Permutation Codes Correcting a Single Burst Deletion I: Unstable Deletions

Yeow Meng Chee, Van Khu Vu, and Xiande Zhang
School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
email: {ymchee, vankhu001, xiandezhang}@ntu.edu.sg

***Abstract*—We construct the first class of permutation codes that are capable of correcting a burst of up to s unstable deletions, for general s. Efficient decoding algorithms are presented to show the correctness of our constructions.**

I. Introduction

Flash memories have become a mature technology for the nonvolatile storage of information that can be electrically erased and/or reprogrammed. They are increasingly replacing hard disks, offering the advantage of speed, noise, power consumption, and physical reliability. However, there remain two primary factors affecting more pervasive applications of flash memories:

(i) high cost per gigabyte of storage, and
(ii) limited number of program-erase (P/E) cycles.

To overcome the high cost per gigabyte, flash memories have moved from single-level cell (SLC) to multi-level cell (MLC) technology, where data in a multi-level cell is represented by $q > 2$ charge levels, instead of two charge levels in a single-level cell [1]. This increases the storage density of flash memories. To address the limited number of P/E cycles, rewritable codes aimed at performing multiple writes per cell before erasure, and hence increasing the overall lifetime of the flash memory have been studied [2]–[4].

The dominant source of errors in flash memories is charge leakage in cells [5]. Since the charge levels in a multi-level cell are closer together than in a single-level cell, MLC flash memories are more prone to errors due to charge leakage. A permutation coded rank modulation scheme was first introduced by Jiang et al. [2] to combat errors in flash memories due to charge leakage. This scheme was later extended to tolerate more severe and other errors [6]–[9].

More recently, Gabrys et al. [10], [11] considered deletion errors in rank modulated flash memories. Such errors can occur when cells are corrupted and the charge levels cannot be read correctly. In particular, Gabrys et al. gave distance properties under which permutation codes are capable of correcting against deletion errors. In this paper, we study the related problem of burst deletions in rank modulated flash memories, that is, a series of deletions that occur in consecutive cells. The motivation behind considering burst deletions is that as flash memory scales, the parasitic capacitance of adjacent cells increases, which can cause corruptions in a cell to bleed to adjacent cells, through capacitative coupling [12], [13].

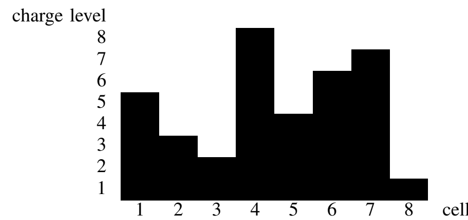

Fig. 1: Charge levels in a block of eight cells in a flash memory. The relative values of the charge levels give the permutation $(5, 3, 2, 8, 4, 6, 7, 1)$, which represents the information stored in the block.

In a rank modulated flash memory, information is written in blocks of n cells, and is represented as the relative values of the charge levels (see Fig. 1). Hence, each block stores a permutation of degree n. A *deletion* occurs when the charge level in a cell cannot be read and the location of that cell is not known. Deletions can be further classified according to their *stability*, a notion introduced by Gabrys et al. [10], [11].

(i) *Stable deletion*: In such a deletion, the absolute values of the remaining components of the permutation are known.
(ii) *Unstable deletion*: In such a deletion, only the relative values of the remaining components of the permutation are known.

Example 1. Suppose a deletion occurs in cell 3 of the block depicted in Fig. 1. In a stable deletion, the remaining components of the permutation gives the vector $(5, 3, 8, 4, 6, 7, 1)$, whereas in an unstable deletion the remaining components of the permutation gives the vector $(4, 2, 7, 3, 5, 6, 1)$.

Our focus in this paper is on *unstable deletions*. The unstable deletion model has an analogue for insertions. For single insertion and deletion, the duality was described in [11]. We explain how multiple insertions work in this model. Now suppose symbols 4 and 2 are going to be inserted between symbols 1 and 4 in permutation $(3, 1, 4, 2)$. We first deal with the smaller symbol 2, and insert it into the third position. Then the resulting permutation is $(4, 1, 2, 5, 3)$, where all the symbols in the original permutation of value greater than or equal to 2 are incremented by one. Second, the next smaller symbol 4 is inserted to the third position of $(4, 1, 2, 5, 3)$, which results in a permutation $(5, 1, 4, 2, 6, 3)$ by increasing each symbol greater than or equal to 4 by one. The duality between multiple insertions and deletions is clear.

A permutation that experiences unstable deletions loses all information on values and locations of the corrupted cells. Hence, codes capable of correcting unstable deletion errors need more constraints than codes for correcting stable deletions. For stable deletions, perfect permutation codes correcting a single deletion have been constructed [14]. For unstable deletions, an asymptotically optimal family of single-deletion correcting codes were constructed by Gabrys et al. [11].

