**SCIENCE CHINA**
Information Sciences

**• RESEARCH PAPER •**
Special Issue on Distributed Storage Coding

# Code constructions for multi-node exact repair in distributed storage

Marwen Zorgui[1*] & Zhiying Wang[1]

[1] *University of California, Irvine, CA* 92697*, USA*

**Abstract** We study the problem of centralized exact repair of multiple failures in distributed storage. We present constructions that achieve a new set of interior points under exact repair. The constructions build upon the layered code construction by Tian et al., designed for exact repair of single failure. We firstly improve upon the layered construction for general system parameters. Then, we extend the improved construction to support the repair of multiple failures, with varying number of helpers. In particular, for some parameters, we prove the optimality of one point in terms of the storage size and the repair bandwidth for multiple erasures. Finally, considering minimum bandwidth cooperative repair (MBCR) codes as centralized repair codes, we determine explicitly the best achievable region obtained by space-sharing among all known points, including the MBCR point.

**Keywords** Regenerating codes, exact repair, multiple failures, interior points, Steiner systems

**Citation** Zorgui M, Wang Z Y. Code constructions for multi-node exact repair in distributed storage. Sci China Inf Sci, 2018, 61(): , https://doi.org/10.1007/s11432-018-9516-6

## 1 Introduction

Driven by the growth of data-centric applications, efficient data storage and retrieval has become of crucial importance for data storage providers. Distributed storage systems (DSS) are currently widely employed for large-scale storage. DSS provide scalable storage and high level of resilience in the face of server failures. To maintain the desired level of failure tolerance, DSS utilize a replacement mechanism known also as the repair mechanism, that allows to recover the content of inaccessible/failed nodes. The repair process of a failed node is performed by downloading data from accessible nodes (or a subset thereof) in the system and recovering the lost data. In this work, we focus on two metrics in evaluating the efficiency of DSS, namely, the overhead required for reliability and the amount of data being transferred for a repair process. The seminal work in [2] proposed a new class of erasure codes, called regenerating codes, that optimally solve the repair bandwidth problem for a single failure. It is shown in [2] that one can significantly reduce the amount of bandwidth required for repair and the bandwidth decreases as each node stores more information. In particular, the optimal tradeoff between the storage size and the bandwidth is derived. Regenerating codes, as presented in [2], achieve *functional repair*. In this case, the replacement nodes are not required to be exact copies of the failed nodes, but the repaired code should satisfy reliability constraints. However, in practice, it is often more desirable to recover the exact same information as the failed node, which is called *exact repair*. Exact-repair codes are easier to implement and maintain, and thus are of more interest.

[*] Corresponding author (email: mzorgui@uci.edu)

For a single failure, extreme points on the functional repair tradeoff correspond to minimum storage regenerating (MSR) codes and minimum bandwidth regenerating (MBR) codes. Interior points are points that lie between the MSR and the MBR points, whose storage size is no less than the MSR point and bandwidth is no less than the MBR point. There has been a flurry of interest in the achievability and constructions of exact-repair regenerating codes for the aforementioned points. Several constructions have been proposed for exact-repair MSR codes [3–11] and for exact-repair MBR codes [3], respectively, hence proved the achievability of the extreme points under exact repair. The authors in [12] first showed that most of the optimal functional repair interior points are non-achievable under exact repair, except maybe for a small part close to the MSR point. Later, code constructions for exact-repair interior points have been proposed in [13–18]. Moreover, there is a growing literature focused on understanding the fundamental limits of exact-repair regenerating codes [20–23], as opposed to the well-understood functional regenerating codes [2].

### 1.1 Multi-node recovery

In many practical scenarios, such as in large-scale storage systems with correlated failure patterns, multiple failures are more frequent than a single failure. Moreover, many systems apply a lazy repair strategy, which waits for several failures before repair and seeks to limit the repair cost of erasure codes. Indeed, it has been demonstrated that jointly repairing multiple failures reduces the overall bandwidth compared to repairing each failure individually [24–27]. We distinguish between two ways of repairing multiple failures.

*Cooperative regenerating codes*: In this framework, each replacement node first downloads information from $d$ nodes (helpers). Then, the replacement nodes exchange information between themselves before regenerating the lost nodes. Of interest to our work, we note that codes corresponding to the extreme points on the cooperative tradeoff have been developed: minimum storage cooperative regenerating (MSCR) codes [26, 29, 30] and minimum bandwidth cooperative regenerating (MBCR) codes [31].

*Centralized regenerating codes*: Upon failure of $e$ nodes, the repair is carried out in a centralized way by contacting any $d$ helpers, and downloading $\beta$ amount of information from each helper. We require the content of any $k$ out of $n$ nodes in the system to be sufficient to reconstruct the entire data. Let $\alpha$ be the size of each node, and $F$ be the size of the entire data. A code satisfying the centralized repair constraints is referred to as an $(F, n, k, d, e, \alpha, \beta)$ code. We also say it is a code of the $(n, k, d, e)$ system. In our previous work [27], we characterized the functional repair tradeoff between $\alpha$ and $\beta$ for multi-node recovery. Let $g = \lceil \frac{k}{e} \rceil, t = k - (g-1)e$, where $\lceil \cdot \rceil$ is the ceiling function. The normalized functional tradeoff can then be written as follows

$$\min(t\bar{\alpha}, d\bar{\beta}) + \sum_{p=0}^{g-2} \min(e\bar{\alpha}, (d-t-pe)\bar{\beta}) \geqslant 1, \tag{1}$$

where $\bar{\alpha} = \frac{\alpha}{F}, \bar{\beta} = \frac{\beta}{F}$. Inequality (1) gives $g-1$ linear bounds:

$$(t+pe)\bar{\alpha} + \sum_{i=p}^{g-2} (d-t-ie)\bar{\beta} \geqslant 1, \quad p = 0, \ldots, g-2. \tag{2}$$

In this work, we are interested in designing *centralized exact-repair* regenerating codes for recovering multiple failures. When $e \geqslant k$, the tradeoff reduces to a single point, which is trivially achievable [27]. We hereafter focus on the case $e < k$.

On the functional tradeoff, centralized minimum storage multi-node repair (MSMR) codes [10, 25, 28] are of particular interest to this work, while the existence of exact-repair centralized minimum bandwidth multi-node repair (MBMR) codes is an open problem [27]. MSMR and MBMR points are given by

$$(\bar{\alpha}_{MSMR}, \bar{\beta}_{MSMR}) = (\frac{1}{k}, \frac{e}{k(d-k+e)}) = (\frac{1}{k}, \frac{e}{d-k+e}\bar{\alpha}_{MSMR}), \tag{3}$$

$$(\bar{\alpha}_{MBMR}, \bar{\beta}_{MBMR}) = (\frac{2(d-k+gt)}{gt(2d-k+t)}, \frac{1}{dg - e\binom{g}{2}}). \tag{4}$$

In this paper, we define interior points to be points that lie between the MSMR and the MBMR points. Precisely, an $(\bar{\alpha}, \bar{\beta})$ point is an interior point if it satisfies: $\bar{\alpha} \geqslant \bar{\alpha}_{\rm MSMR}, \bar{\beta} \geqslant \bar{\beta}_{\rm MBMR}, (\bar{\alpha}, \bar{\beta}) \neq (\bar{\alpha}_{\rm MSMR}, \bar{\beta}_{\rm MSMR}), (\bar{\alpha}, \bar{\beta}) \neq (\bar{\alpha}_{\rm MBMR}, \bar{\beta}_{\rm MBMR})$. We note that for single erasure, several works have investigated construction of exact-repair codes operating at interior points [13, 17–19]. Unlike previous works, our main goal in this paper is to construct codes operating at interior points for the multi-node exact-repair problem.

**Remark 1.** Note that the MBMR point defined in (4) may not be achievable under exact repair when $e \geqslant 2$. In particular, in [27], it was shown that for $e \geqslant 2$, the functional MBMR point is not achievable for linear exact-repair codes.

In [25], it is argued that cooperative regenerating codes can be used to construct centralized repair codes. The total bandwidth in this case is obtained by taking into account the bandwidth obtained from the helper nodes and disregarding the communication between the replacement nodes. In particular, MSCR codes achieve the same performance as MSMR codes. Additionally, MBCR codes can be used as centralized repair codes, but do not correspond to MBMR codes [27]. These points are given by

$$(\bar{\alpha}_{MSCR}, \bar{\beta}_{MSCR}) = (\bar{\alpha}_{MSMR}, \bar{\beta}_{MSMR}) = (\frac{1}{k}, \frac{e}{k(d-k+e)}), \tag{5}$$

$$(\bar{\alpha}_{MBCR}, \bar{\beta}_{MBCR}) = (\frac{2d+e-1}{k(2d-k+e)}, \frac{2e}{k(2d-k+e)}). \tag{6}$$

It can be checked that MBCR codes correspond to an interior point [36].[1)]

### 1.2 Existing universal MSMR constructions

Constructions for MSMR codes have been presented in the literature for specific $(n, k, d, e)$ systems [1, 25, 28]. In this work, we are interested in *universal* MSMR codes. Universal MSMR codes achieve simultaneously the optimal repair bandwidth for all $e \leqslant n-k, k \leqslant d \leqslant n-e$. Universal MSMR codes are the building blocks in our construction in this paper. Based on subspace interference alignment, the authors in [8] presented a maximum distance separable (MDS) code with asymptotic optimal repair for all $e \leqslant n-k$ and $k \leqslant d \leqslant n-e$ simultaneously. In [10], the authors presented two families of universal MSMR codes. Given integers $n$ and $n-k$, the first family of codes satisfy $\alpha = s^n$, where $s = lcm(1, 2, \ldots, n-k)$, $lcm$ being the least common multiple, and they can be constructed over any base field $\mathbb{F}$ of size $|\mathbb{F}| \geqslant sn$. The second family of codes satisfy the optimal access property. That is, the repair can be accomplished by accessing the amount of data that equals the optimal $\beta_{MSMR}$ in (3). The codes in the second family satisfy also $\alpha = s^n$, where $s = lcm(1, 2, \ldots, n-k)$, and they can be constructed over any base field $\mathbb{F}$ of size $|\mathbb{F}| \geqslant n+1$.

