IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 June 2023, accepted 19 July 2023, date of publication 27 July 2023, date of current version 18 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299502

RESEARCH ARTICLE

New Storage Codes Between the MSR and MBR Points Through Block Designs

XIAOFANG WANG AND YUAN LIAO
School of Intelligence Technology, Geely University of China, Chengdu 641423, China
Corresponding author: Yuan Liao (liaoyuan@bgu.edu.cn)

ABSTRACT In distributed storage systems, data are stored across multiple storage nodes which are unreliable and prone to failure. While erasure coding is more efficient than simple replication in terms of storage overhead and reliability, classic erasure codes like Reed-Solomon codes require a large repair bandwidth when repairing a failed node. Therefore, reducing both the storage overhead and repair bandwidth under a given fault tolerance is desired, however, it is not possible to minimize both. In 2007, Dimarkis et al. characterized the storage-bandwidth trade-off under functional repair. While exact repair is preferred in practical systems, it was shown that only two extremal points and a line segment are achievable under exact repair. Up to now, the storage-bandwidth trade-off under exact repair remains unresolved for general parameters. Nevertheless, constructing codes with exact repair between the two extremal points is still of great interest, however, very few such constructions have been reported in the literature. In this paper, we present explicit code constructions based on block designs, which can be viewed as a generalization of a previous work by Tian et al. Such a generalization leads to two new codes, i.e., an $(n, k = n-1, d = n-1)$ storage code based on regular mandatory representation designs (MRDs) and an $(n, k = n-2, d \geq n-2)$ storage code based on 3-designs. It is shown that the new storage codes have a better performance than the ones by Tian et al. in terms of the sub-packetization level and storage-bandwidth trade-off. In addition, the new $(n, k = n-2, d)$ storage code supports two repair degrees, i.e., $d \in \{n-2, n-1\}$.

INDEX TERMS Block designs, distributed storage, interior points, minimum bandwidth regenerating codes, minimum storage regenerating codes.

I. INTRODUCTION

Distributed storage systems are widely deployed in large data centers, such as Google File System [1], Facebook Distributed File System [2], Microsoft Azure [3], and also peer-to-peer storage settings, such as DHash++ [4], OceanStore [5], and Total Recall [6]. In a distributed storage system, data are stored across multiple storage nodes that are unreliable and prone to failure. To ensure reliability in the presence of node failures, redundancy needs to be introduced. Replication is a traditional mechanism for introducing redundancy, but it is inefficient in terms of storage overhead as the amount of data is increasing rapidly.

Erasure coding is a more efficient alternative, with maximum distance separable (MDS) codes being an example that achieves the optimal trade-off between storage overhead and fault tolerance. Consider an original file that comprises k symbols over a finite field $\mathbf{F}_q$, by calling upon an (n, k) MDS code, we get n coded symbols such that any k out of the n symbols can recover the original file. These n coded symbols can then be stored across a distributed storage system of n storage nodes. However, in case of a node failure, the entire data needs to be downloaded from any k surviving nodes, leading to a large *repair bandwidth* γ, which is defined as the amount of data downloaded to regenerate a failed node.

While it is desirable to reduce both the storage overhead and repair bandwidth for a given fault tolerance, it is not possible to minimize both. In the pioneering work in [7], Dimakis et al. characterized the trade-off between the storage overhead α (i.e., the amount of data stored in each node) and the repair bandwidth γ under a symmetric setup, where

P1. Each node stores α symbols;

The associate editor coordinating the review of this manuscript and approving it for publication was Adnan Abid.

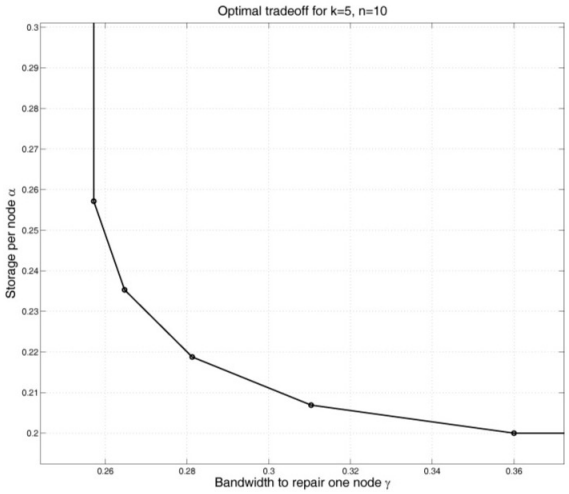

FIGURE 1. Optimal tradeoff curve between storage α and repair bandwidth γ, for $k = 5$, $d = 9$, and $\mathcal{M} = 1$.

P2. A failed node can be repaired by connecting any d surviving nodes, and each of the d nodes transmits the same amount of information, i.e., β symbols;

where d is usually referred to as the *repair degree* in the literature. In addition, to guarantee fault tolerance, it is required that

P3. The file can be reconstructed by connecting any k out of the n nodes.

Consider a file of size $\mathcal{M}$ that is encoded by an (n, k, d) storage code, under the aforementioned settings, it was proved in [7] that α, β, and the file size $\mathcal{M}$ should satisfy

$$\mathcal{M} \leq \sum_{i=0}^{k-1} \min(\alpha, (d-i)\beta). \tag{1}$$

Codes that attain the above bound with equality are referred to as regenerating codes in [7]. For fixed values of parameters $\mathcal{M}, k, d$, there are multiple pairs (α, β) that satisfy (1) with equality. This leads to the storage-bandwidth trade-off which is piece-wise linear, see Fig. 1 for an example [7]. The existence of regenerating codes that can achieve any point on the storage-bandwidth trade-off under functional repair was also shown in [7], where under functional repair the code symbols in the new replacement node can be different from that in the failed node as long as P1–P3 continue to hold. On the optimal trade-off, two extremal points are of particular interest, i.e., the Minimum Storage Regeneration (MSR) and Minimum Bandwidth Regeneration (MBR) points. MSR points are achieved by first minimizing the storage overhead and then the repair bandwidth, while MBR points are achieved on the contrary. The intermediate points between the two extremal points on the curve will be referred to as FR-interior points. Note that exact repair is desired in practical systems, and it is a natural question whether the optimal storage-bandwidth trade-off can be achieved under exact repair.

