NEW MULTIPLE INSERTION-DELETION CORRECTING CODES FOR NON-BINARY ALPHABETS

TUAN A. LE AND HIEU D. NGUYEN

ABSTRACT. We generalize Helberg’s number-theoretic construction of multiple insertion-deletion correcting binary codes to non-binary alphabets. We prove that these codes are able to correct multiple deletion errors and also correct half as many substitution errors. We also present values for the size of the largest code for certain codeword lengths that were found through exhaustive computer search.

1. Introduction

In this paper we extend Helberg’s construction of multiple insertion-deletion correcting binary codes as described in [2] and [3] to non-binary alphabets. We prove that these codes are able to correct multiple deletion errors and also correct half as many substitution errors. Our proof for correction of deletion errors follows the proof given by Abdel-Ghaffar, Palmučić, Ferreira, and Clarke [1], which we adapt for non-binary alphabets. Helberg codes generalize Levenshtein codes (first constructed by Varshamov and Tenengolts [5]), which Levenshtein proved in [4] are capable of correcting a single insertion-deletion error.

Let $A = \{0, 1, \ldots, q-1\}$ be a $q$-ary alphabet and $x = (x_1, \ldots, x_n) \in A^n$ a codeword of length $n$. Each value $x_i$ will be called a bit of $x$. Fix $d$ to be a positive integer and set $p = q - 1$. Following [3], we define the sequence of weights $W(q, d) = \{w_1, w_2, \ldots\}$ by the recursion

$$w_i = 1 + p \sum_{j=1}^{d} w_{i-j},$$

where we assume $w_i = 0$ for $i \leq 0$. Next, we define the moment of $x$ with respect to $W(q, d)$ by

$$M(x) = \sum_{i=1}^{n} w_i x_i.$$

If $m$ and $r$ are fixed positive integers satisfying $m \geq w_{n+1}$ and $0 \leq r < m$, we define the non-binary code $C_n(q, d, m, r)$ to be the set of codewords of length $n$ whose moments have residue $r$ modulo $m$, i.e.,

$$C_n := C_n(q, d, m, r) = \{x \in A^n : M(x) \equiv r \mod m\}.$$

In the case of a binary alphabet where $q = 2$, the codes $C_n(2, d, m, r)$ are known as Helberg codes as defined in [3]. We prove in section 2 that $C_n(2, d, m, r)$ is able to correct $d$-deletion errors and prove in section 3 that it is also able to correct $\lfloor d/2 \rfloor$ substitution errors. In section 4, we present values for the size of the largest code $C_n(q, d, m, r)$ for certain values of $q$, $d$, and $n$. These values were found through exhaustive computer search.

2. Proof of Multiple Insertion-Deletion Correction

Our proof that $C_n(q, d, m, r)$ is a $d$-deletion error-correcting code follows the proof given in [1], where we adapt their arguments for $q$-ary alphabets. We begin with some preliminaries. Given two codewords $x$ and $y$ of length $n$, we shall write $x \equiv y$ to mean $M(x) \equiv M(y) \mod m$. In that case, there exists a residue $r$ such that $x, y \in C_n(q, d, m, r)$. Moreover, if we define $\Delta(x, y) = M(x) - M(y)$, then $x \equiv y$ is equivalent to $\Delta(x, y) \equiv 0 \mod m$.
Define $S(n) = \{1, ..., n\}$. Let $D$ be a non-empty subset of $S(n)$ with $|D| \leq d$. Set $n' = n - |D|$ and define $S' = S(n) - D = \{i_1, ..., i_{n'}\}$ with $i_1 < i_2 < ... < i_{n'}$. Moreover, define $\mathbf{x}^{(D)} = (x_{i_1}, ..., x_{i_{n'}})$ to be the codeword obtained by deleting the elements of $\mathbf{x}$ indexed by $D$. We shall refer to $\mathbf{x}^{(D)}$ to be a deleted codeword of $\mathbf{x}$. We also define the index of $\mathbf{x}^{(D)}$ with respect to $\mathbf{x}$ to be

$$I := I(\mathbf{x}^{(D)}) = M(\mathbf{x}) - M(\mathbf{x}^{(D)})$$

We shall need the following lemma, which allows us to replace the rightmost non-zero bit with the value 0 in any two codewords that are equivalent and have the same deleted codeword. This assumes that the rightmost nonzero bit is the same for both codewords.

