Explicit Constructions and Bounds for Batch Codes with Restricted Size of Reconstruction Sets

Eldho K. Thomas and Vitaly Skachek
Institute of Computer Science
University of Tartu, Estonia
Email: eldho.thomas@ut.ee, vitaly.skachek@ut.ee

Abstract—Linear batch codes and codes for private information retrieval (PIR) with a query size \( t \) and a restricted size \( r \) of the reconstruction sets are studied. New bounds on the parameters of such codes are derived for small values of \( t \) or of \( r \) by providing corresponding constructions. By building on the ideas of Cadambe and Mazumdar, a new bound in a recursive form is derived for batch codes and PIR codes.

I. INTRODUCTION

Batch codes are proposed in [1] for load balancing in the distributed server systems. They can be broadly classified as linear batch codes and combinatorial batch codes. A particular version of the former is known as switch codes and were first appeared in [2–4] in the context of network switches. Some works on combinatorial batch codes can be found in [5–7].

Locally repairable codes (LRCs), or codes with locality, which are deeply studied in [2–13], share lots of similarities with batch codes and therefore many of the bound computation and analysis are expected to be related to each other. In [14], new upper bounds on the parameters of batch codes based on the classical Singleton bound which do not depend on the size of the underlying alphabet are computed. The batch codes turn out to be a special case of private information retrieval (PIR) codes [8], though PIR codes support only queries of type \( (x_i, x_{i_1}, \ldots, x_{i_t}) \), \( 1 \leq i \leq k \), whereas batch codes support queries of a more general form \( (x_{i_1}, x_{i_2}, \ldots, x_{i_t}) \), possibly for different indices \( i_1, i_2, \ldots, i_t \). It follows that batch codes can be used as PIR codes. In [15], the batch codes with unrestricted size of reconstruction sets are considered and some bounds on the optimal length of batch and PIR codes for a given batch size and dimension are proposed.

In this work, we construct new families of batch codes with restricted size \( r \) of reconstruction sets. We also generalize existing bounds on the dimension of LRCs proposed in [12] to batch codes using the connections between the two families. This paper is organized as follows. In Section III, we propose an optimal construction of batch codes with \( t = 2, r \geq 2 \) and a construction with \( t \geq 3, r = 2 \). In Section IV, we present constructions of batch codes with \( t = 3 \) and \( r \geq 3 \). In Section V, we derive a new upper bound on the dimension \( k \) of the batch codes.

II. NOTATIONS

We start with introducing some notations. Denote by \( \mathbb{N} \) the set of nonnegative integers. For \( n \in \mathbb{N} \) we denote \([n] = \{1, 2, \ldots, n\}\). A \( k \times k \) identity matrix will be denoted by \( I_k \), all-one column vector by \( \mathbf{1} \). We use \( 0 \) to denote an all-zero column vector and a zero matrix. The right dimensions will be clear from the context. Let \( x \) be a vector of length \( n \) indexed by \([n]\). Take \( S \subseteq [n] \). Then \( x_S \) stands for a sub-vector of \( x \) indexed by \( S \). If \( A \) is a matrix, then \( A[i] \) denotes the \( i \)-th column in \( A \).

Let \( Q \) be a finite alphabet. Consider an information vector \( x = (x_1, x_2, \ldots, x_k) \in Q^k \). The code is a set of vectors \( \{y = C(x) \mid x \in Q^k\} \subseteq Q^n \), where \( C : Q^k \rightarrow Q^n \) is a bijective mapping, and \( n \in \mathbb{N} \). By slightly abusing the notation, \( C \) will also be used to denote the above code.

In this work, we study (primitive, multiset) batch codes with restricted size of the recovery sets, as they are defined in [14] (see also [15]).

Definition 1. An \((n, k, r, t)\) batch code \( C \) over a finite alphabet \( Q \) is defined by an encoding mapping \( C : Q^k \rightarrow Q^n \), and a decoding mapping \( D : Q^n \times [k]^t \rightarrow Q^t \), such that

1) For any \( x \in Q^k \) and a multiset \( (i_1, i_2, \ldots, i_t) \subseteq [k]^t \),

\[ D(y = C(x), i_1, i_2, \ldots, i_t) = (x_{i_1}, x_{i_2}, \ldots, x_{i_t}). \]

2) The query \( (x_{i_1}, x_{i_2}, \ldots, x_{i_t}) \) can be reconstructed from the data read by \( t \) different users independently, where each user reads at most \( r \) symbols of \( y \).

