

On Sequential Locally Repairable Codes

Wentu Song, Kai Cai, Chau Yuen, Cai Kui and Guangyue Han

Abstract—We consider the locally repairable codes (LRC), aiming at sequentially recovering multiple erasures; in particular, we propose and study the so-called (n, k, r, t) -SLRC (Sequential Locally Repairable Codes) as an $[n, k]$ linear code where any $t' (\leq t)$ erasures can be sequentially recovered, each by $r (2 \leq r < k)$ other code symbols. Here, a sequential recovering means that the erased symbols are recovered one by one, and an already recovered symbol can be used to recover the remaining erased symbols. This important recovering method, in contrast with the extensively studied parallel recovering, is currently far from being thoroughly understood; more specifically, there are to date no codes constructed for arbitrary $t \geq 3$ erasures and bounds to evaluate the performance of such codes.

We first derive a tight upper bound on the code rate of the (n, k, r, t) -SLRC for $t = 3$ and $r \geq 2$. We then propose two constructions of binary (n, k, r, t) -SLRCs for general $r, t \geq 2$ (Existing constructions only deal with $t \leq 7$ erasures). The first construction generalizes the method of direct product construction. The second construction is based on the resolvable configurations and yields SLRCs for any $r \geq 2$ and odd $t \geq 3$. For both constructions, the rates are optimal for $t \in \{2, 3\}$ and are higher than most of the existing LRC families for arbitrary $t \geq 4$.

Index Terms—Distributed storage, locally repairable codes, parallel recovery, sequential recovery.

I. INTRODUCTION

To address the inefficiency of straightforward data replication, various coding techniques are introduced to distributed storage systems (DSSs) [1]–[3], among which the locally repairable codes, also known as locally recoverable codes (LRC) [4]–[7], attracted much attention recently. Experiments of HDFS-Xorbas — a famous family of LRCs — on Amazon EC2 and a cluster in Facebook show that Xorbas enables approximately a $2\times$ reduction in disk I/O and repair network traffic compared to the Reed-Solomon code [8]. Roughly speaking, a linear LRC with locality r is an $[n, k]$ linear code such that the value of each coordinate (code symbol) can be computed from the values of at most r other coordinates.

In a DSS system where a LRC \mathcal{C} is used, the information stored in each storage node corresponds to one coordinate of \mathcal{C} . Hence, each *single* node failure (erasure) can be recovered by a set of at most r other nodes. However, handling two or more node failures is also important because multiple simultaneous node failures are the norm rather than the exception in today's large-scale distributed storage systems [9], [12]. This problem, which has become a central focus for the

LRC society, have recently been investigated by many authors (e.g. [10]–[26]). Basically, when multiple erasures occur, the recovering performance can heavily depend on the recovering strategy in use, say, recovering the erasures simultaneously or one by one. The two strategies were first distinguished as *parallel approach* and *sequential approach* in [22]. In the parallel approach, each failed node (erasure) is recovered by resorting to a set of living nodes and hence the erasures can be recovered simultaneously, whereas in the sequential approach, the erasures are recovered one by one and the already fixed erasure nodes can be used in the next round of recovering. Potentially, for the same LRC, the sequential approach can fix more erasures than the parallel approach, and hence the former is a better candidate than the latter. However, due to a variety of technique difficulties, this more important approach remains far from being thoroughly understood, and there is a severe lack of code constructions and bounds for evaluation of the code performance. In contrast with the extensively studied parallel approach [10]–[21], existing work on the sequential approach to date only deals with $t \leq 7$ erasures. For example, the case of $t = 2$ are considered in [22], where the authors derived upper bounds on the code rate as well as minimum distance and also constructed a family of distance-optimal codes based on Turán graphs; more specifically, for the code rate, they proved that:

$$\frac{k}{n} \leq \frac{r}{r+2}. \quad (1)$$

In [23], a preliminary version of this work, we considered the case of $t = 3$ and gave both constructions and code rate bounds for $t \in \{2, 3\}$ (to be more precise, the constructed codes therein are meant for functional recovering). Of great relevance to the present work are the results recently obtained in [25] and [26], where the authors derived a lower bound on code length n of *binary code* for $t = 3$, an upper bound on code rate of *binary code* for $t = 4$, and constructed a couple of optimal or high rate constructions: rate-optimal codes for $t \in \{2, 3, 4\}$, and high rate codes for $r = 2$ and $t \in \{5, 6, 7\}$. Here, we note that, by using orthogonal Latin squares, the authors in [25] gave an interesting construction of sequential locally recoverable codes for *any odd* $t \geq 3$ with rate $k/n = 1 / (1 + \frac{t-1}{r} + \frac{1}{r^2})$. Obviously, the SLRCs with high code rate that can deal with any t erasures are highly desired in both theory and practice.

A. Our Contribution

In practice, high rate LRCs are desired due to their low storage overhead. In this work, we are interested in the high rate LRCs for sequentially recovering any $t \geq 3$ erasures; more specifically, we propose the (n, k, r, t) -SLRC (Sequential Locally Repairable Code) as an $[n, k]$ linear code in which any

W. Song, C. Yuen and Cai Kui are with Singapore University of Technology and Design, Singapore (e-mails: {wentu_song, yuenchau, cai_kui}@sutd.edu.sg).

Kai Cai and Guangyue Han are with the Department of Mathematics, University of Hong Kong. (e-mails: {kcai, ghan}@hku.hk).

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubpermissions@ieee.org.

t' ($t' \leq t$) erased code symbols can be sequentially recovered, each by at most r ($2 \leq r < k$) other symbols.

Our first contribution is an upper bound on the code rate for an (n, k, r, t) -SLRC with $t = 3$ and any $k > r \geq 2$. The bound is derived by using the following graph-theoretical method: we associate each (n, k, r, t) -SLRC with a set of directed acyclic graphs, called repair graphs, and then obtain the bound by examining the structural properties of the so-called *minimal repair graph* (Here we note that the sprits of this method can be traced back to [29], [30]). For the general case $t \geq 5$, an achievable and explicit upper bound of the rate of (n, k, r, t) -SLRC seems very challenging (see the discussion and conjecture in Section VII).

Our second contribution is two newly constructed families of binary (n, k, r, t) -SLRCs. The first family, which includes the product of m copies of the binary $[r+1, r]$ single-parity code [14] as a special case, is constructed for any positive integers $r (\geq 2)$ and t , and has rate

$$\frac{k}{n} = \frac{1}{\sum_{s=0}^t \frac{1}{r^{|\text{supp}_m(s)|}}},$$

where m is any given positive integer satisfying $t \leq 2^m - 1$ and $\text{supp}_m(s)$ is the support of the m -digit binary representation¹ of s . The second family is constructed for any $r \geq 2$ and any odd integer $t \geq 3$ and is based on the so-called *resolvable configurations*. This family has code rate

$$\frac{k}{n} = \left(1 + \frac{t-1}{r} + \left\lceil \frac{1}{r^2} \right\rceil \right)^{-1},$$

which is the same with the Latin square-based code constructed in [25]. We note that for $t \in \{2, 3\}$, the rates of these two families of codes are optimal.

A basic and important fact revealed by our study is: the sequential approach can have much better performance than the parallel approach, e.g., for the direct product of m copies of the binary $[r+1, r]$ single-parity code, it can recover m erasures with locality r by the parallel approach [14], but $2^m - 1$ erasures with the same locality by the sequential approach.

B. Related Work

Except those mentioned previously, most existing work focus on $[n, k]$ linear LRCs with the parallel approach. In [11], the authors proposed and constructed the $(r, t+1)_a$ code, for which each code symbol i is contained in a punctured code (local code) with length $\leq r+t$ and minimum distance $\geq t+1$. Clearly, for such codes, any t erased code symbols can be recovered in parallel by at most tr other code symbols, among which, each erased symbol can be recovered by at most r symbols. The code rate of this family satisfies [19]

$$\frac{k}{n} \leq \frac{r}{r+t}. \quad (2)$$

Another family is the codes with locality r and availability t [12], [13], for which, each code symbol has t disjoint

¹The m -digit binary representation of any positive integer $s \leq 2^m - 1$ is the binary vector $(\lambda_m, \lambda_{m-1}, \dots, \lambda_1) \in \mathbb{Z}_2^m$ such that $s = \sum_{j=1}^m \lambda_j 2^{j-1}$.

recovering sets of size at most r . An upper bound on the code rate of such codes is proved in [14]:

$$\frac{k}{n} \leq \frac{1}{\prod_{j=1}^t (1 + \frac{1}{jr})}. \quad (3)$$

Unfortunately, for $t \geq 3$, the tightness of bound (3) is not known and most of existing constructions have rate $\leq \frac{r}{r+t}$ (e.g., see [16], [20], [21]). Constructions with rate $> \frac{r}{r+t}$ are proposed only for some very special values, e.g., $(n, k, r, t) = (2^{r+1}-1, 2^r-1, r, r+1)$ [21]. The third family of parallel recovery LRCs is proposed in [10], in which, for any set $E \subseteq [n]$ of erasures of size at most t and any $i \in E$, the i th code symbol has a recovering set of size at most r contained in $[n] \setminus E$. The fourth family, called codes with cooperative local repair, is proposed in [17] and defined by a stronger condition: each subset of t code symbols can be *cooperatively* recovered from at most r other code symbols. For this family, an upper bound of the code rate with exactly the same form as (2) is derived [17]. Codes with cooperative local repair with rate at least $\frac{r-t}{r+t}$ are constructed in [18] based on the regular bipartite graphs with girth at least $t+1$. However, the construction in [18] has block length $n = O(r^{ct})$ for some constant c whereas our construction has a smaller block length $n = O(r^{\log(t)})$. By far, constructing LRCs with high code rate (e.g., $\frac{k}{n} > \frac{r}{r+t}$) is still an open problem, both for parallel recovery and for sequential recovery.

C. Organization

The rest of this paper is organized as follows. In Section II, we define the (n, k, r, t) -SLRC and then present some basic and useful facts. In Section III, we first investigate the (minimal) repair graphs of the SLRC and then prove the upper bound on the code rate of (n, k, r, t) -SLRC for $t \in \{2, 3\}$. Before constructing the first family of SLRCs in Section V, we study an example in Section IV. Then, the second family of SLRCs is constructed in Section VI. Finally, the paper is concluded in Section VII.

D. Notations

For any positive integer n , $[n] := \{1, 2, \dots, n\}$. For any set A , $|A|$ denotes the size (the number of elements) of A . If $B \subseteq A$ and $|B| = t$, then B is called a t -subset of A . For any real number x , $\lceil x \rceil$ denotes the smallest integer greater than or equal to x . If \mathcal{C} is an $[n, k]$ linear code and $A \subseteq [n]$, then $\mathcal{C}|_A$ denotes the punctured code by puncturing coordinates in $\bar{A} := [n] \setminus A$. For any codeword $x = (x_1, x_2, \dots, x_n) \in \mathcal{C}$, $\text{supp}(x) := \{i \in [n]; x_i \neq 0\}$ is the support of x .

II. PRELIMINARIES

A. Sequential Locally repairable code (SLRC)

Let \mathcal{C} be an $[n, k]$ linear code over the finite field \mathbb{F} and $i \in [n]$. A subset $R \subseteq [n] \setminus \{i\}$ is called a *recovering set* of i if there exists an $a_j \in \mathbb{F} \setminus \{0\}$ for each $j \in R$ such that $x_i = \sum_{j \in R} a_j x_j$ for all $x = (x_1, x_2, \dots, x_n) \in \mathcal{C}$. Equivalently, there exists a codeword y in the dual code \mathcal{C}^\perp such that $\text{supp}(y) = R \cup \{i\}$.

Definition 1 (Sequential Locally Repairable Code): For any $E \subseteq [n]$, \mathcal{C} is said to be (E, r) -recoverable if E can be sequentially indexed, i.e., $E = \{i_1, i_2, \dots, i_{|E|}\}$, such that each $i_\ell \in E$ has a recovering set $R_\ell \subseteq \overline{E} \cup \{i_1, \dots, i_{\ell-1}\}$ of size $|R_\ell| \leq r$, where $\overline{E} := [n] \setminus E$; \mathcal{C} is called an (n, k, r, t) -sequential locally repairable code (SLRC) (or simply (r, t) -SLRC) if \mathcal{C} is (E, r) -recoverable for each $E \subseteq [n]$ of size $|E| \leq t$, where r is called the locality of \mathcal{C} .

Particularly, if for each $E \subseteq [n]$ of size $|E| \leq t$ and each $i \in E$, i has a recovering set $R \subseteq \overline{E}$ of size $|R| \leq r$, then \mathcal{C} is called an (n, k, r, t) -parallel locally repairable code (PLRC). This special case is first considered in [10].

By the definition, we can have $r \leq k$ for any (n, k, r, t) -SLRC. Throughout this paper, we assume that a recovering set R has size $2 \leq |R| \leq r < k$. The following equivalent form of Definition 1 will be frequently used in our paper.

Lemma 2: \mathcal{C} is an (n, k, r, t) -SLRC if and only if for any nonempty $E \subseteq [n]$ of size $|E| \leq t$, there exists an $i \in E$ such that i has a recovering set $R \subseteq [n] \setminus E$.

Proof: Let \mathcal{C} be an (n, k, r, t) -SLRC and $\emptyset \neq E \subseteq [n]$ of size $|E| \leq t$. Then by Definition 1, E can be sequentially indexed as $E = \{i_1, i_2, \dots, i_{|E|}\}$ such that i_1 has a recovering set $R_1 \subseteq [n] \setminus E$.

