permutations as

$$\mathsf{d}_\circ^r(\pi,\sigma) = \min_{\pi' \in R(\pi)} \min_{\sigma' \in R(\sigma)} \mathsf{d}_\circ(\pi',\sigma').$$

Note that this distance is a set-distance: it measures the smallest Ulam distance between two permutations in different equivalence classes. Furthermore, the distance $\mathsf{d}_\circ^r(\pi,\sigma)$ equals the minimum number of translocations required to take a permutation in $\mathsf{R}_r(\pi)$ to a permutation in $\mathsf{R}_r(\sigma)$.

Let $\mathsf{U}_r^*(\pi,\sigma)$ denote the set

$$\{(\alpha,\beta) : \alpha \in \mathsf{R}_r(\pi), \beta \in \mathsf{R}_r(\sigma), \mathsf{d}_\circ^r(\pi,\sigma) = \mathsf{d}_\circ(\alpha,\beta)\}.$$

By definition of $\mathsf{d}_\circ^r(\pi,\sigma)$, $\mathsf{U}_r^*(\pi,\sigma)$ is nonempty.

An $\mathsf{MPC}(n,r)$ C has minimum Ulam distance d if for all $\pi,\sigma \in C$ with $\pi \not\equiv_r \sigma$, we have $\mathsf{d}_\circ^r(\pi,\sigma) \geq d$. Such a code is denoted by $\mathsf{MPC}_\circ(n,r,d)$. The code given in (2) is an $\mathsf{MPC}_\circ(4,2,1)$ code, as $\mathsf{d}_\circ((3,2,1,4),(3,1,2,4)) = 1$.

Under minimum distance decoding, an $\mathsf{MPC}_\circ(n,r,d)$ code can correct t translocation errors iff $d \geq 2t+1$. To see this, note that $d < 2t+1$ iff there exists $\pi,\sigma \in C, \pi \not\equiv_r \sigma$, and $\omega \in \mathbb{S}_n$ such that $\mathsf{d}_\circ(\omega,\pi) \leq t$ and $\mathsf{d}_\circ(\omega,\sigma) \leq t$, iff C cannot correct t errors.

For a set P and a permutation π, let π_P denote the projection of π onto P, that is, the sequence obtained by only keeping those elements of π that are in P. We find the following lemma, proved in our companion paper [7], useful in our subsequent derivations.

Lemma 1. *For sets $P \subseteq [n]$ and $Q = [n] \backslash P$, and for permutations $\pi, \sigma \in \mathbb{S}_n$, we have*

$$\mathsf{d}_\circ(\pi,\sigma) \geq \mathsf{d}_\circ(\pi_P,\sigma_P) + \mathsf{d}_\circ(\pi_Q,\sigma_Q).$$

D. Relationship between the Ulam and the Hamming metrics

The following lemma is an immediate consequence of the definition of a translocation.

Lemma 2. *A translocation, applied to a permutation, changes at most one element of each rank. That is, for a translocation φ, a permutation π, and $i \in [n/r]$,*

$$\left|\mathfrak{o}_\pi^r(i) \cap \mathfrak{o}_{\pi\varphi}^r(i)\right| \geq r-1.$$

Since there are n/r ranks, we have

$$\mathsf{d}_H^r(\pi,\pi\varphi) \leq \frac{n}{r},$$

for a translocation φ and a permutation π. Hence,

$$\mathsf{d}_H^r(\pi,\sigma) \leq \frac{n}{r}\mathsf{d}_\circ^r(\pi,\sigma)$$

for $\pi, \sigma \in \mathbb{S}_n$.

We next upper bound $\mathsf{d}_\circ^r(\pi,\sigma)$ in terms of $\mathsf{d}_H^r(\pi,\sigma)$. There exist $\pi' \in \mathsf{R}_r(\pi)$, $\sigma' \in \mathsf{R}_r(\sigma)$, and a common subsequence of π' and σ' that contains the elements of

$$\bigcup_{i=1}^{n/r}\left(\mathfrak{o}_\pi^r(i) \cap \mathfrak{o}_\sigma^r(i)\right).$$

Hence,

$$\begin{aligned}\mathsf{d}_\circ^r(\pi,\sigma) &\leq n - \mathsf{LCS}(\pi',\sigma')\\ &\leq n - \sum_{i=1}^{n/r}\left|\mathfrak{o}_\pi^r(i) \cap \mathfrak{o}_\sigma^r(i)\right|\\ &= \sum_{i=1}^{n/r}\left(r - \left|\mathfrak{o}_\pi^r(i) \cap \mathfrak{o}_\sigma^r(i)\right|\right)\\ &= \mathsf{d}_H^r(\pi,\sigma),\end{aligned}$$

implying that $\mathsf{d}_\circ^r(\pi,\sigma) \leq \mathsf{d}_H^r(\pi,\sigma)$.

Lemma 3. *For $\pi, \sigma \in \mathbb{S}_n$, we have*

$$\frac{r}{n}\mathsf{d}_H^r(\pi,\sigma) \leq \mathsf{d}_\circ^r(\pi,\sigma) \leq \mathsf{d}_H^r(\pi,\sigma).$$

The lemma illustrates the fact that for $r = \Theta(n)$, the Ulam distance is within a constant factor of the Hamming distance, while for $r = o(n)$, the Ulam distance may be much smaller. Consequently, while good codes in the Ulam metric allow for substitution error correction, good codes in the Hamming metric provide resilience under translocation errors only for a certain limited range of parameters.

III. Bounds on Size of Multipermutation Codes

In what follows, we derive bounds on the size of multipermutation codes in the Hamming metric as well as the Ulam metric. For the case of the Hamming distance, we find the asymptotic capacity, while for the Ulam distance we provide lower and upper bounds on the capacity. We point out that a number of bounds on multipermutation codes in the Hamming metric were derived in [15], including some simple and some complicated expressions involving Laguerre polynomials. Nevertheless, these bounds do not allow for finding a capacity formula for the underlying codes.

Let $A_H(n,r,d)$ and $A_\circ(n,r,d)$ denote the maximum cardinalities of an $\mathsf{MPC}_H(n,r,d)$ and an $\mathsf{MPC}_\circ(n,r,d)$ code, respectively. Furthermore, let $\mathcal{C}_H(r,d)$ denote the *capacity*, i.e., maximum achievable rate, of multipermutation codes in the Hamming metric, defined as

$$\mathcal{C}_H(r,d) = \lim_{n\to\infty}\frac{\ln A_H(n,r,d)}{\ln n!}.$$

The capacity of multipermutation codes in the Ulam metric, $\mathcal{C}_\circ(r,d)$, is defined similarly.

In the remainder of the paper, limits are evaluated for $n \to \infty$ unless stated otherwise. We assume that all limits of interest exist and we use $\rho = \rho(r) = \lim \frac{\ln r}{\ln n}$, as well as $\delta = \delta(d) = \lim \frac{d}{n}$.

A. Multipermutation Codes in the Hamming Metric

It was shown by Luo et al. [13] that

$$A_H(n,r,d) \leq \frac{d}{r+d-n}, \quad \text{for } r+d > n, \tag{3}$$

and by Huczynska and Mullen [15] that

$$A_H(n,r,d) \leq \frac{n!}{r(d-1)!}. \tag{4}$$

The first bound, (3), implies that the asymptotic rate is zero if $r+d>n$, while the second bound implies that

$$\mathcal{C}_H(r,d) \leq 1-\delta, \tag{5}$$

which also follows from the fact that $\mathcal{C}_H(r,d) \leq \mathcal{C}_H(1,d)$ and Theorem 11 of our companion paper [7], stating that $\mathcal{C}_H(1,d) = 1-\delta$. We improve next upon the bound in (5) and provide a matching lower bound, thereby establishing the capacity of r-regular multipermutation codes in the Hamming metric.

Let $S(l,m,r)$ denote the number of sequences of length l over the alphabet $[m]$, with no element appearing more than r times. Note that $S(l,m,r)$ equals the number of ordered partitions of a set of size l into m sets such that each part has at most r elements, and where empty subsets are allowed.