As pointed out in the previous subsection, MSCR codes are in particular MSMR codes. The authors in [30] constructed universal MSCR codes. The code can be constructed over a base field of size $|\mathbb{F}| \geqslant (n-k)n$, with $\alpha = \prod_{1 \leqslant e \leqslant n-d \leqslant n-k} ((e+s-1)(s-1)^{e-1})^{\binom{n}{e}}$, where $s = d-k+1$.

Note that for the particular case of $n = k+2$, universal MSMR code are simply MSR codes with $d = k+1 = n-1$. Several MSR code constructions exist in the literature.

**Example 1.** We provide an example of an MSR code with $(n, k, d) = (4, 2, 3)$ in Table 1. Since $n = k+2$, this is also a universal MSMR code. The construction is known as EVENODD code [37]. We refer to the codeword symbols as symbols. Each symbol is a binary vector of length 2, namely, an element of $\mathbb{F}_2^2$. The information bits are $a, b, c, d \in \mathbb{F}_2$. It can be checked that from any 2 symbols, all the information bits can be recovered. Therefore, $k = 2$. Moreover, the repair of EVENODD code in Table 1 satisfies (3). Indeed, the repair of each of the 4 symbols can be carried out as follows.

[1)] In [36], it was shown that for $e \geqslant 2$, when $k$ is a multiple of $e$, $\bar{\beta}_{\rm MBCR} = \bar{\beta}_{\rm MBMR}, \bar{\alpha}_{\rm MBCR} > \bar{\alpha}_{\rm MBMR}$; otherwise, $\bar{\beta}_{\rm MBCR} > \bar{\beta}_{\rm MBMR}$.

| symbol 1 | symbol 2 | symbol 3 | symbol 4 |
|---|---|---|---|
| $a$ | $c$ | $a+c$ | $a+d$ |
| $b$ | $d$ | $b+d$ | $b+c+d$ |

**Table 1** EVENODD code: an example of an MSR code. The code uses the binary field $\mathbb{F}_2$. "+" denotes XOR.

- To repair symbol 1, symbol 2 sends $c$, symbol 3 sends $a+c$, and symbol 4 sends $(a+d)+(b+c+d) = (a+c)+b$.
- To repair symbol 2, symbol 1 sends $b$, symbol 3 sends $b+d$ and symbol 4 sends $b+c+d$.
- To repair symbol 3, symbol 1 sends $b$, symbol 2 sends $d$ and symbol 4 sends $(a+d)+(b+c+d) = (a+c)+b$.
- To repair symbol 4, symbol 1 sends $b$, symbol 2 sends $c+d$ and symbol 3 sends $a+c$.

Since each helper contributes $\frac{1}{2}$ of its size, we say that each helper contributes $\frac{1}{2}$ symbol in $\mathbb{F}_2^2$ in the repair process. More generally, the contribution of a helper in the repair process is a fraction of a symbol, given by $\frac{\bar{\beta}_{MSMR}}{\bar{\alpha}_{MSMR}} = \frac{e}{d-k+e}$ in (3). We use the EVENODD code in Table 1 as the underlying MSMR code in our subsequent illustrative examples.

### 1.3 Contributions and organization of the paper

The main contributions of this paper are summarized as follows.

- We improve upon the layered construction presented in [18], which is concerned with single node repair, to construct a family of regenerating codes that is capable of repairing multiple failures. To the best of our knowledge, this is the first known construction for multiple erasures in the interior points of the tradeoff. Moreover, our code possesses the universal repair property: it allows a varying number of failures and a varying number of helpers, such that $1 \leqslant e \leqslant n-k, k \leqslant d \leqslant n-e$.
- We illustrate our code construction via different examples and evaluate its performance under various scenarios.
- For the $(k+e, k, k, e)$ system, we first prove the optimality of a particular constructed code using the functional repair tradeoff (1); combining the achievable points via our construction and also the MBCR point, we then determine the best achievable region obtained by space-sharing among all known points.

The remainder of the paper is organized as follows. A description of our code construction is provided in Section 2. In Section 3, we analyze the achievability region of the $(k+e, k, k, e)$ system. The last section concludes the paper.

Notation: we denote by $[i]$ the set of integers $\{1, 2, \ldots, i\}$ for $i \geqslant 1$. The notations $\lceil \cdot \rceil, \lfloor \cdot \rfloor$ represent the ceiling and the floor functions.

## 2 Code construction

Exact-repair regenerating codes are characterized by parameters $(F, n, k, d, e, \alpha, \beta)$. We consider a distributed storage system with $n$ nodes storing $F$ amount of information. The data elements are distributed across the $n$ storage nodes such that each node stores up to $\alpha$ amount of information. We use $\bar{\alpha} = \frac{\alpha}{F}, \bar{\beta} = \frac{\beta}{F}$ to denote the normalized storage size and repair bandwidth, respectively. The system should satisfy the following two properties:

*Reconstruction property*: by connecting to any $k \leqslant n$ nodes it should be sufficient to reconstruct the entire data.

*Repair property*: upon failure of $e$ nodes, $1 \leqslant e \leqslant n-k$, a central node is assumed to contact $d$ helpers, $k \leqslant d \leqslant n-e$, and download $\beta$ amount of information from each of them. The exact content of the failed nodes is determined by the central node. $\beta$ is called the repair bandwidth.

We first describe the code construction which is an improvement upon [18]. The construction is based on a collection of subsets of $[n]$, called a Steiner system. Information is first encoded within each subset, and then distributed among the $n$ nodes. We recall the definition of Steiner systems.

**Definition 1.** A Steiner system $S(t,r,n)$, $t \leqslant r \leqslant n$, is a collection of subsets of size $r$, included in $[n]$, such that any subset of $[n]$ of size $t$ appears exactly once across all the subsets.

The existence of Steiner systems is not known in general when $t < r < n$. But for large $n$, [32] proved the existence of Steiner systems as long as the parameters $t, r, n$ satisfy certain divisibility conditions. When $t = r$, Steiner systems always exist, and the subsets in this case are all $r$-combinations of the set $[n]$. We call this case trivial Steiner systems. The family of $(F, n, k, d, e, \alpha, \beta)$ codes we describe below is parameterized by $t, m, r$, for $e \leqslant m < r \leqslant n, t \leqslant r$, where

$$F = N(r-m), N = \frac{\binom{n}{t}}{\binom{r}{t}}, \alpha = \frac{\binom{n-1}{t-1}}{\binom{r-1}{t-1}}, k = n-m. \tag{7}$$

Note here that $N$ is the number of subsets in the Steiner system $S(t,r,n)$, and the integrality of $N$ and $\alpha$ follows from properties of Steiner systems [33].

**Construction 1.** **Precoding step:** We consider a Steiner system $S(t,r,n)$ and generate $N = \frac{\binom{n}{t}}{\binom{r}{t}}$ subsets, also referred to as blocks, such that each block is indexed by a set $J \in S(t,r,n)$. Block $J$ corresponds to $r-m$ information symbols over an alphabet of size $q$, which is then encoded using a universal MSMR code with length $r$ and dimension $r-m$ over an alphabet of size $q$. The MSMR codeword symbols, called the repair group $J$, is comprised of $\{c_{x,J} : x \in J\}$. The total number of information symbols is $F = N(r-m)$, each symbol being of size $\log_2 q$ bits.

**The code matrix:** The code structure can be described by a code matrix $C$, of size $n \times N$. The rows of $C$ are indexed by integers in $[n]$, corresponding to the different storage nodes, and its columns are indexed by sets in $S(t,r,n)$, arranged in some arbitrary chosen order. We formally define $C$ as

$$C_{x,J} = \begin{cases} c_{x,J}, & \text{if } x \in J, \\ -, & \text{otherwise,} \end{cases} \tag{8}$$

where "$-$" denotes an empty symbol. Node $i \in [n]$ stores all the non-empty symbols of row $i$ in the code matrix $C$. It can be checked that the storage per node is given by $\alpha = \frac{Nr}{n} = \frac{\binom{n-1}{t-1}}{\binom{r-1}{t-1}}$.

By abuse of notation, the terms block and repair group are used interchangeably. For each block, the universal MSMR code possesses the optimal repair bandwidth (3) for any number of erasures $l$, $1 \leqslant l \leqslant m$, and any number of helpers $d$, $r-m \leqslant d \leqslant r-l$. The requirement on the alphabet size $q$ is dictated by the existence of a universal MSMR code. Such MSMR codes are known to exist (see Section 1.2).