Conversely, for any $E \subseteq [n]$ of size $|E| \leq t$, by the assumption, one can find an $i_1 \in E$ such that i_1 has a recovering set $R_1 \subseteq [n] \setminus E$. Furthermore, since $|E \setminus \{i_1\}| < |E| \leq t$, there exists an $i_2 \in E \setminus \{i_1\}$ such that i_2 has a recovering set $R_2 \subseteq [n] \setminus (E \setminus \{i_1\}) = \overline{E} \cup \{i_1\}$. Similarly, we can find an $i_3 \in E \setminus \{i_1, i_2\}$ such that i_3 has a recovering set $R_3 \subseteq \overline{E} \cup \{i_1, i_2\}$, and so on. Then E can be sequentially indexed as $E = \{i_1, i_2, \dots, i_{|E|}\}$ such that each $i_\ell \in E$ has a recovering set $R_\ell \subseteq \overline{E} \cup \{i_1, \dots, i_{\ell-1}\}$. So by Definition 1, \mathcal{C} is an (n, k, r, t) -SLRC. ■

The following lemma gives a sufficient condition for a linear code being an (r, t) -SLRC, which, as a byproduct, reflects the difference between the sequential recovery and the parallel recovery.

Lemma 3: Suppose $[n] = A \cup B$ and $A \cap B = \emptyset$. Suppose $t_1, t_2 \geq 0$ and \mathcal{C} is an $[n, k]$ linear code such that

- (1) For any nonempty $E \subseteq A$ of size $|E| \leq t_1$, there exists an $i \in E$ such that i has a recovering set $R \subseteq A \setminus E$;
- (2) For any nonempty $E \subseteq A$ of size $|E| \leq t_1 + t_2 + 1$, there exists an $i \in E$ such that i has a recovering set $R \subseteq [n] \setminus E$;
- (3) For any nonempty $E \subseteq B$ of size $|E| \leq t_2$, there exists an $i \in E$ such that i has a recovering set $R \subseteq B \setminus E$;
- (4) For any nonempty $E \subseteq B$ of size $|E| \leq t_1 + t_2 + 1$, there exists an $i \in E$ such that i has a recovering set $R \subseteq [n] \setminus E$.

Then \mathcal{C} is an (r, t) -SLRC with $t = t_1 + t_2 + 1$.

Proof: We prove, by Lemma 2, that for any nonempty $E \subseteq [n]$ of size $|E| \leq t_1 + t_2 + 1$, there exists an $i \in E$ such that i has a recovering set $R \subseteq [n] \setminus E$. Obviously, this is true when $E \subseteq A$ or $E \subseteq B$ (by condition (2) or (4)). Consequently, we can assume $E \cap A \neq \emptyset$ and $E \cap B \neq \emptyset$ in the remainder of the proof.

We need to consider the following two cases.

Case 1: $0 < |E \cap A| \leq t_1$. By condition (1), there exists an $i \in E$ such that i has a recovering set $R \subseteq A \setminus E \subseteq [n] \setminus E$.

Case 2: $|E \cap A| > t_1$. Since $|E| \leq t_1 + t_2 + 1$ and $A \cap B = \emptyset$, we have $0 < |E \cap B| \leq t_2$. By condition (3), there exists an $i \in E$ such that i has a recovering set $R \subseteq B \setminus E \subseteq [n] \setminus E$.

The proof is completed by combining the above cases. ■

B. Repair Graph and Minimal Repair Graph

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed acyclic graph, where \mathcal{V} is the vertex set and \mathcal{E} is the (directed) edge set. A directed edge e from vertex u to v is denoted by an ordered pair $e = (u, v)$, where u is called the *tail* of e and v the *head* of e . Moreover, u is called 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, v is called an *inner vertex*. Denote by $S(G)$ the set of all sources of G . For any $E \subseteq \mathcal{V}$, let

$$\text{Out}(E) = \bigcup_{v \in E} \text{Out}(v) \setminus E. \quad (4)$$

By (4), we have $E \cap \text{Out}(E) = \emptyset$. For any $v \in \mathcal{V}$, denote

$$\text{Out}^2(v) = \bigcup_{u \in \text{Out}(v)} \text{Out}(u) \setminus \text{Out}(v) \quad (5)$$

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 .

For example, consider the graph depicted in Fig. 1, where vertices are indexed by $\{1, 2, \dots, 16\}$. Then $\text{Out}(3) = \{9, 10\}$, $\text{Out}(4) = \{10, 11\}$ and $\text{Out}^2(3) = \{13, 15, 16\}$. Let $E = \{3, 4\}$. Then $\text{Out}(E) = \{9, 10, 11\}$.

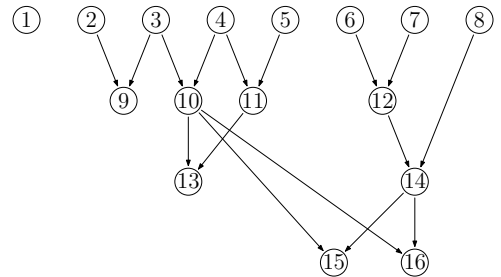


Fig. 1. An example repair graph with $n = 16$, $r = 2$ and $|S(G)| = 8$.

Definition 4 (Repair Graph): Let \mathcal{C} be an (n, k, r, t) -SLRC and $G = (\mathcal{V}, \mathcal{E})$ be a directed acyclic graph such that $\mathcal{V} = [n]$. G is called a *repair graph* of \mathcal{C} if for all inner vertex $i \in \mathcal{V}$, $\text{In}(i)$ is a recovering set of i .

Obviously, an (n, k, r, t) -SLRC may have many repair graphs. If \mathcal{C} is an (n, k, r, t) -SLRC, we usually use $\{G_\lambda; \lambda \in \Lambda\}$ to denote the set of all repair graphs of \mathcal{C} , where Λ is some proper index set. It should be noted that the repair graph defined here has subtle differences with the recovering graph defined in [14]: it is acyclic and an (n, k, r, t) -SLRC may have many repair graphs such that for each $i \in [n]$, at most one recovery set of i is considered in each repair graph. The key ingredient of our technique is the so-called *minimal repair graph* as defined below.

Let \mathcal{C} be an (n, k, r, t) -SLRC and $\{G_\lambda; \lambda \in \Lambda\}$ be the set of all repair graphs of \mathcal{C} . Recall that for each $\lambda \in \Lambda$, $S(G_\lambda)$ is the set of all sources of G_λ . Denote

$$\delta^* \triangleq \min\{|S(G_\lambda)|; \lambda \in \Lambda\}. \quad (6)$$

Definition 5 (Minimal Repair Graph): A repair graph G_{λ_0} , $\lambda_0 \in \Lambda$, is called a *minimal repair graph* of \mathcal{C} if $|S(G_{\lambda_0})| = \delta^*$.

Remark 6: It is easy to see that any (n, k, r, t) -SLRC has at least one minimal repair graph by noticing that the set $\{|S(G_\lambda)|; \lambda \in \Lambda\} \subseteq [n]$ is finite.

III. AN UPPER BOUND ON THE CODE RATE

Before stating the main result of this section, we need to first examine the properties of the minimal repair graphs of (n, k, r, t) -SLRC.

A. Properties of the Minimal Repair Graph

In this subsection, we always assume that \mathcal{C} is an (n, k, r, t) -SLRC and $G_{\lambda_0} = (\mathcal{V}, \mathcal{E})$ is a minimal repair graph of \mathcal{C} . The following two results are of fundamental importance.

Lemma 7:

$$(n - \delta^*)r \geq |\mathcal{E}|. \quad (7)$$

Proof: By definition, G_{λ_0} has $n - \delta^*$ inner vertices, each of which has at most r in-neighbors, and hence the result immediately follows. ■

Lemma 8:

$$k \leq \delta^*. \quad (8)$$

Proof: According to Definition 4, for each $j \in [n]$, the j th code symbol of \mathcal{C} is a linear combination of the code symbols in $\text{In}(j)$. In other words, the code symbols of $\text{In}(j)$ spans the code symbols of $\{j\} \cup \text{In}(j)$. Moreover, since G_{λ_0} is acyclic, then a method of induction will yield that the code symbols of $S(G_{\lambda_0})$ spans \mathcal{C} , which proves $k \leq |S(G_{\lambda_0})| = \delta^*$. ■

The following is a key lemma for the investigation of the structure of G_{λ_0} .

Lemma 9: For any $E \subseteq [n]$ of size $|E| \leq t$,

$$|\text{Out}(E)| \geq |E \cap S(G_{\lambda_0})|. \quad (9)$$

Proof: Suppose, on the contrary, that there exists an $E = \{i_1, i_2, \dots, i_{t'}\} \subseteq [n]$ such that $|E| = t' \leq t$ and $|\text{Out}(E)| < |E \cap S(G_{\lambda_0})|$. By the definition of (n, k, r, t) -SLRC, there exists $R_\ell \subseteq \overline{E} \cup \{i_1, \dots, i_{\ell-1}\}$, a recovering set of i_ℓ , for each $\ell \in [t']$.

We now construct a graph G_{λ_1} from G_{λ_0} by deleting and adding edges as follows: First, for each $i \in E \cup \text{Out}(E)$ and $j \in \text{In}(i)$, delete (j, i) if it is an edge of G_{λ_0} , and denote the resulted graph by G_{λ_1} ; second, for each $i_\ell \in E$ and each $j \in R_\ell$, add a directed edge from j to i_ℓ , yielding a graph G_{λ_1} . Clearly, G_{λ_1} is acyclic because G_{λ_0} is acyclic. Moreover, since $R_\ell \subseteq \overline{E} \cup \{i_1, \dots, i_{\ell-1}\}$ for each $\ell \in [t']$, G_{λ_1} is also acyclic.

We are now ready to conclude that G_{λ_1} is a repair graph of \mathcal{C} and $|S(G_{\lambda_1})| < |S(G_{\lambda_0})|$, which however contradicts to the minimality of G_{λ_0} .

Indeed, by our way of construction, $S(G_{\lambda_1}) = (S(G_{\lambda_0}) \setminus E) \cup \text{Out}(E)$. Then for each inner node i of G_{λ_1} , we have the following two cases:

Case 1: $i \in E$. Then $i = i_\ell$ for some $\ell \in [t']$ and by the construction of G_{λ_1} , $\text{In}(i) = R_\ell$ is a recovering set of i .

Case 2: i is an inner vertex of G_{λ_0} and $i \notin \text{Out}(E)$. Then for G_{λ_0} , $\text{In}(i) \subseteq \overline{E} = [n] \setminus E$ is a recovering set of i .

So $\text{In}(i)$ is always a recovering set of i . Hence, G_{λ_1} is a repair graph of \mathcal{C} .

On the other hand, by definition, $S(G_{\lambda_0}) \cap \text{Out}(E) = \emptyset$ and $E \cap \text{Out}(E) = \emptyset$. So if we assume that $|\text{Out}(E)| < |E \cap S(G_{\lambda_0})|$, then we have

$$\begin{aligned} |S(G_{\lambda_1})| &= |(S(G_{\lambda_0}) \setminus E) \cup \text{Out}(E)| \\ &= |(S(G_{\lambda_0}) \setminus E)| + |\text{Out}(E)| \\ &= |S(G_{\lambda_0})| - |E \cap S(G_{\lambda_0})| + |\text{Out}(E)| \\ &< |S(G_{\lambda_0})|, \end{aligned} \quad (10)$$

which completes the proof. ■

The following example illustrates the construction of G_{λ_1} in the proof of Lemma 9.

Example 10: Consider the graph in Fig. 1, which will be denoted by G_{λ_0} in this example. Suppose it is a repair graph of a $(r = 2, t = 3)$ -SLRC. One verifies that $\{2, 3\}$ is a recovering set of 9, $\{3, 4\}$ is a recovering set of 10, and etc.

Let $E = \{2, 3, 9\}$ and assume the recovering sets of 2, 3 and 9 are $\{1, 10\}$, $\{12, 13\}$ and $\{11, 14\}$, respectively. Then we can construct a graph G_{λ_1} as follows. Bearing in mind the fact that $\text{Out}(E) = \{10\}$, we delete edges $(2, 9)$, $(3, 9)$, $(3, 10)$ and $(4, 10)$ in the first step; and in the second step, we add edges $(1, 2)$, $(10, 2)$, $(12, 3)$, $(13, 3)$, $(11, 9)$ and $(14, 9)$. The resulted graph G_{λ_1} is shown in Fig. 2. It can be verified that $|S(G_{\lambda_1})| = |(S(G_{\lambda_0}) \setminus E) \cup \text{Out}(E)| = |\{1, 4, 5, 6, 7, 8, 10\}| = 7 < 8 = |S(G_{\lambda_0})|$, which means that the graph in Fig. 1 is not a minimal repair graph.

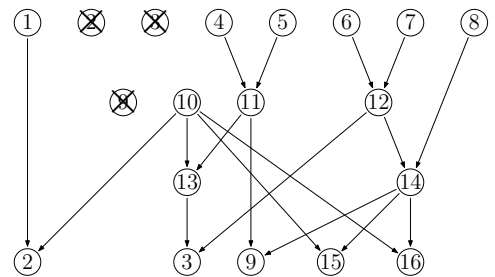


Fig 2. Construction of G_{λ_1} from the graph in Fig. 1.

The following two corollaries give some explicit structural properties of the minimal repair graphs of an (n, k, r, t) -SLRC.

Corollary 11: If $t \geq 3$, for any $v \in S(G_{\lambda_0})$, the following statements hold:

- 1) $|\text{Out}(v)| \geq 1$.
- 2) If $\text{Out}(v) = \{v'\}$, then $\text{Out}^2(v) = \text{Out}(v') \neq \emptyset$.
- 3) If $\text{Out}(v) = \{v_1\}$ and $\text{Out}(v_1) = \{v_2\}$, then $\text{Out}(v_2) \neq \emptyset$.
- 4) If $\text{Out}(v) = \{v_1\}$ and $\text{Out}(v_1) = \{v_2\}$, then $|\text{Out}(u)| \geq 2$ for any source $u \in \text{In}(v_2)$.
- 5) If v, w are two distinct sources and $|\text{Out}(v)| = |\text{Out}(w)| = 1$, then $\text{Out}(v) \neq \text{Out}(w)$.

Proof: We will prove all the statements by way of contradiction.