**Lemma 1.** Let $\mathbf{x}$ and $\mathbf{y}$ be two codewords of length $n$ with the following two properties:

1. $\mathbf{x} \equiv \mathbf{y}$.
2. $\mathbf{x}^{(D)} = \mathbf{y}^{(E)}$ for some subsets $D$ and $E$ of $\{1, ..., n\}$ with $|D| = |E| \leq d$.

Suppose there exists a positive index $L$ such that $x_L = y_L > 0$ and $x_i = y_i = 0$ for all $i > L$. Then there exist codewords $\tilde{x}$ and $\tilde{y}$ where $\tilde{x}_i = x_i$, $\tilde{y}_i = y_i$ for all $i \neq L$ and $\tilde{x}_L = \tilde{y}_L = 0$ such that $\tilde{x}$ and $\tilde{y}$ have the same two properties as $\mathbf{x}$ and $\mathbf{y}$, namely

1. $\tilde{x} \equiv \tilde{y}$.
2. $\tilde{x}^{(D)} = \tilde{y}^{(E)}$ for some sets $\tilde{D}$ and $\tilde{E}$ having the same size as $D$ and $E$.

**Proof.** Define $\tilde{x}$ and $\tilde{y}$ according to the lemma. Since $x_i - y_i = \tilde{x}_i - \tilde{y}_i$ for all $i = 1, ..., n$, it follows that $M(\mathbf{x}) - M(\mathbf{y}) = M(\tilde{x}) - M(\tilde{y})$. But $\mathbf{x} \equiv \mathbf{y}$; hence, $\tilde{x} \equiv \tilde{y}$. This proves (1). To prove (2), consider four cases:

Case I: Assume $i_L \in D \cap E$. In this case, the nonzero bits $x_{i_L}$ and $y_{i_L}$ are deleted from $\mathbf{x}$ and $\mathbf{y}$, respectively, to obtain $\mathbf{x}^{(D)}$ and $\mathbf{y}^{(E)}$. Define $\tilde{D} = D$ and $\tilde{E} = E$. Since $\mathbf{x}^{(D)} = \mathbf{y}^{(E)}$, it follows that $\tilde{x}^{(D)} = \tilde{y}^{(E)}$ since the zero bits $\tilde{x}_{i_L}$ and $\tilde{y}_{i_L}$ are deleted from $\tilde{x}$ and $\tilde{y}$, respectively, as well.

Case II: Assume $i_L \notin D \cup E$. Since $\mathbf{x}^{(D)} = \mathbf{y}^{(E)}$, it follows that $x_{i_L}$ and $y_{i_L}$ appear in $\mathbf{x}^{(D)}$ and $\mathbf{y}^{(E)}$ as the rightmost nonzero bit, respectively. But then replacing $x_{i_L}$ and $y_{i_L}$ by $\tilde{x}_{i_L}$ and $\tilde{y}_{i_L}$, respectively, yields $\tilde{x}^{(D)} = \tilde{y}^{(E)}$. Thus, it suffices to again define $\tilde{D} = D$ and $\tilde{E} = E$.