**Example 2.** Consider a Steiner system $S(t,r,n) = S(3,4,8)$. So the number of blocks is $N = 14$ and each node number appears in $\alpha = \frac{rN}{n} = 7$ blocks. The 14 blocks are given by

$$\begin{aligned} &J_1 = \{1,2,4,8\}, J_2 = \{2,3,5,8\}, J_3 = \{3,4,6,8\}, J_4 = \{4,5,7,8\}, J_5 = \{1,5,6,8\}, J_6 = \{2,6,7,8\}, \\ &J_7 = \{1,3,7,8\}, J_8 = \{3,5,6,7\}, J_9 = \{1,4,6,7\}, J_{10} = \{1,2,5,7\}, J_{11} = \{1,2,3,6\}, J_{12} = \{2,3,4,7\}, \\ &J_{13} = \{1,3,4,5\}, J_{14} = \{2,4,5,6\}. \end{aligned}$$

The code matrix is given by (9).

$$C = \begin{bmatrix} c_{1,J_1} & - & - & - & c_{1,J_5} & - & c_{1,J_7} & - & c_{1,J_9} & c_{1,J_{10}} & c_{1,J_{11}} & - & c_{1,J_{13}} & - \\ c_{2,J_1} & c_{2,J_2} & - & - & - & c_{2,J_6} & - & - & - & c_{2,J_{10}} & c_{2,J_{11}} & c_{2,J_{12}} & - & c_{2,J_{14}} \\ - & c_{3,J_2} & c_{3,J_3} & - & - & - & c_{3,J_7} & c_{3,J_8} & - & - & c_{3,J_{11}} & c_{3,J_{12}} & c_{3,J_{13}} & - \\ c_{4,J_1} & - & c_{4,J_3} & c_{4,J_4} & - & - & - & - & c_{4,J_9} & - & - & c_{4,J_{12}} & c_{4,J_{13}} & c_{4,J_{14}} \\ - & c_{5,J_2} & - & c_{5,J_4} & c_{5,J_5} & - & - & c_{5,J_8} & - & c_{5,J_{10}} & - & - & c_{5,J_{13}} & c_{5,J_{14}} \\ - & - & c_{6,J_3} & - & c_{6,J_5} & c_{6,J_6} & - & c_{6,J_8} & c_{6,J_9} & - & c_{6,J_{11}} & - & - & c_{6,J_{14}} \\ - & - & - & c_{7,J_4} & - & c_{7,J_6} & c_{7,J_7} & c_{7,J_8} & c_{7,J_9} & c_{7,J_{10}} & - & c_{7,J_{12}} & - & - \\ c_{8,J_1} & c_{8,J_2} & c_{8,J_3} & c_{8,J_4} & c_{8,J_5} & c_{8,J_6} & c_{8,J_7} & - & - & - & - & - & - & - \end{bmatrix}. \tag{9}$$

Let $m=2, e=2, d=n-e=6$. For each block, we use the EVENODD code defined in Table 1, so every symbol $c_{x,J} \in \mathbb{F}_2^2$, $q=4$. Then we can repair nodes 1 and 2 simultaneously, by downloading

- symbols $c_{4,J_1}, c_{8,J_1}$ from nodes 4 and 8, respectively. These help repair symbols $c_{1,J_1}$ and $c_{2,J_1}$,
- symbols $c_{5,J_{10}}, c_{7,J_{10}}$ from nodes 5 and 7, respectively. These help repair symbols $c_{1,J_{10}}$ and $c_{2,J_{10}}$,
- symbols $c_{3,J_{11}}, c_{6,J_{11}}$ from nodes 3 and 6, respectively. These help repair symbols $c_{1,J_{11}}$ and $c_{2,J_{11}}$,
- $\frac{1}{2}$ symbol from each of the nodes $5, 6$ and 8, to repair $c_{1,J_5}$,
- $\frac{1}{2}$ symbol from each of the nodes $3, 7$ and 8, to repair $c_{1,J_7}$,
- $\frac{1}{2}$ symbol from each of the nodes $4, 6$ and 7, to repair $c_{1,J_9}$,
- $\frac{1}{2}$ symbol from each of the nodes $3, 4$ and 5, to repair $c_{1,J_{13}}$,
- and similarly for node 2 to repair $c_{2,J_2}, c_{2,J_6}, c_{2,J_{12}}$ and $c_{2,J_{14}}$.

In total, we download 18 symbols. Each helper transmits 3 symbols.

From the example above, we see that each repair group $J$ tolerates the failure of $m$ nodes. Therefore, the code $C$ also tolerates the failure of up to any $m$ nodes. Thus, it can be checked that for Construction 1 from any $k=n-m$ nodes, we can recover the data, which is the *reconstruction parameter*. Namely, the code can recover from any $m$ failures.

Therefore, it is possible to repair any $e$ failures, for a flexible number of failures such that $1 \leqslant e \leqslant m$. The number of helpers is flexible, and satisfies $k \leqslant d \leqslant n-e$. In other words, Construction 1 possesses the *universality* of repair for flexible $e$ and $d$. The repair bandwidth is given in Propositions 1 and 2 for two different scenarios in the next two subsections.

### 2.1 Code construction with trivial Steiner systems

**Proposition 1.** Using Construction 1 with $t=r$, it is possible to repair simultaneously any set of $1 \leqslant e \leqslant m$ nodes, using $n-m \leqslant d \leqslant n-e$ helpers, such that the contribution of each helper, denoted by $\beta_e(d)$, is given by

$$\beta_e(d) = \sum_{s=1}^{e} \binom{e}{s} \sum_{p=\max(s,r-d)}^{\min(n-d-e+s,r-1)} \binom{d-1}{r-p-1} \binom{n-d-e}{p-s} \frac{s}{m-p+s}. \tag{10}$$

*Proof.* In the repair procedure, any subset of missing symbols belonging to the same repair group is repaired via MSMR repair procedure, using *all* available helpers from the same group among the chosen helper nodes. Fixing the set of helper nodes, we argue that the repair is feasible. Indeed, let $H$ be the set of $d$ helpers. For each repair group $J$, we denote the set of remaining nodes in $J$ as $J^{'}$. Using $|H \cup J^{'}| \leqslant n-e$ and $d \geqslant k = n-m$, it follows that

$$\begin{aligned} |J^{'} \cap H| &= |H| + |J^{'}| - |H \cup J^{'}| \\ &\geqslant d + r - e - (n-e) = r + d - n \\ &\geqslant r - n + n - m = r - m. \end{aligned} \tag{11}$$

Thus, for each repair group, we have enough information across the set of helpers to recover the missing components.

We now analyze the contribution of a single helper $h$: $h$ helps in the simultaneous repair of $s$ missing symbols of the same repair group, such that $1 \leqslant s \leqslant e$. For each size $s$, we count all possible cases in which the repair can be done through the help of $r-p$ coded symbols among all the $d$ helpers, because the number of available coded symbols determines the contribution of each helper, as dictated by the MSMR repair bandwidth (3). It follows that, for the corresponding repair group, $r-p-1$ can be chosen from the set of $d-1$ helpers (helper $h$ already belongs to the repair group by assumption), while the remaining $p-s$ elements of the repair group can be chosen from the remaining $n-e-d$ nodes. Figure 1 summarizes the repair situation for given parameters $s$ and $p$. Summing over all repair contributions, and analyzing the limit cases of $p$ for a given $s$, (10) follows. □

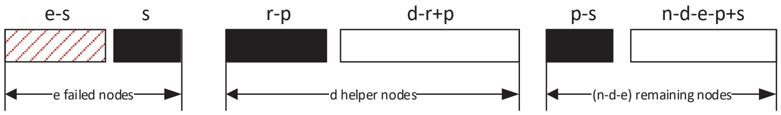

**Figure 1** A repair situation associated to given parameters $s$ and $p$.

**Example 3.** Consider a Steiner system $S(t,r,n)=S(4,4,5)$. So the number of blocks is $N=5$ and each node stores $\alpha=\frac{rN}{n}=5$ symbols. The 5 blocks are given by $J_1=\{2,3,4,5\}, J_2=\{1,3,4,5\}, J_3=\{1,2,4,5\}, J_4=\{1,2,3,5\}, J_5=\{1,2,3,4\}$. The code matrix is given by (12).

$$C=\begin{bmatrix} - & c_{1,J_2} & c_{1,J_3} & c_{1,J_4} & c_{1,J_5} \\ c_{2,J_1} & - & c_{2,J_3} & c_{2,J_4} & c_{2,J_5} \\ c_{3,J_1} & c_{3,J_2} & - & c_{3,J_4} & c_{3,J_5} \\ c_{4,J_1} & c_{4,J_2} & c_{4,J_3} & - & c_{4,J_5} \\ c_{5,J_1} & c_{5,J_2} & c_{5,J_3} & c_{5,J_4} & - \end{bmatrix}. \tag{12}$$

Let $m=2, e=1, d=3$. Then, $k=2$. Consider the repair of node 1 with the helper nodes $2,3,4$. We download: 1 symbol from each of the nodes 3 and 4 to repair $c_{1,J_2}$; 1 symbol from each of the nodes 2 and 4 to repair $c_{1,J_3}$; 1 symbol from each of the nodes 2 and 3 to repair $c_{1,J_4}$; $\frac{1}{2}$ symbol from each of the nodes $2,3$ and 4 to repair $c_{1,J_5}$. Each helper transmits $\frac{5}{2}$ symbols, as given by (10). The first three cases correspond to $p=2$, and the last case corresponds to $p=1$.

**Remark 2.** It can be seen that the repair procedure can benefit from the MSMR repair property in the case $n>k+1$. In particular, the advantages of using MSMR codes in our construction over maximum distance separable (MDS) codes as in [18] are: 1) lower repair bandwidth, 2) symmetric repair among helper nodes, which obviates the need for the expensive procedure of duplicating the block design in [18], and 3) universality, meaning non-trivial repair strategies for multiple erasures, $1\leqslant e\leqslant m$ with the help of varying number of helpers $d$, such that $n-m\leqslant d\leqslant n-e$. Figure 2 shows a comparison between the performance of the layered code in [18] and Construction 1, for an $(n,k,d,e)=(10,7,7,1)$ system. The MSMR repair property clearly helps reduce the bandwidth.

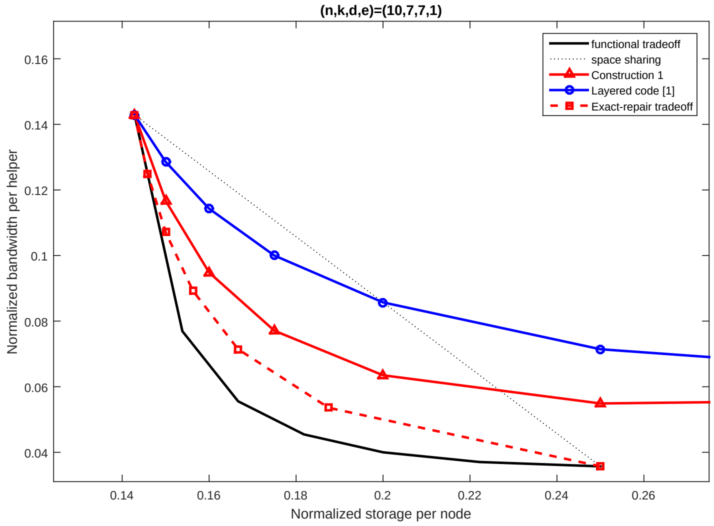

**Figure 2** Using the MSMR repair property improves upon the layered code repair performance. The exact-repair tradeoff is achieved in [13], with lower bound derived in [23].

