On the Labeling Problem of Permutation Group Codes under the Infinity Metric

Itzhak Tamo
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 84105, Israel
tamo@ee.bgu.ac.il

Moshe Schwartz
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 84105, Israel
schwartz@ee.bgu.ac.il

***Abstract*—Codes over permutations under the infinity norm have been recently suggested as a coding scheme for correcting limited-magnitude errors in the rank modulation scheme. Given such a code, we show that a simple relabeling operation, which produces an isomorphic code, may drastically change the minimal distance of the code. Thus, we may choose a code structure for efficient encoding/decoding procedures, and then optimize the code's minimal distance via relabeling.**

We formally define the relabeling problem, and show that all codes may be relabeled to get a minimal distance at most 2. The decision problem of whether a code may be relabeled to distance 1 is shown to be NP-complete, and calculating the best achievable minimal distance after relabeling is proved hard to approximate.

Finally, we consider general bounds on the relabeling problem. We specifically show the optimal relabeling distance of cyclic groups. A specific case of a general probabilistic argument is used to show $\mathrm{AGL}(p)$ may be relabeled to a minimal distance of $p - O(\sqrt{p \ln p})$.

I. INTRODUCTION

In the race to dominate non-volatile information-storage devices, flash memory is a prominent contender. Flash memory is an electronic non-volatile memory that uses floating-gate cells to store information [4]. While initially, flash memory cells used to contain a single bit of information, in the standard multi-level flash-cell technology of today, every cell has $q > 2$ discrete states, $\{0, 1, \ldots, q-1\}$, and therefore can store $\log_2 q$ bits. The flash memory changes the state of a cell by injecting (*cell programming*) or removing (*cell erasing*) charge into/from the cell.

Flash memories possess inherent problems one has to address in order to have a reliable storage medium. Firstly, writing is more time- and energy-consuming than reading [4]. The main reason behind this asymmetry is the iterative cell-programming procedure designed to avoid over-programming [1] (raising the cell's charge level above its target level). Cells can be programmed individually, however when erasing is needed, the flash memory can erase only entire blocks (today, containing approximately 10^5 cells, see [4]). Since over-programming can only be corrected by erasure of entire blocks, a more cautious programming is performed in order to achieve a more accurate cell level. This approach, of course, is costly in both time and energy.

Another major concern for flash memory is data reliability. The stored data can be corrupted due to many reasons such as charge leakage, read disturbance, and write disturbance (see [4]), among other things. Some of the error mechanisms have an asymmetric property, making the cells' charge levels drift in one direction. In addition, error rate increases due to the number of levels in multi-level flash memory, since the safety margins between adjacent levels are reduced.

To address these issues, the *rank-modulation scheme* has been recently suggested [12]. This scheme removes the necessity to measure absolute charge levels, eliminates the over-programming problem, and reduces retention errors. In this scheme the information is stored by the permutation induced by the n distinct charge levels being read from n cells. Each cell has a *rank* which indicates its relative position when ordering the cells in descending charge level. The ranks of the n cells induce a permutation of $\{1, 2, \ldots, n\}$.

While this new scheme alleviates some of the problems associated with current flash technology, the flash-memory channel remains noisy and error correction must be employed to increase reliability. In a recent work [15], spike-error correction was addressed in this model. Such errors are characterized by a limited-magnitude change in charge level in the cells, which readily translates into a limited-magnitude change in the rank of, possibly, *all* cells in the stored permutation. These errors correspond to a bounded distance change in the induced permutation under the ℓ_∞-metric. We call codes protecting against such errors *limited-magnitude rank-modulation codes*, or LMRM-codes. Throughout the paper we consider only LMRM-codes.

A similar error model for flash memory was considered not in the context of rank modulation in [5], while a different error-model (charge-constrained errors for rank modulation) was studied in [13]. Codes over permutations are also referred to as *permutation arrays* and have been studied in the past under different metrics [2], [3], [6], [7], [9], [10], [16]. Specifically, permutation arrays under the ℓ_∞-metric were considered in [14].

A code over permutations, being a subset of the symmetric group S_n, may happen to be a subgroup, in which case we call it a *group code*. Group theory offers a rich structure to be exploited when constructing and analyzing group codes, in an analogy to the case of linear codes over vector spaces. Hence, throughout this paper, we focus on LMRM group codes.