It has been shown that the two extremal points can be achieved under exact repair and there are abundant constructions [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. Further in [30], it was shown that the line segment from the MSR point to the next deflection point is achievable while the other interior points on the optimal trade-off are not achievable under exact repair. In [31], through a computer-aided approach, Tian completely characterized the trade-off for $(n, k, d) = (4, 3, 3)$, which showed that there is a non-vanishing gap between the functional repair trade-off and exact repair trade-off. Since then, there are several works focusing on addressing the optimal storage-bandwidth trade-off under exact repair [32], [33], [34], [35], [36] and exact-repair code constructions between the MSR and MBR points [37], [38], [39], [40]. However, the optimal trade-off under exact repair was only investigated under very restricted parameters. In [32], [33], and [34], bounds among $\mathcal{M}, k$, and d are given under the condition of exact repair for $d = k = n-1$. These bounds demonstrate that the file size $\mathcal{M}$ of the $(n, k = n-1, d = n-1)$ storage code is upper bounded by

$$\mathcal{M} \leq \begin{cases} \lfloor \frac{s(s-1)n\alpha + n(n-1)\beta}{s^2+s} \rfloor, & \frac{d\beta}{s} \leq \alpha \leq \frac{d\beta}{s-1}, \\ & 2 \leq s \leq n-2 \\ (n-2)\alpha + \beta, & \frac{d\beta}{n-1} \leq \alpha \leq \frac{d\beta}{n-2}. \end{cases} \tag{2}$$

The above bound implies that the optimal storage- bandwidth trade-off curve of $(n, k = n-1, d = n-1)$ regenerating codes is also piece-wise linear, with the k corner points satisfying

$$(\bar{\alpha}_i, \bar{\beta}_i) = \left(\frac{i+1}{in}, \frac{i+1}{n(n-1)} \right), \tag{3}$$

where $i = 1, 2, \ldots, k$.

An n-independent achievable optimal trade-off under exact repair was provided in [35] for the case of $d = k$, where it was shown that the first corner point on the trade-off next to the MSR point can be achieved for $d = k$. However, the proof of [35] was just an existence proof, no explicit constructions were given. Up to now, the optimal storage-bandwidth trade-off under exact repair remains open for general parameters $\mathcal{M}, k$, and d.

On the other hand, there are several works that have been dedicated to constructing exact repair storage codes between the MSR and MBR points [36], [37], [38], [40]. In [38], codes between the MSR and MBR points are constructed through block designs, more specifically, by Steiner systems, balanced incomplete block designs (BIBDs). However, explicit constructions were only provided for the parameters $(n, k = n-1, d = n-1)$ and $(n, k = n-2, d = n-1)$. Notably, the $(n, k = n-1, d = n-1)$ code construction is optimal w.r.t. to bound in (2). In [37], $(n' = n + l, k' = k + l, d' = d + l)$ codes between the MSR and MBR points are constructed by taking a known (n, k, d) code as building blocks, i.e., distributing n coded symbols across $n + l$ nodes, it requires to glue $(n + l)!$ permutated copies of the (n, k, d) code to satisfy P1 and P2, which leads to a huge α. In [40], codes are constructed by combining several determinant codes in [39],

a process that also results in a large α. Nevertheless, the $(n, k = n-1, d = n-1)$ code is optimal w.r.t. to bound in (2), while for the other parameters they were conjectured to be optimal in terms of the storage-bandwidth trade-off under exact repair. The parameter α is also referred to as sub-packetization level in [41], which suggests that codes with a large sub-packetization level can lead to reduced design space in terms of various system parameters and make the management of meta-data difficult. As a result, it hinders the implementation of practical systems. Some recent works for codes between the two extremal points were proposed for heterogeneous distributed storage systems [42], [43].

Although the $(n, k = n-1, d = n-1)$ codes proposed in [38] and [40] are optimal w.r.t. to the bound in (2), they can only achieve the k corner points on the optimal storage-bandwidth trade-off curve. To achieve the other points on the curve, space-sharing is necessary, which results in an enlargement of the sub-packetization level. In this paper, motivated by the idea in [38], we present explicit code constructions with parameters $(n, k = n-1, d = n-1)$ and $(n, k = n-2, d \geq n-2)$ based on block designs, more specifically, by regular MRDs and 3-designs. Thus the code constructions in this paper can be viewed as a generalization of the work in [38]. Such a generalization leads to the two new codes with the following advantages:

- We propose a novel approach to directly construct an $(n, k = n-1, d = n-1)$ regenerating code based on an r-regular $(n, w, w+1, \lambda)$-MRD. Unlike previous works such as [38] and [40], our code achieves an interior point other than the k corner points on the optimal storage-bandwidth trade-off curve. This distinguishing characteristic sets it apart from the codes described in previous works such as [38] and [40]. As a result, the new $(n, k = n-1, d = n-1)$ code achieves a smaller sub-packetization level compared to the ones in [38] and [40] while maintaining the same normalized storage-bandwidth trade-off in certain cases.
- For the new $(n, k = n-2, d \geq n-2)$ code, it supports two repair degrees, i.e., a failed node can be repaired by contacting $n-2$ helper nodes or $n-1$ helper nodes, this provides more flexibility since it is not always feasible to connect and download data from all the surviving nodes in a practical system, as some nodes may be unavailable due to other assigned jobs or network congestion [44]. Furthermore, when choosing $d = n-2$, the repair bandwidth is smaller than that of the $(n, k = n-2, d = n-2)$ code in [38]. When $d = n-1$, the repair bandwidth is smaller than that of the $(n, k = n-2, d = n-1)$ code in [38] for some regions. To the best of our knowledge, it is the first time to construct storage codes between MSR and MBR points that can support multiple repair degrees with efficient repair mechanisms.

The remainder of the paper is organized as follows. Section II reviews some necessary preliminaries of block designs. Section III proposes the new code constructions and their asserted properties. Performance analysis and comparisons are carried out in Section IV. Finally, Section V concludes the study.