Lemma 4. *(Singleton bound) For positive integers n, r, d such that r divides n, we have*

$$A_H(n,r,d) \leq \left(\frac{n}{r}\right)^{n-d+1}.$$

Proof: Consider an $\mathsf{MPC}_H(n,r,d)$ code C of size M and let

$$\mathfrak{M} = \mathfrak{M}_r(C) = \{\mathfrak{m}_1,\ldots,\mathfrak{m}_M\}$$

denote its multipermutation representation. Since the minimum Hamming distance d_H^r of C is at least d, for distinct i and j,

$$\sum_{k=1}^{n} \mathbb{I}\left(\mathfrak{m}_i(k) \neq \mathfrak{m}_j(k)\right) \geq d, \tag{6}$$

where the indicator function $\mathbb{I}$ is defined in the standard manner as

$$\mathbb{I}(\text{condition}) = \begin{cases} 1, & \text{if condition is true,} \\ 0, & \text{if condition is false.} \end{cases}$$

By removing the last $d-1$ elements of each multipermutation $\mathfrak{m}_i, i \in [M]$, we obtain the set $\mathfrak{M}' = \{\mathfrak{m}_1', \ldots, \mathfrak{m}_M'\}$ of sequences of length $n-d+1$ over $[n/r]$ where no element appears more than r times.[1]

Since $d-1$ elements are removed, (6) implies that

$$\sum_{k=1}^{n-d+1} \mathbb{I}\left(\mathfrak{m}_i'(k) \neq \mathfrak{m}_j'(k)\right) = \sum_{k=1}^{n-d+1} \mathbb{I}\left(\mathfrak{m}_i(k) \neq \mathfrak{m}_j(k)\right) \geq 1.$$

and thus for distinct $i, j \in [M]$, $\mathfrak{m}_i'$ and $\mathfrak{m}_j'$ are distinct. Hence, we have

$$A_H(n,r,d) \leq S\left(n-d+1, \frac{n}{r}, r\right). \tag{7}$$

Furthermore, since $S(n-d+1, n/r, r) \leq S(n-d+1, n/r, \infty)$, we find

$$A_H(n,r,d) \leq S\left(n-d+1, \frac{n}{r}, \infty\right) = \left(\frac{n}{r}\right)^{n-d+1}.$$

■

As shown in the sequel, the bound given in Lemma 4 is sufficiently tight for capacity derivations. Nevertheless, it may be useful to bound $S(l,m,r)$ more tightly.

It is easy to see that

$$S(l,m,r) = \sum_{\substack{x_1 + \cdots + x_m = l, \\ 0 \leq x_i \leq r}} \frac{l!}{\prod_{i=1}^{m} x_i!}. \tag{8}$$

where the x_i's are integers. The exponential generating function (EGF) of $S(l,m,r)$ is

$$\sum_{l=0}^{\infty} S(l,m,r) \frac{z^l}{l!} = \left(\sum_{i=0}^{r} \frac{z^i}{i!}\right)^m$$

and thus one can write

$$A_H(n,r,d) \leq (n-d+1)! \left[z^{n-d+1}\right] \left(\sum_{i=0}^{r} \frac{z^i}{i!}\right)^{n/r}. \tag{9}$$

The bound given in (9) can be used to find numerical upper bounds on the code size, such as those provided in Table I. In addition, it can be used to obtain simple asymptotic bounds using methods described in the classical text [27, Ch. 8]. As a final note, we point out that the related problems of restricted multisets and restricted integer partitions are far better studied combinatorial entities than the one we addressed above [28, Ch. 21, Sec. 8], although no simple direct connection between these problems and the problem discussed here exist.

Another approach, which is conceptually much simpler and which applies to many other coding-theoretic scenarios is using the Poisson approximation theorem for multinomial variables and the Chernoff bound [29], [30], or alternatively, the Central limit theorem [31]. As shown in [31], [32], the number of terms in the multinomial summation formula, m, may represent the number of labeled urns into which l labeled balls are thrown randomly. The occupancy variables X_i, $i = 1, \ldots, m$, are dependent, since $X_1 + \ldots + X_m = l$. But in the asymptotic central domain regime, with l/m constant, the variables X_i, $i = 1, \ldots, m$, may be viewed as *independent* Poisson variables with mean $\lambda = l/m$. Any result of computations involving independent Poisson variables that satisfies the inversion conditions dictated by Tauberian theorems described in [32] may be *asymptotically converted* into the correct result by simply replacing λ with l/m. Furthermore, the same approach may be used when dealing with urns and balls that satisfy additional constraints [32].

To understand the principles behind the Poisson transform method, we follow the analysis in [31] based on [33]. The key observation is that the Poisson distributions satisfy the additivity (infinite divisibility) property, i.e., the property that the sum of independent Poisson random variables is another Poisson random variable with mean parameter equal to the sum of the parameters of the individual variables in the sum. Then, it is straightforward to show that for two different ball placement processes, the urn occupancy variables have the same distribution: 1) in the first case, each urn receives balls according to a Poisson distribution with parameter λ independently of all other urns; 2) in the second case, balls arrive with a Poisson distribution with parameter λm and are routed with uniform probability $1/m$ to one of the urns.

Assume next that $g(m,\lambda)$ is a quantity of interest where the input to each urn is generated according to model 1). The same quantity under the original urns and balls model with a fixed number l of balls is denoted by $f(m,l)$. Using the equivalence between the two formulations 1) and 2), one can show that

$$g(m,\lambda) = \sum_{l} f(m,l)\, P\left(X_1 + \ldots + X_m = l\right),$$

where $X_1, \ldots, X_m$ are i.i.d Poisson random variables with parameter λ. As a result, it is straightforward to see that $f(m,l) = \frac{l!}{m^l}\left[\lambda^l\right]\left\{e^{\lambda m} g(m,\lambda)\right\}$.

In words, $e^{\lambda m} g(m,\lambda)$ represents the exponential generating function over the number of balls l of $f(m,l)$ evaluated at λm. Evaluating the coefficient in a generating function in the asymptotic domain may be accomplished with the aid of Tauberian theorems (see [32]) or classical asymptotic analysis. In the case of the Poisson transform, provided that some minor technical conditions are met, it can be shown that $f(m,l) \simeq g(m,l/m)$, where $a(x) \simeq b(x)$ stands for $\lim_{x\to\infty} a(x)/b(x) = 1$. Intuitively, the aforementioned result implies that when the dependencies among a large number of random variables are weak – for example, only in terms of a constraint on the total sum of their values – then the

[1] This argument is akin to the approach proposed in [4] for permutation codes in the Kendall metric.

Table I: Bounds on the size 3-regular multipermutation codes in the Hamming metric of length 9.

Upper bound on $A_H(9,3,d)$	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$	$d=9$
(3) [13]	-	-	-	-	-	-	7	4	3
(4) [15]	120960	120960	60480	20160	5040	1008	168	24	3
Lemma 4	19683	6561	2187	729	243	81	27	9	3
(11) (approximate bound)	12077	4560	1700	624	224	79	27	9	3
(9)	1680	1680	1050	510	210	78	27	9	3

variables are asymptotically independent, provided a proper choice of the distribution ensures consistence with the finite-valued parameters.

In the case of interest, we need to find the probability $P\{X_i \leq r,\, i=1,\ldots,m\}$. For $m,l \to \infty$, such that l/m is a constant, and for r fixed, this leads to

$$S(l,m,r) \simeq m^l \left(\sum_{i=0}^{r} \exp(-\lambda)\, \frac{\lambda^i}{i!} \right)^m,$$

where $\lambda = l/m$. The asymptotic formula for $S(l,m,r)$ depends on the relationship between the parameters r,l,m. For $r \leq l/m$, the Chernoff bound reads as

$$\sum_{i=0}^{r} \exp(-\lambda)\, \frac{\lambda^i}{i!} \leq \frac{\exp(-\lambda)(e\,\lambda)^r}{r^r},$$

so that

$$S(l,m,r) \lesssim m^l\, \frac{\exp(-l/m)(e\,l/m)^r}{r^r}. \tag{10}$$

For the case of interest in our derivation, $r > l/m$. Whenever $r > 10$, one may use the straightforward Central Limit Theorem approximation

$$\sum_{i=0}^{r} \exp(-\lambda)\, \frac{\lambda^i}{i!} \simeq \Phi\left(\frac{r+0.5-l/m}{\sqrt{l/m}} \right),$$

so that

$$S(l,m,r) \simeq m^l\, \Phi^m \left(\frac{r+0.5-l/m}{\sqrt{l/m}} \right),$$

where the function $\Phi(\cdot)$ stands for the cumulative distribution function (CDF) of a standard Gaussian random variable. Since $r > \frac{n-d+1}{n/r}$, the preceding relation and (7) imply

$$A_H(n,r,d) \lesssim \left(\frac{n}{r}\right)^{n-d+1} \Phi^{n/r} \left(\frac{n+2(d-1)r}{2\sqrt{n(n-d+1)r}} \right) \tag{11}$$

provided that $\frac{n-d+1}{n/r}$ is a constant larger than 10.

As an example, upper bounds on the size of $\mathsf{MPC}_H(9,3,d)$ are given in Table I. Note that the bounds of (11), Lemma 4, and (9) are very close for small values of d. Indeed, the right side of (11) is bounded above by $(n/r)^{n-d+1}$ and below by

$$\left(\frac{n}{r}\right)^{n-d+1} \left(\frac{1}{2}\right)^{n/r}$$

and we have

$$\lim \frac{\ln\left(\left(\frac{n}{r}\right)^{n-d+1} \left(\frac{1}{2}\right)^{n/r} \right)}{\ln\left(\frac{n}{r}\right)^{n-d+1}} = 1$$

provided that $\delta < 1$ and $r < n$. Therefore, the bounds of (11) and Lemma 4 have the same asymptotic exponent.