1) To reach a contradiction, we suppose $|\text{Out}(v)| = 0$. Let $E = \{v\}$. Then, $|\text{Out}(E)| = |\text{Out}(v)| = 0 < 1 = |\{v\}| = |E \cap S(G_{\lambda_0})|$, which contradicts Lemma 9.

2) Since G_{λ_0} is acyclic and $\text{Out}(v) = \{v'\}$, then from (5), $\text{Out}^2(v) = \text{Out}(v')$. If $\text{Out}(v') = \emptyset$, then by letting $E = \{v, v'\}$, we have $|\text{Out}(E)| = |\emptyset| = 0 < 1 = |\{v\}| = |E \cap S(G_{\lambda_0})|$, which contradicts Lemma 9.

3) If $\text{Out}(v_2) = \emptyset$, then by letting $E = \{v, v_1, v_2\}$, we have $|\text{Out}(E)| = |\emptyset| = 0 < 1 = |\{v\}| = |E \cap S(G_{\lambda_0})|$, which contradicts Lemma 9.

4) It follows from the assumptions that $u \neq v$. Suppose, by way of contradiction, that $|\text{Out}(u)| = 1$. Then $\text{Out}(u) = \{v_2\}$ since $u \in \text{In}(v_2)$. Let $E = \{v, v_1, u\}$. We have $|\text{Out}(E)| = |\{v_2\}| = 1 < 2 = |\{v, u\}| = |E \cap S(G_{\lambda_0})|$, which contradicts Lemma 9.

5) Suppose $\text{Out}(v) = \text{Out}(w) = \{v_1\}$. Let $E = \{v, w\}$. Then we have $|\text{Out}(E)| = |\{v_1\}| = 1 < 2 = |\{v, w\}| = |E \cap S(G_{\lambda_0})|$, which contradicts Lemma 9. ■

Below, we give an example and a counterexample of minimal repair graphs by checking the statements in Corollary 11.

Example 12: Consider the repair graph G_{λ_0} in Fig. 3 whose vertex set is $\mathcal{V} = \{1, 2, \dots, 15\}$. One verifies that $|\text{Out}(E)| \geq |E \cap S(G_{\lambda_0})|$ for each $E \subseteq [n]$ of size $|E| \leq t$. For G_{λ_0} , we verify the statements of Corollary 11 as follows:

- 1) For every $v \in S(G_{\lambda_0})$, $|\text{Out}(v)| \geq 1$.
- 2) For $v = 5$ and $v' = 10$, we have $\text{Out}(v) = \{v'\}$ and $\text{Out}^2(v) = \text{Out}(v') = \{12, 13\}$.
- 3) For $v = 1$, $v_1 = 8$ and $v_2 = 11$, we have $\text{Out}(v) = \{v_1\}$, $\text{Out}(v_1) = \{v_2\}$ and $\text{Out}(v_2) = \{14\}$.
- 4) For $v = 1$, $v_1 = 8$ and $v_2 = 11$, we have $u = 6 \in \text{In}(v_2)$ is a source and $|\text{Out}(u)| = |\{10, 11\}| \geq 2$.
- 5) For $v = 1$ and $w = 5$, we have $|\text{Out}(v)| = |\text{Out}(w)| = 1$ and $\text{Out}(v) = \{8\} \neq \text{Out}(w) = \{10\}$.

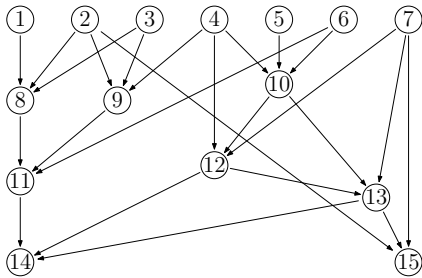


Fig. 3. An example repair graph $G_{\lambda_0} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, 15\}$.

Example 13: Each of the following five observations, which violates one of the corresponding statements of Corollary 11, is sufficient to establish that the graph in Fig. 1 is not a minimal repair graph.

- 1) For the source $v = 1$, we have $\text{Out}(1) = \emptyset$.
- 2) For the source $v = 2$, we have $\text{Out}(2) = \{9\}$ and $\text{Out}(9) = \emptyset$.
- 3) For the source $v = 5$, we have $\text{Out}(5) = \{11\}$, $\text{Out}(11) = \{13\}$ and $\text{Out}(13) = \emptyset$.

4) For the source $v = 6$, we have $\text{Out}(6) = \{12\}$, $\text{Out}(12) = \{14\}$ and there is another source $u = 8 \in \text{In}(14)$ such that $\text{Out}(8) = \{14\}$.

5) For the two sources $v = 6$ and $w = 7$, we have $\text{Out}(6) = \text{Out}(7) = \{12\}$.

Remark 14: In Corollary 11, statement 1) holds for all $t \geq 1$, since the contradiction in the proof is derived from a subset E of size 1. And statements 2) and 5) hold for all $t \geq 2$ since the contradictions are derived from subsets of size 2.

Corollary 15: Suppose $t \geq 3$ and let $v \in S(G_{\lambda_0})$ be such that $\text{Out}(v) = \{v_1, v_2\}$. Then the following statements are true:

- 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 $|\text{Out}(w)| \geq 2$ for any source $w \in \text{In}(v_2)$.

Proof: All the statement above can be proved by assuming the opposite and choosing a proper E as in the proof of Lemma 11, which will lead to a contradiction.

1) Suppose, by way of contradiction, that $\text{Out}(v_1) = \text{Out}(v_2) = \emptyset$. We let $E = \{v, v_1, v_2\}$ and have $|\text{Out}(E)| = |\emptyset| = 0 < 1 = |\{v\}| = |E \cap S(G_{\lambda_0})|$, which contradicts to Lemma 9.

2) Suppose, on the contrary, that $\text{Out}(v_2) = \emptyset$. A similar argument as above will yield a contradiction by letting $E = \{u, v, v_2\}$.

3) Suppose that there exists a source w such that $\text{Out}(w) = \{v_2\}$. A contradiction can be obtained by letting $E = \{u, v, w\}$. ■

We give below an example and a counterexample of minimal repair graphs by checking the statements in Corollary 15.

Example 16: Again consider the repair graph G_{λ_0} in Fig. 3. Let $v = 3$, $v_1 = 8$ and $v_2 = 9$. Then $v \in S(G_{\lambda_0})$ and $\text{Out}(v) = \{v_1, v_2\}$. We now verify the statements in Corollary 15 as follows:

- 1) $\text{Out}(v_1) = \text{Out}(v_2) = \{11\} \neq \emptyset$.
- 2) For $u = 1$, we have $\{v_1\} = \text{Out}(u)$ and $\text{Out}(v_2) = \{11\} \neq \emptyset$.
- 3) For $w = 4$, we can see that $w \in \text{In}(v_2)$ is a source and $|\text{Out}(w)| = |\{9, 10, 12\}| \geq 2$.

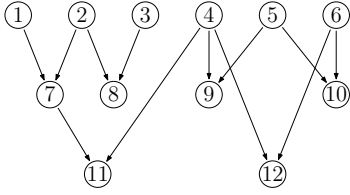
Example 17: Let G be a repair graph as shown in Fig. 4. Then each of the following three observations, which violates one of the corresponding statements of Corollary 15, is sufficient to establish that G is not a minimal repair graph.

- 1) There exists a source $v = 5$ such that $\text{Out}(v) = \{9, 10\}$ and $\text{Out}(9) = \text{Out}(10) = \emptyset$.
- 2) There exists a source $v = 2$ such that $\text{Out}(2) = \{7, 8\}$, and a source $u = 1$ such that $\text{Out}(1) = \{7\}$ and $\text{Out}(8) = \emptyset$.
- 3) There exist three sources $v = 2$, $u = 1$ and $w = 3$ such that $\text{Out}(2) = \{7, 8\}$, $\text{Out}(1) = \{7\}$ and $\text{Out}(3) = \{8\}$.

B. Upper Bound on the Code Rate for $(n, k, r, 3)$ -SLRC

In this subsection, we assume \mathcal{C} is an $(n, k, r, 3)$ -SLRC and $G_{\lambda_0} = (\mathcal{V}, \mathcal{E})$ is a minimal repair graph of \mathcal{C} . Recall that $S(G_{\lambda_0})$ is the set of all sources of G_{λ_0} . We divide $S(G_{\lambda_0})$ into four subsets as follows.

$$A = \{v \in S(G_{\lambda_0}) \mid |\text{Out}(v)| \geq 3\}, \quad (11)$$


 Fig. 4. An example repair graph with $n = 12$ and $r = 2$.

$$B = \{v \in \mathcal{S}(G_{\lambda_0}); |\text{Out}(v)| = 2\}, \quad (12)$$

$$C_1 = \{v \in \mathcal{S}(G_{\lambda_0}); |\text{Out}(v)| = 1 \text{ and } |\text{Out}^2(v)| = 1\} \quad (13)$$

and

$$C_2 = \{v \in \mathcal{S}(G_{\lambda_0}); |\text{Out}(v)| = 1 \text{ and } |\text{Out}^2(v)| \geq 2\}. \quad (14)$$

Clearly, A, B, C_1 and C_2 are mutually disjoint. Moreover, by 1), 2) of Corollary 11, $\mathcal{S}(G_{\lambda_0}) = A \cup B \cup C_1 \cup C_2$. Hence,

$$\delta^* = |\mathcal{S}(G_{\lambda_0})| = |A| + |B| + |C_1| + |C_2|. \quad (15)$$

We define three types of edges of G_{λ_0} , called red edge, green edge and blue edge respectively, as follows.

Firstly, an edge is called a *red edge* if its tail is a source. For each $v \in \mathcal{S}(G_{\lambda_0})$, let $\mathcal{E}_{\text{red}}(v)$ be the set of all red edges whose tail is v and denote

$$\mathcal{E}_{\text{red}} = \bigcup_{v \in \mathcal{S}(G_{\lambda_0})} \mathcal{E}_{\text{red}}(v).$$

Then \mathcal{E}_{red} is the set of all red edges. Clearly, $|\mathcal{E}_{\text{red}}(v)| = |\text{Out}(v)|$ and $\mathcal{E}_{\text{red}}(w) \cap \mathcal{E}_{\text{red}}(v) = \emptyset$ for any source $w \neq v$. So by (11)–(14), we have

$$|\mathcal{E}_{\text{red}}| = \sum_{v \in \mathcal{S}(G_{\lambda_0})} |\text{Out}(v)| \geq 3|A| + 2|B| + |C_1| + |C_2|. \quad (16)$$

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 $\mathcal{E}_{\text{green}}(v)$ be the set of all green edges whose tail is the unique out-neighbor of v . Clearly, $|\mathcal{E}_{\text{green}}(v)| = |\text{Out}^2(v)|$. Let

$$\mathcal{E}_{\text{green}} = \bigcup_{v \in C_1 \cup C_2} \mathcal{E}_{\text{green}}(v)$$

be the set of all green edges. Note that if $v \neq w \in C_1 \cup C_2$, then by 5) of Corollary 11, $v' \neq w'$, where v' (resp. w') is the unique out-neighbor of v (resp. w). So $\mathcal{E}_{\text{green}}(v) \cap \mathcal{E}_{\text{green}}(w) = \emptyset$. Hence, by (13) and (14),

$$|\mathcal{E}_{\text{green}}| = \sum_{v \in C_1 \cup C_2} |\text{Out}^2(v)| \geq |C_1| + 2|C_2|. \quad (17)$$

Thirdly, suppose $e \in \mathcal{E}$ is not a green edge and $v \in B \cup C_1$. e is called a *blue edge belonging to v* if one of the following two 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 $\mathcal{E}_{\text{blue}}(v)$ be the set of all blue edges belonging to v and let

$$\mathcal{E}_{\text{blue}} = \bigcup_{v \in B \cup C_1} \mathcal{E}_{\text{blue}}(v)$$

be the set of all blue edges. Then we have the following lemma.

Lemma 18: The number of blue edges is lower bounded by

$$|\mathcal{E}_{\text{blue}}| \geq \frac{|B| + |C_1|}{r}. \quad (18)$$

Proof: It is sufficient to prove : i) For each $v \in B \cup C_1$, there exists at least one blue edge belonging to v ; and ii) Each blue edge belongs to at most r different $v \in B \cup C_1$. To prove these two statements, we will use the definitions of red edge, green edge and blue edge repeatedly.

We first prove i) by considering the cases of $v \in B$ and $v \in C_1$.

Let $v \in B$, and we look for a blue edge belonging to v . In this case, by (12), we can assume $\text{Out}(v) = \{v_1, v_2\}$ (see Fig. 5(a)). Then, by 1) of Corollary 15, $\text{Out}(v_1) \neq \emptyset$ or $\text{Out}(v_2) \neq \emptyset$. Without loss of generality, assume $\text{Out}(v_1) \neq \emptyset$ and $v_3 \in \text{Out}(v_1)$. Consider (v_1, v_3) . If it is not a green edge, then by definition, it is a blue edge belonging to v . So we assume that (v_1, v_3) is a green edge. Then by definition, $\{v_1\} = \text{Out}(u)$ for some $u \in C_1 \cup C_2$. By 2) of Corollary 15, $\text{Out}(v_2) \neq \emptyset$ and we can let $v_4 \in \text{Out}(v_2)$, as illustrated in Fig. 5(a). Consider (v_2, v_4) . By 3) of Corollary 15, $|\text{Out}(w)| \geq 2$ for any source $w \in \text{In}(v_2)$, which implies (v_2, v_4) is not a green edge. So (v_2, v_4) is a blue edge belonging to v . Hence, for each $v \in B$, we can always find a blue edge belonging to v .

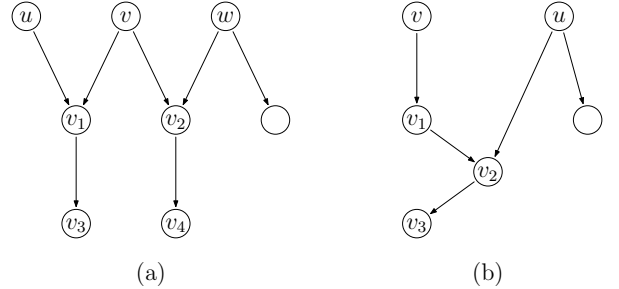


Fig. 5. Two local graphs.