II. BASIC CONCEPTS AND LEMMAS OF BLOCK DESIGNS

Definition 1 ([45]): A t-(n, w, λ) design is a pair $(X, \mathbb{B})$ where X is an n-set of points and $\mathbb{B}$ is a collection of w-subsets of X (called blocks) with the property that every t-subset of X is contained in exactly λ blocks. Particularly, a 2-design is called a balanced incomplete block design (BIBD) and is denoted by (n, w, λ)-BIBD.

Definition 2 ([45]): Let W be a subset of non-negative integers. A regular mandatory representation design (n, W, λ), denoted by r-regular (n, W, λ)-MRD, is a pair $(X, \mathbb{B})$ where X is an n-set and $\mathbb{B}$ is a family of subsets of X that satisfy

(i) $|B| \in W$ for any $B \in \mathbb{B}$;

(ii) For each $w \in W$, there is at least one subset $B \in \mathbb{B}$ with $|B| = w$;

(iii) Each element of X is contained in exactly r blocks of $\mathbb{B}$;

(iv) Every pair of distinct elements of X occurs in exactly λ blocks of $\mathbb{B}$.

Particularly, a r-regular (n, W, λ)-MRD is a BIBD if $|W| = 1$.

For a t-(n, w, λ) design $(X, \mathbb{B})$, every s-subset of X is contained in exactly λ_s blocks where $0 \leq s \leq t$ [45]. For simplicity, we also denote the number of blocks that contain any given element of X by r and $b = |\mathbb{B}|$. For a 3-(n, w, λ) design, by [45], we have

$$\lambda_2 = \frac{b(w-1)}{n(n-1)} \quad \text{and} \quad \lambda = \lambda_3 = \frac{b(w-1)(w-2)}{n(n-1)(n-2)}. \tag{4}$$

If $(X, \mathbb{B})$ is an (n, w, λ)-BIBD, by [45], we similarly have

$$\lambda = \frac{bw(w-1)}{n(n-1)} \quad \text{and} \quad r = \lambda_1 = \frac{bw}{n} = \frac{\lambda(n-1)}{w-1}. \tag{5}$$

Lemma 1 ([45]): For a r-regular (n, W, λ)-MRD with $|W| = s$, assume that there are b_j blocks of size w_j, $j = 1, 2, \cdots, s$, then

$$\lambda n(n-1) = \sum_{j=1}^{s} b_j w_j (w_j - 1).$$

A BIBD with parameters n, w, λ, b, r is also denoted by $(n, w, \lambda; b, r)$-BIBD. For any given n, w_1, and w_2, define

$$r_{\min}^{(1)} = \min\{r_1 : \text{there is an } (n, w_1, \lambda_1; b_1, r_1)\text{-BIBD}\}, \tag{6}$$

$$r_{\min}^{(2)} = \min\{r_2 : \text{there is an } (n, w_2, \lambda_2; b_2, r_2)\text{-BIBD}\}, \tag{7}$$

and

$$r_{\min}^{(3)} = \min\{r_3 \in \mathbb{N} : \text{there is a } r_3\text{-regular } (n, \{w_1, w_2\}, \lambda_3)\text{-MRD}\}, \tag{8}$$

then we have the following result.

Lemma 2: For any given $(n, w_1, \lambda^{(1)}; b_1, r_{\min}^{(1)})$-BIBD $(X, \mathbb{B}_1)$, $(n, w_2, \lambda^{(2)}; b_2, r_{\min}^{(2)})$-BIBD $(X, \mathbb{B}_2)$, and $r_{\min}^{(3)}$-regular $(n, \{w_1, w_2\}, \lambda^{(3)})$-MRD $(X, \mathbb{B}_3)$, where $2 \leq w_1 < w_2 < n$, we have

$$r_{\min}^{(3)} \leq r_{\min}^{(1)} + r_{\min}^{(2)}.$$

Proof: Let $\mathbb{B} = \mathbb{B}_1 \bigcup \mathbb{B}_2$, where the notation $\bigcup$ denotes a multi-set union in this paper. Clearly, $(X, \mathbb{B})$ is a r-regular $(n, \{w_1, w_2\}, \lambda)$-MRD, where $r = r_{\min}^{(1)} + r_{\min}^{(2)}$, $\lambda = \lambda^{(1)} + \lambda^{(2)}$. Therefore, $r_{\min}^{(3)} \leq r = r_{\min}^{(1)} + r_{\min}^{(2)}$ always holds. ■

From here on, we always assume $X = \{1, 2, \cdots, n\}$ for simplicity. Finally, we introduce the very useful notion of the incidence matrix.

Definition 3 ([45]): Let $(X, \mathbb{B})$ be a design where $\mathbb{B} = \{B_1, B_2, \ldots, B_b\}$. The incidence matrix of $(X, \mathbb{B})$ is the $n \times b$ binary matrix $M = (m_{i,j})$ defined by the rule

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

For an $n \times b$ matrix M that denoted by

$$M = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{pmatrix},$$

define the weight of row i by $wt(m_i) = \sum_{l=1}^{b} m_{i,l}$. For two row vectors m_i and m_j, let $m_i \cdot m_j = (m_{i,1}m_{j,1}, \cdots, m_{i,b}m_{j,b})$ be the inner product, then it is obvious that

$$wt(m_i + m_j) = wt(m_i) + wt(m_j) - wt(m_i \cdot m_j). \tag{9}$$

III. CONSTRUCTION OF THE NEW STORAGE CODES

In this section, we present the construction of the new $(n, k = n - \delta, k \leq d < n)$ storage codes, where $\delta = 1, 2$, and analyze the conditions such that P1, P2, and P3 hold for the storage code under two specific cases.

Since quantities α and β scale linearly with $\mathcal{M}$, they can be normalized by $\mathcal{M}$ as follows:

$$\bar{\alpha} := \frac{\alpha}{\mathcal{M}}, \quad \bar{\beta} := \frac{\beta}{\mathcal{M}}.$$

Then (1) can be written as

$$1 \leq \sum_{i=0}^{k-1} \min(\bar{\alpha}, (d-i)\bar{\beta}). \tag{10}$$

Throughout this paper, $(\bar{\alpha}, \bar{\beta})$, which denotes the pair of the normalized storage and repair bandwidth of an (n, k, d) code, will be used as the measure of performance and will be referred to as the normalized storage-bandwidth pair for simplicity if the context is clear.