In this paper, we construct the first family of permutation codes that are capable of correcting a single burst of s unstable deletions, for general s. Our constructions are based on an application of permutation interleaving. For the use of specialized forms of interleaving in other areas of coding theory, interested readers are referred to [6], [8], [15]–[17].

II. Preliminaries

A. Definitions

For integers $a \le b$, $[a,b]$ denotes the set $\{a,\ldots,b\}$. Let n be a positive integer and S_n be the set of all permutations on the set $[1,n]$.

For a permutation $\sigma \in S_n$, let σ_i be the ith component of σ, that is, $\sigma = (\sigma_1,\sigma_2,\ldots,\sigma_n)$. For a set of positions $I \subseteq [1,n]$, define $\sigma(I) = \{\sigma_i : i \in I\}$. For $a \in [1,n]$ and $I \subseteq [1,n]$, the integer $a(I) \in [1,n]$ is defined so that $a(I) = a - |\{i \in I : i < a\}|$. For example, if $\sigma = (5,1,4,2,6,3)$ and $I = \{3,4\}$, then $\sigma(I) = \{2,4\}$, $5(\sigma(I)) = 5-2 = 3$, $1(\sigma(I)) = 1-0 = 1$, $6(\sigma(I)) = 6-2 = 4$ and $3(\sigma(I)) = 3-1 = 2$.

Definition 1. Assume that $\sigma = (\sigma_1,\sigma_2,\ldots,\sigma_n) \in S_n$ and $I \subseteq [1,n]$ is a set of size s. We say the permutation σ suffers s *unstable deletions* in I, resulting in the permutation $\sigma_{\downarrow,I} = (\sigma'_1,\ldots,\sigma'_{n-s}) \in S_{n-s}$, if for all $k \in [1,n]\setminus I$, and $i = k(I)$, we have $\sigma'_i = \sigma_k(\sigma(I))$. If $I = [a,b]$ for some a and b, then σ suffers an unstable *burst deletion* of length s.

As an example, let $\sigma = (5,1,4,2,6,3)$ and $I = \{3,4\}$. If σ suffers an unstable burst deletion in I, then it results in the permutation $\sigma_{\downarrow,I} = (3,1,4,2)$.

Definition 2. A code $\mathcal{C} \subseteq S_n$ is called an *s-UD permutation code* if it can correct up to s unstable deletions, or an *s-UBD permutation code* if it can correct a single unstable burst deletion of length s.

B. 1-UD Permutation Codes

For a positive integer $a \in \mathbb{Z}_n$, let

$$\mathcal{C}_a^n = \left\{ \mathsf{u} \in \{0,1\}^n : \sum_{i=1}^{n} i\mathsf{u}_i \equiv a \bmod (n+1) \right\},$$

where u_i is the ith component of u. Then $\mathcal{C}_a^n$ are the family of binary codes known as the Varshamov-Tenengolts codes [18]. These codes are capable of correcting a single deletion.

For a permutation $\sigma = (\sigma_1,\sigma_2,\ldots,\sigma_n) \in S_n$, its *inverse* is the permutation $\sigma^{-1} = (\sigma_1^{-1},\sigma_2^{-1},\ldots,\sigma_n^{-1})$, where σ_i^{-1} is the location of i in σ. As in [19], the *signature* of σ is the binary vector $\alpha(\sigma) = (\alpha(\sigma)_1,\ldots,\alpha(\sigma)_{n-1})$ of length $n-1$, where $\alpha(\sigma)_i = 1$ if $\sigma_{i+1} \ge \sigma_i$, and 0 otherwise, for all $i \in [1,n-1]$.

The 1-UD permutation codes of Gabrys et al. [11] are defined as follows. Given $a,b \in \mathbb{Z}_n$, let

$$\mathcal{C}_{a,b}^n = \left\{ \sigma \in S_n : \alpha(\sigma) \in \mathcal{C}_a^{n-1} \text{ and } \alpha(\sigma^{-1}) \in \mathcal{C}_b^{n-1} \right\}.$$

Note that there are two defining constraints for $\mathcal{C}_{a,b}^n$. If a codeword in $\mathcal{C}_{a,b}^n$ suffers a single unstable deletion, then the signature constraint helps to get some information about the deleted location, while the inverse signature constraint helps to get some information about the deleted value. Combining both allows the recovery of the original codeword. An efficient decoding algorithm is given by Gabrys et al. [11].

III. An Upper Bound

Let $A_{\mathsf{UBD}}(n,s)$ be the maximum size of an s-UBD permutation code in S_n. We abbreviate $A_{\mathsf{UBD}}(n,1)$ to $A_{\mathsf{UBD}}(n)$. Gabrys et al. [11] gave the following upper bound.