If the alphabet \( Q \) is a finite field, and the associated encoding mapping \( C : Q^k \rightarrow Q^n \) is linear over \( Q \), the corresponding code is termed linear. In that case, for each fixed \((i_1, i_2, \ldots, i_t) \subseteq [k]^t \), the corresponding decoding mapping \( D \) from \( Q^n \) to \( Q^t \) is linear over \( Q \) too. Additionally, if the encoding mapping \( C : x \mapsto y \) is such that \( x \) is a sub-vector of \( y \), then the corresponding code is called systematic.

This setup has first appeared in [16]. It was shown therein that the minimum distance \( d \) of a batch code satisfies \( d \geq t \). Batch codes are closely related to locally-repairable codes, which have been extensively studied in the context of the distributed data storage. The main difference between them is that in batch codes we are interested in the reconstruction of information symbols in \( x \), while in locally-repairable codes we are interested in the recovery of coded symbols in \( y \).
The following property of the batch codes is stated as Corollary III.2 in [14].

**Lemma 1.** Let \( C \) be a linear \((n,k,r,t)\) batch code over \( \mathbb{Q} \), and \( x \in \mathbb{Q}^k \), whose encoding is \( y \in \mathbb{C} \). Let \( R_1, R_2, \ldots, R_r \subseteq \{1,2,\ldots,n\} \) be \( t \) disjoint recovery sets for the coordinate \( x_i \). Then, there exist indices \( a_2 \in R_2, a_3 \in R_3, \ldots, a_t \in R_t \), such that if we fix the values of all coordinates of \( y \) indexed by the sets \( R_1 \setminus \{a_2\}, R_3 \setminus \{a_3\}, \ldots, R_t \setminus \{a_t\} \), then the values of the coordinates of \( y \) indexed by \( \{a_2, a_3, \ldots, a_t\} \) are uniquely determined.

It is proven in [14] that for a linear \((n,k,r,t)\) batch code over \( \mathbb{Q} \),
\[
 n \geq k + d + \max_{1 \leq \beta \leq t} \left( \beta - 1 \right) \left( \left\lfloor \frac{k}{r \beta - \beta + 1} \right\rfloor - 1 \right) - 1.
\]
In particular, when the code is systematic, the bound can be tighten a bit, as follows:
\[
 n \geq k + d + \max_{2 \leq \beta \leq t} \left( \beta - 1 \right) \left( \left\lfloor \frac{k}{r \beta - \beta + 2} \right\rfloor - 1 \right) - 1.
\]

III. **Batch Codes with \( r = 2 \) or \( t = 2 \)**

A. **Optimal batch codes with \( r \geq 2 \) and \( t = 2 \)**

In this and subsequent sections, we construct \((n,k,r,t)\) batch codes for specific values of \( t \) and \( r \). To this end, consider an \((n,k,r,t)\) systematic batch code with \( r \geq 2 \) and \( t = 2 \). Then, from the bound in (2), we have
\[
 n \geq \left\lfloor \frac{k}{r} \right\rfloor + k + d - 2 \geq \left\lfloor \frac{k}{r} \right\rfloor + k,
\]
where the right-most inequality is due to inequality \( d \geq t \). The construction of codes attaining this bound, for \( t = 2 \) and \( r = 2 \), is presented in [14] Example 2]. In the sequel, we generalize that construction to other values of \( t \) and \( r \).

First, we show that the bound (3) is optimal for \( t = 2 \) and any \( r \geq 2 \). We achieve that by constructing corresponding \((n,k,r,t)\) batch codes.

Take \( G \) to be a \( k \times n \) binary systematic generator matrix of a code \( C \) defined as follows:
\[
 G = \begin{pmatrix}
 I_r & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\
 0 & I_r & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\
 0 & 0 & \cdots & I_r & 0 & 0 & 0 & \cdots & 1 & 0 \\
 0 & 0 & \cdots & 0 & I_r & 0 & 0 & \cdots & 0 & 1 \\
 \end{pmatrix},
\]
where \( s = k \mod r \), and recall that 1 denotes all-one column vector.