Case III: Assume $i_L \in D - E$. In this case, the bit $x_{i_L}$ is deleted from $\mathbf{x}$ to obtain $\mathbf{x}^{(D)}$, but the bit $y_{i_L}$ is not deleted from $\mathbf{y}$ and therefore appears in $\mathbf{y}^{(E)}$. Let $z$ denote the number of bits to the right of $y_{i_L}$ in $\mathbf{y}^{(E)}$, which must all be 0 since $y_{i_L} = 0$ for all $i > i_L$. Then the number of bits to the right of $y_{i_L}$ that are deleted from $\mathbf{y}$ to obtain $\tilde{y}^{(E)}$ equals $z' = n - i_L - z$. Let $x_K$ denote the rightmost nonzero bit of $\mathbf{x}^{(D)}$. Since $\mathbf{x}^{(D)} = \mathbf{y}^{(E)}$, it follows that $x_K = y_{i_L} = x_{i_L}$ and the number of zeros to the right of $x_K$ in $\mathbf{x}^{(D)}$ also equals $z$. Therefore, the number of bits to the right of $x_K$ that are deleted from $\mathbf{x}$ to obtain $\mathbf{x}^{(D)}$ equals $n - K - z$. We now define $D' = \{K, K + 1, ..., i_L - 1, i_L + 1, ..., i_{n'} + z'\}$. It follows that $\mathbf{x}^{(D')} = \mathbf{x}^{(D)}$ with $|D'| = |D|$. Since $i_L \notin D' \cup E$, this reduces to Case II where $D$ is replaced by $D'$.

Case IV: Assume $i_L \in E - D$. The argument in this case is the same as Case III with the roles of $D$ and $E$ reversed. \hfill \Box

**Theorem 2.** Let $\mathbf{x}$ and $\mathbf{y}$ be two codewords of length $n$ that satisfy properties (1) and (2) in Lemma[1]. Then $0 < |\Delta(\mathbf{x}, \mathbf{y})| < m$.

**Proof.** We shall first prove that $|\Delta(\mathbf{x}, \mathbf{y})| < m$. To begin, we rewrite $\Delta(\mathbf{x}, \mathbf{y})$ as follows:

$$\Delta(\mathbf{x}, \mathbf{y}) = M(\mathbf{x}) - M(\mathbf{y})$$

$$= \sum_{i \in D} w_i x_i - \sum_{j \in E} w_j y_j + \sum_{k=1}^{n'} (w_{i_k} - w_{j_k}) x_{i_k}$$

$$\leq \sum_{i \in D} w_i x_i + \sum_{k=1}^{n'} (w_{i_k} - w_{j_k}) x_{i_k}$$
Next, we partition \( S(n') = \{1, 2, \ldots, n'\} \) into those elements \( k \) where \( i_k \leq j_k \) and those where \( i_k > j_k \) to obtain

\[
\Delta(x, y) \leq \sum_{i \in D} w_i x_i + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k} + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]

\[
\leq \sum_{i \in D} w_i x_i + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]

\[
\leq \sum_{i \in D} pw_i + \sum_{k \in S(n')} p(w_{i_k} - w_{j_k})
\]

\[
= \sum_{i \in D} pw_i + \sum_{k \in S(n')} pw_{i_k} - \sum_{k \in S(n')} pw_{j_k}
\]

We now add and subtract as follows:

\[
\Delta(x, y) \leq \sum_{i \in D} pw_i + \sum_{k \in S(n')} pw_{i_k} + \sum_{k \in S(n')} pw_{i_k} - \sum_{k \in S(n')} pw_{i_k} = \sum_{i = 1}^{n} pw_i - \sum_{k = 1}^{n'} pw_{\min(w_{i_k}, w_{j_k})} \leq \sum_{i = 1}^{n} pw_i - \sum_{k = 1}^{n'} pw_k
\]

\[
\leq \sum_{i = n' + 1}^{n} pw_i = \sum_{j = 1}^{n - n'} pw_{n+1 - j}
\]

\[
\leq p \sum_{j = 1}^{d} w_{n+1 - j} = w_{n+1} - 1
\]

\[
\leq m - 1
\]

\[
< m
\]

On the other hand, by reversing \( x \) and \( y \), we obtain \( \Delta(y, x) < m \), which implies \( \Delta(x, y) = -\Delta(y, x) > -m \). Hence, \( |\Delta(x, y)| < m \) as desired.