Lemma 1 (Gabrys et al. [11])**.** *For any positive $\epsilon < 1$, there exists an N_ϵ such that for all $n \ge N_\epsilon$, $A_{\mathsf{UBD}}(n) \le \frac{n!}{n(n-\log n)}(1+\epsilon)$.*

The 1-UD permutation codes constructed of Gabrys et al. [11] are of size at least $n!/n^2$. Thus they are asymptotically optimal with respect to this upper bound.

We provide a corresponding upper bound for $A_{\mathsf{UBD}}(n,s)$ in this section.

Let $\sigma = (\sigma_1,\sigma_2,\ldots,\sigma_n) \in S_n$. For each $i \in [1,s]$, let $I_i = [1,n] \setminus \{i, s+i, \ldots, \lfloor\frac{n-i}{s}\rfloor s + i\}$. Then $\sigma_{\downarrow,I_i}$ is a permutation over $\left[1, \lfloor\frac{n-i}{s}\rfloor + 1\right]$, induced by ranking the symbols in the subsequence $\left(\sigma_i, \sigma_{s+i}, \ldots, \sigma_{\lfloor\frac{n-i}{s}\rfloor s+i}\right)$. Let $B_s(\sigma)$ be the set of all permutations in S_{n-s} received as a result of a burst of s unstable deletions in σ. A *consecutive run* of σ is a substring of maximal length in σ that contains consecutively valued symbols, increasing or decreasing. Let $R(\sigma)$ be the number of consecutive runs in σ. For example, $\sigma = (5,1,2,4,3,6)$ has four consecutive runs (5), $(1,2)$, $(4,3)$ and (6). It is easy to see that $|B_1(\sigma)| = R(\sigma)$ [11]. For the number of permutations with exactly r consecutive runs, please see [20].

Lemma 2. *Let $n > s$ be positive integers. For any permutation $\sigma \in S_n$, we have*

$$|B_s(\sigma)| \ge 1 + \sum_{i=1}^{s} (R(\sigma_{\downarrow,I_i}) - 1). \tag{1}$$

Proof. The proof is by induction on n. It is easy to see that (1) holds for $n = s+1$. Assume that (1) holds for $s+1 \le n \le k$. We prove that it also holds for $n = k+1$.

Now, σ is a permutation of length $k+1$ and consider the permutation $\pi = \sigma_{\downarrow,\{k+1\}}$. Let $n_0 \in [1,s]$ be the integer such that $n_0 \equiv k+1 \bmod s$. We note that

$$R(\sigma_{\downarrow,I_i}) = R(\pi_{\downarrow,I_i}), \quad \forall\ i \in [1,s]\setminus\{n_0\}. \tag{2}$$

Moreover, when there is a burst of exactly s unstable deletions in the permutations σ and π, there is exactly one symbol deleted in $\sigma_{\downarrow,I_{n_0}}$ and $\pi_{\downarrow,I_{n_0}}$, respectively. Hence,

$$|B_1(\sigma_{\downarrow,I_{n_0}})| - |B_1(\pi_{\downarrow,I_{n_0}})| \le |B_s(\sigma)| - |B_s(\pi)|. \tag{3}$$

Because of (2), (3), and the induction hypothesis, we have

$$\begin{aligned}
&1+\sum_{i=1}^{s}(R(\sigma_{\downarrow,I_i})-1)\\
=&1+\sum_{i=1}^{s}(R(\pi_{\downarrow,I_i})-1)+R(\sigma_{\downarrow,I_{n_0}})-R(\pi_{\downarrow,I_{n_0}})\\
\le&|B_s(\pi)|+R(\sigma_{\downarrow,I_{n_0}})-R(\pi_{\downarrow,I_{n_0}})\le|B_s(\sigma)|.
\end{aligned}$$

This completes the proof. ■

The following result is useful in the estimate of $A_{\mathsf{UBD}}(n,s)$.

Lemma 3 (Gabrys et al. [11])**.** *The number of permutations in S_n with at most $n-\log n$ consecutive runs is at most $\frac{n!(n-\log n)^2}{(\log n)!}$.*

We are now ready to provide an upper bound for the maximum size of a permutation code correcting a burst of s unstable deletions.

