Locally Repairable Codes with Functional Repair and Multiple Erasure Tolerance

Wentu Song and Chau Yuen

Abstract

We consider the problem of designing \([n, k]\) linear codes for distributed storage systems (DSS) that satisfy the \((r, t)\)-Local Repair Property, where any \(t' < t\) simultaneously failed nodes can be locally repaired, each with locality \(r\). The parameters \(n, k, r, t\) are positive integers such that \(r < k < n\) and \(t \leq n - k\). We consider the functional repair model and the sequential approach for repairing multiple failed nodes. By functional repair, we mean that the packet stored in each newcomer is not necessarily an exact copy of the lost data but a symbol that keep the \((r, t)\)-local repair property. By the sequential approach, we mean that the \(t'\) newcomers are ordered in a proper sequence such that each newcomer can be repaired from the live nodes and the newcomers that are ordered before it. Such codes, which we refer to as \((n, k, r, t)\)-functional locally repairable codes (FLRC), are the most general class of LRCs and contain several subclasses of LRCs reported in the literature.

In this paper, we aim to optimize the storage overhead (equivalently, the code rate) of FLRCs. We derive a lower bound on the code length \(n\) given \(t \in \{2, 3\}\) and any possible \(k, r, t\). For \(t = 3\), our bound generalizes the rate bound proved in [14]. For \(t \leq 3\), our bound improves the rate bound proved in [10]. We also give some constructions of exact LRCs for \(t \in \{2, 3\}\) whose length \(n\) achieves the bound of \((n, k, r, t)\)-FLRC, which proves the tightness of our bounds and also implies that there is no gap between the optimal code length of functional LRCs and exact LRCs for certain sets of parameters. Moreover, our constructions are over the binary field, hence are of interest in practice.

Index Terms

Distributed storage, locally repairable codes, exact repair, functional repair.

I. INTRODUCTION

A distributed storage system (DSS) stores data through a large, distributed network of storage nodes. To ensure reliability against node failure, data is stored in redundancy form so that it can be reconstructed from the system even if some of the storage nodes fail. Moreover, to maintain the data reliability in the presence of node failures, each failed node is replaced by a newcomer that stores a data packet computed from the data packets stored in some available storage nodes. This process is called node repair.

There are two models of node repair, called exact repair and functional repair respectively. By exact repair, each newcomer stores an exact copy of the lost packet. By functional repair, each newcomer stores a packet that is not necessarily an exact copy of the lost data, but a packet that makes the system keep the same level of data reliability and the possibility of node repair in the future. While exact repair is a special case of functional repair and is more preferable in practice for its simplicity, functional repair model has its theoretical interest because potentially it allows us to construct codes with improved code rate or minimum distance.

Modern distributed storage systems employ various coding techniques, such as erasure codes, regenerating codes and locally repairable codes, to improve system efficiency. Classical MDS codes (such as Reed-Solomon codes) are optimal in storage efficiency but are inefficient in node repair—the total amount of data download needed to repair a single failed node equals to the size of the whole file [1]. As improvements of MDS codes, regenerating codes aim to optimize the repair bandwidth [1] and locally repairable codes (LRC) aim to minimize the repair locality, i.e. the number of disk accesses required during a single node repair [2]. In this work, we focus on the metric of repair locality.

Repair locality was initially studied as a metric for repair cost independently by Gopalan et al. [3], Oggier et al. [4], and Papailiopoulos et al. [5]. The \(i\)th coordinate of an \([n, k]_q\) linear code \(C\) (also called the \(i\)th code symbol of \(C\)) is said to have locality \(r\), if its value is computable from the values of a set of at most \(r\) other coordinates of \(C\) (called a repair set of \(i\)). In the literature, an \([n, k]_q\) linear code is called a locally repairable code (LRC) if all of its code symbols have locality \(r\) for some \(r < k\). In a DSS coded by an LRC \(C\), each storage node stores a code symbol of \(C\) and any single failed node can be “locally and exactly repaired” in the sense that the newcomer can recover the lost data by contacting at most \(r\) live nodes, where \(r\) is the symbol locality of \(C\).

A. Local Repair for Multiple Node Failures

In real DSS, it is not uncommon that two or more storage nodes fail simultaneously at one time, which motivates the researchers to study LRCs that can locally repair more than one failed nodes. Studies of LRCs for multiple node failures can be found in [6]—[15] and references therein.

W. Song and C. Yuen are with Singapore University of Technology and Design, Singapore (e-mails: {wentu_song, yuenchau}@sutd.edu.sg).
To repair \( t \) \((t \geq 2)\) simultaneously failed nodes, \( t \) newcomers are added into the system, each downloads data from a set of at most \( r \) available nodes to create its storage content. The authors in [14] distinguished two approaches of how the \( t \) newcomers contact the available nodes, called parallel approach and sequential approach respectively. By the parallel approach, each newcomer download data from a set of live nodes. In contrast, by the sequential approach, the \( t \) newcomers can be properly ordered in a sequence and each newcomer can download data from both the live nodes and the newcomers ordered before it. Clearly, the parallel approach is a special case of the sequential approach. Potentially, the sequential approach allows us to design codes with improved code rate or minimum distance than the parallel approach.

Given the parameters \( n, k, r \) and \( t \), where \( n \) is the code length and \( k \) is the dimension, four subclasses of linear LRCs that can exactly and locally repair up to \( t \) failed nodes by the parallel approach are reported in the literature:  

- a) Codes with all-symbol locality \((r, t+1)\), in which each code symbol is contained in a local code of length at most \( r + t \) and minimum distance at least \( t + 1 \) [7];
- b) Codes with all-symbol locality \( r \) and availability \( t \), in which each code symbol has \( t \) pairwise disjoint repair sets with locality \( r \) [8], [9];
- c) Codes with \((r, t)\)-locality, in which each subset of \( t \) code symbols can be cooperatively repaired from at most \( r \) other code symbols [13];
- d) Codes with overall local repair tolerance \( t \), in which for any \( E \subseteq [n] \) of size \( t \) and any \( i \in E \), the \( i \)th code symbol has a repair set contained in \([n]\backslash E\) and with locality \( r \) [6].

For convenience, we refer to the above four subclasses of LRCs as \((r, \delta)_a\) codes, \((r, \delta)_c\) codes, \((r, t)\)-CLRC and \((r, t)_o\) codes respectively, where \( \delta = t + 1 \). Clearly, the first three subclasses are all contained in the subclass of \((r, t)_o\) codes. Moreover, \((r, t)_o\) codes can exactly and locally repair up to \( t \) failed nodes by the parallel approach. For \((r, \delta)_a\) codes and \((r, t)\)-CLRC, the code rate satisfies (e.g., see [15] and [14]):

\[
\frac{k}{n} \leq \frac{r}{r + t}
\]

and the minimum distance satisfies (see [7] and [13]):

\[
d \leq n - k + 1 - t \left( \frac{k}{r} - 1 \right).
\]

For \((r, \delta)_c\) codes (i.e., codes with all-symbol locality \( r \) and availability \( t \)), it was proved in [10] that the code rate satisfies:

\[
\frac{k}{n} \leq \frac{1}{\prod_{j=1}^{t} (1 + \frac{1}{jr})}
\]

and the minimum distance satisfies:

\[
d \leq n - \sum_{j=0}^{t} \left\lfloor \frac{k - 1}{r^j} \right\rfloor.
\]

For \( t = 2 \), the bound [4] is shown to be achievable for some special case of parameters [10]. However, for the general case, it is not known whether the bounds [3] and [4] are achievable. Recent work by Wang et al. [12] shows that for any positive integers \( r \) and \( t \), there exist \((r, \delta)_c\) codes over the binary field with code rate \( \frac{r}{r + t} \). Unfortunately, the rate does not achieve the bound [3] for \( t \geq 2 \). For the more general case, the \((r, t)_o\) codes, no result is known about the code rate bound or the minimum distance bound for \( t \geq 2 \).

For LRCs that can exactly and locally repair \( t = 2 \) failed nodes by the sequential approach, it was proved in [14] that the code rate satisfies:

\[
\frac{k}{n} \leq \frac{r}{r + 2}.
\]

An upper bound for the minimum distance of such codes was also given in [14]. However, for \( t \geq 3 \), no result is known about the code rate bound or the minimum distance bound.

Fig 1. Relation of the six subclasses of \([n, k]\) linear LRCs, where \( \delta = t + 1 \).
B. LRC with Functional Repair

Vector codes that can locally repair single failed node with functional repair model was considered by Hollmann et al. [10]–[18]. Suppose $\alpha$ is the capacity of each storage node and $\beta$ is the transport capacity, i.e., the amount of data that can be transported from a node contacted during the repair process. It was proved in [18] that if $\alpha = \beta$, the code rate is upper bounded by $r$, where $r$ is the repair locality. However, the study of LRC for multiple node failures under functional repair model is not seen in the literature.

C. Our Contribution

Given positive integers $n, k, r$ and $t$ such that $r < k < n$ and $t \leq n - k$. We consider the problem of designing $[n, k]$ linear codes for distributed storage systems (DSS) that satisfy the $(r, t)$-Local Repair Property, where any $t' (\leq t)$ simultaneously failed nodes can be locally repaired, each with locality $r$. We consider the functional repair model and the sequential approach for repairing multiple failed nodes. By functional repair, we mean that the packet stored in each newcomer is not necessarily an exact copy of the lost data but a symbol that keep the $(r, t)$-local repair property. We call such codes $(n, k, r, t)$-functional locally repairable code (FLRC). A subclass of FLRC, called $(n, k, r, t)$-exact locally repairable code (ELRC), in which the $(r, t)$-local repair property is satisfied by exact repair, is also considered.

Clearly, codes studied in [14] are $(n, k, r, 2)$-ELRC and $(r, t)$, codes (i.e., codes with overall local repair tolerance $t$) studied in [6] are $(n, k, r, t)$-ELRC. The relation of the six subclasses of LRCs mentioned above are depicted in Fig. 1.

It is easy to see that the minimum distance of an $(n, k, r, t)$-FLRC is at least $t + 1$. In this paper, our goal is to optimize the storage overhead (equivalently, the code rate of such codes). When $t = 1$, by the result of [18], the code rate of an $(n, k, r, t)$-FLRC is upper bounded by $\frac{r}{n-1}$. So we focus on the case of $t \geq 2$. Our method is to associate each $(n, k, r, t)$-FLRC with a set of directed acyclic graphs, called repair graph. Then by studying the structural properties of the so called minimal repair graph (similar to the discussion in [19], [20]), we derive a lower bound of the code length $n$. Our main results are listed as bellow:

1) We prove that for $(n, k, r, t = 2)$-FLRC, the code length satisfies

$$n \geq k + \left\lceil \frac{2k}{r} \right\rceil.$$ 

Equivalently, the code rate satisfies

$$\frac{k}{n} \leq \frac{r}{r+2}.$$ 

Note that bound (3) is an upper bound of the code rate of $(n, k, r, t = 2)$-ELRC. Thus, our bound generalizes the bound (3) to the setting of functional repair model.

2) We prove that for $(n, k, r, t = 3)$-FLRC, the code length satisfies

$$n \geq k + \left\lceil \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \right\rceil.$$ 

Note that codes with all-symbol $(r, \delta = 4)$-locality is an $(n, k, r, t = 3)$-ELRC. For $t = 3$, (3) implies that $n \geq \frac{r+1}{2r+1} \frac{2r+1}{3} \frac{3r+1}{k}$. Moreover, we can check that $k + \left\lceil \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \right\rceil \geq k + \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \geq \frac{r+1}{2r+1} \frac{2r+1}{3} \frac{3r+1}{k}$. So our result improves the bound (3) for $t = 3$.

3) We give some constructions of $(n, k, r, t)$-ELRC for $t \in \{2, 3\}$ whose code length $n$ achieves the corresponding bound of FLRC, which proves the tightness of our bounds and also implies that there is no gap between the optimal code length of functional LRCs and exact LRCs for some sets of parameters. Moreover, our constructions are over the binary field, hence are of practical interest.

D. Organization

The rest of this paper is organized as follows. In Section II, we give the basic notations and concepts including functional locally repairable code (FLRC), exact locally repairable code (ELRC) and repair graph of FLRC. In section III, we prove some structural properties of the minimal repair graph of FLRC. Lower bounds on code length of $(n, k, r, t)$-FLRC for $t \in \{2, 3\}$ are derived in Section IV. Constructions of ELRC with optimal code length is presented in Section V. The paper is concluded in Section VI.

II. PRELIMINARY

For any set $A$, we use $|A|$ to denote the size (i.e., the number of elements) of $A$. A set $B$ is called an $r$-subset of $A$ if $B \subseteq A$ and $|B| = r$. For any positive integer $n$, we denote $\{n\} := \{1, 2, \cdots, n\}$. An $[n, k]$ linear code over a field $\mathbb{F}$ is a $k$-dimensional subspace of $\mathbb{F}^n$. 
Let $C$ be an $[n,k]$ linear code over the field $\mathbb{F}$. If there is no confusion in the context, we will omit the base field $\mathbb{F}$ and only say that $C$ is an $[n,k]$ linear code. A $k$-subset $S$ of $[n]$ is called an information set of $C$ if for all codeword $x = (x_1, x_2, \cdots, x_n) \in C$ and all $i \in [n]$, $x_i = \sum_{j \in S} a_{ij} x_j$, where all $a_{ij} \in \mathbb{F}$ and are independent of $x$. The code symbols in $\{x_j, j \in S\}$ are called information symbol of $C$. In contrast, code symbols in $\{x_i, i \in [n]\setminus S\}$ are called parity symbol of $C$. An $[n,k]$ linear code has at least one information set.