Construction 1: Given a design $(X, \mathbb{B})$ with $\mathbb{B} = \{B_1, B_2, \ldots, B_b\}$, where $|B_i| = w_i > \delta$, let $M = (m_{i,j})$ be its incident matrix. The storage code using this block design has $\mathcal{M} = \sum_{i=1}^{b} (w_i - \delta)N$ data symbols in certain finite filed F_q, we arrange these data symbols in b matrices $\mathbf{E}_1, \mathbf{E}_2, \cdots, \mathbf{E}_b$, where $\mathbf{E}_i$ is a $(w_i - \delta) \times N$ matrix, $1 \leq i \leq b$. The structure of the storage code can be inferred from the following two steps:

Step 1. For each $1 \leq i \leq b$, the data matrix $\mathbf{E}_i$ is encoded by a $(w_i, w_i - \delta)$ MSR code with sub-packetization level N, to yield a $w_i \times N$ matrix $\mathbf{U}_i$ with the first $w_i - \delta$ rows storing systematic data and the last δ rows storing parity data.

Step 2. For each $i \in \{1, 2, \ldots, b\}$, we place the data in the w_i rows of $\mathbf{U}_i$ on the w_i nodes in B_i.

In the following, we first give a motivating example which shows the main idea of the construction.

Example 1: Consider a 4-regular $(6, \{3, 4\}, 2)$-MRD $(X, \mathbb{B})$ with incident matrix

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

we set $\delta = N = 1$, and construct an $(n, k, d) = (6, 5, 5)$ storage code using this design. The code has $\mathcal{M} = 17$ data symbols, which can be arranged as

$$\begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{pmatrix}, \begin{pmatrix} u_{1,2} \\ u_{2,2} \end{pmatrix}, \begin{pmatrix} u_{1,3} \\ u_{2,3} \end{pmatrix}, \begin{pmatrix} u_{1,4} \\ u_{2,4} \\ u_{3,4} \end{pmatrix}, \begin{pmatrix} u_{1,5} \\ u_{2,5} \\ u_{3,5} \end{pmatrix}, \begin{pmatrix} u_{1,6} \\ u_{2,6} \end{pmatrix}, \begin{pmatrix} u_{1,7} \\ u_{2,7} \end{pmatrix},$$

then encode the data in the above 7 matrices by a $(4, 3)$ or $(3, 2)$ scalar MDS code (i.e., $N = 1$) to yield

$$\begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{4,1} \end{pmatrix}, \begin{pmatrix} u_{1,2} \\ u_{2,2} \\ u_{3,2} \end{pmatrix}, \begin{pmatrix} u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix}, \begin{pmatrix} u_{1,4} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \end{pmatrix}, \begin{pmatrix} u_{1,5} \\ u_{2,5} \\ u_{3,5} \\ u_{4,5} \end{pmatrix}, \begin{pmatrix} u_{1,6} \\ u_{2,6} \\ u_{3,6} \end{pmatrix}, \begin{pmatrix} u_{1,7} \\ u_{2,7} \\ u_{3,7} \end{pmatrix}.$$

The way how the data are stored across a distributed storage system is depicted in Fig. 2.

Remark 1: Please note that there are some key points regarding the new construction and the choices of the underlying block designs, which are specified in the following subsections and represent the main contributions of this work. The first key point is the observation that the model used in [38] to distribute encoded data to storage nodes based on a BIBD is sufficient but not necessary. Therefore, we generalize the model to distribute encoded data based on a general block design and identify the necessary and sufficient conditions for deploying block designs for given parameters, as described in the following subsections.

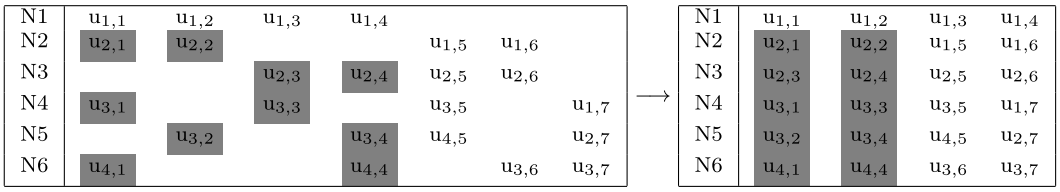

FIGURE 2. The (6,5,5) storage code that is based on a 4-regular (6, {3, 4}, 2)-MRD, where the symbols that need to transmit from the helper nodes when repairing node 1 are indicated in shade.

As we will see in Section III-A, the second key point is that P2 holds for the $(n, k = n-1, d = n-1)$ storage code in Construction 1 if and only if the underlying block design $(X, \mathbb{B})$ satisfies the condition that every pair of distinct elements of X occurs in exactly β blocks of $\mathbb{B}$, and it is not required that each block has the same cardinality. Therefore, P1-P3 holds for the $(n, k = n-1, d = n-1)$ storage code in Construction 1 if and only if $(X, \mathbb{B})$ is a regular MRD, which subsumes BIBDs as a special case.

Similarly, the third key point is that P1-P3 holds for the $(n, k = n-2, d)$ storage code in Construction 1 if and only if $(X, \mathbb{B})$ is a 3-design if $d = n-2$ and a BIBD if $d = n-1$, which will be illustrated in Section III-B.

Based on the structure of the $(n, k = n-\delta, d)$ storage code, it is obvious that the number of the symbols stored in node i is equal to $N \cdot wt(m_i)$, then we have the following result.

Lemma 3: P1 holds for the $(n, k = n-\delta, d)$ storage code in Construction 1 if and only if

$$N \cdot wt(m_i) = \alpha \ (constant), \ 1 \le i \le n. \tag{11}$$

In the following two subsections, we will analyze the necessary and sufficient conditions of P2 and P3 for the $(n, k = n-\delta, k \le d \le n-1)$ storage code in Construction 1 in two cases:

(i) $\delta = N = 1$;

(ii) $\delta = 2, N > 1, w_i \equiv w \ge 2$ for $1 \le i \le b$.