The next lemma provides a lower bound on $A_H(n,r,d)$.

Lemma 5. *(Gilbert-Varshamov Bound) We have*

$$A_H(n,r,d) \geq \frac{n!}{(r!)^{n/r} \binom{n}{d-1} \left(\frac{n}{r}\right)^{d-1}}.$$

Proof: There are $\frac{n!}{(r!)^{n/r}}$ multipermutations of $\mathsf{M}(n,r)$. The size of a ball of radius $d-1$ in the space of multipermutations of $\mathsf{M}(n,r)$ endowed with the Hamming distance is bounded above by $\binom{n}{d-1}\left(\frac{n}{r}\right)^{d-1}$ (an exact and complicated expression for the size of the ball may be found in [15]). The Lemma follows by a standard application of Gilbert's argument. ■

Theorem 6. *We have*

$$\mathcal{C}_H(r,d) = (1-\rho)(1-\delta).$$

Proof: First, recall that $\lim$ expressions with no subscripts stand for $\lim_{n\to\infty}$.

On the one hand, from Lemma 4, we have

$$\begin{aligned} \mathcal{C}_H(r,d) &\leq \lim \frac{(n-d+1)(\ln n - \ln r)}{\ln n!} \\ &= \lim \frac{n\ln n - n\ln r - d\ln n + d\ln r}{n\ln n + O(n)} \\ &= 1-\rho-\delta+\rho\delta. \end{aligned}$$

On the other hand, from Lemma 5, we easily see that

$$\begin{aligned} \mathcal{C}_H(r,d) &\geq \lim \frac{\ln\left(n!(r!)^{-n/r} \binom{n}{d-1}^{-1} \left(\frac{n}{r}\right)^{-d+1} \right)}{\ln n!} \\ &= 1 - \lim \frac{(n/r)\ln r! + (d-1)\ln(n/r)}{\ln n!} \\ &= 1 - \lim \frac{n\ln r + d\ln n - d\ln r + O(n)}{n\ln n + O(n)} \\ &= 1-\rho-\delta+\rho\delta, \end{aligned}$$

where we have used the fact that $\lim \frac{\ln\binom{n}{d-1}}{\ln n!} = 0$. This establishes the claimed result for the asymptotic capacity of multipermutation codes in the Hamming metric. ■

B. Multipermutation Codes in the Ulam Metric

Using Lemma 3 which implies that $A_\circ(n,r,d) \leq A_H(n,r,d)$, we find the following upper bound on $A_\circ(n,r,d)$:

$$A_\circ(n,r,d) \leq A_H(n,r,d) \leq S(n-d+1, \frac{n}{r}, r) \leq \left(\frac{n}{r}\right)^{n-d+1}. \tag{12}$$

The next lemma provides a lower bound on $A_\circ(n,r,d)$.

Lemma 7. *(Gilbert-Varshamov Bound) For positive integers n, r, d such that n is a multiple of r, we have*

$$A_\circ(n,r,d) \geq \frac{(n-d+1)!}{\binom{n}{d-1}(r!)^{2n/r}}.$$

Proof: Let $\mathsf{B}_\circ^r(u)$ denote the size of a ball of radius u in $\mathbb{S}_n$ endowed by $\mathsf{d}_\circ^r$ (note that due to symmetry, i.e., left invariance of the Ulam metric, the volume of the ball is independent on the choice of the center). Equivalently, let $\mathsf{B}_\circ^r(u) = \{\pi \in \mathbb{S}_n : \mathsf{d}_\circ^r(\pi, e) \leq u\}$. The Gilbert bound states that

$$A_\circ(n,r,d) \geq \frac{n!}{\mathsf{B}_\circ^r(d-1)}.$$

We show that $\mathsf{B}_\circ^r(d-1) \leq (r!)^{2n/r}\mathsf{B}_\circ^1(d-1)$. The lemma then follows from a result pertaining to the Ulam metric we derived in [7], namely:

$$\mathsf{B}_\circ^1(d-1) \leq \binom{n}{d-1}\frac{n!}{(n-d+1)!}.$$

The set $\{\pi \in \mathbb{S}_n : \mathsf{d}_\circ^r(\pi, e) \leq u\}$ equals

$$\bigcup_{\sigma \in \mathsf{R}_r(e)} \quad \bigcup_{\pi \in \mathbb{S}_n : \mathsf{d}_\circ(\pi,\sigma) \leq u} \quad \mathsf{R}_r(\pi).$$

Hence,

$$\begin{aligned}
\mathsf{B}_\circ^r(u) &= \left|\cup_{\sigma \in \mathsf{R}_r(e)} \cup_{\pi \in \mathbb{S}_n : \mathsf{d}_\circ(\pi,\sigma) \leq u} \mathsf{R}_r(\pi)\right| \\
&\leq (r!)^{n/r} \left|\cup_{\pi \in \mathbb{S}_n : \mathsf{d}_\circ(\pi,e) \leq u} \mathsf{R}_r(\pi)\right| \\
&\leq (r!)^{n/r}\mathsf{B}_\circ^1(u)(r!)^{n/r} \\
&= (r!)^{2n/r}\mathsf{B}_\circ^1(u),
\end{aligned}$$

which shows that $\mathsf{B}_\circ^r(d-1) \leq (r!)^{2n/r}\mathsf{B}_\circ^1(d-1)$. ■

Next, we improve upon Lemma 7 by finding a sharper bound for $\mathsf{B}_\circ^r(u), u \in \mathbb{Z}^+$. Consider the ball around the identity permutation e. For a permutation π that satisfies $\mathsf{d}_\circ^r(\pi, e) \leq u$, there exists a $\pi' \in \mathsf{R}_r(\pi)$ that has a common subsequence s of length $l = n - u$ with some $e' \in \mathsf{R}_r(e)$. There are

$$A = \sum_{\substack{x_1 + \cdots + x_{n/r} = l, \\ x_i \in [0,r], \forall i}} \prod_{i=1}^{n/r} \binom{r}{x_i} x_i!$$

ways of choosing a sequence s of length l such that it is a subsequence of some $e' \in \mathsf{R}_r(e)$, with $\prod_{i=1}^{n/r}\binom{r}{x_i}x_i!$ counting the number of ways one can choose a subsequence of length l with x_i elements from rank i,

$$\mathfrak{o}_{e'}^r(i) = \mathfrak{o}_e^r(i) = \{(i-1)r+1, \ldots, ir\}.$$

The number of ordered partitions $\mathfrak{o}$ with parts of size equal to r, such that there exists a π' that satisfies $\mathfrak{o} = \mathfrak{o}_{\pi'}^r$ and contains s as a subsequence equals

$$B = \sum_{\substack{x_1 + \cdots + x_{n/r} = l, \\ x_i \in [0,r], \forall i}} \binom{n-l}{r-x_1, \ldots, r-x_{n/r}}.$$

Here, the multinomial $\binom{n-l}{r-x_1,\ldots,r-x_{n/r}}$ accounts for the number of ways of choosing the ordered partition $\mathfrak{o}$ such that the first x_1 elements of s are in the first part, the next x_2 elements are in the second part, and so on.

In addition, there are $(r!)^{n/r}$ permutations π such that $\pi \in \mathsf{R}_r(\pi')$. Hence,

$$\mathsf{B}_\circ^r(u) \leq AB(r!)^{n/r}. \tag{13}$$

With regards to bounding the combinatorial sum A, we observe that

$$\begin{aligned}
A &\leq \binom{l+n/r-1}{n/r-1} \max_{\substack{x_1 + \cdots + x_{n/r} = l, \\ x_i \in [0,r], \forall i}} \prod_{i=1}^{n/r}\binom{r}{x_i}x_i! \\
&\leq \binom{l+n/r-1}{n/r-1}\Big(\frac{r!}{(r-\frac{l}{n/r})!}\Big)^{n/r} \\
&\leq \binom{2n}{n}\Big(\frac{r!}{\frac{r(n-l)}{n}!}\Big)^{n/r}.
\end{aligned}$$

Using a similar approach for B, we find

$$B \leq \binom{2n}{n}\frac{(n-l)!}{\Big(\frac{r(n-l)}{n}!\Big)^{n/r}}.$$

Hence,

$$\mathsf{B}_\circ^r(u) \leq \binom{2n}{n}^2\Big(\frac{r!}{\frac{ru}{n}!}\Big)^{2n/r}u!,$$

and so, for $d > 1$,

$$\begin{aligned}
\ln \mathsf{B}_\circ^r(d-1) &\leq \frac{2n}{r}\ln r! - \frac{2n}{r}\ln\frac{r(d-1)}{n}! \\
&\quad + \ln(d-1)! + O(n) \\
&= 2n\ln r - 2(d-1)\ln\frac{r(d-1)}{n} \\
&\quad + (d-1)\ln(d-1) + O(n) \\
&= 2n\ln r - 2d\ln r + 2d\ln n - d\ln d + O(n).
\end{aligned}$$