If $\mathcal{C}$ and $\mathcal{C}'$ are conjugate subgroups of the symmetric group, then from a group-theoretic point of view, they are almost the same group, having the same properties. However, from a coding point of view these two codes can possess vastly

978-1-4577-0594-6/11/$26.00 ©2011 IEEE

different minimal distance, which is one of the most important properties of a code. For example, let $\mathcal{C} = \{\iota, (1,n)\}$ and $\mathcal{C}' = \{\iota, (1,2)\}$, where ι is the identity permutation and the rest of the permutations are given in a cycle notation. The subgroups $\mathcal{C}$ and $\mathcal{C}'$ are conjugate but the minimal distance of $\mathcal{C}$ and $\mathcal{C}'$ is $n-1$ and 1 respectively, which is the highest and the lowest possible minimal distance in the ℓ_∞-metric.

Hence, we conclude that the minimal distance of a code $\mathcal{C}$ depends crucially on the specific conjugate subgroup. Thus, while a certain group code might be chosen due to its group-theoretic structure (perhaps allowing simple encoding or decoding), we may choose to use an isomorphic conjugate of the group, having the same group-theoretic structure, but with a higher minimal distance. We refer to the problem of finding the optimal minimal distance among all conjugate groups (sets) of a certain group (set) as the *labeling problem*.

Apart from introducing and motivating the labeling problem, we show that this algorithmic problem is hard. However, we are able to show the existence of a labeling with high minimal distance for a variety of codes, based on the size of the code and the number of cycles in certain permutations derived from the code itself.

The rest of the paper is organized as follows. In Section II we define the notation, introduce the error model with the associated ℓ_∞-metric, as well as formally defining the labeling problem. We proceed in Section III to introduce two algorithmic problems related to the labeling problem, and we show their hardness. In Section IV we give some labeling results on ordinary groups and we present our main result of the paper, which gives general labeling results for arbitrary codes based on a probabilistic argument. In addition with give a few corollaries by applying this result to some well-known groups. We conclude in Section V with a summary of the results and short concluding remarks. Due to the page limitation, most proofs are omitted.

II. Definitions and Notations

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

We will use the cycle notation for permutations $f \in S_n$, where $f = (f_0, f_1, \ldots, f_{k-1})$ denotes the permutation mapping $f_i \mapsto f_{i+1 \bmod k}$ for $i \in [0, k-1]$. Given two permutations $f, g \in S_n$, the product fg is a permutation mapping $i \mapsto f(g(i))$ for all $i \in [n]$.

A *code*, $\mathcal{C}$ is a subset $\mathcal{C} \subseteq S_n$. Sometimes $\mathcal{C}$ will be also be a subgroup of S_n, in which case we shall refer to $\mathcal{C}$ as a *group code*. For a code $\mathcal{C}$ and a permutation $f \in S_n$ we call the code $f\mathcal{C}f^{-1} = \{fcf^{-1} : c \in \mathcal{C}\}$ a *conjugate code* of $\mathcal{C}$.

Consider n flash memory cells which we name $1, 2, \ldots, n$. The charge level of each cell is denoted by $c_i \in \mathbb{R}$ for all $i \in [n]$. In the *rank-modulation scheme* defined in [12], the information is stored by the permutation induced by the cells' charge levels: The induced permutation (in vector notation) is $[f_1, f_2, \ldots, f_n]$ iff $c_{f_1} > c_{f_2} > \cdots > c_{f_n}$.

Having stored a permutation in n flash cells, a corrupted version of it may be read due to any of a variety of error sources (see [4]). For a measure of the corruption in the stored permutations one can use any of the well-known metrics over S_n (see [8]). Given a metric over S_n, defined by a distance function $d : S_n \times S_n \to \mathbb{N} \cup \{0\}$, an *error-correcting code* is a subset of S_n with lower-bounded distance between distinct members.

In [13], the Kendall-τ metric was used, where the distance between two permutations is the number of adjacent transpositions required to transform one into the other. This metric is used when we can bound the total difference in charge levels.

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

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

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

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

Loosely speaking, an error of limited magnitude cannot change the *rank* of the cell i (which is simply $f^{-1}(i)$) by d or more positions.