For any $E \subseteq [n]$, let $\overline{E} = [n]\setminus E$ and $C|_E$ be the punctured code of $C$ associated with the coordinate set $E$. That is, $C|_E$ is obtained from $C$ by deleting all code symbols in the set $\{x_i, i \in \overline{E}\}$ for each codeword $x = (x_1, x_2, \cdots, x_n) \in C$.

A. Locally repairable code (LRC)

In this subsection, we always assume that $C$ is an $[n,k]$ linear code over $\mathbb{F}$. We first present the concept of repair set for each coordinate $i \in [n]$.

**Definition 1:** Let $i \in [n]$ and $R \subseteq [n]\setminus \{i\}$. The subset $R$ is called an $(r,C)$-repair set of $i$ if $|R| \leq r$ and $x_i = \sum_{j \in R} a_{ij} x_j$ for all $x = (x_1, x_2, \cdots, x_n) \in C$, where all $a_{ij} \in \mathbb{F}$ and are independent of $x$.

In the following, we will omit the prefix $(r,C)$ and say that $R$ is a repair set of $i$ if there is no confusion in the context.

**Definition 2:** Let $E$ be a $t$-subset of $[n]$. $C$ is said to be $(E,r)$-repairable if there exists an index of $E$, say $E = \{i_1, \cdots, i_t\}$, and a collection of subsets

$$\{R_\ell \subseteq \overline{E} \cup \{i_1, \cdots, i_{\ell-1}\}; |R_\ell| \leq r, \ell \in [t]\}$$

such that for each $\ell \in [t]$, $R_\ell$ is an $(r,C)$-repair set of $i_\ell$.

In this paper, we assume $r < k < n$, which means small repair locality and at least one redundant code symbol. Moreover, if $C$ is $(E,r)$-repairable for some $t$-subset $E$ of $[n]$, then we can easily see that $t \leq n - k$.

**Definition 3:** Let $C'$ be an $[n,k]$ linear code over $\mathbb{F}$ (not necessarily different from $C$) and $E \subseteq [n]$. $C'$ is said to be an $(E,r)$-repair code of $C$ if the following two conditions hold:

(i) $C|_E = C'|_E$;

(ii) $C'$ is $(E,r)$-repairable.

Consider a DSS with $n$ storage nodes where a data file is stored as a codeword of $C$, each node storing one code symbol. Suppose the nodes indexed by $E$ fail. Then the symbols stored in the live nodes form a codeword $x|_E$ of the punctured code $C|_E$. If $C'$ is an $(E,r)$-repair code of $C$, then $x|_E$ is also a codeword of $C'|_E$. Moreover, since $C'$ is $(E,r)$-repairable, then we can construct a codeword of $C'$ from $x|_E$ using the sequential approach, which form a process of functional repair.

**Definition 4:** An $(n,k,r,t)$-functional locally repairable code (FLRC) is a collection of $[n,k]$ linear codes $\{C_\lambda; \lambda \in \Lambda\}$, where $\Lambda$ is an index set, such that for each $\lambda \in \Lambda$ and each $E \subseteq [n]$ of size $|E| \leq t$, there is a $\lambda' \in \Lambda$ such that $C_{\lambda'}$ is an $(E,r)$-repair code of $C_{\lambda}$.

**Definition 5:** An $(n,k,r,t)$-exact locally repairable code (ELRC) is an $[n,k]$ linear code $C$ such that for each $E \subseteq [n]$ of size $|E| \leq t$, $C$ is $(E,r)$-repairable.

Clearly, for any DSS with $n$ storage nodes and a data file of $k$ information symbols being stored, if the $(r,t)$-local repair property is satisfied for functional repair model and the sequential approach, then the coding scheme can be described as an $(n,k,r,t)$-FLRC. Conversely, any $(n,k,r,t)$-FLRC can be used as a coding scheme for such DSS.

Let $\{C_\lambda; \lambda \in \Lambda\}$ be an $(n,k,r,t)$-FLRC. Suppose $i \in [n]$ and $\lambda_1 \neq \lambda_2 \in \Lambda$. It is possible that the $(r,C_{\lambda_1})$-repair set of $i$ is different from the $(r,C_{\lambda_2})$-repair set of $i$. In other words, the repair set of the coordinate $i$ is not fixed, but depends on the state of the system.

From Definition 2 and 4, we can easily see that an $[n,k]$ linear code $C$ is an $(n,k,r,t)$-ELRC if and only if for all $E \subseteq [n]$ of size $|E| \leq t$, $C$ is an $(E,r)$-repair code of itself. So an $(n,k,r,t)$-ELRC is naturally an $(n,k,r,t)$-FLRC. Moreover, we can characterize $(n,k,r,t)$-ELRC by a seemingly simpler condition as follows.

**Lemma 6:** An $[n,k]$ linear code $C$ is an $(n,k,r,t)$-ELRC if and only if for any $E \subseteq [n]$ of size $|E| \leq t$, there exists an $i \in E$ such that $i$ has an $(r,C)$-repair set contained in $[n]\setminus E$.

**Proof:** If $C$ is an $(n,k,r,t)$-ELRC, then by Definition 2 and 4 there exists an index of $E$, say $E = \{i_1, \cdots, i_t\}$, such that $i_1$ has an $(r,C)$-repair set $R_1 \subseteq \overline{E} = [n]\setminus E$.

Conversely, for any $E \subseteq [n]$ of size $|E| = t'$, by assumption, there exists an $i_1 \in E$ such that $i_1$ has an $(r,C)$-repair set $R_1 \subseteq \overline{E} = [n]\setminus E$. Now, let $E_1 = E\setminus \{i_1\}$. Then $|E_1| \leq t$ and by assumption, there exists an $i_2 \in E_1$ such that $i_2$ has an $(r,C)$-repair set $R_2 \subseteq [n]\setminus E_1$. Similarly, we can find an $i_3 \in E\setminus \{i_1, i_2\}$ such that $i_3$ has an $(r,C)$-repair set $R_3 \subseteq \overline{E} \cup \{i_1, i_2\}$. And so on. Then we can index $E$ as $E = \{i_1, i_2, \cdots, i_{t'}\}$ such that each $i_\ell$ has an $(r,C)$-repair set $R_\ell \subseteq \overline{E} \cup \{i_1, i_2, \cdots, i_{\ell-1}\}$. Thus, by Definition 2 and 4 $C$ is an $(n,k,r,t)$-ELRC.

B. Repair graph of LRC

To derive a bound of the code length, we introduce the concepts of repair graph and minimal repair graph of an $(n,k,r,t)$-FLRC and investigate the structural properties of the minimal repair graphs.
Let $G = (\mathcal{V}, \mathcal{E})$ be a directed, acyclic graph with node (vertex) set $\mathcal{V}$ and edge (arc) set $\mathcal{E}$. For any $e = (u, v) \in \mathcal{E}$, we call $u$ the tail of $e$ and $v$ the head of $e$. We also call $u$ an in-neighbor of $v$ and $v$ an out-neighbor of $u$. For each $v \in \mathcal{V}$, let $\text{In}(v)$ and $\text{Out}(v)$ denote the set of in-neighbors and out-neighbors of $v$ respectively. If $\text{In}(v) = \emptyset$, we call $v$ a source. Otherwise, we call $v$ an inner node. We use $S(G)$ to denote the set of all sources of $G$. Moreover, for any $V \subseteq \mathcal{V}$, let

$$\text{Out}(V) = \bigcup_{v \in V} \text{Out}(v) \setminus V.$$  

(6)

And for any $v \in \mathcal{V}$, let

$$\text{Out}^2(v) = \bigcup_{u \in \text{Out}(v)} \text{Out}(u) \setminus \text{Out}(v)$$

(7)
i.e., $\text{Out}^2(v)$ is the set of all $w \in \mathcal{V}$ such that $w$ is an out-neighbor of some $u \in \text{Out}(v)$ but not an out-neighbor of $v$.

As an example, consider the graph as depicted in Fig. 2. We have $\text{Out}(3) = \{9, 10\}$ and $\text{Out}(4) = \{10, 11\}$. So by (6), $\text{Out}(V) = \{9, 10, 11\}$, where $V = \{3, 4\}$. Moreover, by (7), we have $\text{Out}^2(3) = \{13, 15, 16\}$.

Fig. 2. An example repair graph $G_{\lambda_0}$, where $r = 2$ and $n = 16$.

For any linear code $C$ with repair locality, we can associate $C$ with a set of graphs called repair graph of $C$.

**Definition 7:** Let $C$ be an $[n, k]$ linear code and $G = (\mathcal{V}, \mathcal{E})$ be a directed, acyclic graph such that $\mathcal{V} = [n]$. $G$ is called a repair graph of $C$ if for all inner node $i \in \mathcal{V}$, $\text{In}(i)$ is an $(r, C)$-repair set of $i$.

A code $C$ may have many repair graphs. Moreover, in Definition 7, we do not require that $R = \text{In}(i)$ for any $(r, C)$-repair set $R$ of $i$. Thus, it is possible that there exists an $(r, C)$-repair set $R$ of $i$ such that $\text{In}(i) \neq R$. However, we can always construct a repair graph $G'$ of $C$ such that $\text{In}(i) = R$ in $G'$.

**Definition 8:** For any $(n, k, r, t)$-FLRC $\{C_{\lambda}; \lambda \in \Lambda\}$, let

$$\delta^* \triangleq \min \{|S(G_{\lambda})|; \lambda \in \Lambda, G_{\lambda} \in \mathcal{G}_{\lambda}\}$$

(8)

where $\mathcal{G}_{\lambda}$ is the set of all repair graphs of $C_{\lambda}$. If $\lambda_0 \in \Lambda$ and $G_{\lambda_0}$ is a repair graph of $C_{\lambda_0}$ such that $\delta^* = |S(G_{\lambda_0})|$, then we call $G_{\lambda_0}$ a minimal repair graph of $\{C_{\lambda}; \lambda \in \Lambda\}$.

**Remark 9:** Note that for any $(n, k, r, t)$-FLRC $\{C_{\lambda}; \lambda \in \Lambda\}$, $\{|S(G_{\lambda})|; \lambda \in \Lambda, G_{\lambda} \in \mathcal{G}_{\lambda}\} \subseteq [n]$ is a finite set. So by (8), we can always find a $\lambda_0 \in \Lambda$ and a repair graph $G_{\lambda_0}$ of $C_{\lambda_0}$ such that $\delta^* = |S(G_{\lambda_0})|$. Thus, any $(n, k, r, t)$-FLRC has at least one minimal repair graph.

### III. Properties of Minimal Repair Graph

In this section, we investigate the properties of minimal repair graphs of $(n, k, r, t)$-FLRC, which will be used to derive a lower bound on the code length $n$ in the next section. Our discussions are summarized and illustrated in Fig. 3.

In this section, we assume $\{C_{\lambda}; \lambda \in \Lambda\}$ is an $(n, k, r, t)$-FLRC and $G_{\lambda_0} = (\mathcal{V}, \mathcal{E}) \in \mathcal{G}_{\lambda_0}$ is a minimal repair graph of $\{C_{\lambda}; \lambda \in \Lambda\}$, where $\lambda_0 \in \Lambda$. Note that the node set $\mathcal{V} = [n]$.

By Definition 8 and 7, $G_{\lambda_0}$ has $n - \delta^*$ inner nodes and each inner node of $G_{\lambda_0}$ has at most $r$ in-neighbors. So we have

$$\delta^* r \geq |\mathcal{E}|.$$  

(9)

The following lemma shows that the dimension $k$ is upper bounded by the number of sources of $G_{\lambda_0}$.

**Lemma 10:** For any $(n, k, r, t)$-FLRC $\{C_{\lambda}; \lambda \in \Lambda\}$,

$$k \leq \delta^* = |S(G_{\lambda_0})|.$$  

(10)

**Proof:** Consider an arbitrary $\lambda \in \Lambda$ and an arbitrary repair graph $G_{\lambda}$ of $C_{\lambda}$. By Definition 7, $G_{\lambda}$ is acyclic and for each inner node $j$, $\text{In}(j)$ is an $(r, C)$-repair set of $j$. Then by Definition 1 and by induction, for all codeword $x = (x_1, x_2, \cdots, x_n) \in C_{\lambda}$ and all $j \in [n]$, the $j$th code symbol $x_j$ is an $\mathbb{F}$-linear combination of the symbols in $\{x_i; i \in S(G_{\lambda_0})\}$. So the set $S(G_{\lambda})$ contains an information set of $C_{\lambda}$, which implies that $k \leq |S(G_{\lambda})|$. Since $\lambda$ is an arbitrary element of $\Lambda$ and $G_{\lambda}$ is an arbitrary
repair graph of $C_\lambda$, then by Definition 8 we have $k \leq \min\{|S(G_\lambda)|; \lambda \in \Lambda, G_\lambda \in \mathcal{G}_\lambda\} = \delta^* = |S(G_{\lambda_0})|$, which proves the lemma.

The following lemma and its corollaries give some structural properties of $G_{\lambda_0}$.