Now, let $v \in C_1$ and we look for a blue edge belonging to v . By (13), we can assume $\text{Out}(v) = \{v_1\}$ and $\text{Out}^2(v) = \{v_2\}$ (see Fig. 5(b)). By 3) of Corollary 11, we have $\text{Out}(v_2) \neq \emptyset$. Let $v_3 \in \text{Out}(v_2)$. Note that by 4) of Corollary 11, $|\text{Out}(u)| \geq 2$ for any source $u \in \text{In}(v_2)$ (see Fig. 5(b) as illustration), which implies (v_2, v_3) is not a green edge. So (v_2, v_3) is a blue edge belonging to v . Hence, for each $v \in C_1$, we can always find a blue edge belonging to v .

By the above discussion, statement i) holds.

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 . To prove statement ii), we prove that there is an *injection*, say, φ , from S to $\text{In}(u')$. Then ii) follows from the fact that $\text{In}(u')$ has size at most r . The injection of $\varphi(v)$ can be constructed as follows: If $v \in B$, simply let $\varphi(v) = v$. If $v \in C_1$, let $\varphi(v) = v'$, where $\{v'\} = \text{Out}(v)$. It is easy to see that $\varphi(v)$ is an injection (to see this, use 5) of Corollary 11), which completes the proof of statement ii). ■

Example 19: Consider the repair graph in Fig. 3. We have $A = \{2, 4, 7\}$, $B = \{3, 6\}$, $C_1 = \{1\}$ and $C_2 = \{5\}$, and the

edges with tails from 1 to 7 are red edges, as illustrated in Fig. 6.

Moreover, one can check that $\mathcal{E}_{\text{green}}(1) = \{(8, 11)\}$ and $\mathcal{E}_{\text{green}}(5) = \{(10, 12), (10, 13)\}$. As for blue edges, since $1 \in C_1$ and $11 \in \text{Out}^2(1)$, then $(11, 14) \in \mathcal{E}_{\text{blue}}(1)$; Since $11 \in \text{Out}(6)$ and $6 \in B$, then $(11, 14) \in \mathcal{E}_{\text{blue}}(6)$; Since $3 \in B$ and $9 \in \text{Out}(3)$, then $(9, 11) \in \mathcal{E}_{\text{blue}}(3)$. One can check that $\mathcal{E}_{\text{blue}}(1) = \mathcal{E}_{\text{blue}}(6) = \{(11, 14)\}$ and $\mathcal{E}_{\text{blue}}(3) = \{(9, 11)\}$. The green edges and blue edges are also specified in Fig. 6.

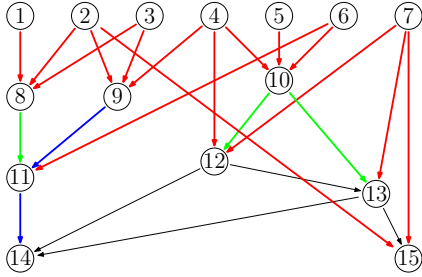


Fig. 6. Illustration of red edge, green edge and blue edge of minimal repair graph.

We are now ready to state the main theorem of this section.

Theorem 20: For any $(n, k, r, 3)$ -SLRC, we have ²

$$\frac{k}{n} \leq \left(\frac{r}{r+1} \right)^2. \quad (19)$$

Proof: By definition, we can easily see that $\mathcal{E}_{\text{red}}, \mathcal{E}_{\text{green}}$ and $\mathcal{E}_{\text{blue}}$ are mutually disjoint. Then by (15)-(18), we have

$$\begin{aligned} |\mathcal{E}| &\geq |\mathcal{E}_{\text{red}}| + |\mathcal{E}_{\text{green}}| + |\mathcal{E}_{\text{blue}}| \\ &\geq (3|A| + 2|B| + |C_1| + |C_2|) \\ &\quad + (|C_1| + 2|C_2|) + \frac{|B| + |C_1|}{r} \\ &= 2(|A| + |B| + |C_1| + |C_2|) \\ &\quad + (|A| + |C_2| + \frac{|B| + |C_1|}{r}) \\ &= 2\delta^* + \frac{r|A| + r|C_2| + |B| + |C_1|}{r} \\ &\geq 2\delta^* + \frac{|A| + |C_2| + |B| + |C_1|}{r} \\ &= 2\delta^* + \frac{\delta^*}{r}, \end{aligned}$$

which, together with Lemma 7, implies that

$$(n - \delta^*)r \geq |\mathcal{E}| \geq 2\delta^* + \frac{\delta^*}{r}.$$

So

$$(n - \delta^*)r \geq 2\delta^* + \frac{\delta^*}{r}.$$

Solving n from the above equation, we have

$$n \geq \delta^* + \frac{2\delta^* + \frac{\delta^*}{r}}{r}. \quad (20)$$

²In [23], a preliminary version of this paper, the bound (19) was presented equivalently in terms of the code length as $n \geq k + \left\lceil \frac{2k + \frac{k}{r}}{r} \right\rceil$.

By Lemma 8, $\delta^* \geq k$. So (20) implies that

$$\begin{aligned} n &\geq k + \frac{2k + \frac{k}{r}}{r} \\ &= k \left(1 + \frac{2}{r} + \frac{1}{r^2} \right) \\ &= k \left(\frac{r+1}{r} \right)^2. \end{aligned}$$

Hence,

$$\frac{k}{n} \leq \left(\frac{r}{r+1} \right)^2,$$

which proves the theorem. \blacksquare

We will later construct two families of $(n, k, r, 3)$ -SLRCs achieving (19) and hence show the tightness of this bound.

C. Code Rate for $(n, k, r, 2)$ -SLRC

In this subsection, we give a new proof of the bound (1) for the $(n, k, r, 2)$ -SLRC using the similar techniques as in Subsection B. Assume that \mathcal{C} is an $(n, k, r, 2)$ -SLRC and $G_{\lambda_0} = (\mathcal{V}, \mathcal{E})$ is a minimal repair graph of \mathcal{C} .

Proof of Bound (1): By Remark 14 and 1) of Corollary 11, each source of G_{λ_0} has at least one out-neighbor. Let A be the set of sources that has only one out-neighbor and let \mathcal{E}_{red} be the set of all edges e , called red edges, such that the tail of e is a source. Then the number of red edges is

$$|\mathcal{E}_{\text{red}}| \geq |A| + 2|S(G_{\lambda_0}) \setminus A| = 2\delta^* - |A|. \quad (21)$$

For each $v \in A$, let v' be the unique out-neighbor of v and $\mathcal{E}_{\text{green}}(v)$ be the set of all edges whose tail is v' . By Remark 14 and 2) of Corollary 11, $\text{Out}^2(v) = \text{Out}(v') \neq \emptyset$. So $|\mathcal{E}_{\text{green}}(v)| = |\text{Out}(v')| \geq 1$. Let $\mathcal{E}_{\text{green}}$ be the set of all green edges. 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 14 and 5) of Corollary 11, $v'_1 \neq v'_2$. So $\mathcal{E}_{\text{green}}(v_1) \cap \mathcal{E}_{\text{green}}(v_2) = \emptyset$. Hence,

$$|\mathcal{E}_{\text{green}}| = \left| \bigcup_{v \in A} \mathcal{E}_{\text{green}}(v) \right| = \sum_{v \in A} |\mathcal{E}_{\text{green}}(v)| \geq |A|. \quad (22)$$

Clearly, $\mathcal{E}_{\text{red}} \cap \mathcal{E}_{\text{green}} = \emptyset$. Then by (21) and (22),

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

On the other hand, by Lemma 7,

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

So $(n - \delta^*)r \geq 2\delta^*$, which implies $n \geq \delta^* + \frac{2\delta^*}{r} \geq k + \frac{2k}{r}$ (Lemma 8). Hence, $\frac{k}{n} \leq \frac{r}{r+2}$, which completes the proof. \blacksquare

IV. AN EXAMPLE OF SLRC

For illustrative purposes, we first give an example of binary (r, t) -SLRCs that will be constructed in the next section. Let $r = 2$, $m = 3$ and \mathcal{C} be the product code of m copies of the binary $[r+1, r]$ single parity check code. Then \mathcal{C} has length $n = (r+1)^3 = 27$ and dimension $k = r^3 = 8$. For notational convenience, we use $\mathbb{Z}_3^3 = \{(i_3, i_2, i_1); i_1, i_2, i_3 \in \mathbb{Z}_3\}$ instead

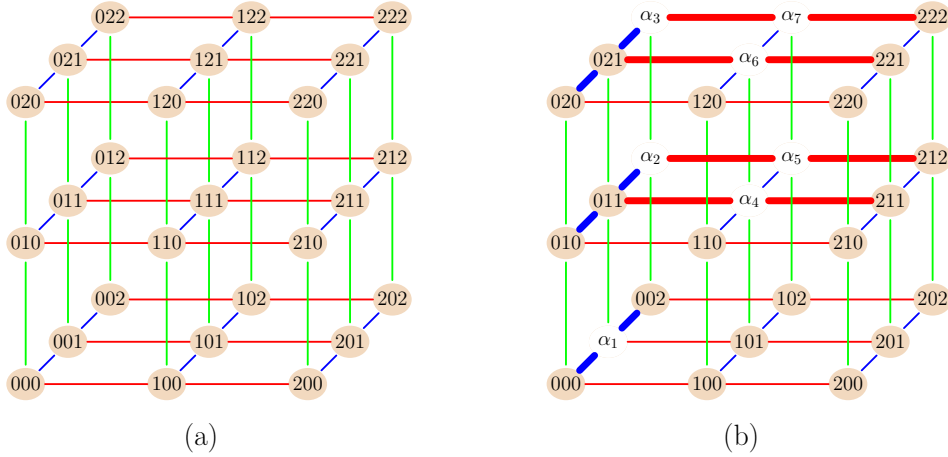


Fig 7. (a) The index set of \mathbb{Z}_3^3 , where (i_3, i_2, i_1) is simply written as $i_3i_2i_1$; (b) The recovering sets R_1, \dots, R_7 of $\alpha_1, \dots, \alpha_7$, where $R_1 = \{(000), (002)\}$, $R_2 = \{(010), (011)\}$, $R_3 = \{(020), (021)\}$, $R_4 = \{(011), (211)\}$, $R_5 = \{\alpha_2, (212)\}$, $R_6 = \{(021), (221)\}$ and $R_7 = \{\alpha_3, (222)\}$ drawn in heavy lines.

of $[n]$ to denote the index set of the coordinates of \mathcal{C} , and let $\mathbb{Z}_2^3 = \{(i_3, i_2, i_1); i_1, i_2, i_3 \in \mathbb{Z}_2\}$ denote the information set of \mathcal{C} . Here, $\mathbb{Z}_3 = \{0, 1, 2\}$ and $\mathbb{Z}_2 = \{0, 1\}$ are simply viewed as two sets (devoid of any algebraic connotation), and obviously, we have $\mathbb{Z}_2 \subseteq \mathbb{Z}_3$.

The index set \mathbb{Z}_3^3 is depicted in Fig. 7(a). By definition, each code symbol (coordinate) of \mathcal{C} can be recovered by all the other symbols on the same (red, green or blue) line. Hence, each code symbol of \mathcal{C} has $m = 3$ disjoint recovering sets (red, green and blue) of size $r = 2$, and \mathcal{C} can recover any 3 erasures by parallel recovery. However, we can prove (see details in the next section) that it can recover any $t = 2^m - 1 = 7$ erasures by sequential recovery. For example, consider an erasure of 7 code symbols, say, $E = \{\alpha_1, \dots, \alpha_7\}$, where $\alpha_1 = (001)$, $\alpha_2 = (012)$, $\alpha_3 = (022)$, $\alpha_4 = (111)$, $\alpha_5 = (112)$, $\alpha_6 = (121)$ and $\alpha_7 = (122)$, as illustrated in Fig. 7(b). We can select a sequence of recovering sets $R_1 = \{(000), (002)\}$, $R_2 = \{(010), (011)\}$, $R_3 = \{(020), (021)\}$, $R_4 = \{(011), (211)\}$, $R_5 = \{\alpha_2, (212)\}$, $R_6 = \{(021), (221)\}$ and $R_7 = \{\alpha_3, (222)\}$ (see Fig. 7(b)). It is easy to check that R_1, \dots, R_7 sequentially repair $\{\alpha_1, \dots, \alpha_7\}$.

In general, by properly puncturing \mathcal{C} , we can obtain (r, t) -SLRC for any $t \in \{1, 2, \dots, 6\}$. As an example, we construct an $(r, 5)$ -SLRC as follows. For each $j \in \mathbb{Z}_2 = \{0, 1\}$, let

$$A_j = \{(j, i_2, i_1); i_2, i_1 \in \mathbb{Z}_3\}$$

and let

$$A = A_0 \cup A_1,$$

$$B = \{(2, i_2, i_1); i_2 \in \mathbb{Z}_2 \text{ and } i_1 \in \mathbb{Z}_3\} \\ = \{(200), (201), (202), (210), (211), (212)\}.$$

Let $\Omega = A \cup B$, as depicted in Fig. 8. Then, the punctured code $\mathcal{C}|_\Omega$ is an $(n', k, r, 5)$ -SLRC, with $n' = |\Omega| = 24$.

In fact, the following statements hold.

- i) For any nonempty $E \subseteq A$ of size $|E| \leq t_1 = 3$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq A \setminus E$.

- ii) For any nonempty $E \subseteq A$ of size $|E| \leq t = 5$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq \Omega \setminus E$.
- iii) For any nonempty $E \subseteq B$ of size $|E| \leq t_2 = 1$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq B \setminus E$.
- iv) For any nonempty $E \subseteq B$ of size $|E| \leq t = 5$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq \Omega \setminus E$.