A. THE NEW $(n, k = n-1, d = n-1)$ STORAGE CODE

When $\delta = 1$, the parameters of the storage code are $(n, k = n-1, d = n-1)$, and the MSR codes used in Step 1 in Construction 1 can be just the scalar MDS codes, i.e., $N = 1$. Under such parameters, we have the following result.

Theorem 1: The $(n, k = n-1, d = n-1)$ storage code in Construction 1 has properties P1, P2 and P3 if and only if its corresponding design $(X, \mathbb{B})$ is a regular MRD.

Proof: To prove this theorem, we first note that P3 holds if P2 holds since the reconstructing process is similar to several repair processes. Therefore, we only need to discuss the conditions under which P2 holds.

To repair node i, denote the set of the indices of the helper nodes by $\Delta = X \backslash \{i\}$. For each $s \in \{1, 2, \ldots, b\}$ such that $i \in B_s$, the lost symbol in U_s can be regenerated by downloading the remaining symbols in U_s from the helper nodes with indices in $B_s \setminus \{i\}$ (see Fig. 2 for an example). Therefore, the number of symbols that are sent from the jth node is $wt(m_i \cdot m_j) = |\{B : i, j \in B, B \in \mathbb{B}\}|$. Thus, P2 holds for the $(n, k = n-1, d = n-1)$ storage code in Construction 1 if and only if

$$wt(m_i \cdot m_j) = \beta \ (constant), \ 1 \le i \ne j \le n.$$

In conjunction with Lemma 3 and Definition 2, we obtain the desired result. ∎

Remark 2: We note that if $(X, \mathbb{B})$ is an (n, w, λ)-BIBD, i.e., $w_i \equiv w$ for $1 \le i \le b$, then the $(n, k = n-1, d = n-1)$ storage code is exactly the code constructed in [38]. However, our construction is more general since BIBDs are a special kind of regular MRDs. Additionally, the normalized storage-bandwidth pair of the storage code based on an (n, w, λ)-BIBD given in [38] is

$$(\bar{\alpha}, \bar{\beta}) = \left(\frac{w}{n(w-1)}, \frac{w}{n(n-1)}\right). \tag{12}$$

B. THE NEW $(n, k = n-2, d)$ STORAGE CODE

In this subsection, we consider the case where $\delta = 2, N > 1$, and $w_i \equiv w \ge 2$ for $1 \le i \le b$, i.e., the parameters of the storage code are $(n, k = n-2, n-2 \le d \le n-1)$, while the MSR codes in Step 1 in Construction 1 can be any MSR codes such as the aforementioned MSR codes introduced in Section I. Then for any $1 \le i \le b$, the $(w-2) \times N$ matrix $\mathbf{U}_i$ in Step 1 in Construction 1 has the following two abilities according to the properties of MSR codes [7]:

- A1 Reconstruction ability: Any $w-2$ out of the w rows of U_i suffice to recover the whole source data in $\mathbf{U}_i$.
- A2 Repair ability: Any row in U_i can be regenerated by downloading half data from each of the $w-1$ remaining rows.

Note that for an $(n, k = n-2, n-2 \le d \le n-1)$ storage code constructed above, P3 always holds from the ability A1. Based on a $(w, w-2)$ MSR code, an advantage of the $(n, k = n-2, d \ge k)$ code is that d can equal to $n-2$ and $n-1$ by A1 and A2, respectively. In what follows, we analyze the necessary and sufficient conditions for the $(n, k = n-2, n-2 \le d \le n-1)$ storage code in Construction 1 to satisfy P2. We first consider the case where $d = n-2$ and prove the following theorem.

Theorem 2: The $(n, k = n-2, d = n-2)$ storage code in Construction 1 satisfies properties P1 and P2 if and only if its corresponding block design $(X, \mathbb{B})$ is a 3-(n, w, λ) design. If $(X, \mathbb{B})$ is a 3-design, then the normalized

storage-bandwidth pair of the code is

$$(\bar{\alpha}, \bar{\beta}) = \left(\frac{w}{n(w-2)}, \frac{w(w-1)(n+w-4)}{2n(n-1)(n-2)(w-2)} \right).$$

Proof: Assume node i is failed and we connect nodes in the set $\Delta = X \backslash \{i, j\}$ to repair node i, where $j \in \{1, 2, \ldots, n\} \backslash \{i\}$. To prove this theorem, we consider the number of symbols sent from each helper node to repair node i. For $s = 1, 2, \cdots, b$, let us consider the blocks B_s with $i \in B_s$.

(i) If $j \notin B_s$, then $|\Delta \cap B_s| = w - 1$, i.e., $w - 1$ rows in $\mathbf{U}_s$ are available. According to A2, the lost row in $\mathbf{U}_s$ can be regenerated by downloading $N/2$ symbols in each of the $w - 1$ remaining rows in $\mathbf{U}_s$. Therefore, for any $t \in X \setminus \{i, j\}$, the number of symbols which sent from node t is $\frac{N}{2} wt(m_i \cdot (m_j + 1) \cdot m_t)$.

(ii) If $j \in B_s$, then $|\Delta \cap B_s| = w - 2$. According to A1, the lost row in $\mathbf{U}_s$ can be regenerated by downloading all the symbols in the $w-2$ rows in $\mathbf{U}_s$ that are respectively stored in the nodes with indices in $\Delta \cap B_s$. Therefore, for any $t \in X \setminus \{i, j\}$, the number of symbols sent from helper node t is $N \cdot wt(m_i \cdot m_j \cdot m_t)$.