**Example 4.** We note that for large $n$ and $n-k$, the complexity of designing universal MSMR may lead to large subpacketization size $\alpha$. However, for small $n$ and $n-k$, universal MSMR can be competitive and may not entail additional overhead, compared to using MDS codes as in [18].

Consider $(n, k, d) = (5, 3, 3), (r, m) = (4, 2)$. We compare the performance achieved by [18] and Construction 1, considering the repair of a single failure.

- The construction in [18, Figure 3] uses an MDS code with dimension 2 and length 4. Assume the MDS code is defined over a field of size 4, then, every codeword symbol of the MDS code corresponds to 2 bits. Then, the code generated in [18, Figure 3] achieves the following: $F = 60$ bits and $\alpha = 24$ bits. In the repair process, a replacement node downloads 48 bits; each helper transmits 16 bits. The code achieves $(\bar{\alpha}, \bar{\beta}) = (\frac{2}{5}, \frac{4}{5})$.
- In Construction 1, assume we use the EVENODD code (c.f. Table 1) as the underlying MSR code. Then, $F = 20$ bits, $\alpha = 8$ bits. In the repair process, a replacement node downloads 15 bits; each helper transmits 5 bits. The code achieves $(\bar{\alpha}, \bar{\beta}) = (\frac{2}{5}, \frac{1}{4})$.

Therefore, for the same $\bar{\alpha} = \frac{2}{5}$, Construction 1 achieves smaller subpacketization size and also strictly smaller normalized repair bandwidth $\bar{\beta}$.

**Remark 3.** The technique of using MSR codes as building blocks for outer code constructions has been used in the literature, for instance in constructing codes with local regeneration [34, 35] and exact-repair regenerating codes for single failure [19]. In particular, it can be checked that the construction in [19] uses copies of trivial Steiner systems, and achieves the same normalized points $(\bar{\alpha}, \bar{\beta})$ as Construction 1. However, the construction in [19] uses higher subpacketization size due to the necessity of code duplication in order to achieve symmetry across all nodes. Moreover, Construction 1 allows the use of general balanced incomplete block designs (c.f, Remark 7) in order to further reduce the subpacketization size. We note that determinant codes in [13] achieve the same normalized interior points as Construction 1 for an $(k+1, k, k, 1)$ system. Moreover, unlike Construction 1, determinant codes achieve the same set of normalized interior points for any number of nodes $n \geqslant k+1$, which is better than Construction 1. The authors in [17] provide a construction that improves upon [15] for certain parameter sets [17, Theorems 3.1, 3.2]. For a numerical example, for a $(9, 7, 8, 1)$ system, when $\bar{\alpha} < 0.1538$, Construction 1 outperforms [17], when $\bar{\alpha} > 0.1538$, [17] is better.

**Remark 4.** Using (3), it can be argued that $\beta_e(d)$ in (10) is decreasing in $d$. For fixed $n, k, m, e$, the minimum bandwidth per helper is then achieved with $d = n - e$, and is given by

$$\beta_e(n-e) = \sum_{s=1}^{e} \binom{e}{s} \binom{n-e-1}{r-s-1} \frac{s}{m} = \frac{e}{m} \sum_{s=1}^{e} \binom{e-1}{s-1} \binom{n-e-1}{r-s-1} = \frac{e}{m} \binom{n-2}{r-2}, \tag{13}$$

where the last equality follows from Vandermonde's identity. Moreover, we obtain after simplification

$$F = (r-2)\binom{n}{r}, \bar{\alpha} = \frac{r}{n(r-2)}, \bar{\beta} = \frac{e}{m} \frac{r(r-1)}{n(n-1)(r-2)}. \tag{14}$$

We argue in the next example that for some parameters, one can use a regenerating code corresponding to an interior point instead of an MSMR code as the inner code per repair group, and achieve the same performance.

**Example 5.** Consider the case $(n, k, d, e) = (5, 4, 4, 1)$. Let $r = t = 4, m = 1$ in (7). The code matrix is given by (12). Thus, the code per column of $C$ is of length $r = 4$ and its reconstruction parameter is $r - m = 3$. We use the interior code: $(\bar{\alpha}_0, \bar{\beta}_0) = (\frac{3}{8}, \frac{1}{4})$ per repair group. Let $F_0$ be the information size per column. Thus, $F = 5F_0$ and $\alpha = \frac{3F_0}{2}$. It follows that $\bar{\alpha} = \frac{3}{10}$. To repair node 1, we download a total bandwidth of $3F_0$. Thus, $\bar{\beta} = \frac{3}{20}$. We obtain the achievable point $(\bar{\alpha}, \bar{\beta}) = (\frac{3}{10}, \frac{3}{20})$. The same point is equally achievable using Construction 1 with $(t, r, n, m, e) = (3, 3, 5, 1, 1)$ with an MSMR code as the interior code. This point is optimal on the exact-repair tradeoff of the $(5, 4, 4, 1)$ system [13, 15], and is the optimal point next to the minimum bandwidth regenerating point.

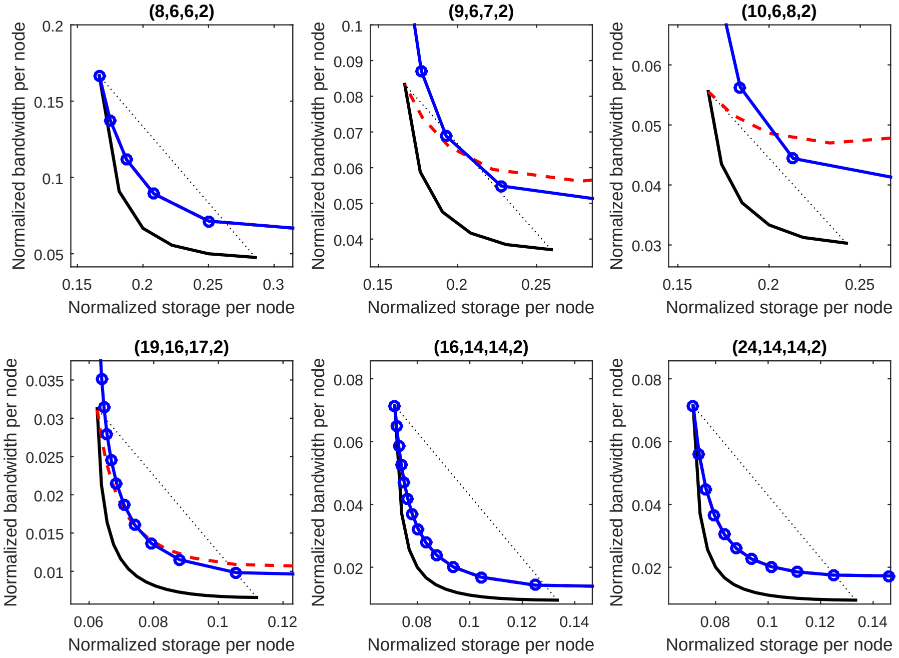

**Figure 3** Comparing achievable schemes for different scenarios. The dashed curves correspond to points achieved with Construction 1, and the solid curves (with open circles) corresponds to Construction 2, coupled with MSMR repair. The lowest solid curve corresponds to the functional repair tradeoff (1). The dotted line corresponds to space-sharing between the MSMR and the MBCR points.

**Remark 5.** In the scenario $d > k$, the authors in [18] presented a modified construction achieving a different set of interior points for a single failure. The repair scheme of the modified construction is however based on the naive repair scheme of MDS code. We do not describe the details here and refer the reader to [18]. We note that one can again use MSMR codes in the modified construction and the repair scheme of Proposition 1, and this leads to smaller repair bandwidth, for the same $F, \alpha$. We refer to this construction as Construction 2. Moreover, Construction 2 can be used to repair $e \leqslant n-k$ node failures and the repair bandwidth is given by (10).

**Example 6.** For two erasures, Figure 3 compares achievable points obtained using Construction 1, and achievable points obtained by Construction 2. The performance of both codes depends on the system parameters. When $d = k$, Construction 1 is identical to Construction 2. In most other cases, Construction 1 performs better closer to the MSMR point while Construction 2 performs better closer to the minimum bandwidth point.

## 2.2 Code construction with general Steiner systems

In Proposition 1, we considered Construction 1 with Steiner systems such that $t = r$. We study next the use of a general Steiner system for the specific $(n, n-m, n-e, e)$ system, such that $1 \leqslant e \leqslant m, d = n-e$.