It is easy to see that \( C \) supports any query of size \( t = 2 \). If \( r \mid k \), then \( n = \frac{k}{r} (r + 1) \), which satisfies the lower bound (3) with equality. When \( r \nmid k \), we have \( s = k - \left\lfloor \frac{k}{r} \right\rfloor r \), and
\[
 n = \left\lfloor \frac{k}{r} \right\rfloor r + s + 1 = \left\lfloor \frac{k}{r} \right\rfloor + k + \frac{k}{r},
\]
which also satisfies (3) with equality.

Since \( B(k,r,t) \geq \mathcal{P}(k,r,t) \), we can summarize the result as in the following proposition.

**Proposition 3.** For any \( k, t = 2 \) and \( r \geq 2 \),
\[
 B(k,r,t) = \mathcal{P}(k,r,t) = \left\lfloor \frac{k}{r} \right\rfloor + k.
\]

Corollary 4. For any \( k \geq 1, t = 2 \) and \( r \geq 2 \), the optimal length of a non-systematic batch and PIR code, \( B_n(k,r,t) \) and \( \mathcal{P}_n(k,r,t) \), respectively, satisfies
\[
 \left\lfloor \frac{k}{r} \right\rfloor + k \geq B_n(k,r,t) \geq \mathcal{P}_n(k,r,t) \geq \left\lfloor \frac{k}{2r-1} \right\rfloor + k.
\]

B. **Batch codes with \( t \geq 2 \) and \( r = 2 \)**

In this section, we propose a construction of a (systematic) binary batch codes with \( t \geq 2 \) and \( r = 2 \) using a generator matrix with columns of weight at most \( 2 \). In particular, for \( t = 2 \) and \( r = 2 \), the construction is identical to the one in the previous section.

Let \( G \) be a binary generator matrix defined as \( G = [I_k | A] \).

In the sequel we describe how to construct the sub-matrix \( A \).

When \( k \) is even or \( t - 1 \) is even, the sub-matrix \( A \) has all its columns of weight 2 and all its rows of weight \( t - 1 \),
such that there is no 1-square pattern. In other words, for any $i_1, i_2, j_1, j_2, i_1 \neq i_2, j_1 \neq j_2$ at least one of the entries $A_{i_1,j_1}, A_{i_1,j_2}, A_{i_2,j_1}$ and $A_{i_2,j_2}$ is zero.

The total number of columns in $G$ is $n = k + (t - 1) \cdot \frac{k}{2}$. In particular, for $r = 2$ and $t = 2$, this bound coincides with (\ref{eq:2}).

When $k$ is odd and $t - 1$ is odd, then $A$ has all (except the last) columns of weight 2, and the last column of weight 1. All its rows are of weight $t - 1$, and there is no 1-square pattern in $A$. Therefore the total number of columns in $G$ is

$$n = k + \left( t - 1 \right) \cdot \frac{k}{2}.$$

**Proposition 5.** The code $C$ defined by the above generator matrix $G$ for $t = 3$ supports any query of the form $(x_i, x_j, x_k)$ with recovery sets of size at most 2, $i, j, \ell \in [k]$.

**Proposition 6.** The code $C$ defined by the above generator matrix $G$ for $r = 4$ supports any query of the form $(x_i, x_j, x_t, x_h)$ with recovery sets of size at most 2, $i, j, \ell, h \in [k]$.

**Proposition 7.** The code $C$ defined by the above generator matrix $G$ for general $t \geq 5$ supports any query of size $t$ of the form $(x_i, x_i, \cdots, x_i)$ with recovery sets of size at most 2.

**Proposition 8.** For $r = 2$ and $t \geq 3$,

$$k + (t - 1) \left\lfloor \frac{k}{t} \right\rfloor \leq \mathcal{P}(k, r, t) \leq k + \left( t - 1 \right) \cdot \frac{k}{2}.$$

The left-most inequality in Proposition 8 is obtained from (2) by substituting $d \geq 1$ and $r = 2$, and by picking $\beta = t$.

**Proposition 9.** For $r = 2$ and $t \in \{3, 4\}$,

$$k + (t - 1) \left\lfloor \frac{k}{t} \right\rfloor \leq \mathcal{B}(k, r, t) \leq k + \left( t - 1 \right) \cdot \frac{k}{2}.$$

The following examples illustrate the above constructions.