This implies that

$$\begin{aligned}
\mathcal{C}_\circ(r,d) &\geq 1 - \lim\frac{2n\ln r - 2d\ln r + 2d\ln n - d\ln d + O(n)}{n\ln n + O(n)} \\
&= 1 - 2\rho + 2\delta\rho - \delta \\
&= (1-\delta)(1-2\rho).
\end{aligned} \tag{14}$$

Theorem 8. *The capacity of multipermutation codes in the Ulam metric is bounded according to*

$$(1-\delta)(1-2\rho) \leq \mathcal{C}_\circ(r,d) \leq (1-\delta)(1-\rho).$$

Proof: The lower bound is given in (14) while the upper bound is a result of (12) and Theorem 6. ■

IV. Constructions

In the next subsections, we present several constructions for multipermutation codes in the Ulam metric. One of the key ingredients of our constructions is permutation interleaving, which we proposed for Ulam metric code design in [7]. The related idea of restricting certain positions in the codewords to certain values was first described in [11], [12], while interleaving in the Chebyshev metric was discussed in [18].

For sequences $\pi_1, \ldots, \pi_k$, let $\pi_1 \circ_r \pi_2 \circ_r \cdots \circ_r \pi_k$ denote the sequence obtained by sequentially interleaving blocks of r elements of $\pi_i, i \in [k]$.

For example, $(1,3,4,2) \circ_2 (6,7,8,5) \circ_2 (12,10,9,11) = (1,3,6,7,12,10,4,2,8,5,9,11)$.

This form of interleaving will henceforth be called block interleaving. Whenever $r = 1$, we simply write $\circ$ instead of $\circ_1$.

A. Constructions based on almost disjoint sets

Two sets A and B are said to be *at most k-intersecting,* if for a given positive integer k, one has

$$|A \cap B| \leq k.$$

When k is smaller than the size of the sets $A, B,$ and the aforementioned bound is true, we say that the sets are *almost disjoint*. The next lemma shows how sets of set partitions with almost disjoint parts can be used for constructing multipermutation codes in the Ulam metric.

Lemma 9. *Let C be an $\mathsf{MPC}(n,r)$ code, and suppose that t is a positive integer such that $2t < r$. If for all $\pi, \sigma \in C$ and $i \in [n/r]$, we either have $\mathfrak{o}^r_\pi(i) = \mathfrak{o}^r_\sigma(i)$ or*

$$|\mathfrak{o}^r_\pi(i) \cap \mathfrak{o}^r_\sigma(i)| < r - 2t, \tag{15}$$

then the code C can correct t translocation errors, that is, C is an $\mathsf{MPC}_\circ(n,r,2t+1)$ code.

Proof: Suppose $\pi \in C$ is the (unknown) stored codeword and ω is the retrieved permutation. The Ulam distance between π and ω is at most t since the codeword π is affected by at most t translocation errors. We show that given ω, $\mathfrak{o}^r_\pi$ can be uniquely identified. Fix $i \in [n/r]$. Since there are at most t translocation errors, by Lemma 2, we have

$$|\mathfrak{o}^r_\pi(i) \cap \mathfrak{o}^r_\omega(i)| \geq r - t. \tag{16}$$

To identify $\mathfrak{o}^r_\pi(i)$ uniquely, it suffices to have $|\mathfrak{o}^r_\sigma(i) \cap \mathfrak{o}^r_\omega(i)| < r - t$ for all $\sigma \in C$ such that $\mathfrak{o}^r_\pi(i) \neq \mathfrak{o}^r_\sigma(i)$.

Suppose that $\sigma \in C$ and $\mathfrak{o}^r_\pi(i) \neq \mathfrak{o}^r_\sigma(i)$. We use (15) and (16) to show that $|\mathfrak{o}^r_\sigma(i) \cap \mathfrak{o}^r_\omega(i)| < r - t$. For simplicity, let $B_\pi = \mathfrak{o}^r_\pi(i)$, $B_\sigma = \mathfrak{o}^r_\sigma(i)$, and $B_\omega = \mathfrak{o}^r_\omega(i)$. We then have

$$\begin{aligned}|B_\sigma \cap B_\omega| &= |B_\sigma \cap B_\omega \cap B^c_\pi| + |B_\sigma \cap B_\omega \cap B_\pi| \\ &\leq |B_\omega \cap B^c_\pi| + |B_\sigma \cap B_\pi| \\ &\overset{(a)}{<} (r - |B_\omega \cap B_\pi|) + (r - 2t) \\ &\overset{(b)}{\leq} t + r - 2t = r - t,\end{aligned}$$

where B^c_π denotes the complement of B_π. Inequality (a) follows from the fact that $|B_\omega| = r$ and (15); and inequality (b) follows from (16). This completes the proof. ■

Remark 10. A code satisfying the condition of Lemma 9 can in fact correct a class of errors that is more general than translocation errors. More precisely, the code can correct errors that lead to the displacement of at most t elements of each rank. In particular, the code can correct t transposition errors, t Hamming errors, or any t errors where each error displaces at most one element from each rank. As an example of the latter type of error, consider

$$(\{3,\mathbf{4}\},\{2,6\},\{\mathbf{7},8\},\{\mathbf{1},5\}) \xrightarrow{\text{error}} (\{3,\mathbf{1}\},\{2,6\},\{\mathbf{4},8\},\{\mathbf{7},5\}),$$

where each rank corresponds to one set in the set partition. We note that each part except for the one listed second has one displaced (moved) element.

A code that satisfies the conditions of Lemma 9 can be decoded in time $O(Mnr)$, where M is the size of the code. As before, suppose that $\pi \in C$ is the unknown stored codeword and ω is the retrieved permutation. For each $i \in [n/r]$, we must identify a unique set $A(i) \in \{\mathfrak{o}^r_\sigma(i) : \sigma \in C\}$ such that

$$|A(i) \cap \mathfrak{o}^r_\omega(i)| \geq r - t. \tag{17}$$

The ordered partition representation of π is then $\mathfrak{o}^r_\pi = (A(1), \ldots, A(n/r))$.

The intersection of $\mathfrak{o}^r_\omega(i)$ and each of the sets in $\{\mathfrak{o}^r_\sigma(i) : \sigma \in C\}$ can be trivially found with time complexity $O(r^2)$. Since there are M sets in $\{\mathfrak{o}^r_\sigma(i) : \sigma \in C\}$, finding $A(i)$ for each $i \in [n/r]$ takes $O(Mr^2)$ steps. Thus $\mathfrak{o}^r_\pi$ can be identified with complexity $O(Mr^2 n/r) = O(Mnr)$.

Since for some code parameters M can be exponential in n, the time needed for exhaustive search decoding may be exponential as well. However, if more information about the structure of the code is available, decoding may be performed much faster, as in the cases of constructions based on grouping elements and Steiner systems discussed in Subsections IV-A1 and IV-A2.

We pause to briefly comment on the relationship between almost disjoint sets of set partitions and intersecting families, in the context of the celebrated Erdős-Ko-Rado (EKR) theorem (see [34], [35] and references therein). A family of subsets of a set is said to be intersecting if each pair of subsets have a non-empty intersection. The EKR theorem establishes upper bounds on the size of the largest intersecting family. This theorem is also extended to the space of permutations where a set of permutations is said to be intersecting if each pair of permutations agree in some coordinate [36]–[38]. In our formulation, we require the intersections to be small, unlike for intersecting families where the intersection size may be arbitrary large as long as it is non-zero. Furthermore, we require our subsets to be organized into ordered partitions, with the intersection property holding only for parts at the same location. Although the code-anticode theorem by Delsarte [39], [40] may help in establishing bounds on families of subsets intersecting in a few elements only, it cannot be used for the specialized ordered set partition setting in a simple manner. To the best of our knowledge, the almost disjoint set partition family problem has not been previously studied in the extremal combinatorics literature.

Next, we describe two methods for constructing codes that satisfy the conditions of Lemma 9.

1) A Construction based on grouping elements: If r is a multiple of $2t+1$, the following simple construction satisfies the conditions of Lemma 9. Partition the set $[n]$ in an arbitrary fashion into $n/(2t+1)$ parts $E_1, \cdots, E_{n/(2t+1)}$, each of size

$2t+1$. Consider all ordered partitions $\mathfrak{o}$ of $[n]$ into n/r parts of size r that place all elements of each $E_j, j \in [n/(2t+1)]$, in the same part. Let C be a code such that its corresponding set of ordered set partitions $\mathfrak{O}_r(C)$ consists of the set of aforementioned partitions $\mathfrak{o}$.