**Proposition 2.** Construction 1 generates an $(F, n, n-m, n-e, e, \alpha, \beta)$ code such that during repair of $e \leqslant m$ nodes, each helper contributes

$$\beta_e = \frac{e}{m} \frac{\binom{n-2}{t-2}}{\binom{r-2}{t-2}}, \tag{15}$$

and the system satisfies

$$F = (r-2)\frac{\binom{n}{t}}{\binom{r}{t}}, \alpha = \frac{\binom{n-1}{t-1}}{\binom{r-1}{t-1}}, \bar{\alpha} = \frac{r}{n(r-2)}, \bar{\beta}_e = \frac{e}{m}\frac{r(r-1)}{n(n-1)(r-2)}. \tag{16}$$

*Proof.* We consider a Steiner system $S(t,r,n)$. Recall that $k=n-m$, which implies that the system can tolerate any $e\leqslant m$ failures. Let $\mathcal{E}=\{e_{i_1},\ldots,e_{i_e}\}$ denote a set of $e$ failed nodes. The set of helpers $\mathcal{H}$ consists of all the remaining nodes, $\mathcal{H}=[n]\backslash\mathcal{E}$. The repair proceeds by recovering the blocks which contain at least one failed node. Consider a helper $h\in\mathcal{H}$ and a failed node $i_j, j\in[e]$. By basic counting argument, nodes $h$ and $i_j$ share $\lambda_2\triangleq\frac{\binom{n-2}{t-2}}{\binom{r-2}{t-2}}$ blocks. Let $J$ be one of these $\lambda_2$ blocks, and denote by $s$ the number of failed nodes within $J$. Note that $1\leqslant s\leqslant e$. As the non-failed nodes within $J$ are helpers, the contribution of node $h$ in the recovery of block $J$ follows from (3), and is given by $\frac{s}{(r-s)-(r-m)+s}=\frac{s}{m}$ symbols. That is, the average amount of information $h$ contributes to the repair of $i_j$ is $\frac{1}{m}$. Thus, node $h$ contributes $\frac{\lambda_2}{m}$ symbols in the repair of node $i_j$ across all blocks containing $(h,i_j)$. Summing over all failed nodes, (15) follows. Using (7) and (15), (16) follows after simplification. □

The repair in Example 2 is an illustration of Proposition 2. Similar to Proposition 1, the repair bandwidth is identical among the helper nodes, and independent of the choice of the failed nodes and helpers. One can observe that for each helper, it transmits 1 symbol from 1 block, and two half symbols from 2 blocks. In particular, for helper $h$ and failure nodes $1,2$, the number of blocks containing any subset of $\{h,1,2\}$ is independent of the choice of $h$. Therefore, we observe this symmetry in the repair. In fact, such symmetry occurs when $t\geqslant e+1$ and is due to the fact that every subset of $t$ appears exactly once in the Steiner system.

**Remark 6.** We note here that $\bar{\alpha},\bar{\beta}$ do not depend on $t$ by (16). The advantage of using Steiner systems with smaller $t$, whenever they exist, is that they induce smaller $\alpha$ and $\beta$, for the same normalized parameters. Indeed, it can be shown that $\alpha$, as given by (16), is strictly increasing in $t$. Therefore, to reduce the storage size per node, and therefore the repair bandwidth, it is advantageous to use a Steiner System with the smallest $t$, $t\leqslant r$. Moreover, when $d=n-e, t=r$, Proposition 1 and Proposition 2 give the same $\bar{\alpha},\bar{\beta}$ (cf. Remark 4). Finally, the value of $\bar{\beta}$ for general Steiner systems and arbitrary $d>n-e$ is an open problem. We conjecture that the normalized bandwidth should be identical to that of Proposition 1.

**Example 7.** Consider a Steiner system $S(t,r,n)=S(2,4,13)$. Then, $N=13,\alpha=4$. The blocks are given by

$$J_1=\{1,2,4,10\},J_2=\{2,3,5,11\},J_3=\{3,4,6,12\},J_4=\{4,5,7,13\},J_5=\{1,5,6,8\},J_6=\{2,6,7,9\},$$
$$J_7=\{3,7,8,10\},J_8=\{4,8,9,11\},J_9=\{5,9,10,12\},J_{10}=\{6,10,11,13\},J_{11}=\{1,7,11,12\},$$
$$J_{12}=\{2,8,12,13\},J_{13}=\{1,3,9,13\}.$$

Let $m=2,e=2,d=n-e=11$. We use the EVENODD code in Table 1. Nodes 1 and 2 can be repaired simultaneously by downloading

- symbols $c_{4,J_1},c_{10,J_1}$ from nodes 4 and 10, respectively. These will help repair symbols $c_{1,J_1}$ and $c_{2,J_1}$.
- $\frac{1}{2}$ symbol from each of the nodes $5,6$ and 8, to repair $c_{1,J_5}$.
- $\frac{1}{2}$ symbol from each of the nodes $7,11$ and 12, to repair $c_{1,J_{11}}$.
- $\frac{1}{2}$ symbol from each of the nodes $3,9$ and 13, to repair $c_{1,J_{13}}$.
- $\frac{1}{2}$ symbol from each of the nodes $3,5$ and 11, to repair $c_{2,J_2}$.
- $\frac{1}{2}$ symbol from each of the nodes $6,7$ and 9, to repair $c_{2,J_6}$.
- $\frac{1}{2}$ symbol from each of the nodes $8,12$ and 13, to repair $c_{2,J_{12}}$.

While each helper transmits one symbol, the nature of the repair is different across helpers. For instance, node 4 transmits an entire symbol that contributes to the recovery of block $J_1$, while node 5 transmits two half symbols that contribute to the recovery of blocks $J_2$ and $J_5$, respectively. This scenario happens because $t\leqslant e$, which implies that some helpers may share a block with the set of $e$ failed nodes, while other helpers may not. In all scenarios, the repair bandwidth is identical across helpers, as shown in Proposition 2.

**Remark 7.** Construction 1 can be extended to general balanced incomplete block design (BIBD) systems. A BIBD system $BIBD(t,r,n,\lambda)$, $t\leqslant r\leqslant n$, is a collection of subsets of size $r$, included in $[n]$, such that any subset of $[n]$ of size $t$ appears exactly $\lambda$ times across all the subsets. In particular, a Steiner system is a BIBD system with $\lambda = 1$. When $t = r$, a BIBD system can simply be obtained by duplicating a Steiner system $S(t, r, n)$ $\lambda$ times. Therefore, in this case we restrict our attention to Steiner systems as they provide smaller storage cost, for the same parameters. Moreover, it can be seen that the proof of Proposition 2 holds for any BIBD system, with the only difference being in the number of blocks any two nodes share. For instance, for a $BIBD(t, r, n, \lambda)$ system, any two nodes share $\lambda\frac{\binom{n-2}{t-2}}{\binom{r-2}{t-2}}$ blocks [33] and we obtain

$$F = \lambda(r-2)\frac{\binom{n}{t}}{\binom{r}{t}}, \alpha = \lambda\frac{\binom{n-1}{t-1}}{\binom{r-1}{t-1}}, \beta_e = \lambda\frac{e}{m}\frac{\binom{n-2}{t-2}}{\binom{r-2}{t-2}}, \bar{\alpha} = \frac{r}{n(r-2)}, \bar{\beta}_e = \frac{e}{m}\frac{r(r-1)}{n(n-1)(r-2)}. \tag{17}$$

Therefore, when designing the code, for fixed parameters $n, r$, one can optimize over existing block designs and chooses the block design that results in the smallest $\alpha = \lambda\frac{\binom{n-1}{t-1}}{\binom{r-1}{t-1}}$. Recall that the trivial design always exists and satisfies $\lambda = 1, t = r$.

## 3 Analysis of the achievability for an $(n, k, d, e) = (k + e, k, k, e)$ system

In this section, we analyze the achievable region for an $(n, k, d, e) = (k + e, k, k, e)$ system by means of Construction 1, using a Steiner system $S(t, r, k + e)$ with the choice of $t \leqslant r$ as indicated in Remark 6.

**Corollary 1.** Construction 1 with $m = e$ generates a set of achievable points for an $(F, k+e, k, k, e, \alpha, \beta)$ system, such that

$$F = (r-2)\frac{\binom{k+e}{t}}{\binom{r}{t}}, \alpha = \frac{\binom{k+e-1}{t-1}}{\binom{r-1}{t-1}}, \beta = \frac{\binom{k+e-2}{t-2}}{\binom{r-1}{t-1}}, \tag{18}$$

$$\bar{\alpha} = \frac{r}{(k+e)(r-e)}, \bar{\beta} = \frac{r(r-1)}{(k+e)(k+e-1)(r-e)}, \quad e+1 \leqslant r \leqslant k+e. \tag{19}$$

*Proof.* follows from Proposition 2. □

### 3.1 Optimality of one achievable point

**Proposition 3.** For the $(k+e, k, k, e)$ system, the point achieved in (19) for $r = k+e-1$ is an optimal interior point.

*Proof.* From (18) when $r = k + e - 1$, we obtain

$$(\bar{\alpha}, \bar{\beta}) = (\frac{k+e-1}{(k+e)(k-1)}, \frac{k+e-2}{(k+e)(k-1)}). \tag{20}$$

Substituting (20) in (2) and setting $p = g - 2$, we obtain

$$(t + ge - 2e)\bar{\alpha} + (k - t - ge + 2e)\bar{\beta} = (k - e)\bar{\alpha} + e\bar{\beta} = 1.$$

Therefore, the above point lies on the functional repair lower bound and hence is optimal. It lies on the first segment of the bound near the MSMR point, and it is not the MSMR point nor the MBCR point, as indicated by (3) and (6). □

We note that the optimality of the point in Proposition 3 for the case of $(k+1, k, k, 1)$ was also shown in [18, Proposition 3]. Figure 4 illustrates the optimality of the point achieved by Proposition 3 for two different systems. The achievable point, corresponding to $(r, m) = (n - 1, e)$, lies on the functional tradeoff, which proves also its optimality under exact repair.

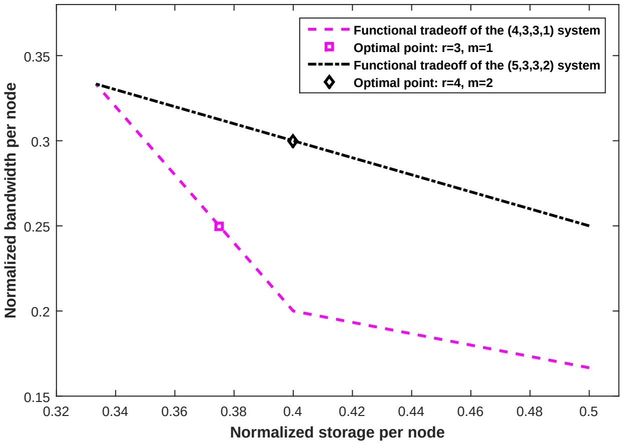

**Figure 4** Functional tradeoff and optimal points achieved by Proposition 3.

### 3.2 Optimal extension property

From Proposition 3, Construction 1 gives us an optimal point for any $(k+e, k, k, e)$ system. Construction 1 also offers the following optimal extension property.