Next, we prove that \( \Delta(x, y) \neq 0 \) by considering four different cases. By Lemma 1 we can assume without loss of generality that there exists \( L \in \{1, \ldots, n\} \) such that \( x_L > y_L \) and \( x_i = y_i = 0 \) for all \( i > L \).

Case 1: Assume \( L \in D \cap E \). Then \( i_k \neq L \) for all \( k = 1, \ldots, n' \). It follows that

\[
\Delta(x, y) = M(x) - M(y) = \sum_{i \in D} w_i x_i - \sum_{j \in E} w_j y_j + \sum_{k = 1}^{n'} (w_{i_k} - w_{j_k}) x_{i_k}
\]

\[
= w_L x_L + \sum_{i \in D} w_i x_i - w_L y_L - \sum_{j \in E} w_j y_j + \sum_{k = 1}^{n'} (w_{i_k} - w_{j_k}) x_{i_k}
\]

\[
\geq w_L (x_L - y_L) - \sum_{j \in E} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k} + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]

\[
\geq w_L - \sum_{j \in E} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]
where we have used the fact that $i_k \neq L$ and $x_{i_k} = 0$ for $i_k > L$. Since $x_{i_k} \leq p$, it follows that

$$\Delta(x, y) \geq w_L - \sum_{j \in E \atop j \leq L-1} pw_j + \sum_{k \in S(n')} p(w_{i_k} - w_{j_k})$$

$$= w_L - \sum_{j \in E \atop j \leq L-1} pw_j + \sum_{k \in S(n')} pw_{i_k} - \sum_{k \in S(n')} pw_{j_k}$$

We now add and subtract as follows:

$$\Delta(x, y) \geq w_L - \sum_{j \in E \atop j \leq L-1} pw_j + \sum_{k \in S(n')} pw_{i_k} - \sum_{k \in S(n')} pw_{j_k} + \sum_{k \in S(n')} pw_{j_k}$$

$$\geq w_L - \sum_{j = 1}^{L-1} pw_j + \sum_{i \in \min(n', L-1)} pw_i - \sum_{k \in S(n')} pw_{j_k}$$

$$\geq w_L - \sum_{i = \min(n', L-1)}^{L-1} pw_i \quad (\text{since } L - d \leq \min(n', L-1); \text{ recall } L \leq n = n' + d \text{ and } d \geq 1)$$

$$= 1$$

Case II: Assume $L \in D - E$. Recall that $x_L > y_L$ and $x_i = y_i = 0$ for $i > L$. Since $L \notin E$, it follows that

$$\Delta(x, y) = M(x) - M(y) = \sum_{i \in D} w_ix_i - \sum_{j \in E} w_jy_j + \sum_{k = 1}^{n'} (w_{i_k} - w_{j_k})x_{i_k}$$

$$= w_Lx_L + \sum_{i \in D \atop i \leq L-1} w_ix_i - \sum_{j \in E \atop j \leq L-1} w_jy_j + \sum_{k = 1}^{n'} (w_{i_k} - w_{j_k})x_{i_k}.$$
Analogously, we partition $S(n')$ into those elements $k$ where $i_k < j_k$ and those where $i_k \geq j_k$ to obtain
\[
\Delta(x, y) \geq w_L x_L - \sum_{j \leq L-1} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k} + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]
\[
\geq w_L x_L - \sum_{j \leq L-1} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]
\[
\geq w_L - \sum_{j \leq L-1} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]

The rest of the argument now follows the same as that in Case I. Therefore, $\Delta(x, y) \geq 1$.

Case III. Assume $L \in E - D$. The argument in this case is the same as Case II by switching the roles of $D$ and $E$.