As an illustration, suppose $t = 1$, $r = 6$, and $n = 12$, and let $\{1, \ldots, 12\}$ be partitioned as $\{E_1, E_2, E_3, E_4\}$, with

$$\begin{aligned} E_1 &= \{1, 2, 3\}, & E_2 &= \{4, 5, 6\}, \\ E_3 &= \{7, 8, 9\}, & E_4 &= \{10, 11, 12\}. \end{aligned}$$

Next, consider ordered partitions of $\{1, \ldots, 12\}$ that place all elements of each E_i in the same part, namely,

$$\begin{aligned} \mathfrak{o}_1 &= \left(\{\underline{1,2,3}, \underline{4,5,6}\}, \{\underline{7,8,9}, \underline{10,11,12}\}\right), \\ \mathfrak{o}_2 &= \left(\{\underline{1,2,3}, \underline{7,8,9}\}, \{\underline{4,5,6}, \underline{10,11,12}\}\right), \\ \mathfrak{o}_3 &= \left(\{\underline{1,2,3}, \underline{10,11,12}\}, \{\underline{4,5,6}, \underline{7,8,9}\}\right), \\ \mathfrak{o}_4 &= \left(\{\underline{4,5,6}, \underline{7,8,9}\}, \{\underline{1,2,3}, \underline{10,11,12}\}\right), \\ \mathfrak{o}_5 &= \left(\{\underline{4,5,6}, \underline{10,11,12}\}, \{\underline{1,2,3}, \underline{7,8,9}\}\right), \\ \mathfrak{o}_6 &= \left(\{\underline{7,8,9}, \underline{10,11,12}\}, \{\underline{1,2,3}, \underline{4,5,6}\}\right). \end{aligned}$$

Then, the code corresponding to the set of ordered partitions $\mathfrak{O} = \{\mathfrak{o}_1, \ldots, \mathfrak{o}_6\}$ can correct one translocation error.

To see that C satisfies the conditions of Lemma 9, consider $\mathfrak{o}, \mathfrak{o}' \in \mathfrak{O}_r(C)$ and $i \in [n/r]$. Suppose that $\mathfrak{o}(i) \neq \mathfrak{o}'(i)$. There exists E_j such that $E_j \subseteq \mathfrak{o}(i)$ but $E_j \cap \mathfrak{o}'(i) = \emptyset$. Since $|E_j| = 2t+1$, we have $|\mathfrak{o} \cap \mathfrak{o}'(i)| < r - 2t$.

The simplicity of this construction allows for fast decoding. Without loss of generality, assume that

$$E_j = \{(j-1)(2t+1)+1, \ldots, j(2t+1)\}.$$

Suppose that π is the stored codeword and ω is the retrieved permutation. For each $i \in [n/r]$, we have $E_j \subseteq \mathfrak{o}_\pi^r(i)$ if $|E_j \cap \mathfrak{o}_\omega^r(i)| \geq t+1$. To compute $|E_j \cap \mathfrak{o}_\omega^r(i)|$, $j \in [n/(2t+1)]$, we compare each element of $\mathfrak{o}_\omega^r(i)$ with $j(2t+1)$, $j \in [n/(2t+1)]$. This can be performed in $O\left(\frac{rn}{2t+1}\right)$ steps. Hence, decoding can be performed in time $O(\frac{n}{r}\frac{rn}{2t+1}) = O(n^2)$.

Let $d = 2t+1$. The cardinality of the code C equals

$$\frac{(n/d)!}{((r/d)!)^{n/r}},$$

and thus the asymptotic rate is

$$\begin{aligned} \lim \frac{\frac{n}{d}\ln\frac{n}{d} - \frac{n}{d}\ln\frac{r}{d} + O(n)}{n\ln n + O(n)} &= \lim \frac{1}{d}\frac{\ln n - \ln r + O(1)}{\ln n + O(1)} \\ &= (1-\rho)\lim\frac{1}{d}. \end{aligned}$$

Hence, the asymptotic rate is nonzero iff d is bounded (constant). While the rate of the code does not approach capacity, it should be noted that, per Remark 10, the code can correct more general errors than translocation errors.

2) Constructions based on combinatorial designs: Several well-known – and a number of significantly lesser known – families of combinatorial objects are closely related to the notion of almost disjoint ordered set partition families. These include block designs and Latin squares. From the first category, we use *Steiner systems* and *resolvable balanced incomplete block designs* and, from the latter category, we mention *semi-Latin squares*, representing a generalization of the well-known family of Latin squares [41]. The constructions are straightforward consequences of the definition of almost disjoint sets, but they provide for a rather limited set of code parameters. A more general method, based on interleaving arguments, will be presented in the next subsection.

A Latin square of order n is an $n \times n$ array such that each element of $[n]$ appears exactly once in each row and exactly once in each column. A semi-Latin square with parameters n and r is an $\frac{n}{r} \times \frac{n}{r}$, array where each cell is an r-subset of $[n]$ such that each element in $[n]$ appears exactly once in each column and exactly once in each row [25]. An example of a semi-Latin square is shown below, with $n = 6$, $r = 2$:

{1,4}	{2,5}	{3,6}
{3,5}	{1,6}	{2,4}
{2,6}	{3,4}	{1,5}

Note that the definition of a semi-Latin square implies that each row and each column of the square represent a partition of $[n]$. Hence, we arrive at the following result.

Lemma 11. *The rows of a semi-Latin square with parameters n and r, viewed as ordered set partitions of $[n]$, form the ordered set partitions of an $\mathsf{MPC}_\circ(n, r, r)$ code of cardinality $\frac{n}{r}$.*

The result is a direct consequence of Lemma 9 and the fact that no element is repeated in a column of a semi-Latin square. Unfortunately, the size of a code based on semi-Latin squares is small, since the row-column restrictions are too strong for the purpose of designing almost disjoint ordered set partition families.

As stated before, a code that satisfies the conditions of Lemma 9 can be decoded in time $O(Mnr)$. This implies that the code of Lemma 11 is decodable in time $O(n^2)$.

Another family of combinatorial objects that allow for constructing almost disjoint ordered set partitions are special types of designs, namely *resolvable balanced incomplete block designs* and *resolvable Steiner systems*.

A k-(n, r, λ)-*design* is a family of r-subsets of a set X of size n, each called a *block*, such that every k-subset of X appears in exactly λ blocks. Such a design is *resolvable* if its blocks can be grouped into m classes, such that each class forms a partition of X. It is known that [41, p. 202]

$$m = \lambda\frac{\binom{n-1}{k-1}}{\binom{r-1}{k-1}}.$$

A *Steiner system* $S(k, r, n)$ is a k-$(n, r, 1)$-design and a *balanced incomplete block design* (BIBD) with parameters (n, r, λ) is a 2-(n, r, λ)-design. For the purpose of code construction, resolvable Steiner systems and Resolvable BIBDs (RBIBDs) are of special interest.

The following lemma shows that resolvable Steiner systems can be used to construct multipermutation codes in the Ulam metric. Resolvable designs may also be used to construct multipermutation codes in the Hamming metric, as described by Chu et al. [16]. The aforementioned construction nevertheless does not cater to the specialized requirements posed by the Ulam metric.

Lemma 12. *If a resolvable Steiner system $S(k,r,n)$ exists, then there exists an $\mathsf{MPC}_\circ(n,r,d)$, where d is an odd number satisfying $d \le r-k+1$, of size*

$$\frac{\binom{n-1}{k-1}}{\binom{r-1}{k-1}}\left(\frac{n}{r}\right)!.$$

Proof: We use a Steiner system $S(k,r,n)$ to construct a family of ordered set partitions satisfying the conditions of Lemma 9.

Let m denote the number of classes of the Steiner system. The blocks of each of the m classes of the Steiner system form an unordered set partition. Each unordered set partition gives rise to $\left(\frac{n}{r}\right)!$ ordered set partitions. Hence, in total, we have $m(\frac{n}{r})!$ ordered set partitions. Let C be a code such that its corresponding set of ordered set partitions $\mathfrak{O}_r(C)$ is the aforementioned set of $m(\frac{n}{r})!$ partitions.