**Example 1.** Consider a binary $(n, k, r, t)$ batch (PIR) code $C$ with $k = 5$, $t = 3$ and $r = 2$. From Proposition 9 we have $9 \leq \mathcal{B}(5, 2, 3) \leq 10$. We construct a batch (PIR) code $C$ of length 10 with the above parameters using the following $5 \times 10$ generator matrix:

$$G = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

**Example 2.** Take a binary $(n, k, r, t)$ batch (PIR) code $C$ with $k = 5$, $t = 4$ and $r = 2$. From Proposition 9 we have $11 \leq \mathcal{B}(5, 2, 4) \leq 13$. We construct a batch (PIR) code $C$ of length 13 with the above parameters using the following $5 \times 13$ generator matrix:

$$G = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

**IV. Batch codes with $t = 3$ and $r \geq 3$**

**A. Batch codes with $t = 3$ and $r : r | k$**

We generalize the construction in Section III to systematic batch codes with $t = 3$ and $r \geq 3$ by using a generator matrix with columns of weight at most $r$.

Assume that $k \geq 2$. From (3), by substituting $d \geq t$, we obtain a lower bound

$$\mathcal{B}(k, r, t) \geq k + \max \left\{ \left\lfloor \frac{k}{r} \right\rfloor + 1, 2 \left\lfloor \frac{k}{2r - 1} \right\rfloor \right\}.$$

Next, we derive upper bounds by constructing corresponding codes.

Let $G$ be a generator matrix of the form $G = [I_k | A | B]$, where $I_k$ is the systematic part. Here, $A$ is a $k \times \frac{k}{r}$ matrix with the $j$th column of the form $A[j] = (a_{1,j}, a_{2,j}, \cdots, a_{k,j})^T$, $1 \leq j \leq \frac{k}{r}$, where $a_{i,j} = 1$ if $(j - 1)r + 1 \leq i \leq jr$, and $a_{i,j} = 0$ otherwise. The matrix $B$ is defined as follows.

- **Case 1:** $\frac{k}{r} \leq r$. In that case we choose $B$ to be a $k \times r$ matrix defined as $[I_r | I_r | \cdots | I_r]^T$. We have that $n = k + \frac{k}{r} + r = (r + 1)\frac{k}{r} + r$.

- **Case 2:** $\frac{k}{r} > r$. We define $B^T = [B_1^T | B_2^T | \cdots | B_{k/r}^T]^T$, where all $B_i$ are $(k/r) \times \zeta$ block matrices. We construct $B$ such that each column has weight $r$, and for each its column at most one non-zero element appears in each block $B_i$. Moreover, we require that every row in $B$ contains at most one non-zero entry. Since $k/r > r$, such $B$ exists. We take $\varsigma = k/r$. To this end, $n = (r + 1)\frac{k}{r} + \varsigma$.

**Proposition 10.** Let $t = 3$ and $r \geq 3$, $r | k$. Denote $\varsigma = \max\{ \frac{k}{r}, r \}$. Then, $\mathcal{B}(k, r, t) \leq (r + 1)\frac{k}{r} + \varsigma$.

**Proposition 11.** The code $C$ defined by the generator matrix $G$ supports any query of the form $(x_i, x_j, x_k)$ with recovery sets of size at most $r$, $i, j, \ell \in [k]$.

**Example 3.** Consider a special case $r = 2$ and $k / r \geq r$. Then, Proposition 11 yields $n = 4k/r$, which is equivalent to the upper bound in Proposition 8.

**Example 4.** Let $k = 8$, $r = 4$, $t = 3$ so that $k/r = 2$. Then the following generator matrix $G$ generates a batch code of length $n = 14$.

$$G = \begin{pmatrix}
I_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & I_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & I_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & I_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & I_4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

**Example 5.** Let $k = 12$, $r = 3$, $t = 3$ so that $k/r = 4$. Then the following generator matrix $G$ generates a batch code of length $n = 12 + 4 + 4 = 20$.

$$G = \begin{pmatrix}
I_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & I_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & I_3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & I_3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & I_3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}.$$
In the sequel, we generalize the results in the previous subsection towards the case where \( r \upharpoonright k \). Let \( G \) be the generator matrix of the block form \( G = [I_k] [A | B | C] \).

A is a \( k \times \lceil \frac{k}{r} \rceil \) matrix with the \( j \)th column of the form \( A^j = (a_{1,j}, a_{2,j}, \ldots, a_{k,j}) \), \( 1 \leq j \leq \lceil \frac{k}{r} \rceil \), where \( a_{i,j} = 1 \) if \((j-1)r + 1 \leq i < jr\), and \( a_{i,j} = 0 \) otherwise.