Case IV. Assume $L \notin D \cup E$. Then $i_K = L$ for some $i_K \in S'$. We claim that $j_K \leq i_K - 1$. Since $x^{(D)} = y^{(E)}$, it follows that $x_{i_K} = y_{j_K}$. On the other hand, we have $y_{jK} < x_{i_K}$ and $y_i = 0$ for all $i \geq L = i_K$. Thus, $j_K \leq i_K - 1$.

We now proceed similarly as in previous cases:
\[
\Delta(x, y) = M(x) - M(y) = \sum_{i \in D} w_i x_i - \sum_{j \in E} w_j y_j + \sum_{k=1}^{n'} (w_{i_k} - w_{j_k}) x_{i_k}
\]
\[
= \sum_{i \in D} w_i x_i - \sum_{j \leq L-1} w_j y_j + \sum_{k=1}^{n'} (w_{i_k} - w_{j_k}) x_{i_k}
\]
\[
\geq - \sum_{j \leq L-1} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k} + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k}
\]
\[
\geq - \sum_{j \leq L-1} w_j y_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k}) x_{i_k} + (w_{i_k} - w_{j_k}) x_{i_k}
\]

Next, since $x_i \leq p$ for all $i \in \mathbb{N}$, we have
\[
\Delta(x, y) \geq w_L - p w_{jK} - \sum_{j \leq L-1} p w_j + \sum_{k \in S(n')} (w_{i_k} - w_{j_k})
\]
\[
= w_L - p w_{jK} - \sum_{j \leq L-1} p w_j + \sum_{k \in S(n')} p w_{i_k} - \sum_{k \in S(n')} p w_{j_k}
\]
\[
= w_L - p w_{jK} - \sum_{j \leq L-1} p w_j + \sum_{k \in S(n')} p w_{i_k} - \sum_{k \in S(n')} p w_{j_k}
\]
\[
+ \sum_{k \in S(n')} p w_{j_k} - \sum_{k \in S(n')} p w_{j_k}
\]
\[
= w_L - \sum_{j=1}^{L-1} p w_j + \sum_{k \in S(n')} p w_{i_k} + \sum_{k \in S(n')} p w_{j_k}
\]

The rest of the proof now follows the same as that in Case I. Therefore, $\Delta(x, y) \geq 1$. Hence, $0 < |\Delta(x, y)| < m$ as desired.
Theorem 3. The code $C_n(q, d, m, r)$ is a $d$-insertion-deletion correcting code.

Proof. Suppose on the contrary that $C_n(q, d, m, r)$ is not capable of correcting up to $d$ deletions. Then there exist codewords $x, y \in C_n(q, d, m, r)$ and subsets $D$ and $E$ with $|D|= |E|\leq d$ such that $x^{(D)} = y^{(E)}$. By Theorem 2, we have $0 < |\Delta(x, y)| < m$. It follows that $x \not\equiv y$, which contradicts the fact that $x$ and $y$ belong to the same code. Thus, $C_n(q, d, m, r)$ is capable of correcting up to $d$ deletions, and therefore, can correct up to $d$ insertion-deletion errors as well due to a result of Levenshtein [3].

3. Correction of Substitution Errors

In this section, we demonstrate the ability of the code $C_n(q, d, m, r)$ to also correct up to $\lfloor d/2 \rfloor$ substitution errors, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to $x$. We define the Hamming distance between any two codewords $x$ and $y$ by

$$h(x, y) = \sum_{i=0}^{n} \delta(x_i, y_i),$$

where $\delta(x_i, y_i)$ is the Kronecker delta function. The minimum Hamming distance of the code $C_n(q, d, m, r)$ is then defined by

$$h_n(q, d, m) = \min\{h(x, y) : x, y \in C_n(q, d, m, r)\}.$$

It is well known that if a code has minimum Hamming distance of $e+1$, then it is able to detect $e$ substitution errors and if the minimum Hamming distance is $2e + 1$, then it able to correct $e$ substitution errors.