**Lemma 11:** For any $E \subseteq [n]$ of size $|E| = t' \leq t$,

$$|\text{Out}(E)| \geq |E \cap S(G_{\lambda_0})|. \quad (11)$$

**Proof:** We can prove this lemma by contradiction.

By Definition 4 there is a $\lambda_1 \in \Lambda$ such that $C_{\lambda_1}$ is an $(E, r)$-repair code of $C_{\lambda_0}$. By Definition 2 there exists an index of $E$, say $E = \{i_1, i_2, \ldots, i_{t'}\}$, and a collection of subsets

$$\{R_{t'} \subseteq \overline{E} \cup \{i_1, \ldots, i_{t-1}\}; |R_{t'}| \leq r, t' \in [t']\}$$

such that $R_{t'}$ is an $(r, C_{\lambda_1})$-repair set of $i_t$ for each $t' \in [t']$. We construct a repair graph $G_{\lambda_1}$ of $C_{\lambda_1}$ as follows: First, for each $i_t \in E \cup \text{Out}(E)$ and $j \in \text{In}(i_t)$, delete the edge $(j, i_t)$; Then for each $i_t \in E$ and each $j \in R_{t'}$, add a direct edge from $j$ to $i_t$.

Clearly, $S(G_{\lambda_1}) = (S(G_{\lambda_1})) \cup \text{Out}(E)$. Here we fix the notation $\text{Out}(E)$ to be defined in $G_{\lambda_0}$. For each inner node $i$ of $G_{\lambda_1}$, we have the following two cases:

**Case 1:** $i \in E$. Then $i = i_t$ for some $i_t \in E$ and by the construction of $G_{\lambda_1}$. In $(i) = R_{t'}$ is an $(r, C_{\lambda_1})$-repair set of $i$.

**Case 2:** $i$ is an inner node of $G_{\lambda_0}$ and $i \notin \text{Out}(E)$. Then In $(i) = \overline{E} = \{j \in E; \lambda \in \Lambda\}$ is an $(r, C_{\lambda_0})$-repair set of $i$. Moreover, since $C_{\lambda_1}$ is an $(E, r)$-repair code of $C_{\lambda_0}$, then by condition (ii) of Definition 3, $C_{\lambda_1} = C_{\lambda_0}$. So In $(i)$ is also an $(r, C_{\lambda_1})$-repair set of $i$.

Thus, for each inner node $i$ of $G_{\lambda_1}$, In $(i)$ is an $(r, C_{\lambda_1})$-repair set of $i$. So $G_{\lambda_1}$ is a repair graph of $C_{\lambda_1}$.

Now, suppose $|\text{Out}(E)| < |E \cap S(G_{\lambda_0})|$. Then we have

$$|S(G_{\lambda_1})| = |S(G_{\lambda_0}) \cup \text{Out}(E)| = |S(G_{\lambda_0}) \cup \text{Out}(E)| + |\text{Out}(E)| = |S(G_{\lambda_0})| - |E \cap S(G_{\lambda_0})| + |\text{Out}(E)| < |S(G_{\lambda_0})|$$

which contradicts to Definition 8. Thus, by contradiction, we have $|\text{Out}(E)| \geq |E \cap S(G_{\lambda_0})|$.

**Example 12:** Let $G_{\lambda_0}$ be as in Fig. 2 and $G_{\lambda_0}$ be a repair graph of $C_{\lambda_0}$ with repair locality $r = 2$. By Definition 7, $\{2, 3\}$ is a $(r, C_{\lambda_0})$-repair set of 9, $\{3, 4\}$ is a repair set of 10, etc. Let $C_{\lambda_1}$ be an $(E = \{2, 3, 9\}, r)$-repair code of $C_{\lambda_0}$ such that the $(r, C_{\lambda_1})$-repair sets of 2, 3 and 9 are $\{1, 10\}, \{12, 13\}$ and $\{11, 14\}$ respectively. As in the proof of Lemma 11 we can construct a graph $G_{\lambda_1}$ as in Fig. 2. In $G_{\lambda_0}$, we have $\text{Out}(E) = \{10\}$. In $G_{\lambda_1}$, we have $S(G_{\lambda_1}) = (S(G_{\lambda_0}) \cup \text{Out}(E)) = (S(G_{\lambda_0}) \cup \{10\})$. Moreover, we can check that $G_{\lambda_1}$ is a repair graph of $C_{\lambda_1}$. In fact, note that by Definition 6 $C_{\lambda_1} = C_{\lambda_0}$. We define $E = \{n\} \setminus E$. Then $\{4, 5\}$ is also an $(r, C_{\lambda_1})$-repair set of 11. Similarly, $\{6, 7\}$ is an $(r, C_{\lambda_1})$-repair set of 12, etc. So $G_{\lambda_1}$ is a repair graph of $C_{\lambda_1}$.

**Corollary 13:** Suppose $t \geq 3$. For any source $v$, the following hold:

1. $|\text{Out}(v)| \geq 1$.
2. If $|\text{Out}(v)| = 1$, then $\text{Out}^2(v) = \text{Out}(v') \neq \emptyset$, where $v'$ is the unique out-neighbor of $v$. 

Fig 3. Relationship of discussions in Section III and IV.
3) If \( \text{Out}(v) = \{v_1\} \) and \( \text{Out}(v_1) = \{v_2\} \) for some inner nodes \( v_1 \) and \( v_2 \), then \( \text{Out}(v_2) \neq \emptyset \).
4) If \( \text{Out}(v) = \{v_1\} \) and \( \text{Out}(v_1) = \{v_2\} \) for some inner nodes \( v_1 \) and \( v_2 \), then \( |\text{Out}(u)| \geq 2 \) for any source \( u \) that belongs to \( \text{In}(v_2) \).
5) If \( v \) and \( w \) are two different sources and \( |\text{Out}(v)| = |\text{Out}(w)| = 1 \), then the unique out-neighbor of \( v \) is different from the unique out-neighbor of \( w \).

**Proof:** We can prove all claims by contradiction.

1) Suppose \( v \) has no out-neighbor. Picking \( E = \{v\} \), then \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{v\}| = 1 \), which contradicts to Lemma 11 (e.g., see 1) of example 14). Thus, \( v \) must have at least one out-neighbor.

2) Suppose \( \text{Out}(v') = \emptyset \). Picking \( E = \{v, v', v_1\} \), then \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{v\}| = 1 \), which contradicts to Lemma 11 (e.g., see 2) of example 14). So it must be that \( \text{Out}(v') \neq \emptyset \). Since \( G_{\lambda_0} \) is acyclic and \( \{v'\} = \text{Out}(v) \), then \( v' \in \text{Out}(v') \).

3) Suppose \( \text{Out}(v_2) = \emptyset \). Picking \( E = \{v, v_1, v_2\} \), then \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{v\}| = 1 \), which contradicts to Lemma 11 (e.g., see 3) of example 14). So it must be that \( \text{Out}(v_2) \neq \emptyset \).

4) Suppose \( |\text{Out}(u)| < 2 \). Since \( u \in \text{In}(v_2) \), then \( \text{Out}(u) = \{v_2\} \). Picking \( E = \{v, v_1, u\} \), we have \( |\text{Out}(E)| = |\{v_2\}| = 1 < |E \cap S(G_{\lambda_0})| = |\{v, u\}| = 2 \), which contradicts to Lemma 11 (See 4) of example 14). So it must be that \( |\text{Out}(u)| \geq 2 \).

5) Suppose \( \text{Out}(v) = \text{Out}(w) = \{v_1\} \). Picking \( E = \{v, w\} \), we have \( |\text{Out}(E)| = |\{v_1\}| = 1 < |E \cap S(G_{\lambda_0})| = |\{v, w\}| = 2 \), which contradicts to Lemma 11 (e.g., see 5) of example 14). Thus, the out-neighbor of \( v \) and \( w \) must be different.

The following example illustrates the arguments in the proof of Corollary 15.

**Example 14:** For the repair graph \( G_{\lambda_0} \) in Fig. 2, we have the following observations:

1) Let \( v = 1 \). Note that \( \text{Out}(1) = \emptyset \). If we pick \( E = \{1\} \), then we have \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{1\}| = 1 \).
2) Let \( v = 2 \) and \( v' = 9 \). Note that \( \text{Out}(9) = \emptyset \). If we pick \( E = \{2, 9\} \), then \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{2\}| = 1 \).
3) Let \( v = 5 \), \( v_1 = 11 \) and \( v_2 = 13 \). Note that \( \text{Out}(13) = \emptyset \). If we pick \( E = \{5, 11, 13\} \), then \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{5\}| = 1 \).
4) Let \( v = 6 \), \( v_1 = 12 \), \( v_2 = 14 \) and \( u = 8 \). Note that \( |\text{Out}(8)| = 1 \). If we pick \( E = \{6, 8, 12\} \), then \( |\text{Out}(E)| = |\{14\}| = 1 < |E \cap S(G_{\lambda_0})| = |\{6, 8, 12\}| = 2 \).
5) Let \( v = 6 \), \( u = 7 \) and \( v_1 = 12 \). If we pick \( E = \{6, 7\} \), then \( |\text{Out}(E)| = |\{12\}| = 1 < |E \cap S(G_{\lambda_0})| = |\{6, 7\}| = 2 \).

**Remark 15:** In Corollary 15 (1) holds for all \( t \geq 1 \) and 2), 5) hold for all \( t \geq 2 \). In fact, in the proof of 1), contradiction is derived from a subset \( E \) of size 1. So the proof is valid for all \( t \geq 1 \). Hence, 1) holds for all \( t \geq 1 \). Similarly, checking the proof of 2) and 5), we can see that they hold for all \( t \geq 2 \).

**Corollary 16:** Suppose \( v \in S(G_{\lambda_0}) \) and \( \text{Out}(v) = \{v_1, v_2\} \) for some inner nodes \( v_1 \) and \( v_2 \). If \( t \geq 3 \), the following hold:
1) \( \text{Out}(v_1) \neq \emptyset \) or \( \text{Out}(v_2) \neq \emptyset \).
2) If \( v_1 = \text{Out}(u) \) for some source \( u \), then \( \text{Out}(v_2) \neq \emptyset \).
3) If \( v_1 = \text{Out}(u) \) for some source \( u \), then \( |\text{Out}(u)| \geq 2 \) for any source \( u \) that belongs to \( \text{In}(v_2) \).

**Proof:** We can prove all claims by contradiction.

1) Suppose \( \text{Out}(v_1) = \emptyset \) and \( \text{Out}(v_2) = \emptyset \). Picking \( E = \{v, v_1, v_2\} \), we have \( |\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G_{\lambda_0})| = |\{v\}| = 1 \), which contradicts to Lemma 11 (See 1) of example 17). So it must be that \( \text{Out}(v_1) \neq \emptyset \) or \( \text{Out}(v_2) \neq \emptyset \).
2) Suppose $\text{Out}(v_2) = \emptyset$. Picking $E = \{u, v, v_2\}$, we have $|\text{Out}(E)| = |\{v_2\}| = 1 < |E \cap S(G)\lambda_0| = |\{v, u\}| = 2$, which contradicts to Lemma 11 (e.g., see 2) of example 17). So it must be that $\text{Out}(v_2) \neq \emptyset$.

3) Suppose $w \in \text{In}(v_2)$ is a source and $|\text{Out}(w)| < 2$. Then $\text{Out}(w) = \{v_2\}$. Picking $E = \{u, v, w\}$, we have $|\text{Out}(E)| = |\{v_1, v_2\}| = 2 < |E \cap S(G)\lambda_0| = |\{u, v, w\}| = 3$, which contradicts to Lemma 11 (e.g., see 3) of example 17). So it must be that $|\text{Out}(w)| \geq 2$.

The following example illustrates the arguments in the proof of Corollary 16.

Example 17: For the repair graph $G$ in Fig. 5, we have the following observations:

1) Let $v = 5, v_1 = 9$ and $v_2 = 10$. Note that $\text{Out}(9) = \text{Out}(10) = \emptyset$. If we pick $E = \{5, 9, 10\}$, then we have $|\text{Out}(E)| = |\emptyset| = 0 < |E \cap S(G)| = |\{5\}| = 1$.

2) Let $v = 2, v_1 = 7, v_2 = 8$ and $u = 1$. Note that $\text{Out}(8) = \emptyset$. If we pick $E = \{1, 2, 8\}$, then $|\text{Out}(E)| = |\{7\}| = 1 < |E \cap S(G)\lambda_0| = |\{1, 2\}| = 2$.

3) Let $v = 2, v_1 = 7, v_2 = 8, u = 1$ and $w = 3$. Note that $\text{Out}(3) = \emptyset$. If we pick $E = \{1, 2, 3\}$, then $|\text{Out}(E)| = |\{7, 8\}| = 2 < |E \cap S(G)\lambda_0| = |\{1, 2, 3\}| = 3$.

IV. Bound of Code Length

In this section, we will prove a lower bound on the code length $n$ for $(n, k, r, t)$-FLRC with $t \in \{2, 3\}$.

A. Code Length for $(n, k, r, 2)$-FLRC

The following theorem gives a lower bound on the code length of $(n, k, r, 2)$-FLRC.

Theorem 18: For $(n, k, r, 2)$-FLRC, we have

$$n \geq k + \left\lceil \frac{2k}{r} \right\rceil. \quad (12)$$

Proof: Suppose $\{C_\lambda; \lambda \in A\}$ is an $(n, k, r, 2)$-FLRC and $G_{\lambda_0} = (\mathcal{V}, \mathcal{E})$ is a minimal repair graph of $\{C_\lambda; \lambda \in A\}$, where $\lambda_0 \in \Lambda$. $\mathcal{V} = [n]$ is the node set of $G_{\lambda_0}$ and $\mathcal{E}$ is the edge set of $G_{\lambda_0}$. We first prove $n \geq \delta^* + \frac{2k}{r}$, where $\delta^* = |S(G_{\lambda_0})|$.