Let $t=(d-1)/2$. We have $k \le r-2t$. Each two blocks in the Steiner system have less than k elements in common, and consequently, have less than $r-2t$ elements in common. It follows that the conditions of Lemma 9 are satisfied. Hence, C is an $\mathsf{MPC}_\circ(n,r,d)$ of the stated size. ■

The code described in the preceding lemma can be decoded in time $O(n^k r)$ as follows. Suppose that π is the stored codeword and ω is the retrieved permutation. For each $i \in [n/r]$, to find $\mathfrak{o}^r_\pi(i)$, one needs to compute the size of the intersection of $\mathfrak{o}^r_\omega(i)$ with the blocks of the Steiner system. Computing each intersection takes $O(r^2)$. Hence, decoding can be performed in time

$$O\left(\frac{n}{r}\frac{\binom{n-1}{k-1}}{\binom{r-1}{k-1}}r^2\right)=O\left(nr\binom{n-1}{k-1}\right)=O\left(n^k r\right).$$

An RBIBD with parameters $(n,r,\lambda=1)$ is a resolvable Steiner system $S(2,r,n)$, and thus can be used for code construction. For $\lambda=1$, the case of interest in all our subsequent derivations, the condition

$$n=r \mod r(r-1)$$

is necessary for the existence of an RBIBD, and it is also known to be asymptotically sufficient for $r \ge 5$ [23].

Two of the most commonly used approaches to constructing RBIBDs are based on finite fields [23] and on a simple combinatorial construction [24]. Using the former construction, one can derive RBIBDs with parameters $\lambda=1$, $n=p^{\alpha v}$, and $r=p^\alpha$, with p a prime and α and v positive integers.

The combinatorial construction of [24] is based on the following straightforward procedure. Assume that r is prime and arrange the $n=r^2$ elements of the n-set into an $r\times r$ array in order. Each row corresponds to one block of size r, and each array represents a class that partitions the n-set. The first class, denoted by C_1, is shown below for $r=3$:

1	2	3
4	5	6
7	8	9

Class C_2 is constructed from class C_1 by taking the transpose. Each subsequent class C_i, for $i \ge 3$, is constructed from the previous class C_{i-1} in the following manner: the cyclically continued diagonals of C_{i-1} are arranged row-wise, starting from the main diagonal, and then moving to the left subdiagonals. For the example with $r=3$, the additional three classes constructed according to the above procedure take the form:

1	4	7
2	5	8
3	6	9

1	5	9
2	6	7
3	4	8

1	6	8
2	4	9
3	5	7

Note that the procedure terminates after $r+1$ steps, resulting in a repetition of class C_2. The total number of blocks in the RBIBD equals $r(r+1)=r^2+r$.

The construction involving cyclic diagonal shifts can be extended for resolvable, *unbalanced* IBDs with parameters $n=p^\alpha r$, $\alpha \ge 1$, p prime, and block size r which may be an arbitrary integer ≥ 2. The only difference between a balanced and unbalanced design is the requirement that any pair of elements appear in *at most* λ blocks [24]. For the case $\lambda=1$, i.e., any pair of elements appearing zero or one time, the designs are known as zero-one concurrence designs; they may be constructed by a combination of variety cutting and the diagonalization procedure described above. The interested reader is referred to [24] for an in-depth treatment of this construction.

The aforementioned procedures show that an RBIBD with parameters $(r^2,r,1)$ exists, provided that r is am odd prime. Hence, using Lemma 12, we can obtain the following lemma, which concludes this subsection.

Lemma 13. *Suppose that r is an odd prime. Then, $A_\circ(r^2,r,r-2) \ge (r+1)r!$.*

B. Construction based on codes with r components

In this subsection, we present a construction for multipermutation codes in the Ulam metric based on r permutation codes of length n/r, interleaved to ensure translocation error protection.

Assume first that $d \le n/r$. Consider a partition $\{P_1,\dots,P_r\}$ of $[n]$ into sets of equal size, and the set of codes $\{C_1,\dots,C_r\}$, with each C_i, $i\in[r]$, being a permutation code of minimum Ulam distance d over P_i. We form a new code C as follows:

$$C=\bigcup_{c_i\in C_i,\,\forall\, i}\mathsf{R}_r(c_1\circ\cdots\circ c_r). \tag{18}$$

Proposition 14. *The code C given in (18) is an $\mathsf{MPC}_\circ(n,r,d)$ code.*

Proof: Consider $\pi',\sigma' \in C$ such that $\pi' \not\equiv_r \sigma'$. By construction, there exists $\pi \in \mathsf{R}_r(\pi')$ and $\sigma \in \mathsf{R}_r(\sigma')$ such that

$$\begin{aligned}\sigma&=\sigma_1\circ\cdots\circ\sigma_r, \quad \sigma_i\in C_i,\\ \pi&=\pi_1\circ\cdots\circ\pi_r, \quad \pi_i\in C_i.\end{aligned}\tag{19}$$

Since $\pi \not\equiv_r \sigma$, there exists an element $j \in [r]$ such that $\pi_j \neq \sigma_j$.

We show that for an arbitrary choice of $\alpha \in \mathsf{R}_r(\pi)$ and $\beta \in \mathsf{R}_r(\sigma)$, we have $\mathsf{d}_\circ(\alpha, \beta) \geq d$. Since α and β are chosen arbitrarily, we find that

$$\mathsf{d}^r_\circ(\pi', \sigma') = \mathsf{d}^r_\circ(\pi, \sigma) = \min_{\alpha \in \mathsf{R}_r(\pi)} \min_{\beta \in \mathsf{R}_r(\sigma)} \mathsf{d}_\circ(\alpha, \beta) \geq d,$$

which completes the proof.

For $\alpha \in \mathsf{R}_r(\pi)$ and $\beta \in \mathsf{R}_r(\sigma)$ and each $i \in [r]$, the order of the elements of P_i is the same in π and in α, i.e., $\pi_{P_i} = \alpha_{P_i}$. Furthermore, since $\pi_i \in C_i$, and C_i is a code over P_i, we have $\pi_i = \pi_{P_i}$. Hence, $\alpha_{P_i} = \pi_i$. A similar argument holds for σ and β, implying that $\beta_{P_i} = \sigma_i$ for each $i \in [r]$. So, by Lemma 1, one can show that

$$\begin{aligned} \mathsf{d}_\circ(\alpha, \beta) &\geq \sum_{i=1}^{r} \mathsf{d}_\circ\left(\alpha_{P_i}, \beta_{P_i}\right) = \sum_{i=1}^{r} \mathsf{d}_\circ\left(\pi_i, \sigma_i\right) \\ &\geq \mathsf{d}_\circ\left(\pi_j, \sigma_j\right) \geq d, \end{aligned}$$

where the last inequality follows from $\pi_j \neq \sigma_j$. ■

As an example, for $n = 6$, $r = 2$, and $d = 2$, consider

$$\begin{aligned} P_1 &= \{1, 2, 3\}, \\ P_2 &= \{4, 5, 6\}, \\ C_1 &= \{(1, 2, 3), (3, 2, 1))\}, \\ C_2 &= \{(4, 5, 6), (6, 5, 4)\}. \end{aligned}$$

Note that C_1 and C_2 both have Ulam distance equal to 2. The code C, constructed according to (18) contains

$$\begin{matrix} (1, 4, 2, 5, 3, 6), & (1, 6, 2, 5, 3, 4), \\ (3, 4, 2, 5, 1, 6), & (3, 6, 2, 5, 1, 4), \end{matrix}$$

and their equivalency classes under $\equiv_2$. For instance, let $\pi = (1, 4, 2, 5, 3, 6)$ and $\sigma = (3, 4, 2, 5, 1, 6)$ and consider $\alpha = (4, 1, 5, 2, 3, 6) \in \mathsf{R}_2(\pi)$ and $\beta = (4, 3, 5, 2, 6, 1) \in \mathsf{R}_2(\sigma)$. It can be observed that $\alpha_{P_1} = \pi_{P_1} = (1, 2, 3)$ and $\alpha_{P_2} = \pi_{P_2} = (4, 5, 6)$. Similar statements hold for β and σ. It can also be verified that

$$\mathsf{d}_\circ(\alpha, \beta) = 2.$$

For several constructions of permutation codes in the Ulam metric, we refer the reader to [7].

The components of the constructed code can be decoded independently. As before, suppose that π is the stored codeword and ω is the retrieved permutation. Since there are at most $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors, we have $\mathsf{d}_\circ(\pi, \omega) \leq t$. By Lemma 1, this implies that $\mathsf{d}_\circ(\pi_P, \omega_P) \leq t$ for all $P \in \{P_1, \ldots, P_r\}$. Hence, one can use a decoder for permutation codes in the Ulam metric that can correct t errors. Consequently, $\mathfrak{o}^r_\pi$ can be identified from $\omega_P, P \in \{P_1, \ldots, P_r\}$, through a parallel decoding process. Note that a simple decoding architecture for a class of codes in the Ulam metric was proposed in our companion paper [7], based on Hamming distance decoding of de-interleaved component codes.