Theorem 4. We have

$$h_n(q, d, m) \geq d + 1.$$ 

Thus, the code $C_n(q, d, m, r)$ is able to correct $\lfloor d/2 \rfloor$ substitution errors.

Proof. Suppose on the contrary that $h_n(q, d, m) < d + 1$. Then there exists codewords $x, y \in C_n(q, d, m, r)$ such that $x \equiv y$ and $h(x, y) = c < d + 1$. Let $1 \leq i_1 < i_2 < \cdots < i_c \leq n$ denote the $c$ positions in which $x$ and $y$ differ, i.e., $x_{i_j} \neq y_{i_j}$ for $j = 1, 2, ..., c$. Then

$$|\Delta(x, y)| = |M(x) - M(y)| = \left| \sum_{j=1}^{n} (x_i - y_i)w_i \right| = \sum_{j=1}^{c} |x_{i_j} - y_{i_j}|w_{i_j} < 1 + p \sum_{j=1}^{d} w_{n+1-j} = w_{n+1} \leq m$$

On the other hand, we claim $\Delta(x, y) \neq 0$. Otherwise, we have

$$\Delta(x, y) = M(x) - M(y) = \sum_{j=1}^{c} (x_{i_j} - y_{i_j})w_{i_j} = 0,$$

which implies

$$\left| \sum_{j=1}^{c-1} (x_{i_j} - y_{i_j})w_{i_j} \right| = |x_{i_c} - y_{i_c}|w_{i_c}.$$ 

But this is impossible since

$$\left| \sum_{j=1}^{c-1} (x_{i_j} - y_{i_j})w_{i_j} \right| \leq p \sum_{j=1}^{c-1} w_{i_j} < 1 + p \sum_{j=1}^{d} w_{i_{c-j}} = w_{i_c} \leq |x_{i_c} - y_{i_c}|w_{i_c}.$$ 

Thus, $0 < |\Delta(x, y)| < m$, which contradicts the fact that $M(x) \equiv M(y) \mod m$. 


Table 1. Binary 2-Deletion Codes: Values of $N_n(2, 2)$ and $R_n(2, 2)$

| $n$ | $N_n(2, 2)$ | $R_n(2, 2)$ |
|-----|-------------|-------------|
| 1   | 1           | 0, 1        |
| 2   | 1           | 0, 1, 2, 3  |
| 3   | 2           | 0          |
| 4   | 2           | 0, 1, 2, 7  |
| 5   | 2           | 0, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 19 |
| 6   | 4           | 12, 13     |
| 7   | 5           | 12, 33     |
| 8   | 6           | 12, 33, 39, 45, 66 |
| 10  | 8           | 66         |
| 11  | 9           | 65, 66, 99, 100, 120, 121, 154, 155 |
| 12  | 11          | 65, 66, 99, 154, 155, 175, 176, 181, 182, 187, 188, 208, 209, 264, 297, 298 |
| 13  | 15          | 297, 298   |
| 14  | 18          | 297, 441, 475, 496, 530, 674 |
| 15  | 22          | 297, 441, 674, 763, 784, 790, 796, 817, 906, 1139, 1283 |
| 16  | 30          | 1283       |