Theorem 1. *Let s be a fixed positive integer, $m_1=\left\lfloor\frac{n}{s}\right\rfloor$ and $m_2=\left\lceil\frac{n}{s}\right\rceil$. For any positive $\epsilon<1$, there exists an N_ϵ such that for all $n\ge N_\epsilon$,*

$$A_{\mathsf{UBD}}(n,s)\le(1+\epsilon)\frac{(n-s)!}{s(m_1-\log m_1)}.$$

Proof. Suppose that $\mathcal{C}\subseteq S_n$ is an s-UBD permutation code. Let $\mathcal{C}_1=\{\sigma\in\mathcal{C}:|B_1(\sigma_{\downarrow,I_i})|>m_1-\log m_1 \text{ for all } i\in[1,s]\}$ and $\mathcal{C}_2=\mathcal{C}\setminus\mathcal{C}_1$. Note that for each $\sigma\in\mathcal{C}_2$, there exists at least one i such that $|B_1(\sigma_{\downarrow,I_i})|\le m_1-\log m_1$. Since $B_s(\sigma)\subseteq S_{n-s}$, by Lemma 2, we have $|\mathcal{C}_1|(s(m_1-\log m_1)+1)\le(n-s)!$. Hence,

$$|\mathcal{C}_1|\le\frac{(n-s)!}{s(m_1-\log m_1)}. \tag{4}$$

Since $m_1\le m_2$, we have

$$\frac{m_1!(m_1-\log m_1)^2}{(\log m_1)!}\le\frac{m_2!(m_2-\log m_2)^2}{(\log m_2)!}.$$

Therefore, for each $i\in[1,s]$, by Lemma 3, $|\{\sigma\in\mathcal{C}_2:|B_1(\sigma_{\downarrow,I_i})|\le m_1-\log m_1\}|\le\frac{m_2!(m_2-\log m_2)^2}{(\log m_2)!}$. Hence,

$$|\mathcal{C}_2|\le s\frac{m_2!(m_2-\log m_2)^2}{(\log m_2)!}. \tag{5}$$

From (4) and (5), we get

$$\begin{aligned}
|C|=|C_1|+|C_2|&\le\frac{(n-s)!}{s(m_1-\log m_1)}+s\frac{m_2!(m_2-\log m_2)^2}{(\log m_2)!}\\
&\le\frac{(n-s)!}{s(m_1-\log m_1)}\left(1+\frac{s^2m_2!(m_2-\log m_2)^3}{(n-s)!(\log m_2)!}\right).
\end{aligned}$$

Since $\lim_{n\to\infty}\frac{s^2m_2!(m_2-\log m_2)^3}{(n-s)!(\log m_2)!}=0$ for fixed s, there exists an N_ϵ such that for all $n\ge N_\epsilon$,

$$A_{\mathsf{UBD}}(n,s)\le(1+\epsilon)\frac{(n-s)!}{s(m_1-\log m_1)}.$$

■

Note that when $s=1$, the upper bound for $A_{\mathsf{UBD}}(n,s)$ in Theorem 1 is exactly the same as the upper bound for $A_{\mathsf{UBD}}(n)$ in Lemma 1.

IV. Constructions

In this section, we apply the permutation interleaving method to construct s-UBD permutation codes for $s\ge2$.

Definition 3. For vectors $\tau^i=(\tau^i_1,\tau^i_2,\ldots,\tau^i_m)$, $i\in[1,s]$, of length m, the *interleaved vector* $\sigma=\tau^1\circ\tau^2\circ\cdots\circ\tau^s$ is obtained by alternatively placing the elements of $\tau^1,\tau^2,\ldots,\tau^s$ in order. That is

$$\sigma_j=\tau^i_{\lceil j/s\rceil},\quad j\in[1,ms]$$

where $i\equiv j \bmod s$. For a class of s codes $\mathcal{C}_i$, $i\in[1,s]$ of same length, the *interleaved code*

$$\mathcal{C}_1\circ\mathcal{C}_2\circ\cdots\circ\mathcal{C}_s=\{\tau^1\circ\tau^2\circ\cdots\circ\tau^s:\tau^i\in\mathcal{C}_i,i\in[1,s]\}.$$

For any integer a, a vector $\tau=(\tau_1,\tau_2,\ldots,\tau_m)$ and a code $\mathcal{C}$, define $\tau+a=(\tau_1+a,\tau_2+a,\ldots,\tau_m+a)$ and $\mathcal{C}+a=\{\tau+a:\tau\in\mathcal{C}\}$.

Since the decoding algorithm for our codes is rather complex for general s, we first consider the case $s=2$ to explain the idea. To simplify notations, we assume that n is even.

Theorem 2. *Let $m\ge3$ and $n=2m$. Suppose that for each $i\in\{1,2\}$, $\mathcal{C}_i\subseteq S_m$ is a 1-UD permutation code. Then the interleaved code $\mathcal{C}=\mathcal{C}_1\circ(\mathcal{C}_2+m)$ is a 2-UBD permutation code in S_n.*

Proof. Suppose we receive a permutation $\pi\in S_{n-t}$, $1\le t\le2$. We want to find the unique permutation $\sigma=\tau\circ(\upsilon+m)\in\mathcal{C}$ such that π is obtained from σ through a burst of at most two unstable deletions.