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

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

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

Definition 1. *A* limited-magnitude rank-modulation code (LMRM-code) *with parameters* (n, M, d), *is a subset* $\mathcal{C} \subseteq S_n$ *of cardinality* M, *such that* $d_\infty(f,g) \geqslant d$ *for all* $f, g \in \mathcal{C}$, $f \neq g$. *(We will sometimes omit the parameter* M.*)*

We note that unlike the charge-constrained rank-modulation codes of [13], in which the codeword is stored in the permutation induced by the charge levels of the cells, here the codeword is stored in the *inverse* of the permutation.

For a code $\mathcal{C}$ we define its minimal distance and denote it by $d(\mathcal{C})$ as

$$d(\mathcal{C}) = \min_{\substack{f,g \in \mathcal{C} \\ f \neq g}} d(f,g).$$

A labeling function is a permutation l over the set $[n]$. We say that the code $\mathcal{C}$ has minimal distance d with a labeling function l when

$$d(l\mathcal{C}l^{-1}) = d.$$

It is well known (see [8]) that the ℓ_∞-metric over S_n is only right invariant and not left invariant, i.e., for any $f,g,h \in S_n$, $d(f,g) = d(fh,gh)$, and usually $d(f,g) \neq d(hf,hg)$, thus we would expect that in many cases $d(\mathcal{C}) \neq d(l\mathcal{C}l^{-1})$. Therefore, the questions of which labeling permutation leads to the optimal minimal distance, and what is the optimal minimal distance, rise naturally in the context of error-correcting codes over permutations under the infinity metric. Note that l is called a labeling function because for a permutation in cycle notation $f = (a_1,\ldots,a_{k_1})\ldots(a_{k_j+1},\ldots,a_n)$ we get

$$lfl^{-1} = (l(a_1),\ldots,l(a_{k_1}))\ldots(l(a_{k_j+1}),\ldots,l(a_n)).$$

The labeled permutation lfl^{-1} has the same cycle structure as f but the elements within each cycle are relabeled by l.

From the right invariance of the ℓ_∞-metric we get that

$$d(\mathcal{C}) = \min_{\substack{f=gh^{-1} \\ g,h\in\mathcal{C}, g\neq h}} d(f,\iota),$$

where ι is the identity element of S_n. This makes it easier to calculate the minimal distance of a group code since f simply goes over all the codewords.

More specifically, we will explore the case where $\mathcal{C}$ is a subgroup of S_n and ask which conjugate group of $\mathcal{C}$ has the largest minimal distance. We denote by $\mathcal{L}_{\min}(\mathcal{C})$ ($\mathcal{L}_{\max}(\mathcal{C})$) the minimal (maximal) achievable minimal distance among all the conjugates of a code $\mathcal{C}$.

III. The Labeling Problem is hard to approximate

In this section we define two algorithmic problems regarding the labeling of codes, and show that they are hard to approximate. We shall begin by showing that for any code $\mathcal{C}$, $\mathcal{L}_{\min}(\mathcal{C}) \leqslant 2$, which means that the minimal distance of a code depends crucially on its labeling. We then continue by showing the decision problem of whether $\mathcal{L}_{\max}(\mathcal{C}) \geqslant 2$ is NP-complete, while finding out what $\mathcal{L}_{\max}(\mathcal{C})$ is hard to approximate.

Theorem 2. *For any code $\mathcal{C} \subseteq S_n$ (not necessarily a subgroup), there exists a labeling of the elements such that the minimum distance is at most 2, i.e., there exists $l \in S_n$ such that $d(l\mathcal{C}l^{-1}) \leqslant 2$. Moreover, $\mathcal{C}$ has a labeling with minimal distance 1 if and only if the set $\{ab^{-1} : a,b \in \mathcal{C}\}$ contains an involution (a permutation of order 2).*

We now show that the algorithmic decision problem of determining whether a certain code $\mathcal{C}$ has $\mathcal{L}_{\max}(\mathcal{C}) = 1$ or $\mathcal{L}_{\max}(\mathcal{C}) \geqslant 2$ is NP-complete.