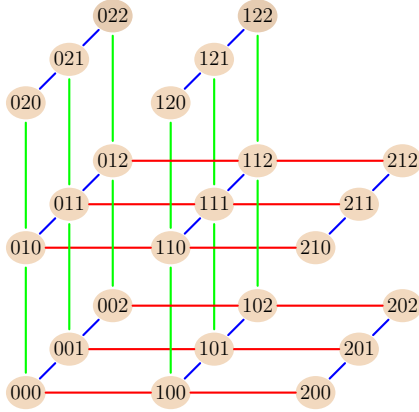
The above statements *i), iii), iv)* can be easily verified. For example, one can see that the punctured codes $\mathcal{C}|_{A_0}$ and $\mathcal{C}|_{A_1}$ are both $(r, 3)$ -SLRCs and $\mathcal{C}|_B$ is a $(r, 1)$ -SLRC, and hence *i)* and *iii)* hold. From Fig. 8, one verifies that each $(2, i_2, i_1) \in B$ has a recovery set (red line) $R = \{(0, i_2, i_1), (1, i_2, i_1)\} \subseteq A$, hence *iv)* holds. To prove *ii)*, we consider the following two cases:

- 1) $E \subseteq A_0$ or $E \subseteq A_1$. Without loss of generality, we assume $E \subseteq A_1$. If $E \subseteq \{(120), (121), (122)\}$, then each $(1, 2, i) \in E$ has a recovery set (green line) $R = \{(1, 0, i), (1, 1, i)\} \subseteq \Omega \setminus E$; otherwise, there exists a $(1, i_2, i_1) \in E \cap \{(100), (101), (102), (110), (111), (112)\}$ which has a recovery set (red line) $R = \{(0, i_2, i_1), (2, i_2, i_1)\} \subseteq \Omega \setminus E$.
- 2) $E \cap A_0 \neq \emptyset$ and $E \cap A_1 \neq \emptyset$. Since $|E| \leq 5$, we have $|E \cap A_0| \leq 3$ or $|E \cap A_1| \leq 3$. Note that both $\mathcal{C}|_{A_0}$ and $\mathcal{C}|_{A_1}$ are $(r, 3)$ -SLRCs, by Lemma 2, there exists an $\alpha \in E$ and $j \in \{0, 1\}$ such that α has a recovering set $R \subseteq A_j \setminus E \subseteq A \setminus E \subseteq \Omega \setminus E$.

Then, by Lemma 3, $\mathcal{C}|_\Omega$ is an $(n', k, r, 5)$ -SLRC. The generalization of this example as well as the formal proof will be given in the next section.

V. CONSTRUCTION OF (n, k, r, t) -SLRC

In this section, we construct a family of binary (n, k, r, t) -SLRCs for any positive integers $r (\geq 2)$ and t . It will be shown that the code rate of this family is greater than $\frac{r}{r+t}$, and in particular, for $t \in \{2, 3\}$, it achieves the bounds (1) and (19), respectively.


 Fig. 8. Graphical illustration of a subset of \mathbb{Z}_3^3 .

We first introduce some notations. For any integers $r \geq 2$ and $m \geq 1$, let $\mathbb{Z}_r = \{0, 1, \dots, r-1\}$ and $\mathbb{Z}_r^m = \{(i_m, i_{m-1}, \dots, i_1) : i_m, i_{m-1}, \dots, i_1 \in \mathbb{Z}_r\}$ (Here, as before, \mathbb{Z}_r is simply treated as a set). So $\mathbb{Z}_r \subseteq \mathbb{Z}_{r+1} = \{0, 1, \dots, r-1, r\}$. We will use α, β, γ , etc., to denote elements (points) of \mathbb{Z}_{r+1}^m . Note that for each $\alpha = (i_m, i_{m-1}, \dots, i_1) \in \mathbb{Z}_r$ and $\ell \in [m]$, i_ℓ is the ℓ th coordinate of α from the right.

For each $\alpha = (i_m, i_{m-1}, \dots, i_1) \in \mathbb{Z}_{r+1}^m$, we let

$$\mathbf{U}^{(m)}(\alpha) = \{\ell \in [m]; i_\ell = r\}, \quad (23)$$

and

$$\mathbf{T}^{(m)}(\alpha) = \{\ell \in [m]; i_\ell \in \mathbb{Z}_r\}. \quad (24)$$

Furthermore, we let

$$\mathcal{L}^{(m)}(\alpha) = \{(j_m, j_{m-1}, \dots, j_1) \in \mathbb{Z}_r^m; j_\ell = i_\ell, \forall \ell \in \mathbf{T}^{(m)}(\alpha)\}. \quad (25)$$

Clearly, $\mathbf{U}^{(m)}(\alpha) \cap \mathbf{T}^{(m)}(\alpha) = \emptyset$ and $\mathbf{U}^{(m)}(\alpha) \cup \mathbf{T}^{(m)}(\alpha) = [m]$. Moreover, for each $\alpha \in \mathbb{Z}_{r+1}^m \setminus \mathbb{Z}_r^m$, $\mathbf{U}^{(m)}(\alpha) \neq \emptyset$ and $\mathcal{L}^{(m)}(\alpha) \neq \emptyset$. In particular, if $\alpha = (r, r, \dots, r)$, then $\mathbf{U}^{(m)}(\alpha) = [m]$ and $\mathcal{L}^{(m)}(\alpha) = \mathbb{Z}_r^m$.

As an example, let $r = 2$, $m = 6$ and $\alpha = (1, 0, 2, 1, 1, 2) \in \mathbb{Z}_3^6$. Then $\mathbf{U}^{(6)}(\alpha) = \{4, 1\}$, $\mathbf{T}^{(6)}(\alpha) = \{6, 5, 3, 2\}$ and $\mathcal{L}^{(6)}(\alpha) = \{(1, 0, i_4, 1, 1, i_1); i_4, i_1 \in \mathbb{Z}_2\} = \{(1, 0, 0, 1, 1, 0), (1, 0, 0, 1, 1, 1), (1, 0, 1, 1, 1, 0), (1, 0, 1, 1, 1, 1)\}$.

Note that for any integer s such that $0 \leq s \leq 2^m - 1$, s has a unique m -digit binary representation, say, $(\lambda_m \lambda_{m-1} \dots \lambda_1)$. That is, $(\lambda_m, \lambda_{m-1}, \dots, \lambda_1) \in \{0, 1\}^m$ and $s = \sum_{\ell=1}^m \lambda_\ell 2^{\ell-1}$. Denote by $\text{supp}_m(s)$ the support of $(\lambda_m, \lambda_{m-1}, \dots, \lambda_1)$. Let

$$\Gamma_s^{(m)} = \{\alpha \in \mathbb{Z}_{r+1}^m; \mathbf{U}^{(m)}(\alpha) = \text{supp}_m(s)\} \quad (26)$$

and

$$\Omega_s^{(m)} = \bigcup_{\ell=0}^s \Gamma_\ell^{(m)}. \quad (27)$$

For example, suppose $r = 2$, $m = 6$ and $s = 22$. Then (010110) is the unique 6-digit binary representation of s and $\text{supp}_6(s) = \{5, 3, 2\}$. From (26), we have $\Gamma_{22}^{(6)} = \{(i_6, 2, i_4, 2, 2, i_1); i_6, i_4, i_1 \in \mathbb{Z}_2\}$.

Clearly, $\Gamma_0^{(m)}, \Gamma_1^{(m)}, \dots, \Gamma_{2^m-1}^{(m)}$ are mutually disjoint and $|\Gamma_s^{(m)}| = r^{m-|\text{supp}_m(s)|}$, for $s = 0, 1, \dots, 2^m - 1$. In particular, $|\Gamma_0^{(m)}| = r^m$ and $|\Gamma_{2^m-1}^{(m)}| = 1$. Moreover, by definition, we have $\Omega_0^{(m)} = \Gamma_0^{(m)} = \mathbb{Z}_r^m$ and $\Omega_{2^m-1}^{(m)} = \mathbb{Z}_{r+1}^m$.

For any integers $r \geq 2$ and $t \geq 1$, we can always pick an integer m such that $t \leq 2^m - 1$ and, using the above notations, define a matrix $H_t^{(m)} = (h_{\alpha, \beta})$ satisfying the following two properties:

- (1) The rows of $H_t^{(m)}$ are indexed by $\Omega_t^{(m)} \setminus \Omega_0^{(m)}$ and the columns of $H_t^{(m)}$ are indexed by $\Omega_t^{(m)}$;
- (2) For each $\alpha \in \Omega_t^{(m)} \setminus \Omega_0^{(m)}$ and $\beta \in \Omega_t^{(m)}$,

$$h_{\alpha, \beta} = \begin{cases} 1, & \text{if } \beta \in \mathcal{L}^{(m)}(\alpha) \cup \{\alpha\}; \\ 0, & \text{Otherwise.} \end{cases} \quad (28)$$

It should be noted that, $H_t^{(m)} = (h_{\alpha, \beta})$ is an $h \times n$ binary matrix, where $h = |\Omega_t^{(m)} \setminus \Omega_0^{(m)}|$, $n = |\Omega_t^{(m)}|$. The sub-matrix of $H_t^{(m)}$, formed by the columns indexed by $\Omega_t^{(m)} \setminus \Omega_0^{(m)}$, is a permutation matrix. Hence, $\text{rank}(H_t^{(m)}) = h$.

Theorem 21: Let $\mathcal{C}_t^{(m)}$ be the binary code that has a parity check matrix $H_t^{(m)}$. Then $\mathcal{C}_t^{(m)}$ is an (n, k, r, t) -SLRC with

$$n = r^m \sum_{s=0}^t \frac{1}{r^{|\text{supp}_m(s)|}} \quad (29)$$

and

$$k = r^m. \quad (30)$$

Hence, the code rate of $\mathcal{C}_t^{(m)}$ is

$$\frac{k}{n} = \frac{1}{\sum_{s=0}^t \frac{1}{r^{|\text{supp}_m(s)|}}}, \quad (31)$$

where the integers $r \geq 2$, $t \geq 1$ and m is any integer satisfying $t \leq 2^m - 1$.

Remark 22: We make the following detailed remarks on the construction above.

- 1) The example codes given in the last section are just $\mathcal{C}_t^{(m)}$ for $r = 2$, $m = 3$ and $t = 7, 5$ respectively. In general, for $t = 2^m - 1$, it is easy to check that $\Omega_{2^m-1}^{(m)} = \mathbb{Z}_{r+1}^m$ and $\mathcal{C}_{2^m-1}^{(m)}$ is the product of m copies of the $[r+1, r]$ binary code. If $t < 2^m - 1$, then $\mathcal{C}_t^{(m)}$ is the punctured code of $\mathcal{C}_{2^m-1}^{(m)}$ with respect to $\Omega_t^{(m)}$.
- 2) For $t \in \{2, 3\}$, we can set $m = 2$. It follows from (31) that the code rates of our construction are $\frac{r}{r+2}$ and $\left(\frac{r}{r+1}\right)^2$ respectively, which are optimal according to (1) and (19). For $t \geq 4$, by (31), the code rate of $\mathcal{C}_t^{(m)}$ is higher than $\frac{r}{r+t}$ for all $r \geq 2$.
- 3) It was shown in [14] that $\mathcal{C}_{2^m-1}^{(m)}$ has locality r and availability m , which implies that it can recover m erasures with locality r using the parallel approach. On the other hand, by Theorem 21, it can recover $t = 2^m - 1$ erasures with the same locality when using the sequential approach, which is a significant advantage of the product code for sequential recovery. In particular, the product of two copies of the $[r+1, r]$ binary code is not optimal (in

rate) among codes with locality r and availability $t = 2$ [16], but optimal among $(r, t = 3)$ -SLRCs.

In the rest of this section, we will prove Theorem 21. To prove that $\mathcal{C}_t^{(m)}$ is an (r, t) -SLRC, we will prove the following more general statement: if a binary linear code \mathcal{C} has a parity check matrix H which contains all rows of $H_t^{(m)}$ (not necessarily $H = H_t^{(m)}$), then it is an (r, t) -SLRC.

We first make the following two simple remarks about the construction.

Remark 23: Let \mathcal{C} be a binary linear code. If the code symbols of \mathcal{C} are indexed by $\Omega_t^{(m)}$, then, by the construction of $H_t^{(m)}$, \mathcal{C} has a parity check matrix which contains all rows of $H_t^{(m)}$ if and only if for each $\alpha \in \Omega_t^{(m)} \setminus \Omega_0^{(m)}$,

$$x_\alpha = \sum_{\beta \in \mathcal{L}^{(m)}(\alpha)} x_\beta. \quad (32)$$

If the code symbols of \mathcal{C} are indexed by S , where $S \neq \Omega_t^{(m)}$, then \mathcal{C} has a parity check matrix which contains all rows of $H_t^{(m)}$ if and only if there is a bijection $\psi : \Omega_t^{(m)} \rightarrow S$ such that for each $\alpha \in \Omega_t^{(m)} \setminus \Omega_0^{(m)}$,

$$x_{\psi(\alpha)} = \sum_{\beta \in \mathcal{L}^{(m)}(\alpha)} x_{\psi(\beta)}. \quad (33)$$

Remark 24: Since $1 \leq t \leq 2^m - 1$, we can find a $m_0 \in [m]$ such that $2^{m_0-1} - 1 < t \leq 2^{m_0} - 1$. Let $t_1 = 2^{m_0-1} - 1$ and $t_2 = t - t_1 - 1$. Then $0 \leq t_2 \leq t_1 \leq 2^{m-1} - 1$ and $\Omega_t^{(m)}$ can be partitioned into two disjoint nonempty subsets

$$A = \Omega_{t_1}^{(m)} = \bigcup_{s=0}^{t_1} \Gamma_s^{(m)}$$

and

$$B = \Omega_t^{(m)} \setminus A = \bigcup_{s=t_1+1}^t \Gamma_s^{(m)}.$$

Moreover, noticing that $\text{supp}_m(s) \subseteq \{1, 2, \dots, m_0 - 1\}$ for $0 \leq s \leq t_1$, then A can be partitioned into r mutually disjoint nonempty subsets, according to the values of the m_0 th coordinate (from the right) of its elements, as follows.

$$A_j = \{(i_m, i_{m-1}, \dots, i_1) \in A; i_{m_0} = j\}, \quad \forall j \in \mathbb{Z}_r.$$

In the following, unless specified otherwise, we always assume that \mathcal{C} is a binary linear code and has a parity check matrix which contains all rows of $H_t^{(m)}$. Without loss of generality, we assume that the code symbols of \mathcal{C} are indexed by $\Omega_t^{(m)}$. To prove Theorem 21, we need the following three lemmas.