From the discussion above, we have that for $t \in X \setminus \{i, j\}$, the total number of symbols sent from node t is

$$\begin{aligned} &\frac{N}{2} wt(m_i \cdot (m_j + 1) \cdot m_t) + N \cdot wt(m_i \cdot m_j \cdot m_t) \\ &= \frac{N}{2} \left(wt(m_i \cdot m_j \cdot m_t) + wt(m_i \cdot m_t) \right) \end{aligned} \tag{13}$$

by (9). Thus, P2 holds for the $(n, k = n-2, d = n-2)$ storage code in Construction 1 if and only if (13) is a constant for any three distinct integers $i, j, t \in X$. It is easy to check that (13) is a constant if and only if both $wt(m_i \cdot m_j \cdot m_t)$ and $wt(m_i \cdot m_t)$ are constants, which together with Lemma 3 imply that P1 and P2 hold for an $(n, k = n - 2, d = n - 2)$ storage code if and only if $(X, \mathbb{B})$ is a 3-(n, w, λ) design.

When $(X, \mathbb{B})$ is a 3-(n, w, λ) design, from (4) and (13) we have

$$\alpha = \frac{bwN}{n} \text{ and } \beta = \frac{N}{2}(\lambda + \lambda_2) = \frac{w(w-1)(n+w-4)}{2n(n-1)(n-2)} bN.$$

Thus the normalized storage-bandwidth pair of the $(n, k = n - 2, d = n - 2)$ code in Construction 1 is

$$(\bar{\alpha}, \bar{\beta}) = \left(\frac{w}{n(w-2)}, \frac{w(w-1)(n+w-4)}{2n(n-1)(n-2)(w-2)} \right).$$

The proof is then completed. ∎

For the case $d = n - 1$, similar to the discussions in subsection III-A, it is not difficult to obtain the following result by (12).

Theorem 3: The $(n, k = n - 2, d = n - 1)$ storage code in Construction 1 satisfies P1 and P2 if and only if its corresponding block design $(X, \mathbb{B})$ is an (n, w, λ)-BIBD. If $(X, \mathbb{B})$ is a BIBD, then the normalized storage-bandwidth pair is $(\bar{\alpha}, \bar{\beta}) = \left(\frac{w}{n(w-2)}, \frac{w(w-1)}{2n(n-1)(w-2)} \right)$.

By Theorems 2, 3, and the fact that a 3-design is also a BIBD, we have the following corollary.

Corollary 1: P1 and P2 hold for the $(n, k = n - 2, d)$ storage code in Construction 1 if its corresponding block design $(X, \mathbb{B})$ is a 3-(n, w, λ) design, where $d \in \{n-2, n-1\}$.

IV. PERFORMANCE ANALYSIS AND COMPARISONS

In this section, we give a detailed comparison between the works in [38] and [40] and ours as they are closely related.

A. COMPARISONS FOR $(n, k = n-1, d = n-1)$ STORAGE CODES

The following theorem demonstrates that the new $(n, k = n-1, d = n-1)$ storage code in Construction 1 outperforms the ones in [38] and [40] in terms of the sub-packetization level for some normalized storage-bandwidth trade-off. For convenience, we refer to the storage codes in [38] and [40] as TSAVK codes and EM codes, respectively.

From [40], we have that the parameters α, β, and $\mathcal{M}$ that EM codes can achieve are

$$\alpha_i^{EM} = \binom{k}{i}, \beta_i^{EM} = \binom{k-1}{i-1}, \mathcal{M}_i^{EM} = k\binom{k}{i} - \binom{k}{i+1} \tag{14}$$

for $i \in \{1, 2, \ldots, k\}$. The normalized storage and repair bandwidth are

$$\begin{aligned} \bar{\alpha}_i^{EM} &= \frac{\alpha_i^{EM}}{\mathcal{M}_i^{EM}} \\ &= \frac{\binom{k}{i}}{k\binom{k}{i} - \binom{k}{i+1}} \\ &= \frac{\binom{k}{i}}{k\binom{k}{i} - \binom{k}{i}(k-i)/(i+1)} \\ &= \frac{i+1}{in}, \end{aligned}$$

and

$$\begin{aligned} \bar{\beta}_i^{EM} &= \frac{\beta_i^{EM}}{\mathcal{M}_i^{EM}} \\ &= \frac{\binom{k-1}{i-1}}{k\binom{k}{i} - \binom{k}{i+1}} \\ &= \frac{\frac{i}{k}\binom{k}{i}}{k\binom{k}{i} - \binom{k}{i}(k-i)/(i+1)} \\ &= \frac{i+1}{n(n-1)}. \end{aligned}$$

By referring to (12), we observe that the normalized storage-bandwidth pair $(\bar{\alpha}_i^{EM}, \bar{\beta}_i^{EM})$ of the EM codes in [40] is also achieved by the TSAVK codes in [38] that based on an $(n, i+1, \lambda)$-BIBD. Therefore, it suffices to compare the normalized storage-bandwidth pair of the new code only with the TSAVK codes in [38].

Theorem 4: For any given integer $s \geq 2$ and any subset of positive integers $W = \{w_1, w_2, \cdots, w_s\}$, where $w_1 < w_2 < \cdots < w_s$, let $(\bar{\alpha}_M, \bar{\beta}_M)$ be the normalized

storage-bandwidth pair of the $(n, k = n-1, d = n-1)$ storage code in Construction 1 that based on an r-regular (n, W, λ)-MRD. For $0 \le j \le w_s - w_1$, let $\left(\bar{\alpha}_j^{TSAVK}, \bar{\beta}_j^{TSAVK}\right)$ be the normalized storage-bandwidth pair of the $(n, k = n-1, d = n-1)$ storage code in [38] with the underlying block design being an $(n, w_1 + j, \lambda^{(j)})$-BIBD. For each $0 \le i < w_s - w_1$, let L_i denote the line passing through $\left(\bar{\alpha}_i^{TSAVK}, \bar{\beta}_i^{TSAVK}\right)$ and $\left(\bar{\alpha}_{i+1}^{TSAVK}, \bar{\beta}_{i+1}^{TSAVK}\right)$, then

(i) If $w_s - w_1 > 1$, the point $(\bar{\alpha}_M, \bar{\beta}_M)$ always lies above the line L_i for $0 \le i < w_s - w_1$;

(ii) If $w_s - w_1 = 1$, the point $(\bar{\alpha}_M, \bar{\beta}_M)$ lies on the line L_0 and between the two points $\left(\bar{\alpha}_i^{TSAVK}, \bar{\beta}_i^{TSAVK}\right)$ and $\left(\bar{\alpha}_{i+1}^{TSAVK}, \bar{\beta}_{i+1}^{TSAVK}\right)$.