Denote \( s = k \mod r \). The matrix B is a \( k \times (s+1) \) block matrix, defined as \( B^T = [B_1^T | B_2^T]^T \), where \( B_1 \) is \((k-s) \times (s+1)\) and \( B_2 \) is an \( s \times (s+1) \) matrix \([I_s] \).

We take \( \gamma \triangleq \min \{ r-s, \lceil \frac{k}{r} \rceil \} \). The first column of \( B_1 \), \( B_1^1 \), has \( \tau \) entries 1, each entry appears in a different block of rows \([j-1)r + 1, jr]\) for \( j = 1, 2, \ldots, \lceil \frac{k}{r} \rceil \). We take also \( \eta \triangleq \min \{ r-1, \lceil \frac{k}{r} \rceil \} \). The columns \( B_1^1, B_1^2, \ldots, B_1^{\eta+1} \), all have \( \eta \) entries 1, each entry appears in a different block of rows. Additionally, every row in \( B_1 \) contains at most one non-zero entry.

We observe that every column in \( B \) has at most \( r \) ones. Denote \( \gamma \triangleq \min \{ r-1, \lceil \frac{k}{r} \rceil \} \). The matrix \( C \) is constructed according to the following rules:

- Each column in \( C \) has \( \gamma \) ones (except possibly for the last column);
- The last \( s \) rows in \( C \) are zeros;
- Each row in \([A | B | C]\) has two ones;
- There are no 1-squares in \([B | C]\).

Next, we estimate the total number of columns in \( G \). The number of ones in the first \( k-s \) positions of \( B_1^1 \) is \( \tau \). The number of ones in the first \( k-s \) positions of each of \( B_2^1, B_2^2, \ldots, B_2^{\eta+1} \) is \( \eta \). Since there are two ones in each of the first \( k-s \) positions of \([A | B | C]\), the total number of columns in \( G \) is

\[
n = (r+1) \left\lceil \frac{k}{r} \right\rceil + 2s + 1 + \left\lceil \frac{k-s}{\gamma} - \eta \cdot s \right\rceil.
\]

**Proposition 12.** For \( t = 3 \) and \( r \geq 3 \),

\[
k + \max \left\{ \left\lceil \frac{k}{r} \right\rceil + 1, 2 \left\lceil \frac{k}{2r-1} \right\rceil \right\} \leq B(k, r, t) \leq \begin{cases} (r+1) \left\lceil \frac{k}{r} \right\rceil + \frac{\gamma}{r} + \zeta & \text{if } r \mid k \\ (r+1) \left\lceil \frac{k}{r} \right\rceil + 2s + 1 + \left\lceil \frac{k-s}{\gamma} - \eta \cdot s \right\rceil & \text{if } r \upharpoonright k \end{cases}
\]

where \( \zeta = \max \{ \frac{s}{r}, \frac{\gamma}{r} \} \).

**Proposition 13.** The code \( C \) defined in this section by the generator matrix \( G \) supports any query of the form \((x_i, x_j, x_k)\) with recovery sets of size at most \( r \cdot i, j, k \in [k] \).

**Example 6.** Let \( k = 11 \), \( r = 3 \), \( t = 3 \) so that \( k/r = 3 \) and \( s = 2 \). Then the following generator matrix \( G \) generates a batch code of length \( n = 19 \).

We summarize the bounds on \( B(k, r, t) \) and \( P(k, r, t) \) for \( t = 2, 3, 4 \) and for all values of \( r \) in Tables I and II.

**Table I**

| \( t \) | \( r \geq 3 \) | \( r = 2 \) | \( r = 3 \) |
|---|---|---|---|
| 2 | \( k + \frac{k}{2} \) | \( k + \frac{k}{2} \) | \( k + 1 \) |
| 3 | \( k + \frac{k}{2} \) | \( k + \frac{k}{2} \) | \( k + \frac{k}{2} \) |

**V. BOUNDS ON THE DIMENSION OF A BATCH CODE**

Let \( k^\text{opt}_q(n, d) \) denote the largest possible dimension of a linear code of length \( n \) and minimum distance \( d \), for a given alphabet \( Q \) of size \( q \). More formally, by following on the notations in [9], denote:

\[
k^\text{opt}_q(n, d) \triangleq \max \frac{\log |C|}{\log q},
\]

where the maximum is taken over all possible linear codes \( C \) of length \( n \) with minimum distance \( d \).