Lemma 25: Suppose $m > 1$. Assuming the notations as in Remark 24, we have:

- 1) For each $j \in \mathbb{Z}_r$, the punctured code $\mathcal{C}|_{A_j}$ has a parity check matrix which contains all rows of $H_{t_1}^{(m-1)}$.
- 2) If $t_2 \geq 1$, the punctured code $\mathcal{C}|_B$ has a parity check matrix which contains all rows of $H_{t_2}^{(m-1)}$.

Proof: For each $j \in \mathbb{Z}_{r+1} = \{0, 1, \dots, r\}$, let

$$\psi_j : \mathbb{Z}_{r+1}^{m-1} \rightarrow \mathbb{Z}_{r+1}^m$$

be such that $\psi_j(\alpha) = (i_{m-1}, \dots, i_{m_0}, j, i_{m_0-1}, \dots, i_1)$ for each $\alpha \in (i_{m-1}, \dots, i_{m_0}, i_{m_0-1}, \dots, i_1) \in \mathbb{Z}_{r+1}^{m-1}$; that is to say, $\psi_j(\alpha)$ is obtained by inserting j as a coordinate between the $(m_0 - 1)$ th and m_0 th coordinate (from the right) of α .

1) For each $j \in \mathbb{Z}_r$, one checks that ψ_j induces a bijection between $\Omega_{t_1}^{(m-1)}$ and A_j such that for each $\alpha \in \Omega_{t_1}^{(m-1)} \setminus \Omega_0^{(m-1)}$,

$$\mathcal{L}^{(m)}(\psi_j(\alpha)) = \{\psi_j(\beta); \beta \in \mathcal{L}^{(m-1)}(\alpha)\}.$$

Since \mathcal{C} has a parity check matrix containing all rows of $H_t^{(m)}$, then by (32), we have

$$\begin{aligned} x_{\psi_j(\alpha)} &= \sum_{\beta' \in \mathcal{L}^{(m)}(\psi_j(\alpha))} x_{\beta'} \\ &= \sum_{\beta \in \mathcal{L}^{(m-1)}(\alpha)} x_{\psi_j(\beta)}. \end{aligned}$$

Hence, by Remark 23, $\mathcal{C}|_{A_j}$ has a parity check matrix which contains all rows of $H_{t_1}^{(m-1)}$.

2) Recall that $t_1 = 2^{m_0-1} - 1$. Then for each $s \in \{t_1 + 1, t_1 + 2, \dots, t\}$, we have

$$\text{supp}_m(s) = \text{supp}_{m-1}(s') \cup \{m_0\},$$

where $s' = s - 2^{m_0-1} \in \{0, 1, \dots, t_2\}$. So similar to 1), we can check that ψ_r induces a bijection between $\Omega_{t_2}^{(m-1)}$ and $B = \Omega_t^{(m)} \setminus \Omega_{t_1}^{(m)}$ such that for each $\alpha \in \Omega_{t_2}^{(m-1)} \setminus \Omega_0^{(m-1)}$,

$$x_{\psi_r(\alpha)} = \sum_{\beta \in \mathcal{L}^{(m-1)}(\alpha)} x_{\psi_r(\beta)}.$$

Hence, by Remark 23, $\mathcal{C}|_B$ has a parity check matrix which contains all rows of $H_{t_2}^{(m-1)}$. \blacksquare

For each $\alpha = (i_m, i_{m-1}, \dots, i_1) \in \mathbb{Z}_{r+1}^m$ and $\ell \in [m]$, let

$$L_\alpha^{(\ell)} = \{(i_m, \dots, i_{\ell+1}, i'_\ell, i_{\ell-1}, \dots, i_1); i'_\ell \in \mathbb{Z}_{r+1}\}. \quad (34)$$

That is, $L_\alpha^{(\ell)}$ consists of α as well as the points in \mathbb{Z}_{r+1}^m which differs from α only at the ℓ th coordinate (from the right).

Lemma 26: For each $\alpha \in \Omega_t^{(m)}$ and $\ell \in [m]$, if $L_\alpha^{(\ell)} \subseteq \Omega_t^{(m)}$, then $R = L_\alpha^{(\ell)} \setminus \{\alpha\}$ is a recovering set of α .

Proof: Let

$$\alpha = (i_m, \dots, i_{\ell+1}, i_\ell, i_{\ell-1}, \dots, i_1).$$

Then by (34),

$$L_\alpha^{(\ell)} = \{\alpha_0, \alpha_1, \dots, \alpha_r\},$$

where $\alpha_j = (i_m, \dots, i_{\ell+1}, j, i_{\ell-1}, \dots, i_1)$ for each $j \in \mathbb{Z}_{r+1}$ and $\alpha = \alpha_{i_\ell}$.

From (25), it is easy to see that

$$\mathcal{L}^{(m)}(\alpha_r) = \bigcup_{j=0}^{r-1} \mathcal{L}^{(m)}(\alpha_j) \quad (35)$$

and for distinct $j_1, j_2 \in \mathbb{Z}_r$,

$$\mathcal{L}^{(m)}(\alpha_{j_1}) \cap \mathcal{L}^{(m)}(\alpha_{j_2}) = \emptyset. \quad (36)$$

So combining (32), (35) and (36), we have

$$\begin{aligned} x_{\alpha_r} &= \sum_{\beta \in \mathcal{L}^{(m)}(\alpha_r)} x_\beta \\ &= \sum_{j=0}^{r-1} \left(\sum_{\beta \in \mathcal{L}^{(m)}(\alpha_j)} x_\beta \right) \\ &= \sum_{j=0}^{r-1} x_{\alpha_j} \end{aligned}$$

which is equivalent to (notice that \mathcal{C} is a binary code)

$$x_\alpha = \sum_{\beta \in L_\alpha^{(\ell)} \setminus \{\alpha\}} x_\beta.$$

Note that from (34), $L_\alpha^{(\ell)}$ has size $r+1$. So $R = L_\alpha^{(\ell)} \setminus \{\alpha\}$ has size r , and hence it is a recovering set of α . ■

Lemma 27: For any nonempty $E \subseteq \Omega_t^{(m)} \setminus \Omega_0^{(m)}$, there exists an $\alpha \in E$ which has a recovering set $R \subseteq \Omega_t^{(m)} \setminus E$.

Proof: Let s be the smallest number such that $E \cap \Gamma_s^{(m)} \neq \emptyset$. Since $E \subseteq \Omega_t^{(m)} \setminus \Omega_0^{(m)}$, we have $s \geq 1$ and $\text{supp}_m(s) \neq \emptyset$. Hence, we can always find an $\ell \in \text{supp}_m(s)$ and an $s' < s$ such that

$$\text{supp}_m(s) = \text{supp}_m(s') \cup \{\ell\}. \quad (37)$$

Pick $\alpha \in E \cap \Gamma_s^{(m)}$. Then by (26), $\mathbf{U}^{(m)}(\alpha) = \text{supp}_m(s)$. Furthermore, by (34) and (37), $\mathbf{U}^{(m)}(\beta) = \text{supp}_m(s')$ for each $\beta \in L_\alpha^{(\ell)} \setminus \{\alpha\}$. Then again by (26), we have

$$L_\alpha^{(\ell)} \setminus \{\alpha\} \subseteq \Gamma_{s'}^{(m)}. \quad (38)$$

Since $s' < s$ and s is the smallest number such that $E \cap \Gamma_s^{(m)} \neq \emptyset$, then $E \cap \Gamma_{s'}^{(m)} = \emptyset$. Hence,

$$L_\alpha^{(\ell)} \setminus \{\alpha\} \subseteq \Gamma_{s'}^{(m)} \setminus E \subseteq \Omega_{s'}^{(m)} \setminus E \subseteq \Omega_t^{(m)} \setminus E,$$

and by Lemma 26, $R = L_\alpha^{(\ell)} \setminus \{\alpha\}$ is a recovering set of α . ■

Now, we are ready to prove Theorem 21.

Proof of Theorem 21: By the construction, it is easy to see that the code length of $\mathcal{C}_t^{(m)}$ is

$$\begin{aligned} n &= \left| \Omega_t^{(m)} \right| \\ &= \sum_{s=0}^t \left| \Gamma_s^{(m)} \right| \\ &= \sum_{s=0}^t r^{m-|\text{supp}_m(s)|} \\ &= r^m \sum_{s=0}^t \frac{1}{r^{|\text{supp}_m(s)|}}, \end{aligned}$$

and the dimension of $\mathcal{C}_t^{(m)}$ is

$$k = \left| \Omega_0^{(m)} \right| = r^m.$$

So the code rate is

$$\frac{k}{n} = \frac{1}{\sum_{s=0}^t \frac{1}{r^{|\text{supp}_m(s)|}}}.$$

We then need to prove that $\mathcal{C}_t^{(m)}$ is an (r, t) -SLRC. It is sufficient to prove that if a binary linear code \mathcal{C} has a parity check matrix containing all rows of $H_t^{(m)}$, then it is an (r, t) -SLRC. We will prove this by an induction on m .

First, for $m = 1$, since $1 \leq t \leq 2^m - 1$, we have $t = 1$. By (26) and (27), $\Gamma_0^{(1)} = \mathbb{Z}_r$, $\Gamma_1^{(1)} = \{r\}$ and $\Omega_1^{(1)} = \mathbb{Z}_{r+1}$. So

$$H_1^{(1)} = (1, 1, \dots, 1)_{1 \times (r+1)}.$$

Clearly, the binary linear code \mathcal{C} with parity check matrix containing $H_1^{(1)}$ is an $(r, 1)$ -SLRC.

Now, suppose $m > 1$ and the induction assumption holds for all $m' < m$ and $t' \leq 2^{m'} - 1$. We consider m and $t \leq 2^m - 1$. Using the same notations as in Remark 24, we have the following four claims.

- i) For any nonempty $E \subseteq A$ of size $|E| \leq t_1$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq A \setminus E$.
- ii) For any nonempty $E \subseteq A$ of size $|E| \leq t$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq \Omega_t^{(m)} \setminus E$.
- iii) For any nonempty $E \subseteq B$ of size $|E| \leq t_2$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq B \setminus E$.
- iv) For any nonempty $E \subseteq B$ of size $|E| \leq t$, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq \Omega_t^{(m)} \setminus E$.

We will prove them one by one as follows.

i): Since $E \subseteq A$ and, by Remark 24, $A = \bigcup_{j=0}^{r-1} A_j$, then $E \cap A_{j_0} \neq \emptyset$ for some $j_0 \in \mathbb{Z}_r$. By 1) of Lemma 25, $\mathcal{C}|_{A_{j_0}}$ has a parity check matrix containing all rows of $H_{t_1}^{(m-1)}$. So by the induction assumption, $\mathcal{C}|_{A_{j_0}}$ is an (r, t_1) -SLRC. Moreover, it follows from the fact $|E \cap A_{j_0}| \leq |E| \leq t_1$ and Lemma 2 that there exists an $\alpha \in E \cap A_{j_0}$ such that α has a recovering set $R \subseteq A_{j_0} \setminus E \subseteq A \setminus E$.

ii): By Remark 24, $\{A_j; j \in \mathbb{Z}_r\}$ is a partition of A . We then consider the following two cases.

Case 1: There are $j_1, j_2 \in \mathbb{Z}_r$, $j_1 \neq j_2$, such that $E \cap A_{j_1} \neq \emptyset$ and $E \cap A_{j_2} \neq \emptyset$. According to Remark 24, $t \leq 2^{m_0} - 1 = 2t_1 + 1$. Then either $|E \cap A_{j_1}| \leq t_1$ or $|E \cap A_{j_2}| \leq t_1$. Without loss of generality, assume $|E \cap A_{j_1}| \leq t_1$. Similar as in the proof of 1), $\mathcal{C}|_{A_{j_1}}$ is an (r, t_1) -SLRC and there exists an $\alpha \in E \cap A_{j_1}$ such that α has a recovering set $R \subseteq A_{j_1} \setminus E \subseteq A \setminus E$.

Case 2: $E \subseteq A_{j_1}$ for some $j_1 \in \mathbb{Z}_r$. In this case, if $E \cap \Omega_0^{(m)} = \emptyset$. Then the desired α exists by Lemma 27. So we assume $E \cap \Omega_0^{(m)} \neq \emptyset$. Pick an $\alpha \in E \cap \Omega_0^{(m)}$. Recall that $t_1 = 2^{m_0-1} - 1$. By (34), one verifies that $L_\alpha^{(m_0)} \subseteq \Omega_0^{(m)} \cup \Gamma_{t_1+1}^{(m)}$ and $L_\alpha^{(m_0)} \cap A_{j_1} = \{\alpha\}$. Since $E \subseteq A_{j_1}$, then $R = L_\alpha^{(m_0)} \setminus \{\alpha\} \subseteq \Omega_{t_1+1}^{(m)} \setminus E \subseteq \Omega_t^{(m)} \setminus E$. By Lemma 26, R is a recovering set of α .

iii): If $t_2 = 0$, the claim is naturally true. Assume $t_2 \geq 1$. By 2) of Lemma 25, $\mathcal{C}|_B$ has a parity check matrix containing all rows of $H_{t_2}^{(m-1)}$. So by the induction assumption, $\mathcal{C}|_B$ is an (r, t_2) -SLRC. Hence, by Lemma 2, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq B \setminus E$.

iv): In this case, by the definition of B , we have $E \cap \Omega_0^{(m)} = \emptyset$. Hence, by Lemma 27, there exists an $\alpha \in E$ such that α has a recovering set $R \subseteq \Omega_t^{(m)} \setminus E$.