If $t=1$, that is, $\pi\in S_{n-1}$, then we know that only one symbol is deleted from σ. Let f_1 be the subsequence of π with values from $[1,m-1]$, which is a permutation in S_{m-1}. Note that f_1 is obtained from τ by experiencing a single unstable deletion, thus we can recover τ from f_1 since $\mathcal{C}_1$ is a 1-UD permutation code. Let f_2 be the subsequence of π with values from $[m+1,n-1]$, which is a permutation over $[m+1,n-1]$. Note that values in f_2 are originally from $\upsilon+m$ before the unstable deletion. Thus f_2-m is a permutation in S_{m-1} obtained from υ by experiencing a single unstable deletion. Note that $f_2-m=\pi_{\downarrow,I}$, where $I=\{i\in[1,n-1]:\pi_i\le m\}$. Since $\mathcal{C}_2$ is a 1-UD permutation code, we can recover υ from f_2. Hence σ is the interleaved vector $\tau\circ(\upsilon+m)$ determined uniquely.

If $t=2$, that is, the permutation π has length $n-2$, then we know that there are exactly two symbols deleted. Since the deletions are adjacent, by definition of $\mathcal{C}$, there are exactly one symbol from τ and one symbol from $\upsilon+m$ that are deleted. Now we de-interleave permutation π. Let $f_1=(\pi_1,\pi_3,\ldots,\pi_{n-3})$ and $f_2=(\pi_2,\pi_4,\ldots,\pi_{n-2})$. Then it must be the case that $f_1\in S_{m-1}$ and f_2 is a permutation over $[m,n-2]$. Similarly, we can recover τ and υ from f_1 and $f_2-(m-1)$ respectively, where $f_2-(m-1)$ is actually $\pi_{\downarrow,I}$ with $I=\{i\in[1,n-2]:\pi_i\le m-1\}$. Thus $\sigma=\tau\circ(\upsilon+m)$ is uniquely determined.

Therefore, the interleaved code $\mathcal{C}=\mathcal{C}_1\circ(\mathcal{C}_2+m)$ is a 2-UBD permutation code in S_n. ■

We are now ready to give the decoding algorithm for the codes in Theorem 2.

Algorithm 1 $S_{n-t} \to S_n, 2 \geq t \geq 1$

Input: $m \geq 3, n = 2m, \pi = (\pi_1, \pi_2 \ldots, \pi_{n-t})$
Output: $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \in \mathcal{C}$

$m' \longleftarrow n - t - m + 1$
$f_1 \longleftarrow$ subsequence of π with entries from $[1, m-1]$
$\tau \longleftarrow$ correcting f_1 by a decoder of $\mathcal{C}_1$
$f_2 \longleftarrow$ subsequence of π with entries from $[m'+1, n-t]$
$\upsilon \longleftarrow$ correcting $f_2 - m'$ by a decoder of $\mathcal{C}_2$
$\sigma \longleftarrow \tau \circ (\upsilon + m)$

We illustrate the preceding algorithm with the following example.

Example 2. Let $m = 4$ and $n = 8$. Suppose that $\tau = (2,1,4,3) \in \mathcal{C}_1$ and $\upsilon = (1,3,4,2) \in \mathcal{C}_2$. Then $\sigma = \tau \circ (\upsilon + m) = (2,5,1,7,4,8,3,6) \in \mathcal{C}$.

If only one symbol 4 is deleted from σ, then $\pi = (2,4,1,6,7,3,5)$ is received. By Algorithm 1, $m' = 4$, $f_1 = (2,1,3)$, and $f_2 = (6,7,5)$. From f_1 and $f_2 - 4 = (2,3,1)$, we can recover $\tau = (2,1,4,3)$ and $\upsilon = (1,3,4,2)$ by the decoders of $\mathcal{C}_1$ and $\mathcal{C}_2$ in [11], respectively. Hence $\sigma = \tau \circ (\upsilon + m)$ is determined in this case.

If two adjacent symbols 7 and 4 are deleted, then $\pi = (2,4,1,6,3,5)$ is received. By Algorithm 1, $m' = 3$, $f_1 = (2,1,3)$, and $f_2 = (4,6,5)$. From f_1 and $f_2 - 3 = (1,3,2)$, again we can recover $\tau = (2,1,4,3)$ and $\upsilon = (1,3,4,2)$ by the decoders of $\mathcal{C}_1$ and $\mathcal{C}_2$. Hence $\sigma = \tau \circ (\upsilon + m)$ is also uniquely determined in this case.