Let \( I \subseteq [n] \) be a set of coordinates. Define

\[
H(I) = \frac{\log \{|x_I : x \in C|\}}{\log q}.
\]

The following result appears as Lemma 2 in [9].

**Lemma 14.** Consider an \([n, k, d] \) code over \( Q \) where there exists a set \( I \subseteq [n] \) such that \( H(I) \leq m \). Then, there exists an \([n-|I|, (k-m)\uparrow, d] \) code over \( Q \), where the \( \uparrow \) symbol denotes the dimension is at least \( k-m \).

Cadambe and Mazumdar show in [9] that, for any \( r \) locally recoverable \((n, k, d) \) code over the alphabet \( Q \), it holds:

\[
k \leq \min \left\{ tr + k^\text{opt}_q(n-t(r+1), d) \right\}.
\]

By using similar techniques, in the sequel we show a bound on \( k \) for a linear \((n, k, r, t) \) batch code. We restrict our discussion to linear codes only.

**Proposition 15.** Let \( C \) be a linear \((n, k, r, t) \) batch code over an alphabet \( Q \) of size \( q \) with minimum distance \( d \), and \( n-1 \geq d \). Then,

\[
k \leq tr - (t-1) + k^\text{opt}_q(n-t, d).
\]
Proof: Since \( C \) is an \((n,k,r,t)\) batch code, for a query \((x_1,\ldots,x_t)\) of size \( t \), for some \( i \in [k] \), there exist \( t \) disjoint recovery sets \( R_1,\ldots,R_t \) with \(|R_j| \leq r\) for all \( j \in [t] \).

Denote \( I \triangleq R_1 \cup R_2 \cup \cdots \cup R_t \). Clearly, \(|I| \leq t\). Take an arbitrary word \( y \in C \). By Lemma 11 there exist indices \( a_2 \in R_2, a_3 \in R_3,\ldots, a_t \in R_t \) such that if we fix the values of all coordinates of \( y \) indexed by the sets \( R_1 \setminus \{a_2\}, R_3 \setminus \{a_3\},\ldots, R_t \setminus \{a_t\} \), then the values of the coordinates of \( y \) indexed by \( S \triangleq \{a_2,a_3,\ldots,a_t\} \) are uniquely determined. It follows that there is one-to-one mapping between \( y_I \setminus S \) and \( y_T \). Therefore, \( H(I) \leq tr - t - 1 \).

The main statement now follows by applying Lemma 14.

We remark that the condition \( n - tr \geq d \) in Proposition 15 is necessary for existence of a code of length \( n - tr \) and minimum distance \( d \). However, it should be noted that any \((n,k,r,t)\) batch code is a \((n,k,r,\beta)\) batch code for \( 1 \leq \beta \leq t \). Thus, if \( t \) is too large to satisfy this condition, we can take a smaller value \( \beta \) instead. The following result follows.

**Corollary 16.** Let \( C \) be a linear \((n,k,r,t)\) batch code over an alphabet \( Q \) of size \( q \) with minimum distance \( d \). Then,
\[
  k \leq \min_{1 \leq \beta \leq t} \left\{ \beta r - (\beta - 1) + k_{q}^{\text{opt}}(n - \beta r, d) \right\}. \tag{8}
\]

Comparison with the bounds in the literature

**Example 7.** The asymptotic versions of the classical Singleton and the sphere-packing bounds for a code over alphabet \( Q \) of size \( q \) are \( R \leq 1 - \delta + o(1) \) and \( R \leq 1 - h_q(\delta/2) + o(1) \), respectively, where \( R = k/n \), \( \delta = d/n \), and \( h_q(\cdot) \) denotes the \( q \)-ary entropy function [17] Chapter 4.

The asymptotic versions of the bounds (7) and (8) can be rewritten as (when ignoring the \( o(1) \) term, for any specific value of \( \beta \)):
\[
  R \leq 1 - \delta - \frac{1}{n} \left( k \frac{1}{br - \beta + 1} - 1 \right), \tag{9}
\]
and
\[
  R \leq \frac{\beta(r - 1)}{n} + R_q^{\text{opt}}(n - \beta r, d), \tag{10}
\]
respectively, where \( R_q^{\text{opt}}(n,d) \) denotes the maximum rate of any code of length \( n \) and minimum distance \( d \) over \( Q \).