**Proposition 4.** Consider a $(k+e, k, k, e)$ system and consider the optimal point achieved by Construction 1 in Proposition 3 with $t = r$, one can *extend* the system to a $(k+e+1, k, k, e+1)$ system, operating at the optimal point of Proposition 3, by adding another node to the system and increasing the storage per node, while keeping the initial storage content.

*Proof.* Let $\alpha_i, \beta_i, F_i$, for $i = 1, 2$, refer to the parameters of the old and the new systems, respectively. Then, $\alpha_2 - \alpha_1 = 1, \beta_2 - \beta_1 = 1, F_2 - F_1 = k - 1$. Moreover, the number of blocks $N$ is increased by 1. Let $k+e+1$ be the index of the new node to be added. The new code is obtained by simply adding another block, whose set is $\{1, \ldots, k+e\}$, and adding to each of the old sets the element $(k+e+1)$, and thus generating another coded symbol for the corresponding repair group. Note here that the dimension of the MSMR code in each block does not change. A key requirement is to assume the use of an MSMR code that can accommodate the addition of extra coded symbols, when needed. This can be done by choosing the number of symbols of the MSMR code to be as large as needed (this may result in an increase in the underlying field size). Each old node will store an extra symbol coming from the new repair group, while the new node stores the newly generated coded symbols from the old repair groups. □

**Example 8.** We illustrate the process of extending a $(4, 3, 3, 1)$ system to a $(5, 3, 3, 2)$ system. Initially, each repair group is of size 3. The code blocks are given by

$$J_1 = \{2, 3, 4\}, J_2 = \{1, 3, 4\}, J_3 = \{1, 2, 4\}, J_4 = \{1, 2, 3\}.$$

The code matrix is given by

$$C_1 = \begin{bmatrix} - & c_{1,J_2} & c_{1,J_3} & c_{1,J_4} \\ c_{2,J_1} & - & c_{2,J_3} & c_{2,J_4} \\ c_{3,J_1} & c_{3,J_2} & - & c_{3,J_4} \\ c_{4,J_1} & c_{4,J_2} & c_{4,J_3} & - \end{bmatrix}.$$

Adding node 5 to the system, we add another block $J_5 = \{1, 2, 3, 4\}$, whose symbols will be distributed across the old nodes $\{1, 2, 3, 4\}$. The old blocks become

$$J_1 = \{2, 3, 4, 5\}, J_2 = \{1, 3, 4, 5\}, J_3 = \{1, 2, 4, 5\}, J_4 = \{1, 2, 3, 5\}.$$

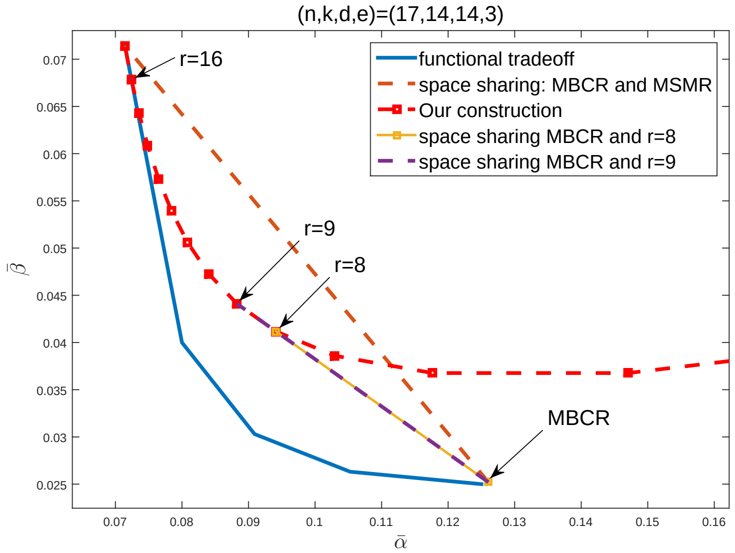

**Figure 5** Achievable points by Construction 1 for a $(n, k, d, e) = (17, 14, 14, 3)$ system. The x-axis is the normalized storage per node $\bar\alpha$ and the y-axis is the normalized bandwidth $\bar\beta$.

The new node 5 stores newly generated coded symbols of each of the old repair groups $\{J_1, \ldots, J_4\}$. The new code matrix is given by (12). The two points corresponding to this example are illustrated in Figure 4.

The above property is useful for systems for which the fault tolerance may be deemed insufficient. Therefore, one can increase the fault tolerance of the system without sacrificing the optimality on the exact-repair tradeoff, or changing the existing data. We note also that by a successive application of Proposition 4, we can increase the fault tolerance of the system by any desirable factor.

### 3.3 Acheivability region for the $(k + e, k, k, e)$ system

In this subsection, we seek to determine the convex hull of the known achievable points for the $(k+e, k, k, e)$ system, which corresponds to the best known achievable points. The convex hull, denoted by $\mathcal{R}$, is the smallest convex set containing all known achievable points, obtained by all convex combinations (i.e., space-sharing) among the points achieved by Construction 1, described in (19), and also the MBCR point given by (6). Therefore, the objective is to determine which points are sufficient to describe $\mathcal{R}$. We refer to these points as *corner points* of $\mathcal{R}$.

Figure 5 presents the achievable points for a $(17, 14, 14, 3)$ system. The achievable points of (18) are parameterized by $r$, such that $e + 1 \leqslant r \leqslant e + k$. For each $r$, we denote the corresponding point as $(\bar\alpha_r, \bar\beta_r)$. As $r$ decreases, the storage $\alpha_r$ increases. By abuse of notation, we refer to the point $(\bar\alpha_r, \bar\beta_r)$ as point $r$. We state some guiding observations for our subsequent analysis. First, one can eliminate some of the achievable points obtained by Construction 1. For instance, point $r = 5$, with $\bar\alpha = 0.1471$, achieves a similar bandwidth as its neighbor point $r = 6$, but a larger storage size. Points to the right of $\bar\alpha = 0.1471$, such that $r < 5$, can be also immediately eliminated, because they can be outperformed by space-sharing between the MBCR point and some interior point. Interestingly, we observe that point $r = 8$ lies exactly on the segment joining point $r = 9$ and the MBCR point. This means that, while point $r = 8$ is not outperformed by space-sharing, it is nonetheless not necessary for the description of $\mathcal{R}$, and thus it is not considered as a corner point. In the following, we show that the observations from Figure 5 can be generalized and we explicitly determine the corner points of $\mathcal{R}$, depending on the system parameters $e$ and $k$. Our characterization of the corner points is presented in Propositions 5 and 6. In order to prove them, we first prove some facts about the corner points in Lemmas 1 to 4.

**Lemma 1.** The achievable points in (18), with $r < 2e$, are not corner points in $\mathcal{R}$.

*Proof.* From (18), it can be seen that $\bar{\alpha}(r)$, seen as a function of $r$, is decreasing. $\bar{\beta}(r)$ is a fractional function in $r$, with a pole at $r = e$. For $r > e$, $\bar{\beta}(r)$ is convex in $r$. It can be shown that it decreases and then increases monotonically. Therefore, as $\bar{\alpha}(r)$ is decreasing, the points of interest are those for which $\bar{\beta}(r)$ increases. Moreover, by noticing that $\bar{\beta}(2e) = \bar{\beta}(2e-1)$, it follows that points with $r \leqslant 2e-1$ do not contribute to the acheivability region $(\bar{\alpha}, \bar{\beta})$ as they are outperformed by the point $r = 2e$ in terms of both, storage and bandwidth. □

Lemma 1 implies that it is sufficient to consider the range $2e \leqslant r \leqslant k+e$. We define the non-negative integer $p$ such that $r = 2e + p$. We now show that the achievable points $r = 2e, \ldots, k+e$ can not be eliminated by space-sharing between themselves, when not considering the MBCR point.

**Lemma 2.** The achievability region of the points $(\bar{\alpha}_r, \bar{\beta}_r), r = e+1, \ldots, k+e$ has points with $r \in \{2e, \ldots, k+e\}$ as corner points, when not considering the MBCR point.

*Proof.* By virtue of Lemma 1, points with $e+1 \leqslant r < 2e$ can be eliminated. We consider the segment joining the points $(\bar{\alpha}_r, \bar{\beta}_r)$ and $(\bar{\alpha}_{r+1}, \bar{\beta}_{r+1})$. The slope of the segment, denoted $sl(r)$, is given by

$$sl(r) = \frac{\bar{\beta}_{r+1} - \bar{\beta}_r}{\bar{\alpha}_{r+1} - \bar{\alpha}_r} = \frac{-r(r-2e+1)}{e(k+e-1)}.$$

The slope $sl(r)$ is strictly decreasing in $r$ for $r \geqslant 2e$. This means, for any three consecutive points $(\bar{\alpha}_{r+2}, \bar{\beta}_{r+2})$, $(\bar{\alpha}_{r+1}, \bar{\beta}_{r+1})$ and $(\bar{\alpha}_r, \bar{\beta}_r)$, the point $(\bar{\alpha}_{r+1}, \bar{\beta}_{r+1})$ lies below the segment joining the other two extreme points. Therefore, space-sharing between $(\bar{\alpha}_{r+2}, \bar{\beta}_{r+2})$ and $(\bar{\alpha}_r, \bar{\beta}_r)$ is suboptimal. □

Now, we analyze the achievability region when adjoining the MBCR point to the points in (18) with $2e \leqslant r \leqslant k+e$.

**Lemma 3.** The MBCR point is a corner point for $\mathcal{R}$ .

*Proof.* Noting that $\bar{\alpha}_{\mathrm{MBCR}} = \frac{2k+e-1}{(k+e)k} > \bar{\alpha}_{2e} = \frac{2}{(k+e)}$ and $\bar{\beta}_{\mathrm{MBCR}} = \frac{2e}{(k+e)k} < \bar{\beta}_{2e} = \frac{2(2e-1)}{(k+e)(k+e-1)}$, along with Lemma 2 concludes the result. □

By Lemma 3, we only need to analyze whether space-sharing between the MBCR point and any other point $r$ may outperform some of the other achievable points $r^{'}$.