Now we generalize the interleaving method in Theorem 2 to construct s-UBD permutation codes. We sketch the main idea of the decoding algorithm first. In Theorem 3, the code is constructed by interleaving s 1-UD permutation codes. Suppose we receive a permutation $\pi \in S_{n-t}$ with $t < s$. The main task is to find an appropriate permutation $\pi' \in S_{n-s}$ which is obtained from π by experiencing a burst of $s - t$ unstable deletions, and in such a way that π' is simultaneously obtained from the original permutation σ by suffering a burst of s unstable deletions. Hence we can recover σ by correcting the de-interleaved components of π'. Note that values from $[1, m-1]$ and $[n-t-m+2, n-t]$ in π are originally from $[1, m]$ and $[n-m+1, n]$ in σ, which are periodically placed in σ, respectively. The permutation π' is obtained by carefully checking the change of placement of these values in π. Due to lack of space, some explanations of facts are omitted in the proof.

Theorem 3. *Let $m, s \geq 3$ and $n = ms$. For each $i \in [1, s]$, let $\mathcal{C}_i$ be a 1-UD permutation code over $[1, m]$. Then the interleaved code $\mathcal{C} = \mathcal{C}_1 \circ (\mathcal{C}_2 + m) \circ \cdots \circ (\mathcal{C}_i + (i-1)m) \circ \cdots \circ (\mathcal{C}_s + (s-1)m)$ is an s-UBD permutation code over $[1, n]$.*

Proof. Suppose that a permutation $\sigma = \tau^1 \circ (\tau^2 + m) \circ \cdots \circ (\tau^i + (i-1)m) \circ \cdots \circ (\tau^s + (s-1)m) \in \mathcal{C}$, where $\tau^i \in \mathcal{C}_i$ for $i \in [1, s]$, suffers a burst of unstable deletions and a permutation $\pi = (\pi_1, \ldots, \pi_{n-t}) \in S_{n-t}$ is received. We show that σ is uniquely identifiable from π as follows. Let $P_i = [(i-1)m + 1, im]$ for each $i \in [1, s]$.

Case 1 ($t = s$): In this case, exactly one symbol in each P_i is deleted from σ since the unstable deletions are adjacent. By de-interleaving the permutation π, we have $f_i = (\pi_i, \pi_{i+s}, \ldots, \pi_{i+(m-2)s})$ for each $i \in [1, s]$. Then τ^i is uniquely determined from the permutation $f_i - (i-1)(m-1)$ by a decoder of $\mathcal{C}_i$ and consequently σ is recovered. Note that $f_i - (i-1)(m-1)$ is actually $\pi_{\downarrow, I_i}$ with $I_i = [1, n-s] \setminus \{i, i+s, \ldots, i+(m-2)s\}$.

Case 2 ($t < s$): In this case, we will get $\pi' = \pi_{\downarrow, I}$ for some positions set I of size $s - t$ and then apply Case 1 with π' to recover σ. Since there is a burst of t ($< s$) unstable deletions from σ, there is at most one symbol in each P_i that is deleted from σ. Let k_i be the positions of values from $[1, m-1]$ in π such that $k_i < k_{i+1}$, $i \in [1, m-1]$. Define $k_m := n - t + 1$ and let $d_j = k_{j+1} - k_j$, $j \in [1, m-1]$. Note that $k_1 \leq s + 1$.

Suppose $1 < k_1 \leq s$. Then the deleted locations are among the positions $[1, s]$ in σ. Let $\pi' = \pi_{\downarrow, [1, s-t]}$.

Suppose $k_1 = s + 1$. Then there must be a unique $j \in [1, m-1]$ such that $d_j = s - t$, and for all $i \in [1, m-1] \setminus \{j\}$, $d_i = s$. Let $\pi' = \pi_{\downarrow, [k_j, k_{j+1}-1]}$.

Suppose $k_1 = 1$. There are only two possible cases: (1) there is a unique $j \in [1, m-1]$ such that $d_j = s - t$; (2) there is a unique $j \in [1, m-1]$ such that $d_j = 2s - t$. If (1) happens, let $\pi' = \pi_{\downarrow, [k_j, k_{j+1}-1]}$. If (2) happens, let $R = [k_j, k_{j+1} - 1]$. Note that $|R| = 2s - t$ and a burst of unstable deletions have occurred in some positions from R in π.

Now we turn to the positions h_i of values from $[n - t - m + 2, n - t]$ in π such that $h_i < h_{i+1}$, $i \in [1, m-1]$. By similarly considering differences between h_i and h_{i+1}, we can recover σ for almost all cases except one case where a set $R' = [h_j, h_{j+1} - 1]$ of $2s - t$ positions in π is found for correction. Let $D = R \cap R'$, then errors occur in D. Suppose $|D| < s$, then let $\pi' = \pi_{\downarrow, [k_j, h_{j+1}]}$ if $h_j < k_j$ and $\pi' = \pi_{\downarrow, [h_j+1, k_{j+1}-1]}$ if $h_j > k_j$. It comes to the worst case if $|D| > s$. In this case, $h_j = k_j - 1$ and $h_{j+1} = k_{j+1} - 1$. We split it in two cases further.