By Remark 15 and 1) of Corollary 13 each source of $G_{\lambda_0}$ has at least one out-neighbor. Let $\mathcal{E}_{\text{red}}$ be the set of all edge $e$ such that the tail of $e$ is a source. We call each edge in $\mathcal{E}_{\text{red}}$ a red edge. Let $A$ be the set of all source $v$ such that $v$ has only one out-neighbor. Then the number of all red edges is $|\mathcal{E}_{\text{red}}| \geq |A| + 2(|S(G_{\lambda_0}) \cap A|) = |A| + 2(|S(G_{\lambda_0})| - |A|) = 2|S(G_{\lambda_0})| - |A| = 2\delta^* - |A|$. Thus, we have

$$|\mathcal{E}_{\text{red}}| \geq 2\delta^* - |A|. \quad (13)$$

For each $v \in A$, since $v$ has only one out-neighbor, by Remark 15 and 2) of Corollary 13 $\text{Out}^2(v) = \text{Out}(v') \neq \emptyset$, where $v'$ is the unique out-neighbor of $v$. Let $\mathcal{E}_{\text{green}}(v)$ be the set of all edges whose tail is $v'$. Then $\mathcal{E}_{\text{green}}(v) \neq \emptyset$. Let $\mathcal{E}_{\text{green}} = \bigcup_{v \in A} \mathcal{E}_{\text{green}}(v)$. We call each edge in $\mathcal{E}_{\text{green}}$ a green edge. For any two different $v_1, v_2 \in A$, let $v_1', v_2'$ be the unique out-neighbor of $v_1, v_2$ respectively. By Remark 15 and 5) of Corollary 13 $v_1' \neq v_2'$. So we have $\mathcal{E}_{\text{green}}(v_1) \cap \mathcal{E}_{\text{green}}(v_2) = \emptyset$. Thus, the number of all green edges is $|\mathcal{E}_{\text{green}}| = |\bigcup_{v \in A} \mathcal{E}_{\text{green}}(v)| = \sum_{v \in A} |\mathcal{E}_{\text{green}}(v)| \geq |A|$, i.e.,

$$|\mathcal{E}_{\text{green}}| \geq |A|. \quad (14)$$

Clearly, $\mathcal{E}_{\text{red}} \cap \mathcal{E}_{\text{green}} = \emptyset$. Then by (13) and (14), we have

$$|\mathcal{E}| \geq |\mathcal{E}_{\text{red}} \cup \mathcal{E}_{\text{green}}| = |\mathcal{E}_{\text{red}}| + |\mathcal{E}_{\text{green}}| \geq 2\delta^*.$$

On the other hand, by (9), we have

$$|(n - \delta^*)r \geq |\mathcal{E}|.$$

Thus, we have $(n - \delta^*)r \geq 2\delta^*$, which implies that $nr \geq \delta^*(r + 2)$. So $n \geq \frac{\delta^*(r+2)}{r} = \delta^* + \frac{2\delta^*}{r}$.

By Lemma 10 $k \leq \delta^* = |S(G_{\lambda_0})|$. So $n \geq \delta^* + \frac{2\delta^*}{r} \geq k + \frac{2k}{r}$. Moreover, since $n$ is an positive integer, then we have $n \geq k + \left\lceil \frac{2k}{r} \right\rceil$, which proves (12).

In [14], it was proved that the code rate of an $(n, k, r, 2)$-FLRC satisfies bound (5). Note that (12) also implies $\frac{k}{n} \leq \frac{r}{n+2}$. So our result generalizes bound (5) to $(n, k, r, 2)$-FLRC.
B. Code Length for \((n, k, r, 3)\)-FLRC

The following theorem gives a lower bound on the code length of \((n, k, r, 3)\)-FLRC.

**Theorem 19:** For \((n, k, r, 3)\)-FLRC, we have

\[
n \geq k + \left[\frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r}\right].
\]

Before proving Theorem 19 we first prove the following Lemma 20. In the rest of this subsection, we always assume \(\{C_{\lambda}; \lambda \in \Lambda\}\) is an \((n, k, r, 3)\)-FLRC and \(G_{\lambda_0} = \langle V, E \rangle\) is a minimal repair graph of \(\{C_{\lambda}; \lambda \in \Lambda\}\), where \(\lambda_0 \in \Lambda, V = [n]\) is the node set of \(G_{\lambda_0}\) and \(E\) is the edge set of \(G_{\lambda_0}\). Then \(\delta^* = |S(G_{\lambda_0})|\), where \(\delta^*\) is defined by (8).

**Lemma 20:** For \((n, k, r, 3)\)-FLRC, we have

\[
(n - \delta^*)r \geq |E| \geq 2\delta^* + \left\lceil \frac{\delta^*}{r} \right\rceil.
\]

**Proof:** By (9), we have \((n - \delta^*)r \geq |E|\), which proves the first inequality of (16). So we only need to prove the second inequality of (16). To do this, we will divide the source set \(S(G_{\lambda_0})\) and the edge set \(E\) into mutually disjoint subsets.

We can divide the source set \(S(G_{\lambda_0})\) into four subsets \(A, B, C_1\) and \(C_2\) as follows:

\[
A = \{v \in S(G_{\lambda_0}); |\text{Out}(v)| \geq 3\},
\]

\[
B = \{v \in S(G_{\lambda_0}); |\text{Out}(v)| = 2\},
\]

\[
C_1 = \{v \in S(G_{\lambda_0}); |\text{Out}(v)| = 1 \text{ and } |\text{Out}^2(v)| = 1\}
\]

and

\[
C_2 = \{v \in S(G_{\lambda_0}); |\text{Out}(v)| = 1 \text{ and } |\text{Out}^2(v)| \geq 2\}.
\]

Clearly, \(A, B, C_1\) and \(C_2\) are mutually disjoint. Moreover, by 1), 2) of Corollary 13 \(S(G_{\lambda_0}) = A \cup B \cup C_1 \cup C_2\). Hence,

\[
\delta^* = |S(G_{\lambda_0})| = |A| + |B| + |C_1| + |C_2|.
\]

We can divide the edge set \(E\) into three subsets as follows.

Firstly, an edge is called a **red edge** if its tail is a source. For each \(v \in S(G_{\lambda_0})\), let \(E_{\text{red}}(v)\) be the set of all red edges whose tail is \(v\) and \(E_{\text{red}} = \bigcup_{v \in S(G_{\lambda_0})} E_{\text{red}}(v)\) be the set of all red edges. Clearly, \(|E_{\text{red}}(v)| = |\text{Out}(v)|\) and \(E_{\text{red}}(w) \cap E_{\text{red}}(v) = \emptyset\) for any source \(w \neq v\). So by (17) - (20), we have

\[
|E_{\text{red}}| = \sum_{v \in S(G_{\lambda_0})} |\text{Out}(v)| \geq 3|A| + 2|B| + |C_1| + |C_2|.
\]

Secondly, an edge is called a **green edge** if its tail is the unique out-neighbor of some source in \(C_1 \cup C_2\). For each \(v \in C_1 \cup C_2\), let \(E_{\text{green}}(v)\) be the set of all green edges whose tail is the unique out-neighbor of \(v\) and \(E_{\text{green}} = \bigcup_{v \in C_1 \cup C_2} E_{\text{green}}(v)\) be the set of all green edges. Note that by 2) of Corollary 13 \(Out^2(v) = \text{Out}(v') \neq \emptyset\), where \(v'\) is the unique out-neighbor of \(v\). Then \(|E_{\text{green}}(v)| = |\text{Out}^2(v)|\). Moreover, if \(v, w \in C_1 \cup C_2\) are different, then by 5) of Corollary 13 their out-neighbors are different. So \(E_{\text{green}}(v) \cap E_{\text{green}}(w) = \emptyset\). Hence, by (19) and (20), we have

\[
|E_{\text{green}}| = \sum_{v \in C_1 \cup C_2} |\text{Out}^2(v)| \geq |C_1| + 2|C_2|.
\]

Thirdly, suppose \(v \in B \cup C_1\) and \(e \in E\) such that \(e\) is neither a red edge nor a green edge. Then \(e\) is called a **blue edge** belonging to \(v\) if one of the following conditions hold:

(a) \(v \in B\) and the tail of \(e\) belongs to \(\text{Out}(v)\).
(b) \(v \in C_1\) and the tail of \(e\) belongs to \(\text{Out}^2(v)\).

Let \(E_{\text{blue}}(v)\) denote the set of all blue edges belonging to \(v\) and \(E_{\text{blue}} = \bigcup_{v \in B \cup C_1} E_{\text{blue}}(v)\). We have the following claim 1, whose proof is given in Appendix A.

**Claim 1:** The number of blue edges is bounded by

\[
|E_{\text{blue}}| \geq \frac{|B| + |C_1|}{r}.
\]
Clearly, \( E_{\text{red}}, E_{\text{green}} \) and \( E_{\text{blue}} \) are mutually disjoint. Then by (21)-(24), we have

\[
|E| \geq |E_{\text{red}}| + |E_{\text{green}}| + |E_{\text{blue}}| \\
\geq (3|A| + 2|B| + |C_1| + |C_2|) \\
+ (|C_1| + 2|C_2|) + \frac{|B| + |C_1|}{r} \\
= 2(|A| + |B| + |C_1| + |C_2|) \\
+ (|A| + |C_2| + \frac{|B| + |C_1|}{r}) \\
= 2\delta^* + r|A| + r|C_2| + \frac{|B| + |C_1|}{r} \\
\geq 2\delta^* + \frac{|A| + |C_2| + |B| + |C_1|}{r} \\
= 2\delta^* + \frac{\delta^*}{r}.
\]

Note that \(|E|\) is an integer. Then we have \(|E| \geq 2\delta^* + \left\lceil \frac{\delta^*}{r} \right\rceil\), which proves the second inequality of (16).

By the above discussion, we proved (16), which in turn proves Lemma 20.

To help the reader to understand the proof of Lemma 20, we give an example as follows.

**Example 21:** Consider the graph in Fig. 6. Using the notations defined in the proof of Lemma 20, we have \( A = \{2, 4, 7\}, B = \{3, 6\}, C_1 = \{1\} \) and \( C_2 = \{5\} \).

It is easy to find all red edges. We can also easily find that \( E_{\text{green}}(1) = \{(8, 11)\} \) and \( E_{\text{green}}(5) = \{(10, 12), (10, 13)\} \).

Since \( 1 \in C_1 \) and \( 11 \in \text{Out}^2(1) \), then \((11, 14) \in E_{\text{blue}}(1) \). Since \( 11 \in \text{Out}(6) \) and \( 6 \in B \), then \((11, 14) \in E_{\text{blue}}(6) \). Since \( 3 \in B \) and \( 9 \in \text{Out}(3) \), then \((9, 11) \in E_{\text{blue}}(3) \). We can further check that \( E_{\text{blue}}(1) = E_{\text{blue}}(6) = \{(11, 14)\} \) and \( E_{\text{blue}}(3) = \{(9, 11)\} \).

![Fig 6. An example of partitioning the edge set of minimal repair graph: The red (resp. green, blue) edges are colored by red (resp. green, blue).](image)

Now, using Lemma 20 and Lemma 10, we can give a simple proof of Theorem 19.

**Proof of Theorem 19.** By Lemma 20, we have

\[
(n - \delta^*)r \geq |E| \geq 2\delta^* + \left\lceil \frac{\delta^*}{r} \right\rceil.
\]

So

\[
(n - \delta^*)r \geq 2\delta^* + \left\lceil \frac{\delta^*}{r} \right\rceil.
\]

Solving \( n \) from the above equation, we can obtain

\[
n \geq \delta^* + \frac{2\delta^* + \left\lceil \frac{\delta^*}{r} \right\rceil}{r}.
\]

(25)

By Lemma 10 we have \( \delta^* \geq k \). So

\[
\delta^* + \frac{2\delta^* + \left\lceil \frac{\delta^*}{r} \right\rceil}{r} \geq k + \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r}.
\]

(26)

From (25) and (26), we have

\[
n \geq k + \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r}.
\]
Since \( n \) is a positive integer, then we have \( n \geq k + \left\lceil \frac{2k + \left\lfloor \frac{k}{r} \right\rfloor}{r} \right\rceil \), which proves Theorem 19. 

We next show that the bound (15) improves the bound (3) for codes with all-symbol \((r, 4)\)-locality. Note that for such codes, the bound (3) is equivalent to

\[
n \geq \frac{r + 1}{r - 2r} + \frac{1}{2r} + \frac{1}{3r} - k.
\]

Also note that codes with all-symbol \((r, 4)\)-locality are \((n, k, r, 3)\)-ELRC. Then by (15), we have

\[
n \geq k + \left\lceil \frac{2k + \left\lfloor \frac{k}{r} \right\rfloor}{r} \right\rceil \geq k + \frac{2k + \left\lfloor \frac{k}{r} \right\rfloor}{r}.
\]

It is easy to check that

\[
\left( k + \frac{2k + \left\lfloor \frac{k}{r} \right\rfloor}{r} \right) - \left( \frac{r + 1}{r - 2r} + \frac{1}{2r} + \frac{1}{3r} - k \right) = \frac{1}{r} \left( \frac{k}{r} - \frac{k}{r} \right) + \frac{k}{6r} \left( 1 - \frac{1}{r^2} \right) \\
\geq 0.
\]

So (28) is an improvement of (27).