Combining i)-iv) and by Lemma 3, we complete the proof. \blacksquare

VI. CONSTRUCTION FROM RESOLVABLE CONFIGURATIONS

In [25], using $t-3$ mutually orthogonal latin squares (MOLS) of order r , the authors constructed a family of binary (r, t) -SLRC with $k = r^2$ and code rate $\frac{k}{n} = 1/(1 + \frac{t-1}{r} + \frac{1}{r^2})$ for odd t . A limitation of this construction is $t \leq r + 2$, which is an unfortunate consequence of the fact that the existing ℓ MOLS of order r ($r > 1$) necessarily requires $\ell \leq r-1$ [31]. In this section, using the so-called resolvable configurations, we give a new family of binary (r, t) -SLRC achieving the same rate $\frac{k}{n} = 1/(1 + \frac{t-1}{r} + \frac{1}{r^2})$ for any r and any odd $t \geq 3$ (not limited by $t \leq r + 2$).

We start with the following definition [31], [32].

Definition 28: Let X be a set of k elements, called points, and \mathcal{A} be a collection of subsets of X , called lines. The pair (X, \mathcal{A}) is called a (k_{t-1}, b_r) configuration if the following three conditions hold:

- (1) Each line contains r points;
- (2) Each point belongs to $t-1$ lines;
- (3) Every pair of distinct points belong to at most one line.

Clearly, condition (3) is equivalent to the following condition.

- (3') Every pair of distinct lines have at most one point in common.

The configuration (X, \mathcal{A}) is called *resolvable*, if further

- (4) All lines in \mathcal{A} can be partitioned into $t-1$ parallel classes, where a parallel class is a set of lines that partition X .

For any (k_{t-1}, b_r) resolvable configuration (X, \mathcal{A}) , one can see that $r|k$ and each parallel class contains $s = \frac{k}{r}$ lines. So, $b = \frac{k}{r}(t-1) = s(t-1)$ in such a case. As usual, the incidence matrix of a (k_{t-1}, b_r) configuration (X, \mathcal{A}) , where $X = \{x_1, \dots, x_k\}$ and $\mathcal{A} = \{A_1, \dots, A_b\}$, is defined as a $b \times k$ binary matrix $M = (m_{i,j})$ such that

$$m_{i,j} = \begin{cases} 1, & \text{if } x_j \in A_i; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, a configuration is uniquely determined by its incidence matrix.

Example 29: It can be verified that the following matrix determines a (k_{t-1}, b_r) resolvable configuration (X, \mathcal{A}) with $k = 9$, $t-1 = 4$, $b = 12$ and $r = 3$. Indeed, $\{A_1, A_2, A_3\}$, $\{A_4, A_5, A_6\}$, $\{A_7, A_8, A_9\}$ and $\{A_{10}, A_{11}, A_{12}\}$ are four parallel classes of (X, \mathcal{A}) and any pair of lines in different

parallel classes have one point in common.

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

Resolvable configurations have recently been used for constructing codes whose information symbols have locality r and availability t by Su [33]. The author also constructed some resolvable configurations in the paper, for example, the (k_{t-1}, b_r) resolvable configurations with $k = r^m$ and $t-1 \leq \frac{r^{m-1}}{r-1}$, where $m \geq 2$ and r is a prime power. The following construction, using the free \mathbb{Z}_r -module [35], not only generalizes the result of [33], but also enables us to construct (k_{t-1}, b_r) resolvable configuration for any $r, t \geq 2$ (here r need not be a prime power), and further (r, t) -SLRCs for any $r \geq 2$ and odd integer $t \geq 3$.

Lemma 30: For any $r, t \geq 2$, there exists a (k_{t-1}, b_r) resolvable configuration with $k = r^m$, where m is an arbitrary integer such that $m \geq \log_2 t$.

Proof: Consider the free \mathbb{Z}_r -module $X = \mathbb{Z}_r^m$, where \mathbb{Z}_r is the ring of integers modulo r . For any $\alpha \in \mathbb{Z}_r^m$, we use $\alpha(j)$ to denote the j th coordinate of α . For example, if $\alpha = (i_1, i_2, \dots, i_m)$, then $\alpha(j) = i_j$.

For each nonempty $S \subseteq [m]$, let $\alpha_S \in \mathbb{Z}_r^m$ be such that $\alpha_S(j) = 1$ for $j \in S$ and $\alpha_S(j) = 0$ otherwise. Let

$$A_{S,0} \triangleq \{i \cdot \alpha_S; i \in \mathbb{Z}_r\}.$$

Clearly, $A_{S,0}$ is a submodule of \mathbb{Z}_r^m with r elements and $A_{S,0} \cap A_{S',0} = (0, 0, \dots, 0)$ for any two distinct nonempty subsets S and S' of $[m]$. Let

$$\mathcal{A}_S = \{A_{S,\ell}, \ell = 0, 1, \dots, r^{m-1} - 1\}$$

be the collection of all cosets of $A_{S,0}$. Then $\alpha_1 - \alpha_2 \in A_{S,0}$ for any $\ell \in \{0, 1, \dots, r^{m-1} - 1\}$ and any $\alpha_1, \alpha_2 \in A_{S,\ell}$.

Note that $m \geq \log_2 t$ (i.e., $t-1 \leq 2^m - 1$) and $[m]$ has $2^m - 1$ nonempty subsets. We can always pick $t-1$ nonempty subsets of $[m]$, say, S_1, S_2, \dots, S_{t-1} . Let

$$\mathcal{A} = \bigcup_{i=1}^{t-1} \mathcal{A}_{S_i}.$$

We claim that $(X = \mathbb{Z}_r^m, \mathcal{A})$ is a (k_{t-1}, b_r) resolvable configuration, which can be seen as follows.

Firstly, note that for each nonempty $S \subseteq [m]$, \mathcal{A}_S is a partition of X , and thereby conditions (1), (2), (4) of Definition 28 hold. Secondly, if S, S' are two distinct nonempty subsets of $[m]$ and $\ell, \ell' \in \{0, 1, \dots, 2^m - 1\}$, then we have $|A_{S,\ell} \cap A_{S',\ell'}| \leq 1$. To see this, suppose that $\alpha_1, \alpha_2 \in A_{S,\ell} \cap A_{S',\ell'}$. Then, $\alpha_1 - \alpha_2 \in A_{S,0} \cap A_{S',0} = (0, 0, \dots, 0)$, and we have

$\alpha_1 = \alpha_2$, i.e., $|A_{S,\ell} \cap A_{S',\ell'}| \leq 1$. Moreover, since for each nonempty $S \subseteq [m]$, \mathcal{A}_S is a partition of X , so Condition (3) of Definition 28 holds, which completes the proof. ■

In the rest of this section, we always assume that (X, \mathcal{A}) is a (k_{t-1}, b_r) resolvable configuration and $\mathcal{A} = \{A_1, \dots, A_b\}$.

We will need the following lemma, which gives an property of the resolvable configuration (X, \mathcal{A}) with odd t .

Lemma 31: Let E be a t -subset of X and t be an odd integer. Then there exists an $A_j \in \mathcal{A}$ such that $|E \cap A_j| = 1$.

Proof: Consider a parallel class of (X, \mathcal{A}) . Since it is a partition of X and $|E| = t$ is odd, there exists some A_{j_1} in the class such that $|E \cap A_{j_1}|$ is odd. If $|E \cap A_{j_1}| = 1$, then we have done. So suppose $E = \{i_1, \dots, i_t\}$ and $\{i_1, i_2, i_3\} \subseteq E \cap A_{j_1}$. Since each point belongs to $t-1$ lines, we can assume i_1 belongs to lines $A_{j_1}, A_{j_2}, \dots, A_{j_{t-1}}$, where $A_{j_1}, A_{j_2}, \dots, A_{j_{t-1}}$ belong to different parallel classes. Moreover, since every pair of distinct points belong to at most one line, then $i_2, i_3 \notin A_{j_\ell}, \forall \ell \in \{2, \dots, t-1\}$ and each point $i_\ell, \ell \in \{4, \dots, t\}$, belongs to at most one line in $\{A_{j_2}, \dots, A_{j_{t-1}}\}$. Hence, there exists a line $A_j \in \{A_{j_2}, \dots, A_{j_{t-1}}\}$ that contains no point in $\{i_2, \dots, i_t\}$. That is to say, $E \cap A_j = \{i_1\}$, which completes the proof. ■

From now on, we let $X = [k]$ and $\mathcal{A}_1 = \{A_1, \dots, A_s\}$, $\mathcal{A}_2 = \{A_{s+1}, \dots, A_{2s}\}, \dots, \mathcal{A}_{t-1} = \{A_{(t-2)s+1}, \dots, A_b\}$ be the $t-1$ parallel classes of (X, \mathcal{A}) . We further partition $[s]$ into $\lceil \frac{s}{r} \rceil$ nonempty subsets, say, $B_1, \dots, B_{\lceil \frac{s}{r} \rceil}$, such that $|B_i| \leq r$ for all $i \in \{1, \dots, \lceil \frac{s}{r} \rceil\}$. Such a partition plays a subtle role in our construction, as will become clear later. Now, let $W = (w_{i,j})$ be a $\lceil \frac{s}{r} \rceil \times b$ matrix defined by

$$w_{i,j} = \begin{cases} 1, & \text{if } j \in B_i \\ 0, & \text{otherwise.} \end{cases}$$

Let M be the incidence matrix of (X, \mathcal{A}) and

$$H = \begin{pmatrix} M & I_b & O_{b \times \lceil \frac{s}{r} \rceil} \\ O_{\lceil \frac{s}{r} \rceil \times k} & W & I_{\lceil \frac{s}{r} \rceil} \end{pmatrix}, \quad (40)$$

where I_ℓ denotes the $\ell \times \ell$ identity matrix and $O_{\ell \times \ell'}$ denotes the $\ell \times \ell'$ all-zero matrix for any positive integers ℓ and ℓ' . Clearly, H has $b + \lceil \frac{s}{r} \rceil$ rows, $n = k + b + \lceil \frac{s}{r} \rceil$ columns and rank $b + \lceil \frac{s}{r} \rceil$.

As an example, consider the resolvable configuration (X, \mathcal{A}) in Example 29. We have $s = \frac{k}{r} = 3$ and $\lceil \frac{s}{r} \rceil = 1$. So we can construct $W = (1, 1, 1, 0, \dots, 0)_{1 \times 12}$ and $I_{\lceil \frac{s}{r} \rceil} = (1)_{1 \times 1}$, and according to (40), further construct a matrix H as in (39).

Let \mathcal{C} be a binary linear code with parity check matrix H as in (39). Then from the first 12 rows of H , we can see that the coordinate 1 has 4 disjoint recovering sets, i.e., $\{2, 3, 10\}$, $\{4, 7, 13\}$, $\{6, 8, 16\}$ and $\{5, 9, 19\}$, and the coordinate 10 has a recovering set $\{1, 2, 3\} \subseteq \{1, \dots, 9\}$. Moreover, from the last row of H , we can see that $\{11, 12, 22\}$ is a recovering set of 10 and $\{10, 11, 12\}$ is a recovering set of 22. In general, we have the following lemma.

Lemma 32: Let \mathcal{C} be an $[n, k]$ binary linear code with parity check matrix H as in (40). Then, the following hold:

- 1) Each $i \in [k]$ has $t-1$ disjoint recovering sets, i.e., $A_{j_\ell} \cup \{k + j_\ell\} \setminus \{i\}$, where $A_{j_\ell}, \ell = 1, \dots, t-1$, are lines containing i .

- 2) Each $i \in \{k+1, \dots, k+b\}$ has a recovering set $R \subseteq [k]$.
- 3) Each $i \in \{k+1, \dots, k+s\}$ has a recovering set $R \subseteq \{k+1, \dots, k+s\} \cup \{k+b+1, \dots, n\} \setminus \{i\}$.
- 4) Each $i \in \{k+b+1, \dots, n\}$ has a recovering set $R \subseteq \{k+1, \dots, k+s\}$.

Proof: 1) and 2) can be established by considering the first b rows of H , and 3) and 4) are by considering the last $\lceil \frac{s}{r} \rceil$ rows of H . ■

Theorem 33: If t is odd, then the binary linear code \mathcal{C} with parity check matrix H as in (40) is an (n, k, r, t) -SLRC with rate

$$\frac{k}{n} = \left(1 + \frac{t-1}{r} + \left\lceil \frac{1}{r^2} \right\rceil \right)^{-1}.$$

Proof: By the construction, \mathcal{C} has block length

$$\begin{aligned} n &= k + b + \left\lceil \frac{s}{r} \right\rceil \\ &= k \left(1 + \frac{t-1}{r} + \left\lceil \frac{1}{r^2} \right\rceil \right). \end{aligned}$$

and dimension $n - (b + \lceil \frac{s}{r} \rceil) = k$. So the code rate is

$$\frac{k}{n} = \left(1 + \frac{t-1}{r} + \left\lceil \frac{1}{r^2} \right\rceil \right)^{-1}.$$

We now prove, according to Lemma 2, that for any $E \subseteq [n]$ with $|E| \leq t$, there exists an $i \in E$ such that i has a recovering set $R \subseteq [n] \setminus E$. Consider the following cases.

Case 1: $E \cap [k] = \emptyset$. Then we have $E \subseteq \{k+1, \dots, n\}$. If $E \cap \{k+1, \dots, k+b\} \neq \emptyset$, then by 2) of Lemma 32, each $i \in E \cap \{k+1, \dots, k+b\}$ has a recovering set $R \subseteq [k] \subseteq [n] \setminus E$; Otherwise, $E \subseteq \{k+b+1, \dots, n\}$, then by 4) of Lemma 32, each $i \in E$ has a recovering set $R \subseteq \{k+1, \dots, k+s\} \subseteq \{k+1, \dots, k+b\} \subseteq [n] \setminus E$.