Case 2a ($t = 1$): If $\pi_{k_{j+1}-s} \in P_1$, then we know that an error has occurred in a position from $[k_j, h_{j+1} - s]$ in π. Let $\pi' = \pi_{\downarrow, [k_j, h_{j+1}-s]}$. If not, let $\pi' = \pi_{\downarrow, [k_j+s, h_{j+1}]}$.

Case 2b ($1 < t < s$): We know that $\pi_{k_j} \in P_1$. If for all $i \in [1, s-1]$, $\pi_{k_j+i} \in P_i \cup P_{i+1}$, then let $\pi' = \pi_{\downarrow, [k_j+s, h_{j+1}]}$. Otherwise, find the smallest index $i \in [1, s-1]$ such that $\pi_{k_j+i} \notin (P_i \cup P_{i+1})$. Let $\pi' = \pi_{\downarrow, [k_j+i, k_j+i+s-t-1]}$.

Therefore, the interleaved code $\mathcal{C}$ is an s-UBD permutation code over $[1, n]$. ■

We only present an algorithm of computing π' from π when $t < s$ as Case 2 in the proof of Theorem 3. Once π' is computed, it is easy to recover σ by applying Case 1.

Now we give an example to illustrate Case 2b, which is the worst case in the proof of Theorem 3.

Example 3. Let $m = 5$, $s = 3$ and $n = 15$. Then $P_1 = [1, 5]$, $P_2 = [6, 10]$ and $P_3 = [11, 15]$. Suppose that $\tau^1 = (3,1,2,5,4) \in \mathcal{C}_1$, $\tau^2 = (2,5,1,4,3) \in$

Algorithm 2 $S_{n-t} \to S_{n-s}, s > t \geq 1$

Input: $m \geq 3, n = sm, \pi = (\pi_1, \pi_2 \dots, \pi_{n-t})$
Output: $\pi' = (\pi'_1, \pi'_2, \dots, \pi'_{n-s})$
  $m' \longleftarrow n - t - m + 2$
  $k_j \longleftarrow$ positions of entries $[1, m-1]$ in π, $j \in [1, m-1]$
  $h_j \longleftarrow$ positions of entries $[m', n-t]$ in π, $j \in [1, m-1]$
  $k_m \longleftarrow n - t + 1$; $k_0 \longleftarrow 1$
  for $j \in [0, m-1]$ **do**
    $d_j \longleftarrow k_{j+1} - k_j$
    if $d_j = s - t$ **then**
      return $\pi' \longleftarrow \pi_{\downarrow,[k_j,k_{j+1}-1]}$
    else if $d_j = 2s - t$ **then**
      $a \longleftarrow k_j, b \longleftarrow k_{j+1}$
  for $j \in [1, m-1]$ **do**
    if $a < h_j < a + s - 1$ **then**
      return $\pi' \longleftarrow \pi_{\downarrow,[a,h_j]}$
    else if $h_j = a + s - 1$ **then**
      return $\pi' \longleftarrow \pi_{\downarrow,[h_j+1,b-1]}$
  if $t = 1$ **then**
    if $\pi_{b-s} \leq m$ **then**
      return $\pi' \longleftarrow \pi_{\downarrow,[a,b-s-1]}$
    else
      return $\pi' \longleftarrow \pi_{\downarrow,[a+s,b-1]}$
  if $t > 1$ **then**
    for $i \in [1, s-1]$ **do**
      if $\pi_{a+i} \notin [(i-1)m + 1, (i+1)m]$ **then**
        return $\pi' \longleftarrow \pi_{\downarrow,[a+i,a+i+s-t-1]}$
    return $\pi' \longleftarrow \pi_{\downarrow,[a+s,b-1]}$

C_2 and $\tau^3 = (4, 2, 5, 1, 3) \in \mathcal{C}_3$. Then $\sigma = (3, 7, 14, 1, 10, 12, 2, 6, 15, 5, 9, 11, 4, 8, 13) \in \mathcal{C}$.

Let $t = 2$. Suppose the two adjacent symbols 12 and 2 are deleted, then $\pi = (2, 6, 12, 1, 9, 5, 13, 4, 8, 10, 3, 7, 11)$ is received. By checking the positions of values from $[1, 4]$ and $[10, 13]$ in π, we have $R = [4, 7]$ and $R' = [3, 6]$. That is $a = k_j = 4$, $b = 8$ in Algorithm 2, and there is no h_j in $[a, a + s - 1] = [4, 6]$. Since the smallest index $i \in [1, 2]$ such that $\pi_{k_j+i} \notin (P_i \cup P_{i+1})$ is 2, $\pi' = (2, 5, 11, 1, 8, 12, 4, 7, 9, 3, 6, 10) = \pi_{\downarrow,\{6\}}$. De-interleaving π', we have $f_1 = (2, 1, 4, 3)$, $f_2 = (5, 8, 7, 6)$ and $f_3 = (11, 12, 9, 10)$. Thus τ^i, $i = 1, 2, 3$ can be recovered from f_1, $f_2 - 4 = (1, 4, 3, 2)$ and $f_3 - 8 = (3, 4, 1, 2)$ by decoders of $\mathcal{C}_i$ respectively, and consequently σ is uniquely determined.