An illustration of the gap between the bounds (15) and (3) for the parameters \( t = r = 3 \) is given in Fig. 7 from which we can see that (15) is tighter than (3) for \( t = 3 \).

V. Code construction

In this section, we give some constructions of \((n, k, r, 2)\)-ELRCs and \((n, k, r, 3)\)-ELRCs whose length \( n \) achieve the bounds (12) and (15) respectively. We call such codes optimal \((n, k, r, 2)\)-ELRC and optimal \((n, k, r, 3)\)-ELRC respectively. By these constructions, we prove the tightness of the bound (12) and (15). Moreover interestingly, our results show that for some sets of parameters, exact LRCs is sufficient to achieve the optimal code length of functional LRCs. Our discussions are summarized and illustrated in Fig. 8.

We begin with a lemma that gives a method to construct subsets of \([n]\) that can be used to construct repair set for LRC.

Lemma 22: Let \( \mathcal{L} = \{C_1, \ldots, C_N\} \) be a collection of pairwise disjoint subsets of \([n]\) and \((r_1, r_2, \ldots, r_K)\) be a \(K\)-tuple of positive integers such that \( \sum_{i=1}^{N} |C_i| = \sum_{i=1}^{K} r_i \). Let \( M \) be a \( K \times N \) binary matrix such that for each \( i \in [K] \) and each \( j \in [N] \), the sum of the \( i \)th row is \( r_i \) and the sum of the \( j \)th column is \( |C_j| \). Then there exists a collection \( \{B_1, \ldots, B_K\} \) of subsets of \( \bigcup_{j=1}^{N} C_j \) such that:

(i) \( B_1, \ldots, B_K \) are pairwise disjoint and \( \bigcup_{i=1}^{K} B_i = \bigcup_{j=1}^{N} C_j \);

(ii) \( |B_i| = r_i \) for all \( i \in [K] \);

(iii) \( |B_i \cap C_j| \leq 1 \) for all \( i \in [K] \) and \( j \in [N] \).

Proof: For each \( j \in [N] \), since the sum of the \( j \)th column of \( M \) is \( |C_j| \), we can replace the ones of the \( j \)th column by elements of \( C_j \) such that each element of \( C_j \) appears exactly once. Denote the resulted matrix by \( M' \). Now for each \( i \in [K] \), let \( B_i \) be the elements of the \( i \)th row of \( M' \) except the zeros.
Since $C_1, \ldots, C_N$ are pairwise disjoint and for each $j \in [N]$, each element of $C_j$ appears exactly once in the $j$th column of $M'$, then each element of $\bigcup_{j=1}^N C_j$ appears exactly once in $M'$, which implies conditions (i) and (iii). Moreover, since the sum of the $i$th row of $M$ is $r_i$, then $|B_i| = r_i$ for all $i \in [K]$. So condition (ii) is satisfied.

We give an example in the below to demonstrate the construction method used in the proof Lemma 22.

**Example 23:** Let $C_1 = \{1, 2, 3, 4, 5\}$, $C_2 = \{6, 7, 8, 9, 10\}$, $C_3 = \{11, 12, 13, 14, 15\}$, $C_4 = \{16, 17, 18, 19, 20\}$, $C_5 = \{22, 23, 24, 25\}$, $C_6 = \{27, 28, 29, 30\}$ and $C_7 = \{31, 32, 33\}$. Let $r_1 = \cdots = r_5 = 5$ and $r_6 = r_7 = 3$. Then we have $\sum_{i=1}^7 |C_i| = 31 = \sum_{i=1}^7 r_i$. Let

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

We can check that for each $i, j \in \{1, 2, \cdots, 7\}$, the sum of the $i$th row is $r_i$ and the sum of the $j$th column is $|C_j|$. Replacing the ones of the $j$th column of $M$ by elements of $C_j$, we obtain

$$M' = \begin{pmatrix}
1 & 6 & 11 & 16 & 22 & 0 & 0 \\
0 & 7 & 12 & 17 & 0 & 27 & 31 \\
2 & 8 & 13 & 18 & 23 & 0 & 0 \\
3 & 0 & 14 & 19 & 0 & 28 & 32 \\
4 & 9 & 0 & 0 & 24 & 29 & 33 \\
5 & 10 & 15 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 20 & 25 & 30 & 0
\end{pmatrix}.$$ 

From $M'$, we can obtain subsets $B_1 = \{1, 6, 11, 16, 22\}$, $B_2 = \{7, 12, 17, 27, 31\}$, $B_3 = \{2, 8, 13, 18, 23\}$, $B_4 = \{3, 14, 19, 28, 32\}$, $B_5 = \{4, 9, 24, 29, 33\}$, $B_6 = \{5, 10, 15\}$ and $B_7 = \{20, 25, 30\}$. It is easy to check that conditions (i)–(iii) of Lemma 22 are satisfied.

**Corollary 24:** Let $\mathcal{L} = \{C_1, \ldots, C_N\}$ be a collection of pairwise disjoint $\delta$-subsets of $[n]$ and $\vec{r} = (r_1, \cdots, r_K)$ be a $K$-tuple of positive integers such that $\sum_{i=1}^K r_i = N$ and $r_i \leq |\mathcal{L}| = N$ for all $i \in [K]$. Then there exists a collection $\{B_1, \ldots, B_K\}$ of subsets of $\bigcup_{j=1}^N C_j$ such that:

(i) $B_1, \ldots, B_K$ are pairwise disjoint and $\bigcup_{i=1}^K B_i = \bigcup_{j=1}^N C_j$;

(ii) $|B_i| = r_i$ for all $i \in [K]$;

(iii) $|B_i \cap C_j| \leq 1$ for all $i \in [K]$ and $j \in [N]$.

**Proof:** Since $\sum_{i=1}^K r_i = N \delta$ and $r_i \leq N$ for all $i \in [K]$, using the Gale-Ryser Theorem (see Manfred 21), we can construct a $K \times N$ binary matrix $M$ such that for each $i \in [K]$ and each $j \in [N]$, the sum of the $i$th row of $M$ is $r_i$ and the
sum of the $j$th column of $M$ is $δ = |C_j|$. By Lemma 22 there exists a collection $\{B_1, \cdots, B_K\}$ of subsets of $\bigcup_{j=1}^NC_j$ that satisfies the conditions (i)—(iii).

The following two lemmas give a sufficient condition of $(n, k, r, 2)$-ELRC and $(n, k, r, 3)$-ELRC respectively.

Lemma 25: Let $C$ be an $[n,k]$ linear code and $[n] = S \cup T$ such that $S \cap T = \emptyset$. Then $C$ is an $(n,k,r,2)$-ELRC if the following two conditions hold:
(i) Each $i \in S$ has two disjoint $(r,C)$-repair sets;
(ii) Each $i \in T$ has an $(r,C)$-repair set $R i \subseteq [n] \setminus E$.

Proof: We will prove that for any $E \subseteq [n]$ of size $|E| \leq 2$, there exists an $i \in E$ such that $i$ has an $(r,C)$-repair set $R i \subseteq [n] \setminus E$. We have the following two cases:
Case 1: $E \cap S = \emptyset$. Then $E \subseteq T$ and by condition (ii) each $i \in E$ has an $(r,C)$-repair set $R i \subseteq [n] \setminus E$.
Case 2: $E \cap S \neq \emptyset$. Suppose $i \in E \cap S$. By condition (i) $i$ has two disjoint $(r,C)$-repair sets, say $R_1$ and $R_2$. Note that $|E| \leq 2$ and $i \notin R_1 \cup R_2$, then either $E \cap R_1 = \emptyset$ or $E \cap R_2 = \emptyset$. Without loss of generality, assume $E \cap R_1 = \emptyset$. Then we have $R_1 \subseteq [n] \setminus E$.

Thus, we can always find an $i \in E$ that has an $(r,C)$-repair set $R i \subseteq [n] \setminus E$. Thus, we can find an element of $\{i\}$ contained in $[n] \setminus E$. Lemma 26: Let $C$ be an $[n,k]$ linear code and $[n] = S \cup T$ such that $S \cap T = \emptyset$. Then $C$ is an $(n,k,r,3)$-ELRC if the following two conditions hold:
(i) Each $i \in S$ has two disjoint $(r,C)$-repair sets, say $R_1$ and $R_2$, such that each $j \in R_1$ has an $(r,C)$-repair set $R \cap (R_2 \cup \{i\}) = \emptyset$;
(ii) Each $i \in T$ has an $(r,C)$-repair set $R i \subseteq [n] \setminus E$.

Proof: For any $E \subseteq [n]$ of size $|E| \leq 3$, similar to the proof of Lemma 25 we have the following two cases:
Case 1: $E \cap S = \emptyset$. Then $E \subseteq T$ and by condition (ii) each $i \in E$ has an $(r,C)$-repair set $R i \subseteq [n] \setminus E$.
Case 2: $E \cap S \neq \emptyset$. Let $i \in E \cap S$. By condition (i) $i$ has two disjoint $(r,C)$-repair sets, say $R_1$ and $R_2$, such that each $j \in R_1$ has an $(r,C)$-repair set $R \cap (R_2 \cup \{i\}) = \emptyset$. Then we have the following two subcases:
Case 2.1: $E \cap R_1 = \emptyset$ or $E \cap R_2 = \emptyset$. If $E \cap R_1 = \emptyset$, then $R_1 \subseteq [n] \setminus E$; if $E \cap R_2 = \emptyset$, then $R_2 \subseteq [n] \setminus E$. So in this subcase, $i$ has an $(r,C)$-repair set contained in $[n] \setminus E$.
Case 2.2: $E \cap R_1 \neq \emptyset$ and $E \cap R_2 \neq \emptyset$. Assume $j \in E \cap R_1$ and $j' \in E \cap R_2$. Then by condition (i), $j$ has an $(r,C)$-repair set $R \cap (R_2 \cup \{i\}) = \emptyset$. So
\[ R \cap (R_2 \cup \{i,j\}) = \emptyset. \]
On the other hand, since $R_1 \cap R_2 = \emptyset$ and $|E| \leq 3$, then $j \neq j'$ and
\[ E = \{i,j,j'\} \subseteq R_2 \cup \{i,j\}. \]
Combining (29) and (28), we have $R \subseteq [n] \setminus E$. So in this subcase, $j \in E$ has an $(r,C)$-repair set $R \subseteq [n] \setminus E$.

Thus, we can find an element of $E$ that has an $(r,C)$-repair set $R \subseteq [n] \setminus E$. By Lemma 26 $C$ is an $(n,k,r,3)$-ELRC.

A. Optimal $(n,k,r,2)$-ELRC

In this subsection, we give a method for constructing $(n = k + \lceil 2k \rceil, k, r, 2)$-ELRC. Our construction is based on the following lemma.

Lemma 27: Suppose $\lceil \frac{k}{r} \rceil \geq r$. There exists a collection $\mathcal{A} = \{A_1, \cdots, A_N\}$ of $\eta = \lceil \frac{2k}{r} \rceil$ subsets of $[k]$ such that:
(i) $|A_i| \leq r$ for each $i \in [\eta]$;
(ii) $|A_i \cap A_j| \leq 1$ for all $\{i,j\} \subseteq [\eta]$;
(iii) Each $i \in [k]$ belongs to exactly two subsets in $\mathcal{A}$;

Proof: The proof is given in Appendix B.

The following are two examples of subsets that satisfy conditions (i)—(iii) of Lemmas 27.

Example 28: For $k = 12$ and $r = 3$, we have $\eta = \lceil \frac{297}{2} \rceil = 8$. Let $\mathcal{A} = \{A_1, \cdots, A_8\}$ be as in Fig. 29(a), where each subset in $\{A_1, \cdots, A_4\}$ is represented by a red line and each subset in $\{A_5, \cdots, A_8\}$ is represented by a blue line. We can check that conditions (i)—(iii) of Lemmas 27 are satisfied.

Example 29: For $k = 10$ and $r = 3$, we have $\eta = \lceil \frac{220}{2} \rceil = 7$. Let $\mathcal{A} = \{A_1, \cdots, A_7\}$ be as in Fig. 29(b), where each subset in $\{A_1, A_2, A_3\}$ is represented by a red solid line, $A_4$ is represented by a red dashed line and each subset in $\{A_5, A_6, A_7\}$ is represented by a blue line. We can check that conditions (i)—(iii) of Lemmas 27 are satisfied.

Now we have the following construction.

Construction 1: Let $\lceil \frac{k}{r} \rceil \geq r$ and $\mathcal{A} = \{A_1, \cdots, A_\eta\}$ be constructed as in Lemma 27 where $\eta = \lceil \frac{2k}{r} \rceil$. Let $x_1, \cdots, x_k$ be $k$ information symbols. Then we can construct a $[k+\eta,k]$ systematic linear code $C$ over $\mathbb{F}_2$ with $\eta$ parities $x_{k+1}, \cdots, x_{k+\eta}$ such that $x_{k+i} = \sum_{j \in A_i} x_j$ for each $i \in [\eta]$.