For \( n - d \gg tr \geq \beta r \), the bound (9) becomes \( R \leq (1 - \delta) \cdot (1 - \frac{1}{br - \beta + 1}) \). For comparison, when we use the sphere-packing bound for \( R_q^{\text{opt}}(n,d) \), then (10) can be rewritten as
\[
  R \leq \frac{\beta(r - 1)}{n} + 1 - h_q(\delta/2) \approx 1 - h_q(\delta/2).
\]

Therefore, for large blocklength, \( n - d \gg tr \), and for a range of \( \delta \) and \( r \), the bound (9) is tighter than (10) (for any \( \beta \geq 1 \)).

**ACKNOWLEDGMENT**

The work of E. Thomas and V. Skachek is supported by the Estonian Research Council under the grants PUT405 and IUT2-1.

**REFERENCES**

[1] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, “Batch codes and their applications”, Proc. of the 36th ACM Symposium on Theory of Computing (STOC), Chicago, June 2004.

[2] Z. Wang, H. M. Kiah and Y. Cassuto, “Optimal binary switch codes with small query size”, Proc. of the IEEE International Symposium on Information Theory (ISIT), Hong Kong, pp. 636–640, June 2015.

[3] Z. Wang, O. Shakedy, Y. Cassuto and J. Bruck, “Codes for network switches”, Proc. of the IEEE International Symposium on Information Theory (ISIT), Istanbul, 2013.

[4] Y.M. Chee, F. Gao, S. T. H. Teo and H. Zhang, “Combinatorial systematic switch codes”, Proc. of the IEEE International Symposium on Information Theory (ISIT), Hong Kong, China, pp. 241–245, 2015.

[5] S. Bhattacharya, S. Ruj and B. Roy, “Combinatorial batch codes: a lower bound and optimal constructions”, Advances in Mathematics of Communications, vol. 6, no. 2, pp. 165–174, 2012.

[6] R.A. Brualdi, K. Kiernan, S.A. Meyer and M.W. Schroeder, "Combinatorial batch codes and transversal matroids", Advances in Mathematics of Communications, vol. 4, no. 3, pp. 419–431, 2010.

[7] C. Butuš and Z. Tuza, “Batch codes and their applications”, Electronic Notes in Discrete Mathematics, vol. 38, pp. 201–206, 2011.

[8] A. Fazeli, A. Vardy, and E. Yaakobi, “PIR with low storage overhead: coding instead of replication”, arXiv:1505.06241, May 2015.

[9] V. Cadambe and A. Mazumdar, “Bounds on the Size of Locally Recoverable Codes,” IEEE Transactions on Information Theory, vol. 61(11), pp. 5787–5794, Nov. 2015.

[10] M. Forbes and S. Yekhanin, “On the locality of codeword symbols in non-linear codes”, Discrete Math, vol. 324, pp. 78–84, 2014.

[11] P. Gopalan, C. Huang, H. Simitci and S. Yekhanin, “On the locality of codeword symbols”, IEEE Trans. on Inform. Theory, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[12] A.S. Rawat, A. Mazumdar and S. Vishwanath, “Cooperative local repair in distributed storage”, EURASIP Journal on Advances in Signal Processing, December 2015.

[13] A.S. Rawat, D.S. Papailiopoulos, A.G. Dimakis, and S. Vishwanath, “Locality and availability in distributed storage”, IEEE Trans. on Information Theory, vol. 62, no. 8, pp. 4481–4493, Aug. 2016.

[14] H. Zhang and V. Skachek, “Bounds for batch codes with restricted query size”, Proc. of the IEEE International Symposium on Information Theory (ISIT), Barcelona, pp. 1192–1196, 2016.

[15] A. Vardy and E. Yaakobi, “Constructions of batch codes with near-optimal redundancy”, IEEE International Symposium on Information Theory, Barcelona, pp. 1197–1201, July 2016.

[16] H. Lipmaa and V. Skachek, “Linear batch codes”, Proceedings of the 4th International Castle Meeting on Coding Theory and Applications, Palma, Portugal, September 2014.

[17] R.M. Roth, "Introduction to coding theory", Cambridge University Press, 2006.

[18] A.G. Dimakis, A. Gal, A.S. Rawat and Z. Song, “Batch codes through dense graphs without short cycles”, arXiv:1410.2920, Oct. 2014.