V. Conclusion

We present the first class of permutation codes that are capable of correcting a burst of up to s unstable deletions, for general s. Efficient decoding algorithms are provided to show the correctness of our constructions.

Since each 1-UD permutation code of length n/s has size at least $\frac{(n/s)!}{(n/s)^2}$ [11], the code we construct in Theorem 3 has size at least $\left(\frac{(n/s)!}{(n/s)^2}\right)^s$. Although this is not optimal with respect to the upper bound we derive in Theorem 1, its rate is asymptotically

$$\frac{s \ln (n/s)! - 2s \ln (n/s)}{\ln n!} \sim \frac{(n - 2s) \ln n + O(n)}{n \ln n + O(n)} \sim 1,$$

for fixed s.

Similar results for the case of stable deletions have also been obtained recently by Chee et al. [21], and will be reported elsewhere.

References

[1] B. Eitan and A. Roy, "Binary and multilevel flash cells," in *Flash Memories*, P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, Eds. Kluwer Acad. Publ., 1999, pp. 91–152.

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

[3] E. E. Gad, E. Yaakobi, A. Jiang, and J. Bruck, "Rank-modulation rewriting codes for flash memories," in *ISIT 2013 – Proceedings of the 2013 IEEE International Symposium on Information Theory*, 2013, pp. 704–708.

[4] Y. M. Chee, H. M. Kiah, and P. Purkayastha, "Rewritable coset coding for flash memories," in *ISIT 2014 – Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014, pp. 2082–2086.

[5] Y. Cai, G. Yalcin, O. Mutlu, E. F. Haratsch, A. Cristal, O. S. Unsal, and K. Mai, "Error analysis and retention-aware error management for NAND flash memory," *Intel Technology Journal*, vol. 17, no. 1, pp. 140–164, 2013.

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

[7] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," *IEEE Trans. Inform. Theory*, vol. 56, no. 7, pp. 3158–3165, 2010.

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

[9] S. Buzaglo, E. Yaakobi, T. Etzion, and J. Bruck, "Error-correcting codes for multipermutations," in *ISIT 2013 – Proceedings of the 2013 IEEE International Symposium on Information Theory*, 2013, pp. 724–728.

[10] R. Gabrys, E. Yaakobi, F. Farnoud, and J. Bruck, "Codes correcting erasures and deletions for rank modulation," in *ISIT 2014 – Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014, pp. 2759–2763.

[11] R. Gabrys, E. Yaakobi, F. Farnoud, F. Sala, J. Bruck, and L. Dolecek, "Single-deletion-correcting codes over permutations," in *ISIT 2014 – Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014, pp. 2764–2768.

[12] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron. Device Lett.*, vol. 23, no. 5, pp. 264–266, 2002.

[13] K. Prall, "Scaling non-volatile memory below 30nm," in *IEEE NVSMW 2007 – Proceedings of the 21st IEEE Non-Volatile Semiconductor Memory Workshop*, 2007, pp. 5–10.

[14] V. I. Levenshteĭn, "Perfect codes in the metric of deletions and insertions," *Diskret. Mat.*, vol. 3, no. 1, pp. 3–20, 1991.

[15] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Permutation arrays under the Chebyshev distance," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2611–2617, 2010.

[16] M. Blaum, J. Bruck, and A. Vardy, "Interleaving Schemes for Multidimensional Cluster Errors," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp.730–743, 1988.

[17] W. Q. Xu and S. W. Golomb, "Optimal interleaving schemes for correcting two-dimensional cluster errors," *Discrete Applied Mathematics*, vol. 155, no. 10, pp 1200–1212, 2007.

[18] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors," *Autom. Remote Control*, vol. 26, no. 2, pp. 286–290, 1965.

[19] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion," *IEEE Trans. Inform. Theory*, vol. 30, no. 5, pp. 766–769, 1984.

[20] C. Homberger, "Counting fixed-length permutation patterns," *Online J. Anal. Comb.*, no. 7, pp. 12, 2012.

[21] Y. M. Chee, T. T. Nguyen, and V. K. Vu, "Permutation codes correcting a single burst deletion II: Stable deletions," *preprint*, 2014.