Theorem 30: The code $C$ obtained by Construction 1 is an $(n = k + \lceil \frac{2k}{r} \rceil, k, r, 2)$-ELRC.
Proof: Let $S = [k]$ and $T = \{k + 1, \ldots, k + \eta\}$, where $\eta = \left\lceil \frac{2k}{r} \right\rceil$. Then we have $S \cap T = \emptyset$. By conditions (ii), (iii) of Lemma 27 for each $i \in S$, there exist two subsets, say $A_{i_1}$ and $A_{i_2}$, such that $A_{i_1} \cap A_{i_2} = \{i\}$. By Construction 1 and condition (i) of Lemma 27 $R_1 = A_{i_1} \cup \{k + i_1\}$ and $R_2 = A_{i_2} \cup \{k + i_2\}$ are two disjoint $(r, C)$-repair sets of $i$. Moreover, for each $i \in T$, again by Construction 1 and condition (i) of Lemma 27 $A_{i \cdot k}$ is an $(r, C)$-repair set of $i$. So by Lemma 25 $C$ is an $(n, k, r, 2)$-ELRC.

Note that the code $C$ obtained by Construction 1 has length $n = k + \eta = k + \left\lceil \frac{2k}{r} \right\rceil$, which meets the bound (12). So from Theorem 10 we can directly obtain the following theorem.

**Theorem 31:** If $\left\lceil \frac{k}{r} \right\rceil \geq r$, then there exist $(n, k, r, 2)$-ELRC over the binary field that meet the bound (12).

The authors in [12] constructed binary codes with all-symbol locality $r$, availability $t$ and code rate $\frac{r}{r+t}$ for $n = \left(\begin{array}{c} r + t \\ r \end{array}\right)$ and any positive integer $r$ and $t$ (such codes are a subclass of $(n, k, r, t)$-ELRC). For $t = 2$, we have $n = \left(\begin{array}{c} r + 2(r+1) \\ 2 \end{array}\right)$ and $k = \frac{r(r+1)}{2}$. In our construction, we require that $\left\lceil \frac{k}{r} \right\rceil \geq r$, which implies that $k \geq r^2 - \frac{2(r+1)}{2}$ if $r > 1$.

### B. Optimal $(n, k, r, 3)$-ELRC

In this subsection, we give a method for constructing $(n = k + \left\lceil \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \right\rceil, k, r, 3)$-ELRC. We always denote

$$m = \left\lceil \frac{k}{r} \right\rceil$$

and

$$\ell = \left\lceil \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \right\rceil - \left\lceil \frac{k}{r} \right\rceil = \left\lceil \frac{2k + m}{r} \right\rceil - m.$$

Then we have

$$n = k + \left\lceil \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \right\rceil = k + m + \ell.$$

Our construction is closely related to the following concept.

**Definition 32:** A mesh of $[n]$ is a collection $\mathcal{R} \cup \mathcal{B}$ of subsets of $[n]$, where $\mathcal{R} = \{RL_1, \ldots, RL_m\}$ and $\mathcal{B} = \{BL_1, \ldots, BL_\ell\}$ are called red lines and blue lines respectively, that satisfies the following conditions:

(i) For each $i \in [m]$, $RL_i \subseteq [k + m]$, $|RL_i| = r + 1$ and $RL_i \cap \{k + 1, \ldots, k + m\} = \{k + i\}$;

(ii) For each $j \in [\ell]$, $BL_j \cap \{k + n + 1, \ldots, n\} = \{k + m + j\}$ and $|BL_j| \leq r + 1$;

(iii) Each $i \in [k + m]$ belongs to exactly two lines, at least one is a red line;

(iv) Any two different lines have at most one point in common;

(v) Any two different lines that intersect with the same red line are disjoint.

Here a line means a subset in $\mathcal{R} \cup \mathcal{B}$ (i.e., a red line or a blue line) and a point means an element of $[n]$.

**Example 33:** For $k = 12$ and $r = 3$, we have $m = \left\lceil \frac{k}{r} \right\rceil = 4$, $\ell = \left\lceil \frac{2k + \left\lceil \frac{k}{r} \right\rceil}{r} \right\rceil - \left\lceil \frac{k}{r} \right\rceil = 6$ and $n = k + m + \ell = 22$. Let $\mathcal{R} = \{RL_1, \ldots, RL_4\}$ be the red lines and $\{B_1, \ldots, B_6\}$ be the blue lines in Fig. 10(a). Then extend each $B_i$ to a blue line $BL_i$ as in Fig. 10(b). Let $\mathcal{B} = \{BL_1, \ldots, BL_6\}$. We can check that $\mathcal{R} \cup \mathcal{B}$ is a mesh of $[n]$.

**Example 34:** For $k = 16$ and $r = 3$, we have $m = 6$, $\ell = 7$ and $n = 29$. Let $\mathcal{R} = \{RL_1, \ldots, RL_5, RL_6\}$, where $RL_1, \ldots, RL_5$ are the red solid lines in Fig. 11(a) and $R_6$ is the red dashed line in Fig. 11(a). We partition the first three columns into $B_1 = \{2, 6, 9\}$, $B_2 = \{18, 19\}$, $B_3 = \{17\}$, $B_4 = \{3\}$, $B_5 = \{8\}$, $B_6 = \{1, 5\}$ and $B_7 = \{4, 7\}$. In Fig. 11(a), each $B_i$ of size $|B_i| \geq 2$ is represented by a blue line and the other points of the first three columns represent the $B_i$s of size 1. Further, we extend each $B_i$ to a blue line $BL_i$ as in Fig. 11(b). Let $\mathcal{B} = \{BL_1, \ldots, BL_6\}$. Then we can check that $\mathcal{R} \cup \mathcal{B}$ is a mesh of $[n]$.

The following two lemmas and their proofs give some constructions of mesh of $[n]$.

**Lemma 35:** If $r|k$ and $m \geq r$, there exists a mesh of $[n]$.
Lemma 36: Suppose \( \lambda = r \mod k > 0 \). If \( \ell \geq r + \lambda + 1 \) and \( m \geq 2r - \lambda + 1 \), then there exists a mesh of \([n]\).

Proof: The proof is given in Appendix D.

Construction 2: Let \( R \cup B \) be a mesh of \([n]\), where \( R = \{RL_1, \ldots, RL_m\} \) is the set of red lines and \( B = \{BL_1, \ldots, BL_\ell\} \) is the set of blue lines. Let \( x_1, \ldots, x_n \) be \( k \) information symbols. Then we can construct an \([n = k + m + \ell, k]\) systematic linear code \( C \) over \( \mathbb{F}_2 \) such that the parities are \( x_{k+1}, \ldots, x_n \) and are computed as follows:

- For each \( i \in [m] \),
  \[
  x_{k+i} = \sum_{j \in RL_i \setminus \{k+i\}} x_j.
  \]

- For each \( i \in [\ell] \),
  \[
  x_{k+m+i} = \sum_{j \in BL_i \setminus \{k+m+i\}} x_j.
  \]

Note that by condition (i) of Definition 32, for each \( i \in [m] \), we have \( RL_i \setminus \{k+i\} \subseteq [k] \). So by (31), \( x_{k+i} \) is computable from information symbols. Similarly, for each \( i \in [\ell] \), by condition (ii) of Definition 32, \( BL_i \setminus \{k+m+i\} \subseteq [k+m] \). So by (32), \( x_{k+m+i} \) is computable from \( \{x_j; j \in [k+m]\} \). Hence, Construction 2 is reasonable.

Theorem 37: The code \( C \) obtained by Construction 2 is an \((n = k + m + \ell, k, \ell, 3)\)-ELRC.

Proof: Let \( S = [k+m] \) and \( T = [k+m+1, \ldots, n] \). Then \( S \cap T = \emptyset \).

For each \( i \in S \), by conditions (iii) and (iv) of Definition 32 there exists a red line \( L \in R \) and a line \( L' \in R \cup B \) such that \( L \cap L' = \{i\} \). By conditions (i) and (ii) of Definition 32, \(|L \setminus \{i\}| = r\) and \(|L' \setminus \{i\}| \leq r \). By (31) and (32), \( R_1 = L \setminus \{i\} \) and \( R_2 = L' \setminus \{i\} \) are two disjoint \((r, C)\)-repair sets of \( i \). Moreover, for each \( j \in L \setminus \{i\} \), by condition (i) of Definition 32, \( j \in L \subseteq [k+m] \). Then by condition (iii) of Definition 32 there exists an \( L'' \in R \cup B \) such that \( L'' \neq L \) and \( j \in L'' \). Clearly, \( L'' \neq L' \). Without loss of generality, \( \{i, j\} \subseteq L \cap L' = L \cap L'' \), which contradicts to condition (iv) of Definition 32. So by condition (v) of Definition 32, \( L'' \cap L' = \emptyset \). Let \( R = L'' \setminus \{j\} \). Then \( R \cap (R_2 \cup \{i\}) \subseteq L'' \cap L' = \emptyset \) and by (31), (32), \( R \) is an \((r, C)\)-repair set of \( j \).
For each \( i \in T \), let \( i' = i - (k + m) \). Then \( i' \in [\ell] \). Let \( R = BL_{i'} \setminus \{i\} \). Then by condition (ii) of Definition 32 and by (32), \( R \subseteq [k + m] = S \) is an \((r, C)\)-repair sets of \( i \).

By Lemma 26, \( C \) is an \((n, k, r, 3)\)-ELRC.

Note that the code \( C \) obtained by Construction 2 has length \( n = k + m + \ell = k + \left\lceil \frac{2k + |E|}{r} \right\rceil \), which meets the bound (15).

So the following theorem is a direct consequence of Lemma 35,36 and Theorem 37.

Theorem 38: Suppose one of the following conditions hold:

(i) \( r|k \) and \( m \geq r \).

(ii) \( \ell \geq r + \lambda + 1 \) and \( m \geq 2r - \lambda + 1 \), where \( \lambda = r \mod k > 0 \).

Then there exist \((n, k, r, 3)\)-ELRC over the binary field that meet the bound (15).

Binary codes with all-symbol locality \( r \), availability \( t \) and code rate \( \frac{1}{r - t} \) are constructed in [12] for any positive integers \( r \) and \( t \) (such codes are a subclass of \((n, k, r, t)\)-ELRC). For \( t = 3 \), the code length is \( n = k + \frac{3}{r} = k + 3k > k + \left\lceil \frac{2k + |E|}{r} \right\rceil \).

Hence is not optimal according to the bound (15).

VI. Conclusions

We investigate the problem of coding for distributed storage system that can locally repair up to \( t \) failed nodes, where \( t \) is a given positive integer. Given the code dimension \( k \), the repair locality \( r \) and \( t \in \{2, 3\} \), we derive a lower bound on the code length \( n \) under the functional repair model. We also give some constructions of exact LRCs for \( t \in \{2, 3\} \) with binary field and whose length \( n \) achieves the corresponding bounds, which proves the tightness of our bounds and also implies that there is no gap between the optimal code length of functional LRCs and exact LRCs for certain sets of parameters.

Some problems are still open. For example, what is the optimal code length for \( t \geq 4 \)? Given \( n, k, r \) and \( t \), what is the upper bound of the minimum distance \( d^* \)? Another interesting problem is to construct functional locally repairable codes \( \{C_{\lambda}; \lambda \in \Lambda\} \) with small size of \( \Lambda \).

APPENDIX A

PROOF OF CLAIM 1

To prove Claim 1, the key is to prove the following two statements: a) For each \( v \in B \cup C_1 \), \( |\mathcal{E}_{\text{blue}}(v)| \geq 1 \); b) Each blue edge belongs to at most \( r \) different \( v \in B \cup C_1 \).

For each \( v \in B \), by (18), \( |\text{Out}(v)| = 2 \). So we can assume \( \text{Out}(v) = \{v_1, v_2\} \). Then \( v_1, v_2 \) are two inner nodes of \( G_{\lambda} \). By 1) of Corollary 16 \( \text{Out}(v_1) \neq \emptyset \) or \( \text{Out}(v_2) \neq \emptyset \). Without loss of generality, we can assume \( \text{Out}(v_1) \neq \emptyset \) and \( v_3 \in \text{Out}(v_1) \). Then we have the following two cases:

Case 1: \( (v_1, v_3) \) is not a green edge. Since \( v_1 \) is an inner node, then \( (v_1, v_3) \) is not a red edge. Note that \( v \in B \) and \( v_1 \in \text{Out}(v) \). Then \( (v_1, v_3) \) is a blue edge belonging to \( v \).

Case 2: \( (v_1, v_3) \) is a green edge. Then \( \{v_1\} = \text{Out}(u) \) for some \( u \in C_1 \cup C_2 \). By 2) of Corollary 16 \( \text{Out}(v_2) \neq \emptyset \). Let \( v_4 \in \text{Out}(v_2) \). Since \( v_2 \) is an inner node, then \( (v_2, v_4) \) is not a red edge. Note that by 3) of Corollary 16 \( |\text{Out}(w)| \geq 2 \) for any source \( w \in \text{In}(v_2) \). (As illustrated in Fig. 12(a).) Then \( (C_1 \cup C_2) \cap \text{In}(v_2) = \emptyset \), which implies that \( v_2 \notin \text{Out}(m) \) for any \( m \in C_1 \cup C_2 \). So \( (v_2, v_4) \) is not a green edge. Since \( v \in B \) and \( v_2 \in \text{Out}(v) \), then \( (v_2, v_4) \) is a blue edge belonging to \( v \).

![Fig 12](image-url) Illustration of the local graph in the proof of Claim 1.