Case 2: $E \cap [k] \neq \emptyset$. Pick an $i_1 \in E \cap [k]$. Let

$$R_\ell = A_{j_\ell} \cup \{k + j_\ell\} \setminus \{i_1\}, \quad \ell = 1, \dots, t-1, \quad (41)$$

where $A_{j_1}, \dots, A_{j_{t-1}}$ are the $t-1$ lines containing i_1 . By 1) of Lemma 32, R_1, \dots, R_{t-1} are $t-1$ disjoint recovering sets of i_1 . If $R_\ell \subseteq [n] \setminus E$ for some $\ell \in \{1, \dots, t-1\}$, then we are done. So in the following we assume $E \cap R_\ell \neq \emptyset$ for all $\ell \in \{1, \dots, t-1\}$. Since all R_ℓ 's are disjoint, so $|E| = t$, $|E \cap R_\ell| = 1, \ell = 1, \dots, t-1$, and $E \subseteq \{i_1\} \cup \left(\bigcup_{\ell=1}^{t-1} R_\ell \right)$. We have the following three subcases:

Case 2.1: $E \cap R_\ell \subseteq [k], \forall \ell \in \{1, \dots, t-1\}$. Then $E \subseteq [k]$. Since $|E| = t$ is odd, by Lemma 31, $|E \cap A_i| = 1$ for some $A_i \in \mathcal{A}$. Let $E \cap A_i = \{i_2\}$. By 1) of Lemma 32, $R = A_i \cup \{k+i\} \setminus \{i_2\} \subseteq [n] \setminus E$ is a recovering set of i_2 .

Case 2.2: $E \cap R_{\ell_1} \subseteq [k]$ and $E \cap R_{\ell_2} \not\subseteq [k]$ for some $\{\ell_1, \ell_2\} \subseteq \{1, \dots, t-1\}$. Without loss of generality, assume $i_2 \in E \cap R_{\ell_1} \subseteq [k]$ and $E \cap R_{\ell_2} \not\subseteq [k]$. Then according to (41), we have $E \cap R_{\ell_2} = \{k + j_2\}$. By 1) of Lemma 32, we can let R'_1, \dots, R'_{t-1} are $t-1$ disjoint recovering sets of i_2 , where $R'_1 = A_{j_1} \cup \{k + j_1\} \setminus \{i_2\}$ and $R'_\ell = A_{j_\ell} \cup \{k + j_\ell\} \setminus \{i_2\}, \ell = 2, \dots, t-1$, such that A_{j_1} together with $A_{j_2}, \dots, A_{j_{t-1}}$ are the $t-1$ lines containing i_2 (see Fig. 9). Note that A_{j_1} is the only line containing both i_1 and i_2 , one can see that $\{i_1, i_2, k + j_2\} \cap R'_\ell = \emptyset, \forall \ell \in \{2, \dots, t-1\}$. Since $|E| = t$,

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (39)$$

then there exists an $\ell_0 \in \{2, \dots, t-1\}$ such that $E \cap R'_{\ell_0} = \emptyset$. Hence, $R'_{\ell_0} \subseteq [n] \setminus E$ is a recovering sets of i_2 .

Case 2.3: $E \cap R_\ell \not\subseteq [k]$ for all $\ell \in \{1, \dots, t-1\}$. Then we have $E \cap R_\ell = \{k + j_\ell\}$. Note that $A_{j_1}, \dots, A_{j_{t-1}}$ belong to distinct parallel classes (since all of them contain i_1). Without loss of generality, we assume $A_{j_\ell} \in \mathcal{A}_\ell$, $\ell \in \{1, \dots, t-1\}$. Then $j_1 \leq s$ and $s < j_\ell \leq b$, $\ell = 2, \dots, t-1$. By 3) of Lemma 32, $k + j_1$ has a recovering set $R \subseteq \{k+1, \dots, k+s\} \cup \{k+b+1, \dots, n\} \setminus \{j_1\} \subseteq [n] \setminus E$.

By the above discussion, for any $E \subseteq [n]$ of size $|E| \leq t$, there exists an $i \in E$ such that i has a recovering set $R \subseteq [n] \setminus E$. So by Lemma 2, \mathcal{C} is an (n, k, r, t) -SLRC. ■

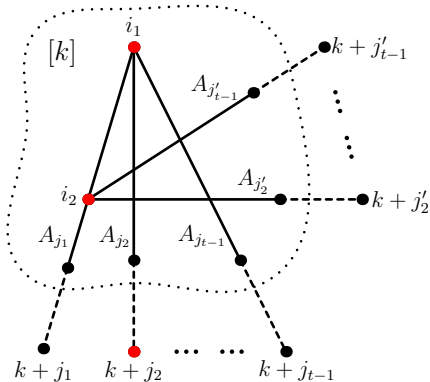


Fig 9. Illustration of points and recovering sets: $R_\ell = A_{j_\ell} \cup \{k + j_\ell\} \setminus \{i_1\}$, $\ell = 1, \dots, t-1$, are $t-1$ recovering sets of i_1 ; R_1 and $R'_\ell = A_{j'_\ell} \cup \{k + j'_\ell\} \setminus \{i_2\}$, $\ell = 2, \dots, t-1$, are $t-1$ recovering sets of i_2 .

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we examined sequential locally repairable codes (SLRCs), for which we have established an upper bound on the code rate of (n, k, r, t) -SLRC for $t = 3$, and constructed two families of (n, k, r, t) -SLRC for $r, t \geq 2$ (for the second family, t is odd). Both of our constructions have code rate $> \frac{r}{r+t}$ and are optimal for $t \in \{2, 3\}$ with respect to the proposed bound.

It is still an open problem to determine the optimal code rate of (n, k, r, t) -SLRCs for general t , i.e., $t \geq 5$. Here, we

conjecture that an achievable upper bound of the code rate of (n, k, r, t) -SLRCs has the following form:

$$\frac{k}{n} \leq \left(1 + \sum_{i=1}^m \frac{a_i}{r^i} \right)^{-1}, \quad (42)$$

where $m = \lceil \log_r k \rceil$, all $a_i \geq 0$ are integers such that $\sum_{i=1}^m a_i = t$. This conjecture can be verified for $t \in \{1, 2, 3, 4\}$, for which the values of the m -tuple (a_1, \dots, a_m) , denoted by α_t for each t , are listed in the following table, where, the cases of $t = 2, 3$ are due to [22] and this work, respectively. The case of $t = 4$ (for binary code) is recently due to Balaji et al [26].

t	a_1	a_2	a_3	\dots	a_m
1	1	0	0	\dots	0
2	2	0	0	\dots	0
3	2	1	0	\dots	0
4	2	2	0	\dots	0

Table 1. The known values of $\alpha_t = (a_1, a_2, \dots, a_m)$.

It is worthwhile to point out that in a recent work by Balaji et al. [27], an achievable upper bound on the rate of (n, k, r, t) -SLRC matching the conjecture (42) for any $r, t \geq 3$ is proven.

ACKNOWLEDGMENT

This research is supported by SUTD-MIT International Design Center and partly supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08).

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] J. Pemas, B. Gaston, C. Yuen and J. Pujol, "Non-homogeneous two-rack model for distributed storage systems," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, 2013, pp. 1237-1241.
- [3] V. T. Van, C. Yuen, J. Li, "Non-homogeneous distributed storage systems," in *Allerton Conference*, 2012, pp. 1133-1140.
- [4] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, USA, Jul. 2012, pp. 2771-2775.
- [5] 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.

- [6] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *INFOCOM, 2011 Proceedings IEEE*, pp. 1215-1223, IEEE, 2011.
- [7] 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.
- [8] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen and D. Borthakur, "XORing elephants: Novel erasure codes for big data," in *Proc. Very Large Data Bases Endowment*, vol. 6, no. 5, pp. 325-336, Mar. 2013.
- [9] D. Ford, F. Labelle, F. Popovici, M. Stokely, V. -A. Truong, L. Barroso, C. Grimes and S. Quinlan, "Availability in globally distributed storage systems," in *Proc. the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 61-74, 2010.
- [10] L. Pamies-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.
- [11] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, "Optimal linear codes with a local-error-correction property," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, USA, Jul. 2012, pp. 2776-2780.
- [12] A. Wang and Z. Zhang, "Repair locality with multiple erasure tolerance," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6979-6987, Nov. 2014.
- [13] 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.
- [14] I. Tamo, A. Barg, "Bounds on locally recoverable codes with multiple recovering sets," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Honolulu, HI, USA, June. 2014, pp. 691-695.
- [15] I. Tamo, A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 80, pp. 4661-4676, Aug. 2014.
- [16] A. Wang, Z. Zhang, and M. Liu, "Achieving Arbitrary Locality and Availability in Binary Codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 1866-1870, 2015.
- [17] A. S. Rawat, A. Mazumdar, and S. Vishwanath, "On cooperative local repair in distributed storage," in *Proc. 48th Annu. Conf. Inf. Sci. Syst.*, Mar. 2014, pp. 1-5.
- [18] A. S. Rawat, A. Mazumdar, and S. Vishwanath, "Cooperative local repair in distributed storage," *EURASIP Journal on Advances in Signal Processing*, pp. 1-17, 2015:107.
- [19] 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.
- [20] P. Huang, E. Yaakobi, H. Uchikawa, and P. H. Siegel, "Linear locally repairable codes with availability," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 1871-1875, 2015.
- [21] A. Wang, Z. Zhang, and D. Lin, "Two Classes of (r, t) -Locally Repairable Codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 445 - 449.
- [22] N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with locality for two erasures," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Honolulu, HI, USA, June. 2014, pp. 1962-1966.
- [23] W. Song and C. Yuen, "Locally Repairable Codes with Functional Repair and Multiple Erasure Tolerance," <http://arxiv.org/abs/1507.02796>, 2015.
- [24] W. Song and C. Yuen, "Binary locally repairable codes - sequential repair for multiple erasures," in *Proc. Globecom 2016*, also available at: <http://arxiv.org/abs/1511.06034>, 2015.
- [25] S. B. Balaji, K. P. Prasanth, and P. V. Kumar, "Binary codes with locality for multiple erasures having short block length," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 655 - 659.
- [26] S. B. Balaji, K. P. Prasanth, and P. V. Kumar, "Binary Codes with Locality for Four Erasures," Available at: <https://arxiv.org/abs/1607.02817>
- [27] S. B. Balaji, G. R. Kini, and P. V. Kumar, "A Tight Rate Bound and a Matching Construction for Locally Recoverable Codes with Sequential Recovery From Any Number of Multiple Erasures," Available at: <https://arxiv.org/abs/1611.08561>
- [28] A. Wang and Z. Zhang, "An Integer Programming-Based Bound for Locally Repairable Codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5280-5294, Oct. 2015.
- [29] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 829-848, Mar. 2006.
- [30] 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.
- [31] C. J. Colbourn and J. H. Denitz, *Handbook of Combinatorial Designs*, 2nd ed. Chapman and Hall/CRC, 2006.
- [32] T. Pisanski and B. Servatius, *Configurations from a graphical viewpoint*, Birkhauser, Basel, 2013.
- [33] Y.S. Su, "Design of Membership Matrices for (r, t) -Availability in Distributed Storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 998 - 1002.
- [34] H. S. Dau, W. Song and C. Yuen, "On Block Security of Regenerating Codes at the MBR Point for Distributed Storage Systems," in *Proc. Int. IEEE Symp. Inf. Theory (ISIT)*, 2014, pp. 1967-1971.
- [35] W.C. Brown, *Matrices over Commutative Rings*, Dekker, New York, 1993.

Wentu Song received the BS and MS degrees in Mathematics from Jilin University, China in 1998 and 2006, respectively, and the Ph.D. degree in Mathematics from Peking University in 2012, China. He is currently a Post Doc research fellow in SUTD-MIT International Design Centre, Singapore University of Technology and Design, where he is working on coding theory, network coding and network distributed storage.

Kai Cai received the Ph.D. degree in Mathematics from Peking University, Beijing, China, in 2004. From 2007 to 2011, he was an associate professor in the Institute of Computing Technology, Chinese Academy of Sciences. He is now with the Department of Mathematics, the University of Hong Kong. His research interests are in algebraic combinatorics, coding theory, and so forth.

Chau Yuen (S'02M'08SM'12) received the B. Eng and PhD degree from Nanyang Technological University, Singapore in 2000 and 2004 respectively. Dr Yuen was a Post Doc Fellow in Lucent Technologies Bell Labs, Murray Hill during 2005. He was a Visiting Assistant Professor of Hong Kong Polytechnic University in 2008. During the period of 2006 - 2010, he worked at the Institute for Infocomm Research (Singapore) as a Senior Research Engineer. He joined Singapore University of Technology and Design from June 2010. He serves as an Editor for IEEE Transactions on Communications and IEEE Transactions on Vehicular Technology. On 2012, he received IEEE Asia-Pacific Outstanding Young Researcher Award.

Kui Cai received B.E. degree in information and control engineering from Shanghai Jiao Tong University, Shanghai, China, M.Eng degree in electrical engineering from National University of Singapore, and joint Ph.D. degree in electrical engineering from Technical University of Eindhoven, The Netherlands, and National University of Singapore. Currently, she is an Associate Professor with Singapore University of Technology and Design (SUTD). She received 2008 IEEE Communications Society Best Paper Award in Coding and Signal Processing for Data Storage. She served as the Vice-Chair (Academia) of IEEE Communications Society, Data Storage Technical Committee (DSTC) during 2015 and 2016. Her main research interests are in the areas of coding theory, information theory, and signal processing for various data storage systems and digital communications.

Guangyue Han received the B.S. and M.S. degrees in mathematics from Peking University, China, and the Ph.D. degree in mathematics from the University of Notre Dame, U.S.A. in 1997, 2000 and 2004, respectively. After three years with the department of mathematics at the University of British Columbia, Canada, he joined the department of mathematics at the University of Hong Kong, China in 2007. His main research areas are coding and information theory.