2-DISTANCE PROBLEM:

- INPUT: A subset of permutations $\mathcal{C} \subseteq S_n$ given as a list of permutations, each given in vector notation.
- OUTPUT: The correct Yes or No answer to the question "Does $\mathcal{C}$ have a labeling that leads to a minimal distance at least 2, i.e., is $\mathcal{L}_{\max}(\mathcal{C}) \geqslant 2$? ".

We start with a few definitions. For a code $\mathcal{C} \subseteq S_n$, define its associated set of involutions as

$$I(\mathcal{C}) = \{g \in S_n : g^2 = \iota,\ \ g = ab^{-1} \neq \iota,\ \ a,b \in \mathcal{C}\}.$$

For any $g \in I(\mathcal{C})$ we define a set of edges, $E(g)$, in the complete graph on n vertices, K_n, where the vertices are conveniently called $1,2,\ldots,n$, as

$$E(g) = \{uv \in E(K_n) : g(u) = v, u \neq v\}.$$

Recall that a Hamiltonian path in an undirected graph G is a path which visits each vertex exactly once. The following theorem shows an equivalence between the property of a code having a labeling with minimal distance at least 2 and the existence of a certain Hamiltonian path in the complete graph K_n.

Theorem 3. *Let $\mathcal{C} \subseteq S_n$ be a code, then $\mathcal{L}_{\max}(\mathcal{C}) \geqslant 2$ if and only if there exists a Hamiltonian path in K_n which does not include all the edges $E(g)$ for any $g \in I(\mathcal{C})$.*

By the last theorem we conclude that any algorithm that finds a labeling of $\mathcal{C}$ with minimal distance at least 2, actually finds a Hamiltonian path in K_n which does not include all the edges $E(g)$, for any $g \in I(\mathcal{C})$. We are now able to show that the 2-DISTANCE problem is NP-complete.

Theorem 4. *The 2-DISTANCE problem is NP-complete.*

Proof: First, we show that 2-DISTANCE is in NP. For any given verifier, $l \in S_n$, which is a labeling function, we compute the distance between ι and all the elements of $I(\mathcal{C})$. Note that $|I(\mathcal{C})| \leqslant |\mathcal{C}|^2$ and constructing $I(\mathcal{C})$ may be easily done in polynomial time. Thus, the question can be verified in polynomial time.

In order to verify the completeness we shall reduce the HAMILTONIAN-PATH problem (see [11]) to our problem. Let $G(V,E)$ be a graph on n vertices (given as an adjacency matrix) in which we want to decide whether a Hamiltonian path exists. Define the code

$$\mathcal{C} = \{(u,v) : uv \notin E\} \cup \{\iota\},$$

where (u,v) is the permutation that fixes everything in place except commuting the elements u and v. Obviously, we can construct $\mathcal{C}$ from G in polynomial time. We then run the 2-DISTANCE algorithm on $\mathcal{C}$ and return its answer.

We observe that

$$I(\mathcal{C}) = \{(u,v)(k,l) : (u,v),(k,l) \in \mathcal{C}, \{u,v\} \neq \{k,l\}\} \cup \mathcal{C} \setminus \{\iota\}.$$

If $a_1,a_2,\ldots,a_n$ is a Hamiltonian path in G, then it is also a Hamiltonian path in K_n not containing all of $E(g)$ for any $g \in I(\mathcal{C})$ (and in fact, not containing *any* edge of $E(g)$). This is true because the edges $E(g)$ are a subset of the edges that are not in E.

For the other direction, if there is a Hamiltonian path in K_n which does not include all the edges of $E(g)$ for any $g \in I(\mathcal{C})$, then, in particular, this path does not include all of $E(g)$, $g \in \mathcal{C}$, $g \neq \iota$. Since for any such $g = (u, v) \in \mathcal{C}$, $E(g) = uv \notin E$, this path is also a Hamiltonian path in G. ∎

We now define a harder algorithmic question and deduce by Theorem 4 that this problem is hard to approximate.

OPTIMAL-DISTANCE PROBLEM:

- INPUT: A subset of permutations $\mathcal{C} \subseteq S_n$ given in a vector notation.
- OUTPUT: The integer $\mathcal{L}_{\max}(\mathcal{C})$.