Assuming that the cardinality of the codes C_i equals $A_\circ(n/r, 1, d)$, the cardinality of C equals $A_\circ(n/r, 1, d)^r$. Recall that we define the cardinality of a multipermutation code as the number of its equivalency classes and not the number of its elements. It was proved in [7] that

$$A_\circ(m, 1, d) \geq \frac{(m - d + 1)!}{\binom{m}{d-1}}.$$

Hence,

$$A_\circ(n, r, d) \geq \left(\frac{(n/r - d + 1)!}{\binom{n/r}{d-1}} \right)^r.$$

Furthermore, from the fact that $\mathcal{C}_\circ(1, d) = 1 - \delta$ [7], we find that

$$\begin{aligned} \mathcal{C}_\circ(r, d) &= \lim \frac{\ln A_\circ(n, r, d)}{\ln n!} \\ &\geq \lim \frac{r \ln A_\circ(n/r, 1, d)}{\ln n!} \\ &= \lim \frac{\ln A_\circ(n/r, 1, d)}{\ln (n/r)!} \lim \frac{r \ln (n/r)!}{\ln n!} \\ &= (1 - \lim \frac{rd}{n})(1 - \rho). \end{aligned}$$

In particular, if $\lim \frac{rd}{n} = 0$, then $\mathcal{C}_\circ(r, d) = (1 - \rho)$.

C. Construction based on codes in the Hamming metric

Recall that $\mathsf{d}^r_H(\pi, \sigma) \geq \mathsf{d}^r_\circ(\pi, \sigma)$. Thus, if C is an $\mathsf{MPC}_\circ(n, r, d)$ code, then it is also an $\mathsf{MPC}_H(n, r, d)$ code. We now show that an $\mathsf{MPC}_\circ(n, r, d)$ code can be obtained using multipermutation Hamming codes of shorter lengths. We refer the reader to [13]–[16] for constructions of multipermutation codes in the Hamming metric.

Proposition 15. *Suppose that n/r is even and that $d \leq r$. Let $P = \left[\frac{n}{2}\right]$, and $Q = [n] \backslash P$. Additionally, let C'_1 be an $\mathsf{MPC}_\circ(\frac{n}{2}, r, d)$ code over P and C_1 be an $\mathsf{MPC}_H(\frac{n}{2}, r, d)$ code over Q. The code $C = C'_1 \circ_r C_1$ is an $\mathsf{MPC}_\circ(n, r, d)$ code.*

Proof: Let $\pi, \sigma \in C$ with $\pi \not\equiv_r \sigma$. Assume that

$$\pi = \pi'_1 \circ_r \pi_1, \qquad \sigma = \sigma'_1 \circ_r \sigma_1,$$

where $\pi'_1, \sigma'_1 \in C'_1$ and $\pi_1, \sigma_1 \in C_1$.

First, suppose that $\pi'_1 \not\equiv_r \sigma'_1$. Then,

$$\begin{aligned} \mathsf{d}^r_\circ(\pi, \sigma) &= \min_{\alpha \in \mathsf{R}_r(\pi)} \min_{\beta \in \mathsf{R}_r(\sigma)} \mathsf{d}_\circ(\alpha, \beta) \\ &\geq \min_{\alpha \in \mathsf{R}_r(\pi)} \min_{\beta \in \mathsf{R}_r(\sigma)} \mathsf{d}_\circ(\alpha_P, \beta_P) \\ &\geq d, \end{aligned}$$

where the first inequality follows from Lemma 1, and the second inequality follows from the facts that $\alpha_P \in \mathsf{R}_r(\pi'_1) \subseteq C'_1$, $\beta_P \in \mathsf{R}_r(\sigma'_1) \subseteq C'_1$, and that C'_1 is an $\mathsf{MPC}_\circ(n/2, r, d)$ code.

Next, suppose that $\pi'_1 \equiv_r \sigma'_1$. Since $\pi \not\equiv_r \sigma$, we have $\pi_1 \not\equiv_r \sigma_1$. Let

$$D = \left\{ x \in Q : x \in \mathfrak{o}^r_{\pi_1}(i), x \in \mathfrak{o}^r_{\sigma_1}(j), i \neq j \right\}$$

be the set of elements of Q that are of different ranks in π_1 and σ_1. Note that $|D| = \mathsf{d}^r_H(\pi_1, \sigma_1)$.

Consider $\alpha \in \mathsf{R}_r(\pi)$ and $\beta \in \mathsf{R}_r(\sigma)$. For odd values of i, we have $\mathfrak{o}^r_\alpha(i) = \mathfrak{o}^r_\beta(i)$, as $\pi'_1 \equiv_r \sigma'_1$.

On the one hand, for any common subsequence of α and β that contains an element of D, there exists some odd i such that $\mathfrak{o}^r_\alpha(i) = \mathfrak{o}^r_\beta(i)$ is not in that subsequence. This implies that the length of the given common subsequence is at most $n - r$. On the other hand, for any common subsequence of α and β that does not contain any element of D, the length of that subsequence is at most

$$n - |D| = n - \mathsf{d}^r_H(\pi_1, \sigma_1) \le n - d.$$

Hence, the length of any common subsequence of α and β is at most

$$\max\{n - d, n - r\} = n - d$$

and thus $\mathsf{d}_\circ(\alpha, \beta) \ge d$. Since α and β are arbitrary elements of $\mathsf{R}_r(\pi)$ and $\mathsf{R}_r(\sigma)$, respectively, we find that $\mathsf{d}^r_\circ(\pi, \sigma) \ge d$, which completes the proof. ■

One particularly simple choice for C'_1 is

$$C'_1 = \mathsf{R}_r\left((1, \ldots, n/2)\right), \qquad (20)$$

which is a code with cardinality 1.

As an example, let $n = 8$, $r = 2$, $d = 4$, and

$$\begin{aligned} C'_1 &= \mathsf{R}_2\left((1,2,3,4)\right) \\ &= \left\{(1,2,3,4), (2,1,3,4), (1,2,4,3), (2,1,4,3)\right\}, \\ C_1 &= \mathsf{R}_2\left((5,6,7,8)\right) \cup \mathsf{R}_2\left((7,8,5,6)\right), \end{aligned}$$

which leads to

$$C = \mathsf{R}_2\left((1,2,5,6,3,4,7,8)\right) \cup \mathsf{R}_2\left((1,2,7,8,3,4,5,6)\right),$$

an $\mathsf{MPC}_\circ\,(8, 2, 4)$ code.

For the case of (20), the cardinality of C equals the cardinality of C_1, which may be as large as $A_H(n/2, r, d)$. Hence,

$$A_\circ(n, r, d) \ge A_H(n/2, r, d)$$

if n/r is even and $d \le r$. With similar arguments, one can show that, if n/r is odd and $d \le r$, then

$$A_\circ(n, r, d) \ge A_H\left((n + r)/2, r, d\right).$$

For $d \le r$ and $\rho < 1$, we have $\delta = 0$. Hence, for $d \le r$ and $\rho < 1$,

$$\mathcal{C}_\circ(r, d) \ge \frac{1}{2}(1 - \rho)(1 - 2\delta) = \frac{1}{2}(1 - \rho).$$

To construct larger codebooks, one may recursively use the construction of Prop. 15 to design C'_1. For simplicity, suppose that n and r are both powers of 2. Let

$$C = \left(\left(C'_k \circ_r C_k\right) \circ_r C_{k-1} \circ_r \cdots\right) \circ_r C_1,$$

where each $C_i, i \in [k]$, is an $\mathsf{MPC}_H\left(n/2^i, r, d\right)$ code, $C'_k = \mathsf{R}_r\left((1, \ldots, n/2^k)\right)$, and k is a positive integer satisfying $k \le \lg(n/r)$. The condition $k \le \lg(n/r)$ is required since we need $n/2^k \ge r$. Note that this condition also implies that $n/2^k \ge d$, since $d \le r$. The code C is an $\mathsf{MPC}_\circ(n, r, d)$ code. The cardinality of C_i can be as large as $A_H\left(n/2^i, r, d\right)$. Hence, if n and r are powers of 2 and $d \le r$, it holds that

$$A_\circ(n, r, d) \ge \prod_{i=1}^{k} A_H\left(n/2^i, r, d\right).$$

Let $n = 2^j$, $r = 2^{\rho j}$, $d \le r$, where ρ is a constant less than 1, and suppose that k is a constant such that $k \le \lg(n/r) = j(1 - \rho)$. For this regime, we have

$$\begin{aligned} \lim_{j\to\infty} \frac{\ln A_\circ(2^j, 2^{\rho j}, d)}{\ln n!} \ge & \lim_{j\to\infty} \frac{\lg A_\circ(2^j, 2^{\rho j}, 2^{\rho j})}{\lg 2^j!} \\ \ge & \sum_{i=1}^{k} \lim_{j\to\infty} \frac{\lg A_H(2^{j-i}, 2^{\rho j}, 2^{\rho j})}{\lg 2^{j-i}!} \frac{\lg 2^{j-i}!}{\lg 2^j!} \\ = & \sum_{i=1}^{k} \left(1 - \lim_{j\to\infty} \frac{\rho j}{j - i}\right) 2^{-i} \\ = & (1 - \rho)(1 - 2^{-k}). \end{aligned}$$

Since $\rho < 1$, k can be chosen arbitrarily large. Hence, the asymptotic rate can be made arbitrary close to $(1 - \rho)$.