In both cases, we can find a blue edge belonging to \( v \).

For each \( v \in C_1 \), by (19), \( |\text{Out}(v)| = |\text{Out}^2(v)| = 1 \). We can assume \( \text{Out}(v) = \{v_1\} \) and \( \text{Out}^2(v) = \{v_2\} \). Then \( v_1, v_2 \) are two inner nodes. By 2) of Corollary 13 we have \( \text{Out}^2(v) = \text{Out}(v_1) = \{v_2\} \). Further, by 3) of Corollary 13 we have \( \text{Out}(v_2) \neq \emptyset \). Let \( v_3 \in \text{Out}(v_2) \). Since \( v_2 \) is an inner node, the edge \( (v_2, v_3) \) is not a red edge. Not that by 4) of Corollary 13 \( |\text{Out}(u)| \geq 2 \) for any source \( u \in \text{In}(v_2) \). (As illustrated in Fig. 12(b).) Then we have \( (C_1 \cup C_2) \cap \text{In}(v_2) = \emptyset \), which implies that \( v_2 \notin \text{Out}(u) \) for any \( u \in C_1 \cup C_2 \). So \( (v_2, v_3) \) is not a green edge. Note that \( v \in C_1 \) and \( \text{Out}^2(v) = \text{Out}(v_1) = \{v_2\} \). So \( (v_2, v_3) \) is a blue edge belonging to \( v \).
By the above discussion, we proved that \(|E_{\text{blue}}(v)| \geq 1\) for each \(v \in B \cup C_1\), which proves the statement a).
Let \((u', u'')\) be a blue edge and \(S\) be the set of all \(v \in B \cup C_1\) such that \((u', u'')\) belongs to \(v\). For each \(v \in S\), we pick a \(\varphi(v) \in \text{In}(u')\) depending on the following two cases:

Case 1: \(v \in B\). Since \((u', u'')\) is a blue edge belongs to \(v\), then \(u' \in \text{Out}(v)\), which implies \(v \in \text{In}(u')\). Pick \(\varphi(v) = v\).

Case 2: \(v \in C_1\). By (19), \(|\text{Out}^2(v)| = |\text{Out}(v)| = 1\). Denote \(\text{Out}(v) = \{v'\}\). Then by 2) of Corollary 13 \(|\text{Out}^2(v) = \text{Out}(v)|\). Moreover, since \((u', u'')\) is a blue edge belongs to \(v\), then \(u' \in \text{Out}^2(v) = \text{Out}(v)\). So \(v' \in \text{In}(u')\). Pick \(\varphi(v) = v'\).

If \(v\) and \(w\) are two different sources in \(S \cap C_1\), by 5) of Corollary 13 their out-neighbors are different. So \(\varphi(v) \neq \varphi(w)\).

Thus, \(\varphi\) is a one-to-one correspondence between \(S\) and a subset of \(\text{In}(u')\). Note that \(|\text{In}(u')| \leq r\). So \(|S| \leq |\text{In}(u')| \leq r\). Thus, \((u', u'')\) belongs to at most \(r\) different \(v \in B \cup C_1\), which proves the statement b).

By statements a) and b), we have \(|E_{\text{blue}}| \geq \frac{|B| + |C_1|}{r} \cdot\frac{r}{27}\), which proves Claim 1.

**APPENDIX B**

**PROOF OF Lemma 27**

We need to consider two cases, i.e., \(r \mid k\) and \(r \nmid k\).

Case 1: \(r \mid k\). We can let \(k = mr\). Then \(\eta = \left[\frac{2k}{r}\right] = 2m\) and \(m = \left[\frac{k}{r}\right] \geq r\). By assumption of Lemma 27 \(m = \left[\frac{k}{r}\right] \geq r\). We assign the elements of \([k]\) in a \(r \times m\) array \(D = (a_{i,j})_{i \in [r], j \in [m]}\) as in Fig. 13 such that \([k] = \{a_{i,j}; i \in [r], j \in [m]\}\). For each \(j \in [m]\), let \(A_j = \{a_{i,j}; i \in [r]\}\). Then \(|A_j| = r, \forall i \in [m]\). In Fig. 13 each subset \(A_i\) is represented by a red line.

Fig 13. Partition of \([n]\): Each subset is represented by a red line.

Let \(\delta = r\) and \(\mathcal{L} = \{A_1, \ldots, A_m\}\). Then \(|A_i| = \delta\) for each \(i \in [m]\). Let \(r_i = r, \forall i \in [m]\). Then \(r_i = r, \forall i \in [m]\). Then \(\sum_{i=1}^{m} r_i = mr = \sum_{j=1}^{m} |A_j|\).

Since \(m \geq r = r_i, \forall i \in [m]\), then by Corollary 24 there exists a collection \(\{B_1, \ldots, B_m\}\) of subsets of \(\bigcup_{j=1}^{m} A_j = [k]\) that satisfies the following three properties:

- \(B_1, \ldots, B_m\) are pairwise disjoint and \(\bigcup_{i=1}^{m} B_i = \bigcup_{j=1}^{m} A_j = [k]\);
- \(|B_i| = r_i = r\) for all \(i \in [m]\);
- \(|B_i \cap A_j| \leq 1\) for all \(i, j \in [m]\).

For each \(i \in [m]\), let \(A_{m+i} = B_i\). Then it is easy to check that \(A = \{A_1, \ldots, A_m\}\) satisfies conditions (i)–(iii) of Lemma 27, where \(\eta = \left[\frac{2k}{r}\right] = 2m\).

Case 2: \(r \nmid k\). Let \(m = \left[\frac{k}{r}\right]\). Since \(r \nmid k\), then \(m - 1 = \left[\frac{k}{r}\right]\) and \(k = (m - 1)r + \lambda\), where \(0 < \lambda < r\). By assumption of Lemma 27 we have \(m - 1 = \left[\frac{k}{r}\right]\).

Let \(\alpha = m - 1 - (r - \lambda)\). We can assign elements of \([k]\) in an \(r \times m\) array \(D = (a_{i,j})_{i \in [r], j \in [m]}\) as in Fig. 14 such that \(\{a_{i,j}; i \in [r], j \in [m]\} \cup \{a_{i,m}; i \in [\lambda]\} = [k]\) and \(a_{i,m} = 0, \forall i \in [\lambda, + \ldots, r]\). Let

\[A_0 = \{a_{1,j}; j \in \{\alpha + 1, \ldots, m - 1\}\}.\]

Then \(|A_0| = (m - 1) - \alpha = r - \lambda\). Let

\[A_j = \begin{cases} \{a_{i,j}; i \in [r]\}, & \text{if } j \in [m - 1]; \\ \{a_{i,m}; i \in [\lambda]\} \cup A_0, & \text{if } j = m. \end{cases}\]

In Fig. 14 each subset in \(\{A_1, \ldots, A_{m-1}\}\) is represented by a red solid line and \(A_m\) is represented by a red dashed line. For convenience, we call each subset in \(\{A_1, \ldots, A_m\}\) a red line. Clearly, \(|A_j| = r\) and \(|A_j \cap A_j| \leq 1\) for all \(j \neq j' \in [m]\).

For each \(j \in [m]\), let \(C_j = A_j \setminus A_0\). Then \(C_1, \ldots, C_m\) are pairwise disjoint and \(\bigcup_{j=1}^{m} C_j = [k] \setminus A_0\). So \(|\bigcup_{j=1}^{m} C_j| = |[k]\setminus A_0| = k - r + \lambda\). Moreover, we have

\[|C_j| = \begin{cases} r, & \text{if } j \in [\alpha]; \\ r - 1, & \text{if } j \in \{\alpha + 1, \ldots, m - 1\}; \\ \lambda, & \text{if } j = m. \end{cases}\]
Let \( \rho = \left\lceil \frac{k - r + \lambda}{r} \right\rceil \). Then \( k - r + \lambda \) can be represented as the sum of \( \rho \) positive integers (not necessarily different) \( r_1, \ldots, r_\rho \) such that \( r_i \leq r, \forall i \in [\rho] \). Since \( m - 1 \geq r \), using the Gale-Ryser Theorem, we can construct an \( m \times \rho \) binary matrix \( M \) such that for each \( i \in [\rho] \) and each \( j \in [m] \), the sum of the \( i \)th row is \( r_i \) and the sum of the \( j \)th column is \( |C_j| \). Let \( \mathcal{L} = \{C_1, \ldots, C_m\} \). By Lemma \[22\] there exists a collection \( \{B_1, \ldots, B_\rho\} \) of subsets of \( \bigcup_{j=1}^m C_j = [k] \setminus A_0 \) such that

- \( B_1, \ldots, B_\rho \) are pairwise disjoint and \( \bigcup_{i=1}^\rho B_i = \bigcup_{j=1}^m C_j = [k] \setminus A_0 \);
- \( |B_i| = r_i \) for all \( i \in [\rho] \);
- \( |B_i \cap C_j| \leq 1 \) for all \( i \in [\rho] \) and \( j \in [m] \).

Now, for each \( i \in [\rho] \), let \( A_{m+i} = B_i \). Note that \( k = (m - 1)r + \lambda \) and \( \rho = \left\lceil \frac{k - r + \lambda}{r} \right\rceil \). Then \( m + \rho = m + \left\lceil \frac{k - r + \lambda}{r} \right\rceil = \left\lceil \frac{2k}{r} \right\rceil = \eta \). Thus, we obtain a collection \( \mathcal{A} = \{A_1, \ldots, A_\eta\} \) of \( \eta \) subsets of \( [k] \). For convenience, we call each subset in \( \{A_{m+1}, \ldots, A_\eta\} \) a blue line.

By the construction, we have \( |A_i| \leq r \) for each \( i \in [\eta] \). So condition (i) of Lemma \[27\] is satisfied.

Again by the construction, we have the following observations: 1) Each \( i \in A_0 \) belongs to exactly two red lines and each \( i \in [k] \setminus A_0 \) belongs to one red line and one blue line; 2) Any two different red lines has at most one point (element) in common; 3) Any two different blue lines have no point (element) in common; 4) A red line and a blue line have at most one point (element) in common.

Observation 1) implies that each \( i \in [k] \) belongs to exactly two subsets in \( \mathcal{A} \). So condition (iii) of Lemma \[27\] is satisfied. Moreover, observations 2)–4) imply that any two different lines have at most one point (element) in common. So condition (ii) of Lemma \[27\] is satisfied.

Thus, we can always construct a collection of \( \eta = \left\lceil \frac{2k}{r} \right\rceil \) subsets of \( [k] \) that satisfies conditions (i)–(iii) of Lemma \[27\]

**APPENDIX C**

**PROOF OF LEMMA \[35\]**

We will construct a set \( \mathcal{R} = \{RL_1, \ldots, RL_m\} \) of red lines and a set \( \mathcal{B} = \{BL_1, \ldots, BL_\ell\} \) of blue lines and prove that \( \mathcal{R} \cup \mathcal{B} \) is a mesh of \([n]\).

Since \( m = \left\lceil \frac{2k}{r} \right\rceil \) and by assumption of Lemma \[35\] \( r|k \), then \( k = mr \) and \( k + m = (r + 1)m \). We can assign the elements of \([k + m]\) in an \((r + 1) \times m\) array \( D = (a_{i,j})_{i \in [r+1], j \in [m]} \) as in Fig. \[15\] such that \( \mathcal{K} = \{a_{i,j} ; i \in [r], j \in [m]\} \) and \( a_{r+1,j} = k + j, \forall j \in [m] \). For each \( j \in [m] \), we let \( RL_j = \{a_{i,j} ; i \in [r+1]\} \). In Fig. \[15\] each subset in \( \{RL_1, \ldots, RL_m\} \) is represented by a red solid line.

Fig 15. Construction of red lines: Each red line is a column of the array.

Since \( k = mr \), then \( \ell = \left\lceil \frac{2k+m}{r} \right\rceil - m = \left\lceil \frac{k+m}{r} \right\rceil \). Hence, \( k + m \) can be represented as the sum of \( \ell \) positive integers \( r_1, \ldots, r_\ell \) such that \( r_i \leq r \) for each \( i \in [\ell] \). Let \( \mathcal{L} = \{RL_1, \ldots, RL_m\} \) and \( \delta = r + 1 \). Note that by assumption of Lemma \[35\] \( m \geq r \). So we have \( r_i \leq r \leq m \) for each \( i \in [\ell] \). By Corollary \[24\] there exists a collection \( \{B_1, \ldots, B_\ell\} \) of subsets of \( \bigcup_{j=1}^m RL_j \) that satisfies the following properties:
• \( B_1, \ldots, B_\ell \) are pairwise disjoint and \( \bigcup_{i=1}^{\ell} B_i = \bigcup_{j=1}^{m} RL_j = [k + m] \);
• \(|B_i| = r_i\) for all \( i \in [\ell] \);
• \(|B_i \cap RL_j| \leq 1\) for all \( i \in [\ell] \) and \( j \in [m] \).

For each \( i \in [\ell] \), let \( BL_i = B_i \cup \{k + m \} \) and let \( \mathcal{B} = \{BL_1, \ldots, BL_\ell\} \).

By the construction, it is easy to check that conditions (i), (ii), (iv) of Definition 32 are satisfied.

By the construction, we also have the following observations: 1) \( \mathcal{R} \) is a partition of \([k + m]\); 2) \( \mathcal{B} \) is a partition of \([n]\); 3) \(|BL_i \cap RL_j| \leq 1\) for all \( i \in [\ell] \) and \( j \in [m] \).