For a constant $\epsilon > 1$ we say the problem may be ϵ-approximated if there exists an efficient algorithm that for any input $\mathcal{C}$ computes $f(\mathcal{C})$ which satisfies

$$\frac{1}{\epsilon}\mathcal{L}_{\max}(\mathcal{C}) \leqslant f(\mathcal{C}) \leqslant \epsilon\mathcal{L}_{\max}(\mathcal{C}).$$

Corollary 5. *For any $1 < \epsilon < 2$, the OPTIMAL-DISTANCE problem cannot be ϵ-approximated unless $P = NP$.*

IV. Bounds on Optimal Labelings

In the previous section we have shown that the 2-DISTANCE and OPTIMAL-DISTANCE problems are hard. We are therefore motivated to focus on solving and bounding the latter problem for specific families of codes, and in particular, codes that form a subgroup of the symmetric group S_n. The rich structure offered by such codes makes them easier to analyze, in much the same way as linear codes in vector space. Furthermore, knowing good labelings for certain groups is of great interest since one can use them as building blocks when constructing larger codes (see for example the direct and semi-direct product constructions in [15]).

The most simple basic groups one can think of are cyclic groups. Recall that for a cyclic group G there is an element $g \in G$ such that G is generated by the powers of g, i.e., $G = \{g^k : k \in \mathbb{N}\}$. The following theorem gives an exact optimal labeling for transitive cyclic group over the set $[n]$.

Theorem 6. *Let $\mathcal{C} \subseteq S_n$ be a transitive cyclic group over the set $[n]$, then the optimal minimal distance for $\mathcal{C}$ is*

$$\mathcal{L}_{\max}(\mathcal{C}) = n - \left\lceil \frac{\sqrt{4n-3}-1}{2} \right\rceil.$$

We now turn to address the optimal labeling problem for the $\mathrm{AGL}(p)$ group, p a prime, but first we give some definitions.

Definition 7. *Let $p \in \mathbb{N}$ be a prime, then $\mathrm{AGL}(p)$ is defined by the subgroup of permutations that acts on the set $[0, p-1]$ and is generated by the permutations $f(x) = x+1$ and $g(x) = ax$, where all calculations are over $\mathrm{GF}(p)$ and a is primitive in $\mathrm{GF}(p)$.*

Throughout we shall consider only $\mathrm{AGL}(p)$ for $p \geqslant 3$. We refer to the natural labeling of $\mathrm{AGL}(p)$ as the labeling derived from the permutations f and g described above. For example, the natural labeling of $\mathrm{AGL}(5)$ is the group generated by the permutations (in cycle notation) $f = (0,1,2,3,4)$ and $g = (1,2,4,3)$. The following theorem gives us the minimal distance of the natural labeling.

Theorem 8. *$\mathrm{AGL}(p)$ with the natural labeling has minimal distance $(p-1)/2$.*

Theorem 9. *For any prime $3 \leqslant p < 8$,*

$$\mathcal{L}_{\max}(\mathrm{AGL}(p)) = \frac{p-1}{2}.$$

Proof: Let I be the set of involutions of $\mathrm{AGL}(p)$. It is easy to verify that any permutation $g \in I$ is of the form $g(x) = -x + b$ for some $b \in \mathrm{GF}(p)$, and so $|I| = p$. We note also that for any $x_1, x_2 \in \mathrm{GF}(p)$ there is exactly one involution $g \in I$ such that $g(x_1) = x_2$ (finding g is by solving the equation $x_2 = -x_1 + b$).

Assume that we have a labeling of $\mathrm{AGL}(p)$ with minimal distance more than the natural minimal distance. In particular, with this labeling every involution has minimal distance at least $(p+1)/2$ from the identity permutation. Let

$$B = \left\{ \{x, y\} : x, y \in \mathrm{GF}(p), |x - y| \geqslant \frac{p+1}{2} \right\}.$$

Now, for any $g \in I$ there is at least one unordered pair $\{x, y\} \in B$ such that $g(x) = y$. It follows that

$$|B| = \frac{p^2-1}{8} \geqslant |I| = p.$$

Solving the inequality we get $p \geqslant 4 + \sqrt{17} > 8$. ∎

We now present a general method we call the neighboring-sets method. With this method, lower and upper bounds on $\mathcal{L}_{\max}(\mathcal{C})$ may be obtained provided certain neighboring sets of indices exist. We shall first describe the general method, and then apply it, using further probabilistic arguments, to show strong bounds on $\mathcal{L}_{\max}(\mathrm{AGL}(p))$.