V. Conclusion

We studied a novel rank modulation scheme based on multipermutation codes in the Ulam metric. We also highlighted the close connection between multipermutation codes in the Hamming metric, also known as constant composition codes and frequency permutation arrays, and codes in the Ulam metric.

The presented results included bounds on the size of multipermutation codes in both the Ulam metric and the Hamming metric; for the case of the Hamming metric, these bounds led to the capacity of the codes, while for the Ulam metric, the bounds led to upper bounds and lower bounds for the capacity, with a gap equal to $\rho(1 - \delta)$. We also presented several construction methods for codes in the Ulam metric using permutation interleaving, semi-Latin squares, resolvable Steiner systems, and resolvable balanced incomplete block designs, among other techniques.

Acknowledgment: The work was supported by NSF grants CCF 0809895, CCF 1218764, and the Emerging Frontiers for Science of Information, CCF 0939370. The authors would like to thank anonymous reviewers for their insightful comments and Eyal En Gad for several useful discussions.

References

[1] D. Slepian, "Permutation modulation," *Proc. IEEE*, vol. 53, pp. 228–236, Mar. 1965.

[2] A. J. Han Vinck, "Coded modulation for power line communications," *AEÜ Journal*, pp. 45–49, January 2000, Available: http://arxiv.org/abs/1104.1528.

[3] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. Information Theory*, vol. 55, pp. 2659–2673, June 2009.

[4] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," *IEEE Trans. Information Theory*, vol. 56, pp. 3158–3165, July 2010.

[5] E. En Gad, M. Langberg, M. Schwartz, and J. Bruck, "Constant-weight Gray codes for local rank modulation," *IEEE Trans. Information Theory*, vol. 57, pp. 7431–7442, Nov. 2011.

[6] A. Jiang, M. Schwartz, and J. Bruck, "Error-correcting codes for rank modulation," in *Proc. IEEE Int. Symp. Information Theory*, (Toronto, Canada), pp. 1736–1740, July 2008.

[7] F. Farnoud (Hassanzadeh), V. Skachek, and O. Milenkovic, "Error-correction in flash memories via codes in the Ulam metric," *IEEE Trans. Information Theory*, vol. 59, no. 5, pp. 3003–3020, 2013.

[8] V. I. Levenshtein, "On perfect codes in deletion and insertion metric," *Discrete Mathematics and Applications*, vol. 2, no. 3, pp. 241–258, 1992.

[9] P. Gopalan, T. S. Jayram, R. Krauthgamer, and R. Kumar, "Estimating the sortedness of a data stream," in *Proc. 18th annu. ACM-SIAM symposium on Discrete algorithms (SODA)*, (New Orleans, Louisiana), pp. 318–327, January 2007.

[10] W. Chu, C. Colbourn, and P. Dukes, "Constructions for permutation codes in powerline communications," *Designs, Codes and Cryptography*, vol. 32, no. 1-3, pp. 51–64, 2004.

[11] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Permutation arrays under the Chebyshev distance," *IEEE Trans. Information Theory*, vol. 56, pp. 2611–2617, June 2010.

[12] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," *IEEE Trans. Information Theory*, vol. 56, pp. 2551–2560, June 2010.

[13] Y. Luo, F.-W. Fu, A. J. Han Vinck, and W. Chen, "On constant-composition codes over Zq," *IEEE Trans. Information Theory*, vol. 49, no. 11, pp. 3010–3016, 2003.

[14] C. Ding and J. Yin, "Combinatorial constructions of optimal constant-composition codes," *IEEE Trans. Information Theory*, vol. 51, no. 10, pp. 3671–3674, 2005.

[15] S. Huczynska and G. L. Mullen, "Frequency permutation arrays," *Journal of Combinatorial Designs*, vol. 14, no. 6, pp. 463–478, 2006.

[16] W. Chu, C. J. Colbourn, and P. Dukes, "On constant composition codes," *Discrete Applied Mathematics*, vol. 154, no. 6, pp. 912 – 929, 2006.

[17] E. En Gad, A. Jiang, and J. Bruck, "Trade-offs between instantaneous and total capacity in multi-cell flash memories," in *Proc. IEEE Int. Symp. Information Theory*, pp. 990–994, 2012.

[18] M.-Z. Shieh and S.-C. Tsai, "Decoding frequency permutation arrays under Chebyshev distance," *IEEE Trans. Information Theory*, vol. 56, pp. 5730–5737, Nov. 2010.

[19] M.-Z. Shieh and S.-C. Tsai, "Computing the ball size of frequency permutations under Chebyshev distance," in *Proc. IEEE Int. Symp. Information Theory*, pp. 2100–2104, July/Aug. 2011.

[20] S. Buzaglo, E. Yaakobi, T. Etzion, and J. Bruck, "Error-correcting codes for multipermutations," in *Proc. IEEE Int. Symp. Information Theory*, 2013.

[21] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," *IEEE Trans. Communications*, vol. 61, no. 7, pp. 2624–2634, 2013.

[22] F. Farnoud, V. Skachek, and O. Milenkovic, "Rank modulation for translocation error correction," in *Proc. IEEE Int. Symp. Information Theory*, pp. 2988–2992, July 2012.

[23] B. Rumov, "Existence of resolvable block designs," *Mathematics of the USSR-Sbornik*, vol. 28, no. 3, p. 325, 1976.

[24] M. Khare and W. Federer, "A simple construction procedure for resolvable incomplete block designs for any number of treatments," *Biometrical Journal*, vol. 23, no. 2, pp. 121–132, 1981.

[25] D. A. Preece and G. H. Freeman, "Semi-Latin squares and related designs," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 45, no. 2, pp. pp. 267–277, 1983.

[26] L. Grupp, A. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. Siegel, and J. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in *42nd Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO-42)*, pp. 24–33, Dec. 2009.

[27] A. M. Odlyzko, *Asymptotic Enumeration Methods*, vol. 2, pp. 1063–1229. Amsterdam: Elsevier, 1995.

[28] R. L. Graham, M. Grötschel, and L. Lovász, *Handbook of Combinatorics*, vol. 2. Elsevier, 1995.

[29] P. Deheuvels and D. Pfeifer, "Poisson approximations of multinomial distributions and point processes," *Journal of Multivariate Analysis*, vol. 25, no. 1, pp. 65–89, 1988.

[30] N. Arenbaev, "Asymptotic behavior of the multinomial dstribution," *Theory of Probability & Its Applications*, vol. 21, no. 4, pp. 805–810, 1976.

[31] M. Hofri, *Analysis of algorithms: computational methods and mathematical tools*. Oxford University Press, 1995.

[32] O. Milenkovic and K. J. Compton, "Probabilistic transforms for combinatorial urn models," *Combinatorics, Probability and Computing*, vol. 13, pp. 645–675, July 2004.

[33] G. H. Gonnet and J. Ian Munro, "The analysis of linear probing sort by the use of a new mathematical transform," *Journal of Algorithms*, vol. 5, no. 4, pp. 451–470, 1984.

[34] M. Deza and P. Frankl, "Erdős-Ko-Rado theorem—22 years later," *SIAM Journal on Algebraic Discrete Methods*, vol. 4, no. 4, pp. 419–431, 1983.

[35] D. Ellis, "Setwise intersecting families of permutations," *Journal of Combinatorial Theory, Series A*, vol. 119, no. 4, pp. 825–849, 2012.

[36] P. Frankl and M. Deza, "On the maximum number of permutations with given maximal or minimal distance," *J. Combinatorial Theory, Series A*, vol. 22, pp. 352–360, 1977.

[37] P. J. Cameron and C. Ku, "Intersecting families of permutations," *European Journal of Combinatorics*, vol. 24, no. 7, pp. 881–890, 2003.

[38] C. Godsil and K. Meagher, "A new proof of the Erdős-Ko-Rado theorem for intersecting families of permutations," *European Journal of Combinatorics*, vol. 30, no. 2, pp. 404–414, 2009.

[39] P. Delsarte, "An algebraic approach to the association schemes of coding theory," *Philips research reports supplements*, no. 10, p. 103, 1973.

[40] M. Schwartz and I. Tamo, "Optimal permutation anticodes with the infinity norm via permanents of (0,1)-matrices," *Journal of Combinatorial Theory, Series A*, vol. 118, no. 6, pp. 1761–1774, 2011.

[41] D. R. Stinson, *Combinatorial designs: construction and analysis*. New York: Springer, 2004.