Table 2. Ternary 2-Deletion Codes: Values of $N_n(3, 2)$ and $R_n(3, 2)$

| $n$ | $N_n(3, 2)$ | $R_n(3, 2)$ |
|-----|-------------|-------------|
| 1   | 1           | 0, 1, 2     |
| 2   | 1           | 0, 1, 2, 3, 4, 5, 6, 7, 8 |
| 3   | 2           | 0, 1        |
| 4   | 2           | 0, 1, 2, 3, 4, 5, 6, 7, 25, 26, 50, 51 |
| 5   | 3           | 0, 25       |
| 6   | 4           | 25, 50      |
| 7   | 4           | 24, 25, 50, 69, 70, 71, 72, 73, 74, 75, 94, 119, 138, 139, 140, 141, 142, 143, 144, 163, 188, 189, 542, 567, 1059, 1084 |
| 8   | 5           | 24, 25, 49, 50, 69, 70, 71, 72, 73, 74, 75, 188, 189, 213, 214, 377, 378, 402, 403, 517, 518, 519, 520, 521, 522, 541, 542, 566, 567 |
| 9   | 7           | 541, 542, 566, 567, 1058, 1059, 1083, 1084 |
| 10  | 8           | 517, 518, 519, 520, 521, 541, 542, 566, 567, 1437, 1482, 1483, 1484, 1485, 1486, 1487, 1551, 1552, 1553, 1554, 1555, 1556, 1601, 2850, 2859, 2896, 2897, 2898, 2899, 2900, 2964, 2965, 2966, 2967, 2968, 2969, 3014, 3884, 3885, 3909, 3910, 3930, 3931, 3932, 3933, 3934 |

4. Maximum Code Size

In this section we present values for the size of the largest code in terms of the codeword length. Given positive integers $q$, $d$, and $n$, we set $m = w_{n+1}$ and denote the size of the largest code $C_n(q, d, m, r)$, where $r$ ranges through all residues, by

$$N_n(q, d) = \max \{|C_n(q, d, w_{n+1}, r)|: r = 0, 1, \ldots, w_{n+1} - 1\}.$$  

Also, let $R_n(q, d)$ denote the set of values $r$ for which $|C_n(q, d, w_{n+1}, r)| = N_n(q, d)$.

Through exhaustive computer search, we computed the values of $N_n(q, d)$ and $R_n(q, d)$ for certain values of $q$, $d$, and $n$. Table 1 gives values for $N_n(2, 2)$ and $R_n(2, 2)$ for binary 2-deletion codes ($q = 2, d = 2$) with $n$ ranging from 1 to 15. Tables 2 and 3 give values for ternary 2-deletion codes ($q = 3, d = 2$) and quaternary 2-deletion codes ($q = 4, d = 2$), respectively, but over a shorter range for $n$.  

7
Table 3. Quaternary 2-Deletion Codes: Values of \( N_n(4,2) \) and \( R_n(4,2) \)

| \( n \) | \( N_n(4,2) \) | \( R_n(4,2) \) |
|---|---|---|
| 1 | 1 | 0, 1, 2, 3 |
| 2 | 1 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 |
| 3 | 2 | 0, 1, 2 |
| 4 | 2 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 61, 62, 63, 122, 123, 124, 183, 184, 185 |
| 5 | 3 | 0, 1, 61, 62 |
| 6 | 4 | 61, 62, 122, 123, 183, 184 |
| 7 | 5 | 61, 880 |
| 8 | 6 | 61, 122, 183, 880, 941, 1760, 1821, 2640, 2701, 3398, 3459, 3520 |

References

[1] K. A. S. Abdel-Ghaffar, F. Palunčić, H. C. Ferreira, and W. A. Clarke, *On Helberg’s generalization of the Levenshtein code for multiple deletion/insertion error correction*, IEEE Trans. Inf. Theory 58 (2012), No. 3, 1804-1808.

[2] A. S. J. Helberg, *Coding for the correction of synchronization errors*, Ph.D. dissertation, Rand Afrikaans Univ., Johannesburg, South Africa, 1993.

[3] A. S. J. Helberg and H. C. Ferreira, *On multiple insertion/deletion correcting codes*, IEEE Trans. Inf. Theory, vol. 48, no. 1, pp. 305-308, Jan. 2002.

[4] V. I. Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals*, Sov. Phys.-Dokl., vol. 10, no. 8, pp. 707-710, Feb. 1966.

[5] R. R. Varshamov and G. M. Tenengol’ts, *Codes which correct single asymmetric errors*, Autom. Remote Control, vol. 26, no. 2, pp. 286-290, 1965.

Department of Mathematics, Rowan University, Glassboro, NJ 08028.

E-mail address: let5@students.rowan.edu, nguyen@rowan.edu