By the above observations, we can easily check that conditions (iii), (v) of Definition 32 are satisfied. So \( \mathcal{R} \cup \mathcal{B} \) is a mesh of \([n]\).

**APPENDIX D**

**PROOF OF LEMMA 36**

We will construct a set \( \mathcal{R} = \{RL_1, \ldots, RL_m\} \) of red lines and a set \( \mathcal{B} = \{BL_1, \ldots, BL_\ell\} \) of blue lines and prove that \( \mathcal{R} \cup \mathcal{B} \) is a mesh of \([n]\).

Since \( m = \left\lceil \frac{k}{\lambda} \right\rceil \) and \( \lambda = r \mod k > 0 \), then

\[
k = (m - 1)r + \lambda. \tag{33}
\]

Hence, \( k + m = (m - 1)r + \lambda + m = (m - 1)(r + 1) + (\lambda + 1) \). We can assign the elements of \([k + m]\) in an \((r + 1) \times m\) array \( D = (a_{i,j})_{i \in [r+1], j \in [m+1]} \) as in Fig. 16 such that \( [k + m] = \{a_{i,j}; i \in [r + 1], j \in [m - 1]\} \cup \{a_{i,m}; i \in [\lambda + 1]\} \) and \( a_{i,m+1} = 0 \) for \( i \in \{\lambda + 2, \ldots, r + 1\} \). Moreover, by proper permutation (if necessary), we can let \( a_{r+1,j} = k + j \) for each \( j \in [m - 1] \) and \( a_{\lambda+1,m} = k + m \). We can construct \( \mathcal{R} = \{RL_1, \ldots, RL_m\} \) and \( \mathcal{B} = \{BL_1, \ldots, BL_{m+\lambda}\} \) by the following three steps.

- Fig 16. Construction of red lines of \([n]\): The first \( m - 1 \) red lines are the first \( m - 1 \) columns of the array and the last red line is depicted by a dashed red line, where \( \alpha = m - 1 - (r - \lambda) \).

**Step 1:** Construct \( \mathcal{R} = \{RL_1, \ldots, RL_m\} \).

Denote

\[
\alpha = m - 1 - (r - \lambda) \tag{34}
\]

and for each \( i \in [r + 1] \), let

\[
A_i = \{a_{i,j}; j \in \{\alpha + 1, \cdots, m - 1\}\}.
\]

Then we have \(|A_i| = m - 1 - \alpha = r - \lambda, \forall i \in [r + 1]|.

For each \( j \in [m] \), let

\[
RL_j = \begin{cases} \{a_{i,j}; i \in [r + 1]\}, & \text{if } j \in [m - 1]; \\ \{a_{i,m}; i \in [\lambda + 1]\} \cup A_1, & \text{if } j = m + 1. \end{cases}
\]

In Fig. 16 each subset in \( \{RL_1, \ldots, RL_{m-1}\} \) is represented by a red solid line and \( RL_m \) is represented by a red dashed line. Clearly, \(|RL_i| = r + 1\) for all \( i \in [m - 1] \). Moreover, by the construction, \(|RL_m| = |A_1| + \lambda + 1 = (r - \lambda) + (\lambda + 1) = r + 1 \).

So we have \(|RL_i| = r + 1\) for all \( i \in [m] \).

**Step 2:** Partition \( \bigcup_{j=1}^{m} RL_j \).

By assumption of this lemma, \( m \geq 2r - \lambda + 1 \), which implies that \( m - 1 - (r - \lambda) \geq r \). So by (34), we have

\[
\alpha = m - 1 - (r - \lambda) \geq r.
\]
Let
\[ \beta = \alpha(r + 1) - (\lambda + 1)(r - 1) - r\lambda \] (35)
and
\[ h = \ell - (\lambda + 1) - r. \] (36)

By assumption of this lemma, \( \ell \geq \lambda + 1 + r. \) So we have \( h \geq 0. \) Moreover, note that
\[
\left\lfloor \frac{\beta}{r} \right\rfloor = \left\lfloor \frac{\alpha(r + 1) - r\lambda - (\lambda + 1)(r - 1)}{r} \right\rfloor
= \left\lfloor \frac{(m - 1 - r + \lambda)(r + 1) - r\lambda - (\lambda + 1)(r - 1)}{r} \right\rfloor
= \left\lfloor \frac{2(m - 1)r + \lambda + m}{r} - m - (\lambda + 1) - r \right\rfloor
= \left\lfloor \frac{2k + m}{r} \right\rfloor - m - (\lambda + 1) - r
= \ell - (\lambda + 1) - r
= h.
\]

So \( \beta \) can be represented as the sum of \( h \) positive integers, say \( r_1, \cdots, r_h, \) such that \( r_i \leq r, \forall i \in [h]. \) Moreover, we let
\[
r_i = \begin{cases} 
  r - 1, & \text{if } i \in \{h + 1, \cdots, h + \lambda + 1\}; \\
  \lambda, & \text{if } i \in \{h + \lambda + 2, \cdots, \ell\}.
\end{cases}
\]

Then by (35) and (36), we have
\[
\sum_{i=1}^{\ell} r_i = \sum_{i=1}^{h} r_i + \sum_{i=h+1}^{h+\lambda+1} r_i + \sum_{i=h+\lambda+2}^{\ell} r_i
= \beta + (\lambda + 1)(r - 1) + (\ell - h - \lambda - 1)\lambda
= \beta + (\lambda + 1)(r - 1) + r\lambda
= \alpha(r + 1)
= \alpha \sum_{i=1}^{\ell} RL_i.
\]

Let \( \mathcal{L} = \{RL_1, \cdots, RL_\alpha\} \) and \( \delta = r + 1. \) Note that \( r_i \leq r \leq \alpha = |\mathcal{L}|, \forall i \in [\ell]. \) Then by Corollary 24, there exists a collection \( \{B_1, \cdots, B_\ell\} \) of subsets of \( \bigcup_{i=1}^{h} RL_i \) that satisfies the following three properties:
- \( B_1, \cdots, B_\ell \) are pairwise disjoint and \( \bigcup_{i=1}^{\ell} B_i = \bigcup_{i=1}^{\alpha} RL_i; \)
- \( |B_i| = r_i \) for all \( i \in [\ell]; \)
- \( |B_i \cap RL_j| \leq 1 \) for all \( i \in [\ell] \) and \( j \in [\alpha]. \)

**Step 3:** For each \( i \in [\ell], \) extend \( B_i \) to \( BL_i. \)

For each \( i \in [h], \) let
\[
BL_i = B_i \cup \{k + m + i\};
\]
For each \( i \in \{h + 1, \cdots, h + \lambda + 1\}, \) let
\[
BL_i = B_i \cup \{a_{i-h,m+1}, k + m + i\};
\]
For each \( i \in \{h + \lambda + 2, \cdots, \ell\}, \) let
\[
BL_i = B_i \cup A_{i-h-\lambda} \cup \{k + m + i\}.
\]

Note that by (36), we have \( \ell - h - \lambda = r + 1. \) So for each \( i \in \{h + \lambda + 2, \cdots, \ell\}, \) we have \( i - h - \lambda \in \{2, \cdots, r + 1\}. \) Hence, \( BL_i \) is reasonably constructed and \( A_1 \cap BL_i = \emptyset. \)

By the construction, it is easy to see that conditions (i), (ii) of Definition 32 are satisfied. Moreover, we can see that each point in \( A_1 \) belongs to two red lines and each point in \( [k + m] \setminus A_1 \) belongs to a red line and a blue line. So condition (iii) of Definition 32 is satisfied.

By the construction, we also have the following observations: 1) \( |RL_m \cap RL_i| = 0 \) for \( i \in [\alpha]; \) 2) \( |RL_m \cap RL_j| = 1 \) for \( j \in \{\alpha + 1, \cdots, m - 1\}; \) 3) If \( i, j \in [m - 1] \) and \( i \neq j, \) then \( RL_i \) and \( RL_j \) have no point in common; 4) A red line and a
blue line have at most one point in common; 5) Two different blue lines have no point in common; 6) If a blue line intersects with $RL_i$, then it does not intersect with $RL_i$ for all $i \in \{\alpha+1, \cdots, m-1\}$.

Note that observations 1)–3) imply that any two different red lines have at most one point in common. Hence observations 1)–5) imply that condition (iv) of Definition 3.2 is satisfied. Now suppose that two lines, say $L_1$ and $L_2$, intersect with $RL_i$ for some $i \in [m]$. We have the following three cases:

Case 1: $i \in [\alpha]$. Then by observations 1) and 3), $L_1$ and $L_2$ are two different blue lines. So by observation 5), $L_1$ and $L_2$ have no point in common.

Case 2: $i \in \{\alpha+1, \cdots, m-1\}$. Then by observations 2) and 3), we have the following two subcases.

Case 2.1: $L_1$ is $RL_i$ and $L_2$ is a blue line. By observation 6), $L_1$ and $L_2$ have no point in common.

Case 2.2: $L_1$ and $L_2$ are two different blue lines. Then by observation 5), $L_1$ and $L_2$ have no point in common.

Case 3: $i = m$. Then by observations 1) and 2), we have the following three subcases.

Case 3.1: $L_1$ is $RL_i$ for some $i \in \{\alpha+1, \cdots, m-1\}$ and $L_2$ is a blue line. By observation 6), $L_1$ and $L_2$ have no point in common.

Case 3.2: $L_1 = RL_i$ and $L_2 = RL_j$ for some $i, j \in [m-1]$ and $i \neq j$. By observation 3), $L_1$ and $L_2$ have no point in common.

Case 3.3: $L_1$ and $L_2$ are two different blue lines. Then by observation 5), $L_1$ and $L_2$ have no point in common.

By above discussion, we proved that condition (v) of Definition 3.2 is satisfied.

So $R \cup B$ is a mesh of $[n]$.

REFERENCES

[1] A. G. Dimakis, B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539-4551, Sep. 2010.

[2] D. S. Papailiopoulos and A. G. Dimakis, “Locally repairable codes,” in Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on, pp. 2771-2775, IEEE, 2012.

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

[4] F. Oggier and A. Datta, “Self-repairing homomorphic codes for distributed storage systems,” in INFOCOM, 2011 Proceedings IEEE, pp. 1215-1223, IEEE, 2011.

[5] D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, “Simple regenerating codes: Network coding for cloud storage,” in INFOCOM, 2012 Proceedings IEEE, pp. 2801-2805, IEEE, 2012.

[6] L. Fuentes-Juarez, H. D. L. Hollmann, and F. Oggier, “Locally repairable codes with multiple repair alternatives,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Istanbul, Turkey, Jul. 2013, pp. 892-896.

[7] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, “Optimal linear codes with a local-error-correction property,” in Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on, pp. 2776-2780, IEEE, 2012.

[8] A. Wang and Z. Zhang, “Repair locality with multiple erasure tolerance,” IEEE Trans. Inf. Theory, vol. 60, no. 11, pp. 6979-6987, Nov. 2014.

[9] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, “Locality and Availability in Distributed Storage,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, HI, USA, June 2014, pp. 681-685.

[10] I. Tamo, A. Barg, “Bounds on locally repairable codes with multiple recovering sets,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, HI, USA, June 2014, pp. 691-695.

[11] I. Tamo, A. Barg, “A family of optimal locally recoverable codes,” IEEE Trans. Inf. Theory, vol. 60, no. 80, pp. 4661-4676, Aug. 2014.

[12] A. Wang, Z. Zhang, and M. Liu, “Achieving Arbitrary Locality and Availability in Binary Codes,” in arXiv preprint arXiv: 1501.04264, 2015.

[13] A. S. Rawat, A. Mazumdar, and S. Vishwanath, “Cooperative Local Repair in Distributed Storage,” [http://arxiv.org/abs/1409.3900](http://arxiv.org/abs/1409.3900), 2015.

[14] N. Prakash, V. Lalitha, and P. V. Kumar, “Codes with locality for two erasures,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, HI, USA, June 2014, pp. 1962-1966.

[15] W. Song, S. H. Dau, C. Yuen, and T. J. Li, “Optimal locally repairable linear codes,” IEEE J. Sel. Areas Commun., vol. 32, no. 5, pp. 1019-1036, May 2014.

[16] H. D. Hollmann, “Storage codes C coding rate and repair locality,” in 2013 International Conference on Computing, Networking and Communications (ICNC13 - Invited), San Diego, USA, Jan. 2013, pp. 830-834.

[17] H. D. Hollmann and W. Poh, “Characterizations and construction methods for linear functional-repair storage codes,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Istanbul, Turkey, Jul. 2013, pp. 336-340.

[18] H. D. Hollmann, “On the minimum storage overhead of distributed storage codes with a given repair locality,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, HI, USA, June 2014, pp. 1401-1405.

[19] C. Fragouli and E. Soljanin, “Information flow decomposition for network coding,” IEEE Trans. Inf. Theory, vol. 52, no. 3, pp. 829-848, Mar. 2006.

[20] W. Song, K. Cai, R. Feng and C. Yuen, “The Complexity of Network Coding With Two Unit-Rate Multicast Sessions,” IEEE Trans. Inf. Theory, vol. 59, no. 9, pp. 5692-5707, Sept. 2013.

[21] K. Manfred, “A simple proof of the Gale-Ryser Theorem,” The American Mathematical Monthly, vol. 103, no. 4, pp. 335-337, 1996.