Proof: By (12), we have

$$\left(\bar{\alpha}_i^{TSAVK}, \bar{\beta}_i^{TSAVK}\right) = \left(\frac{w_1 + i}{n(w_1 + i - 1)}, \frac{w_1 + i}{n(n-1)}\right)$$

for $0 \le i \le w_s - w_1$. Therefore the equation of L_i is

$$\begin{aligned}\bar{\beta} &= f_i(\bar{\alpha}) \\ &= -\frac{(w_1 + i)(w_1 + i - 1)}{n-1}\bar{\alpha} + \frac{(w_1 + i)(w_1 + i + 1)}{n(n-1)},\end{aligned}$$

where $0 \le i < w_s - w_1$.

Given an r-regular (n, W, λ_M)-MRD, assume that there are b_j blocks of size w_j, $j = 1, 2, \cdots, s$, then by Lemma 1, we have

$$\lambda n(n-1) = \sum_{j=1}^{s} b_j w_j (w_j - 1).$$

Therefore,

$$(\bar{\alpha}_M, \bar{\beta}_M) = \left(\frac{\sum_{i=1}^{s} b_i w_i}{n\sum_{i=1}^{s} b_i(w_i - 1)}, \frac{\sum_{i=1}^{s} b_i w_i (w_i - 1)}{n(n-1)\sum_{i=1}^{s} b_i(w_i - 1)}\right).$$

Now we only need to compare the values of $f(\bar{\alpha}_M)$ and $\bar{\beta}_M$ to prove our statement. It is not difficult to obtain that

$$\begin{aligned}&(f_i(\bar{\alpha}_M) - \bar{\beta}_M) n(n-1) \sum_{j=1}^{s} b_j(w_j - 1) \\ &= -(w_1 + i - 1)(w_1 + i) \sum_{j=1}^{s} b_j w_j - \sum_{j=1}^{s} b_j w_j (w_j - 1) \\ &\quad + (w_1 + i)(w_1 + i + 1) \sum_{j=1}^{s} b_j(w_j - 1) \\ &= -b_1 i(i+1) \\ &\quad - \sum_{j=2}^{s} b_j[(w_j - w_1)(w_j - w_1 - 1 - 2i) + i^2 + i] \\ &= -b_1 i(i+1) - \sum_{j=2}^{s} b_j[(w_j - w_1 - \frac{2i+1}{2})^2 - \frac{1}{4}].\end{aligned}$$

Note that $n(n-1)\sum_{j=1}^{s} b_j(w_j - 1) > 0$, therefore,

(i) If $w_s - w_1 > 1$, then $f(\bar{\alpha}_M) - \bar{\beta}_M < 0$ always holds, i.e., the point $(\bar{\alpha}_M, \bar{\beta}_M)$ lies above the line L_i for any $0 \le i < w_s - w_1$;

(ii) If $w_s - w_1 = 1$, then $w_1 < w_s$ forces s to be 2. Therefore, $f_i(\bar{\alpha}_M) - \bar{\beta}_M = 0$ since $i = 0$ in this case, i.e., the point $(\bar{\alpha}_M, \bar{\beta}_M)$ lies on the line L_0. It is easy to see that $(\bar{\alpha}_M, \bar{\beta}_M) \ne \left(\bar{\alpha}_0^{TSAVK}, \bar{\beta}_0^{TSAVK}\right), \left(\bar{\alpha}_1^{TSAVK}, \bar{\beta}_1^{TSAVK}\right)$ in this case. Furthermore, the point $(\bar{\alpha}_M, \bar{\beta}_M)$ lies between the two points $\left(\bar{\alpha}_0^{TSAVK}, \bar{\beta}_0^{TSAVK}\right)$ and $\left(\bar{\alpha}_1^{TSAVK}, \bar{\beta}_1^{TSAVK}\right)$.

The proof is then completed. ■

Since the $(n, k = n-1, d = n-1)$ EM codes and the TSAVK codes are optimal w.r.t the bound in (2), and they achieve the k corner points on the optimal tradeoff curve, which together with Theorem 4 implies that the new $(n, k = n-1, d = n-1)$ storage code in Construction 1 based on an r-regular (n, W, λ)-MRD is optimal w.r.t the bound in (2) if and only if $W = \{w, w+1\}$, where $2 \le w \le k$. Additionally, we provide a rigorous proof to demonstrate the optimality of the new storage code.

Theorem 5: The $(n, k = n-1, d = n-1)$ storage code in Construction 1 that based on an r-regular (n, W, λ)-MRD is optimal w.r.t. the bound in (2) if $W = \{w, w+1\}$, where $2 \le w \le k$. In addition, the new code does not operate on the k corner points of the optimal storage-bandwidth tradeoff curve characterized in (2).

Proof: Clearly, the storage code has the following parameters

$$\begin{aligned}&\alpha = \frac{b_1 w + b_2(w+1)}{n}, \beta = \frac{b_1 w(w-1) + b_2 w(w+1)}{n(n-1)}, \\ &\mathcal{M} = b_1(w-1) + b_2 w. \qquad (15)\end{aligned}$$

Then,

$$\begin{aligned}\frac{d\beta}{\alpha} &= \frac{b_1 w(w-1) + b_2 w(w+1)}{n} \frac{n}{b_1 w + b_2(w+1)} \\ &= \frac{w(b_1 w + b_2(w+1)) - b_1 w}{b_1 w + b_2(w+1)} \\ &= w - \frac{b_1 w}{b_1 w + b_2(w+1)}.\end{aligned}$$

This implies that $\frac{d\beta}{w} \le \alpha \le \frac{d\beta}{w-1}$. Now, let's consider the case when $2 \le w \le n-2$,

$$\lfloor \frac{w(w-1)n\alpha + n(n-1)\beta}{w^2 + w} \rfloor = b_1(w-1) + b_2 w = \mathcal{M}.$$