**Lemma 4.** If a point $r$, $r \geqslant 2e$, is not outperformed by space-sharing between the point $r+1$ and the MBCR point, then, all points $r'$ such that $r' \geqslant r$, are corner points of the achievability region.

*Proof.* The assumption of the lemma implies that the slope of the segment joining the points $(\bar{\alpha}_r, \bar{\beta}_r)$ and $(\bar{\alpha}_{\mathrm{MBCR}}, \bar{\beta}_{\mathrm{MBCR}})$ is smaller than the slope of the segment between $(\bar{\alpha}_{r+1}, \bar{\beta}_{r+1})$ and $(\bar{\alpha}_{\mathrm{MBCR}}, \bar{\beta}_{\mathrm{MBCR}})$. As from Lemma 2, the slope of the segment between $(\bar{\alpha}_r, \bar{\beta}_r)$ and $(\bar{\alpha}_{r+1}, \bar{\beta}_{r+1})$ is decreasing in $r$, it follows that no point $r' \geqslant r$ can be outperformed by space-sharing across any two other achievable points, including the MBCR point. □

Therefore, to determine the corner points of $\mathcal{R}$, we need to successively test for increasing values of $p$, such that $0 \leqslant p \leqslant k-e$, whether the point $r = 2e+p$ is outperformed by space-sharing of MBCR and point $r+1$. Let $p^*$ denote the smallest $p$ such that $r = 2e+p$ is not outperformed by space-sharing, it follows by Lemma 4 the following achievability region.

**Proposition 5.** The achievability region $\mathcal{R}$ is given by the corner points

$$\{(\bar{\alpha}_r, \bar{\beta}_r) : r \in \{r : r = 2e+p \text{ and } p^* \leqslant p \leqslant k-e\}\} \cup \{\mathrm{MBCR}\}, \tag{21}$$

where $1 \leqslant p^* \leqslant k-e$, and $p^*$ is given by

$$\begin{aligned} p^* &= \left\lfloor \frac{e-k-2e^2+1+\sqrt{\Delta}}{2(e+k-1)} \right\rfloor + 1, \\ \Delta &= (2e^2-e+k-1)^2 + 8(k+e-1)e(e-1)(k-e-1). \end{aligned} \tag{22}$$

*Proof.* Consider $r = 2e+p, 0 \leqslant p \leqslant k-e-1$. We consider space-sharing between the MBCR point and the point $r+1$. We compute the normalized bandwidth, denoted by ${\bar{\beta}'}_r$, achieved by the considered space-sharing, at the intermediate point $\alpha = \alpha_r$, and then determine whether $\bar{\beta}'_r > \bar{\beta}_r$. Using (19) and (6), we obtain after simplification

$$\bar{\beta}'_r - \bar{\beta}_r = \frac{k(-2e^2+2e+p^2+p) - p(-2e^2+e+1) + 2e(e^2-1) + p^2(e-1)}{(e+k)(e+p)(e+k-1)(e^2+pe+k-p+kp-1)} \triangleq \frac{N_1(k)}{D} \tag{23}$$

$$= \frac{(k+e-1)p^2 + p(2e^2+k-e-1) + 2e(e-1)(e+1-k)}{(e+k)(e+p)(e+k-1)(e^2+pe+k-p+kp-1)} \triangleq \frac{N_2(p)}{D}. \tag{24}$$

We regard $N_1$ as a function of $k$, for fixed $e$ and $p$, and $N_2$ as a function of $p$, for fixed $e$ and $k$. In this proof, we are interested in analyzing $N_2$. We analyze $N_1$ in a later proof.

Clearly $D > 0$. Thus, $\mathrm{sign}(\bar{\beta}'_r - \bar{\beta}_r) = \mathrm{sign}(N_2(p))$. Therefore, it suffices to study the sign of $N_2(p)$. We note that $\bar{\beta}'_r - \bar{\beta}_r \leqslant 0$ implies that point $r = 2e + p$ can be eliminated by space-sharing and thus it is a not a corner point. $N_2(p)$ is a quadratic function in $p$. Let $\Delta$ denote the discriminant of $N_2(p)$. It can be checked that

$$\Delta = (2e^2 - e + k - 1)^2 + 8(k + e - 1)e(e - 1)(k - e - 1) > 0.$$

Thus, there exists $p_{0,1}, p_{0,2}$ such that $N_2(p_{0,1}) = N_2(p_{0,2}) = 0$. As the leading coefficient of $N_2(p)$ is positive, and $N_2(0) = -2e(e-1)(k-e-1) \leqslant 0$, it follows that one solution, say $p_{0,1}$, is negative and the other solution $p_{0,2}$ is non-negative. That is, $p_{0,1} < 0$ and $p_{0,2} \geqslant 0$. Then, it follows that $\forall 0 \leqslant p \leqslant p_{0,2}, N_2(p) \leqslant 0$, which implies that the set $\{p : p \leqslant p_{0,2}\}$ can be eliminated. In particular, $p = 0$ is always eliminated. Let $p^* = \lfloor p_{0,2} \rfloor + 1$, as in (22). Thus, $p^*$ outperforms space-sharing and so do all $p \geqslant p^*$. As $N_2(k-e-1) = (k-e)(k+e-1)(k-e-1) \geqslant 0$, it follows that $p_{0,2} \leqslant k - e - 1$, and thus $p^* \leqslant k - e$. □

Proposition 5 agrees with known particular cases. 1) When $e = 1$, we have $p^* = 1$ and the only eliminated point ($p = 0$) coincides with the MBCR point, in agreement with [18]. 2) The optimal point in Proposition 3 ($p = k - e - 1$) is not a corner point for $k = e + 1$, because of $p^* = k - e > p$ and Proposition 5. Indeed, the point with $p = k - e - 1$ lies exactly on the segment joining the MBCR and the MSMR point. 3) When $k > e + 1$, the optimal point in Proposition 3 is a corner point, as $\bar{\beta}'_{k+e-1} - \bar{\beta}_{k+e-1} = \frac{(k-e)(k-e-1)}{k(k+e)(k-1)^2} > 0$.

While Proposition 5 describes exactly $\mathcal{R}$, it does not give insight into when a particular point $r = 2e+p$ is a corner point or not. We focus on the analysis of the sign of $N_1(k)$ in (23). $N_1(k)$ is linear in $k$. Depending on the sign of its the leading coefficient $-2e^2 + 2e + p^2 + p$, there may exist an integer $k_{\mathrm{th}}$ such that when $k \geqslant k_{\mathrm{th}}$ space-sharing enhances the achievability region (i.e., $N_1(k) \leqslant 0$ ) and does not enhance it when $k < k_{\mathrm{th}}$. That is, a point with the same $r$ may be a corner point for some $(k+e, k, k, e)$ systems and may be not a corner point for other systems, with higher reconstruction parameter $k$.

For example, for $e > 1$, let $p = e - 1$, we have $N_1(k) = e(1-e)(k - 5e + 1)$. It follows that, for systems with $k \geqslant 5e - 1$, the point $r = 2e + (e-1) = 3e - 1$ is outperformed by space-sharing. For systems with $2e - 1 \leqslant k < 5e - 1$, the point $r = 3e - 1$ is a corner point.

The next proposition addresses the cases in which a particular point $r = 2e+p$ is a corner point, using a similar argument as the above example.

**Proposition 6.** Consider the achievable point $r = 2e + p$, for fixed $(e, k), e > 1$. Let $p_{\max} = \left\lfloor \frac{1}{2}(\sqrt{8e(e-1)-1} - 1) \right\rfloor$ and $k_{\mathrm{th}}(p) = \left\lceil (1-e)\frac{\binom{p+1}{2} + 2\binom{e+1}{2} + ep}{\binom{p+1}{2} - 2\binom{e}{2}} \right\rceil$. Then, Table 2 specifies the scenarios in which $(\bar{\alpha}_r, \bar{\beta}_r)$ is a corner point in $\mathcal{R}$.

| $(\bar{\alpha}_r, \bar{\beta}_r)$ | $k < k_{\mathrm{th}}(p)$ | $k \geqslant k_{\mathrm{th}}(p)$ |
|---|---|---|
| $p \leqslant p_{\max}$ | ✓ | ✗ |
| $p > p_{\max}$ | ✓ | |

**Table 2** Summary of cases for which $(\bar{\alpha}_r, \bar{\beta}_r)$ is a corner point in $\mathcal{R}$. The symbol ✓ means $(\bar{\alpha}_r, \bar{\beta}_r)$ is a corner point while the symbol ✗ denotes the other case.