Definition 10. *Let $\mathcal{C} \subseteq S_n$ be any set of permutations acting on $[n]$. Two disjoint subsets $A, B \subseteq [n]$ are called $\mathcal{C}$-*neighboring sets *if for any $f \in \mathcal{C}$, $f \neq \iota$, the following holds*

$$(f(A) \cap B) \cup (f(B) \cap A) \neq \emptyset.$$

We define $O(\mathcal{C})$ to be the smallest integer $O(\mathcal{C}) = |A| + |B|$, where A and B are $\mathcal{C}$-neighboring sets. If there are no such sets then we say $O(\mathcal{C}) = \infty$.

First we show that if $\mathcal{C}$ is a group then, $O(\mathcal{C})$ is closely related to its optimal minimal distance.

Theorem 11. *Let $\mathcal{C} \subseteq S_n$ be a group that acts on $[n]$ with $O(\mathcal{C}) < \infty$, then*

$$n - O(\mathcal{C}) + 1 \leqslant \mathcal{L}_{\max}(\mathcal{C}).$$

Moreover, if $\mathcal{L}_{\max}(\mathcal{C}) \geqslant \frac{n}{2}$ then also

$$\mathcal{L}_{\max}(\mathcal{C}) \leqslant n - \frac{O(\mathcal{C})}{2}.$$

It is pointed out in the definition that some groups $\mathcal{C} \subseteq S_n$ might have $O(\mathcal{C}) = \infty$, e.g., $O(S_n) = \infty$. The following theorem shows that for any prime $p > 5$, $O(\mathrm{AGL}(p))$ is finite while also showing a lower bound.

Theorem 12. *If $p = 3,5$, then $O(\mathrm{AGL}(p)) = \infty$. For any prime $p \geqslant 7$,*

$$O(\mathrm{AGL}(p)) \geqslant \max\left\{\sqrt{2(p-1)}, 6\right\}.$$

For primes $p \geqslant 37$ we also have

$$O(\mathrm{AGL}(p)) \leqslant p.$$

The following theorem is our main result of this paper. It gives a generic labeling result for a code $\mathcal{C}$ over the set $[n]$ based solely on the size of the code and the number of cycles in the set of permutations $\{gh^{-1} : g,h \in \mathcal{C}\}$.

Theorem 13. *Let $\mathcal{C} \subseteq S_n$ be a code. If there is $0 < p < 1$ and $t > 0$ such that*

$$e^{-\frac{2t^2}{n}} + e^{-np^2/(1-p)} \sum_{\substack{f=gh^{-1} \\ g,h\in\mathcal{C}, g\neq h}} e^{c(f)p^2/(1-p)} < 1, \quad (1)$$

where $c(f)$ is the number of cycles in the permutation f, then there exists a labeling l such that

$$\mathcal{L}_{\max}(\mathcal{C}) \geqslant d(l\mathcal{C}l^{-1}) \geqslant n + 1 - \lfloor 2pn + t \rfloor.$$

We say that $a \in [n]$ is a fixed point of a permutation $f \in S_n$ if $f(a) = a$. The minimal degree of a subgroup $\mathcal{C}$ is the minimum number of non-fixed points among the non-identity permutations in $\mathcal{C}$. The following corollary connects the minimal degree of a group and an achievable distance by applying Theorem 13.

Corollary 14. *Let $\mathcal{C}$ be a subgroup of S_n with minimal degree d, such that there exist $t > 0, 0 < p < \frac{1}{2}$, satisfying*

$$e^{-\frac{2t^2}{n}} + |\mathcal{C}| e^{-\frac{dp^2}{2(1-p)}} < 1,$$

then $\mathcal{C}$ has a labeling l with $d(l\mathcal{C}l^{-1}) \geqslant n + 1 - \lfloor 2pn + t \rfloor$.