*Proof.* We examine $N_1(k)$. First, we note that when $-2e^2+2e+p^2+p>0$, the point $r=2e+p$ is a corner point for all systems. Indeed, as $N_1(e+1)=2ep(e+p)>0, p>0$, we have $N_1(k)>0, \forall k \geqslant e+1, p>0$. It follows that, for a fixed $(e,p)$, we need to determine the sign of $-2e^2+2e+p^2+p$. We have

$$-2e^2+2e+p^2+p<0 \iff p(p+1)<2e(e+1) \iff \binom{p+1}{2}<2\binom{e}{2}, \tag{25}$$

$$\iff p<\sqrt{2e^2-2e-\frac{1}{4}}-\frac{1}{2}=\frac{1}{2}(\sqrt{8e(e-1)-1}-1). \tag{26}$$

We note that RHS of (26) can not be an integer, as otherwise $\sqrt{8e(e-1)-1}$ should be an odd integer, implying $8e(e-1)-1 \equiv 1 \mod 4$, which leads to a contradiction as $8e(e-1)-1 \equiv 3 \mod 4$. This also implies that the slope of $N_1(k)$ cannot be 0, for $e>0, \forall p \geqslant 0$. The maximum value of $p$ satisfying (26) is given by

$$p_{\max}=\left\lfloor \frac{1}{2}(\sqrt{8e(e-1)-1}-1)\right\rfloor. \tag{27}$$

Thus, a point $r=2e+p, p>p_{\max}$ is a corner point for any $(k+e,k,k,e)$ system such that $r \leqslant k+e$. For each $0 \leqslant p \leqslant p_{\max}$, the point $r=2e+p$ is a corner point if and only if $\text{sign}(\bar{\beta'}_r-\bar{\beta}_r)=\text{sign}(N_2(k))>0$. From (23), Let $k_0$ be the solution to the linear equation $N_1(k)=0$. Then, after simplification, we have

$$k_0=\frac{(1-e)(2e^2+2ep+2e+p^2+p)}{-2e^2+2e+p^2+p}=(1-e)\frac{\binom{p+1}{2}+2\binom{e+1}{2}+ep}{\binom{p+1}{2}-2\binom{e}{2}}. \tag{28}$$

As $p \leqslant p_{\max}$, we have $-2e^2+2e+p^2+p<0$, which also implies that $k_0>0$. As $N_1(e+1)=2ep(e+p)$, we have $k_{\text{th}} \geqslant e+1$, with equality iff $p=0$. It can be checked from (28) that when $p=e-1$, $k_0=5e-1$. For $k \geqslant k_0$, point $r$ is not a corner point. As $k$ is an integer and $k_0$ is not necessarily an integer, it follows that $k \geqslant k_0 \iff k \geqslant \lceil k_0 \rceil \triangleq k_{\text{th}}$. □

Using Proposition 6, Corollary 2 follows.

**Corollary 2.** For a $(k+e,k,k,e)$ system with $e \geqslant 2$, we have

- $p^*$ in (22) can also be expressed as

$$p^*=1+\max\{p : p \leqslant p_{max} \text{ and } k \geqslant k_{\text{th}}(p)\} \tag{29}$$

$$=1+\max\left\{p : p \leqslant \left\lfloor \frac{1}{2}(\sqrt{8e(e-1)-1}-1)\right\rfloor \text{ and } k \geqslant \left\lceil (1-e)\frac{\binom{p+1}{2}+2\binom{e+1}{2}+ep}{\binom{p+1}{2}-2\binom{e}{2}}\right\rceil\right\}. \tag{30}$$

- The number of corner points in $\mathcal{R}$ is given by $n_c \triangleq |\{r : 2e+p^* \leqslant r \leqslant k+e\}|+1=k-e+2-p^*$.
- As a function of $k$, $p^*$ levels out at $k=k_{\text{th}}(p_{max})$ and its final value is given by $1+p_{max}$.

**Example 9.** We consider the setting of Figure 5: $e=3, k=14$. We obtain $p_{max}=2, p^*=3$. This means the points $r$, for $6 \leqslant r \leqslant 2e+p^*-1=8$ are not corner points in $\mathcal{R}$ and the number of corner points is $n_c=10$. This clearly matches the observations made in Figure 5.

## 4 Conclusion

We studied the problem of centralized exact repair of multiple failures in distributed storage. We first described a construction that achieves a new set of interior points. In case all non-failed nodes participate in the repair process, we showed that the use of a general Steiner system may reduce the storage size as well as the repair bandwidth, compared to a trivial Steiner system for the same normalized achievable point by our construction. For the $(k+e,k,k,e)$ system, we proved the optimality of one point on the functional centralized repair tadeoff. Moreover, considering minimum bandwidth cooperative repair codes as centralized repair codes, we determined explicitly the best achievable region obtained by space-sharing among all known points.

One future work is the analysis of the repair bandwidth of our construction for general Steiner systems and arbitrary $d \leqslant n-e$. Our conjecture is that the normalized bandwidth is identical to the trivial Steiner systems. Another future direction is to investigate outer bounds for the centralized exact-repair problem. In particular, it is an open problem to understand the minimum bandwidth point under centralized exact repair.

## References

1 Wang Z, Tamo I, Bruck J. Optimal rebuilding of multiple erasures in MDS codes. IEEE Trans. Inf. Theory, 2017, 63:1084–1101

2 Dimakis A G, Godfrey P, Yunnan W, et al. Network coding for distributed storage systems. IEEE Trans. Inf. Theory, 2010, 9: 4539–4551

3 Rashmi K V, Shah N B, Kumar PV. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. IEEE Trans. Inf. Theory, 2012, 57: 5227-39

4 Shah N B, Rashmi K, Kumar P V, et al. Interference alignment in regenerating codes for distributed storage: Necessity and code constructions. IEEE Trans. Inf. Theory, 2012, 58: 2134–2158

5 Suh C, Ramchandran K. Exact-repair MDS code construction using interference alignment. IEEE Trans. Inf. Theory, 2011, 57: 1425–1442

6 Rawat A S, Koyluoglu O O, Vishwanath S. Progress on high-rate MSR codes: Enabling arbitrary number of helper nodes. In: Information Theory and Applications Workshop, San Diego, CA, USA, 2016. 1–6

7 Goparaju S, Fazeli A, Vardy A. Minimum storage regenerating codes for all parameters. IEEE Trans. Inf. Theory, 2017, 63: 6318–6328

8 Cadambe V R, Jafar S A, Maleki H, et al. Asymptotic interference alignment for optimal repair of MDS codes in distributed storage. IEEE Trans. Inf. Theory, 2013, 59: 2974–2987

9 Tamo I, Wang Z, Bruck J. Zigzag codes: MDS array codes with optimal rebuilding. IEEE Trans. Inf. Theory, 2013, 59: 1597–1616

10 Ye M, Barg A. Explicit constructions of high-rate MDS array codes with optimal repair bandwidth. IEEE Trans. Inf. Theory, 2017, 63: 2001–2014

11 Vajha M, Babu BS, Kumar PV. Explicit MSR Codes with Optimal Access, Optimal Sub-Packetization and Small Field Size for $d = k + 1, k + 2, k + 3$. arXiv preprint arXiv: 1804.00598, 2018

12 Shah N B, Rashmi K V, Kumar P V, et al. Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff. IEEE Trans. Inf. Theory, 2012, 58: 1837-52

13 Elyasi M, Mohajer S. Determinant coding: A novel framework for exact-repair regenerating codes. IEEE Trans. Inf. Theory, 2016, 62: 6683–6697

14 Elyasi M, Mohajer S. A probabilistic approach towards exact-repair regeneration codes. In: Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2015. 865-872

15 Tian C. A note on the rate region of exact-repair regenerating codes. arXiv preprint arXiv: 1503.00011, 2015

16 Tian C. Characterizing the rate region of the $(4, 3, 3)$ exact-repair regenerating codes. IEEE Journal on Selected Areas in Communications, 2014, 32: 967–975

17 Senthoor K, Sasidharan B, Kumar PV. Improved layered regenerating codes characterizing the exact-repair storage-repair bandwidth tradeoff for certain parameter sets. In: IEEE Information Theory Workshop, Jerusalem, Israel, 2015. 1-5

18 Tian C, Sasidharan B, Aggarwal V, et al. Layered exact-repair regenerating codes via embedded error correction and block designs. IEEE Trans. Inf. Theory, 2015, 61: 1933–1947

19 Goparaju S, El Rouayheb S, Calderbank R. New codes and inner bounds for exact repair in distributed storage systems. In: IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014. 1036–1040

20 Sasidharan B, Senthoor K, Kumar P V. An improved outer bound on the storage-repair-bandwidth tradeoff of exact-repair regenerating codes. In: IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014. 2430–2434

21 Duursma I M. Outer bounds for exact repair codes. arXiv preprint arXiv:1406.4852, 2014

22 Sasidharan B, Prakash N, Krishnan M N, et al. Outer bounds on the storage-repair bandwidth trade-off of exact-repair regenerating codes. International Journal of Information and Coding Theory, 2016, 3: 255–298

23 Duursma I M. Shortened regenerating codes. IEEE Trans. Inf. Theory, 2018

24 Kermarrec A M, Le Scouarnec N,Straub G. Repairing multiple failures with coordinated and adaptive regenerating codes. In: International Symposium on Network Coding, Beijing, China, 2011. 1–6

25 Rawat A S,Koyluoglu O O, Vishwanath S. Centralized repair of multiple node failures with applications to communication efficient secret sharing. arXiv preprint arXiv:1603.04822, 2016

26 Shum K W, Hu Y. Cooperative regenerating codes. IEEE Trans. Inf. Theory, 2013, 59: 7229–7258

27 Zorgui M, Wang Z. Centralized multi-node repair in distributed storage. In: Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2016. 617–624.

28 Zorgui M, Wang Z. Centralized multi-node repair for minimum storage regenerating codes. In: IEEE International Symposium on Information Theory, Aachen, Germany, 2017. 2213–2217

29 Li J, Li B. Cooperative repair with minimum-storage regenerating codes for distributed storage. In: IEEE INFOCOM, Toronto, ON, Canada, 2014. 316–324

30 Ye M, Barg A. Optimal MDS codes for cooperative repair. arXiv preprint arXiv:1801.09665, 2018

31 Wang A, Zhang Z. Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage. In: IEEE INFOCOM, Turin, Italy, 2013. 400–404.

32 Keevash P. The existence of designs. arXiv preprint arXiv:1401.3665, 2014

33 Colbourn CJ. CRC handbook of combinatorial designs. CRC press, 2010

34 Kamath G M, Prakash N, Lalitha V, et al. Codes with local regeneration and erasure correction. IEEE Trans. Inf. Theory, 2014, 60: 4637–4660

35 Rawat A S, Silberstein N, Koyluoglu O O, et al. Optimal locally repairable codes with local minimum storage regeneration via rank-metric codes. In: Information Theory and Applications Workshop, San Diego, CA, USA, 2013. 1–8

36 Zorgui M, Wang Z. Centralized Multi-Node Repair Regenerating Codes. arXiv preprint arXiv:1706.05431, 2017

37 Blaum M, Brady J, Bruck J, et al. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. IEEE Transactions on computers, 1995, 44: 192-202