Proof: If $\mathcal{C}$ has minimal degree d, then the number of cycles of any $g \in \mathcal{C}$, $g \neq \iota$, is at most $n - \frac{d}{2}$ and the claim follows by Theorem 13. ∎

The following corollary is the main result of this section, showing strong bounds on $\mathcal{L}_{\max}(\mathrm{AGL}(p))$.

Corollary 15. *For p, a large enough prime,*

$$p - O(\sqrt{p \ln p}) \leqslant \mathcal{L}_{\max}(\mathrm{AGL}(p)) \leqslant p - \sqrt{\frac{p-1}{2}}.$$

Proof: For the upper bound we simply combine Theorem 11 and Theorem 12. For the lower bound we recall that $\mathrm{AGL}(p)$ is sharply 2-transitive, hence, its minimal degree is $p - 1$. By Corollary 14,

$$e^{-\frac{2t^2}{p}} + |\mathrm{AGL}(p)| e^{-\frac{(p-1)q^2}{2(1-q)}} \leqslant e^{-\frac{2t^2}{p}} + p^2 e^{-\frac{(p-1)q^2}{2}}. \quad (2)$$

For $t = \sqrt{p\ln(p+1)}$ and $q = \sqrt{\frac{4\ln(p+1)}{p-1}}$, we get

$$e^{-\frac{2t^2}{p}} + p^2 e^{-\frac{(p-1)q^2}{2}} = \frac{1}{(q+1)^2} + \frac{q^2}{(q+1)^2} < 1.$$

We note that for p large enough, $q < \frac{1}{2}$. It follows that

$$\begin{aligned}\mathcal{L}_{\max}(\mathrm{AGL}(p)) &\geqslant p - 2pq - t \\ &= p - 2p\sqrt{\frac{4\ln(p+1)}{p-1}} - \sqrt{p\ln(p+1)} \\ &= p - O(\sqrt{p\ln p}).\end{aligned}$$

∎

V. Summary

In this work we defined the relabeling problem and showed its hardness. We studied bounds on the achievable relabeling distance for general groups, and showed strong bounds on some known groups. Finding out how the best achievable minimal distance after relabeling depends on certain group properties, and finding its exact value for other well-known groups, are still open problems.

References

[1] 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.

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

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

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

[5] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for asymmetric limited-magnitude errors with applications to multilevel flash memories," *IEEE Trans. on Inform. Theory*, vol. 56, no. 4, pp. 1582–1595, Apr. 2010.

[6] H. D. Chadwick and L. Kurz, "Rank permutation group codes based on Kendall's correlation statistic," *IEEE Trans. on Inform. Theory*, vol. IT-15, no. 2, pp. 306–315, Mar. 1969.

[7] C. J. Colbourn, T. Kløve, and A. C. H. Ling, "Permutation arrays for powerline communication and mutually orthogonal latin squares," *IEEE Trans. on Inform. Theory*, vol. 50, no. 6, pp. 1289–1291, Jun. 2004.

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

[9] C. Ding, F.-W. Fu, T. Kløve, and V. K. Wei, "Construction of permutation arrays," *IEEE Trans. on Inform. Theory*, vol. 48, no. 4, pp. 977–980, Apr. 2002.

[10] F.-W. Fu and T. Kløve, "Two constructions of permutation arrays," *IEEE Trans. on Inform. Theory*, vol. 50, no. 5, pp. 881–883, May 2004.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[12] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. on Inform. Theory*, vol. 55, no. 6, pp. 2659–2673, Jun. 2009.

[13] A. Jiang, M. Schwartz, and J. Bruck, "Correcting charge-constrained errors in the rank-modulation scheme," *IEEE Trans. on Inform. Theory*, vol. 56, no. 5, pp. 2112–2120, May 2010.

[14] T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Efficient encoding and decoding with permutation arrays," in *Proceedings of the 2008 IEEE International Symposium on Information Theory (ISIT2008), Toronto, Canada*, 2008, pp. 211–214.

[15] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," *IEEE Trans. on Inform. Theory*, vol. 56, no. 6, pp. 2551–2560, Jun. 2010.

[16] H. Vinck, J. Haering, and T. Wadayama, "Coded M-FSK for power line communications," in *Proceedings of the 2000 IEEE International Symposium on Information Theory (ISIT2000), Sorrento, Italy*, 2000, p. 137.