If $w = n-1$, we have

$$\begin{aligned}&(n-2)\alpha + \beta \\ &= (w-1)\frac{b_1 w + b_2(w+1)}{w+1} + \frac{b_1 w(w-1) + b_2 w(w+1)}{w(w+1)} \\ &= \frac{w(w-1)b_1 w + b_1 w(w-1)}{w(w+1)} \\ &\quad + \frac{b_2(w+1)w(w-1) + b_2 w(w+1)}{w(w+1)} \\ &= b_1(w-1) + b_2 w = \mathcal{M}.\end{aligned}$$

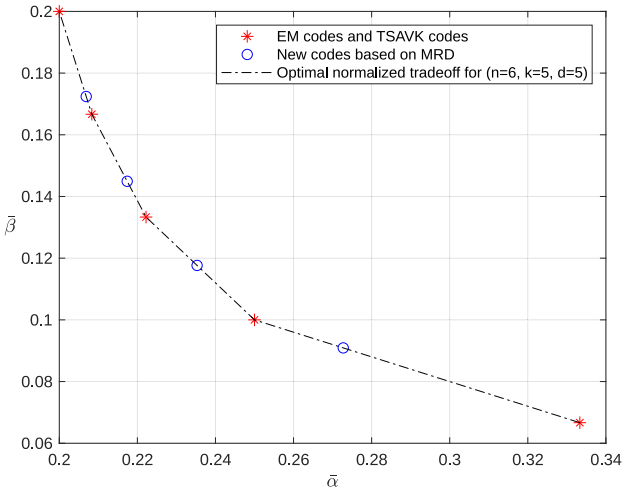

FIGURE 3. Comparison of the normalized storage-bandwidth pairs achieved by the EM codes, TSAVK codes, and the new codes for ($n = 6, k = 5, d = 5$).

Based on the above analysis, we can conclude that (2) holds with equality for the new $(n, k = n-1, d = n-1)$ storage code in Construction 1 that based on an r-regular $(n, \{w, w+1\}, \lambda)$-MRD.

In addition, it is worth noting that according to (2) and (3), the k corner points satisfy $\alpha = \frac{d\beta}{i}$ for $i = 1, 2, \ldots, k$. However, based on (15), it is evident that $\alpha \nmid d\beta$ for the new code, which implies that the new code does not operate on the k corner points of the optimal storage-bandwidth trade-off curve characterized in (2). ■

Note that all of the $(n, k = n-1, d = n-1)$ EM codes, TSAVK codes, and the new storage code based on an r-regular $(n, \{w, w+1\}, \lambda)$-MRD are optimal w.r.t. the bound in (2). However, there are some differences between the new code and EM, TSAVK codes. The EM codes and TSAVK codes operate exactly on the k corner points of the optimal trade-off curve, the other points on the curve can only be achieved through space-sharing. On the other hand, the new storage code based on an r-regular $(n, \{w, w+1\}, \lambda)$-MRD operates on a point in stead of the k corner points of the optimal trade-off curve. An example is illustrated in Figure 3, where the points achieved by the new storage codes are based on the following MRDs $(X, \mathbb{B})$

- A 3-regular $(6, \{2, 3\}, 1)$-MRD, where

$$\mathbb{B} = \{\{1,2,4\}, \{1,3\}, \{1,5,6\}, \{2,3,5\}, \{2,6\}, \{3,4,6\}, \{4,5\}\}.$$

- A 4-regular $(6, \{3, 4\}, 2)$-MRD, where

$$\mathbb{B} = \{\{1,2,3,6\}, \{1,2,5\}, \{1,3,4,5\}, \{1,4,6\}, \{2,3,4\}, \{2,4,5,6\}, \{3,5,6\}\}.$$

- A 15-regular $(6, \{4, 5\}, 10)$-MRD, where

$$\mathbb{B} = \{\{1,2,3,4,5\}, \{1,2,3,4,6\}, \{1,2,3,4\}, \{1,2,3,5,6\}, \{1,2,3,5\}, \{1,2,3,6\}, \{1,2,4,5,6\}, \{1,2,4,5\}, \{1,2,4,6\}, \{1,2,5,6\}, \{1,3,4,5,6\}, \{1,3,4,5\}, \{1,3,4,6\}, \{1,3,5,6\}, \{1,4,5,6\}, \{2,3,4,5,6\}, \{2,3,4,5\}, \{2,3,4,6\}, \{2,3,5,6\}, \{2,4,5,6\}, \{3,4,5,6\}\}.$$

- A 6-regular $(6, \{5, 6\}, 5)$-MRD, where

$$\mathbb{B} = \{\{1,2,3,4,5,6\}, \{1,2,3,4,5\}, \{1,2,3,4,6\}, \{1,2,3,5,6\}, \{1,2,4,5,6\}, \{1,3,4,5,6\}, \{2,3,4,5,6\}\}.$$

Through space-sharing, the EM codes and TSAVK codes can also achieve points on the optimal storage-bandwidth trade-off curve beyond the k corner points. However, it is important to note that space-sharing significantly increases the sub-packetization level compared to that of the new storage code based on an r-regular $(n, w, w+1, \lambda)$-MRD.

To further illustrate this point, let's consider an example that demonstrates how the sub-packetization level of the new storage code $\mathcal{C}^M$ can be lower than that of the codes obtained by space-sharing TSAVK codes and EM codes in certain situations.

Example 2: By substituting $n = 12$, $w_1 = 3$ and $w_2 = 4$ into (6)–(8), we obtain $r_{min}^{(1)} = r_{min}^{(2)} = 11$ and $r_{min}^{(3)} = 4$ [45]. Let $(X, \mathbb{B}_1)$, $(X, \mathbb{B}_2)$, and $(X, \mathbb{B}_3)$ be a $(12, 3, 2; 44, 11)$-BIBD, a $(12, 4, 3; 33, 11)$-BIBD and a 4-regular $(12, \{3, 4\}, 1)$-MRD, respectively. Using these three designs, we can obtain two $(n = 12, k = 11, d = 11)$ storage codes from [38] and an $(n = 12, k = 11, d = 11)$ storage code by Construction 1, denoted by $\mathcal{C}_2^{TSAVK}$, $\mathcal{C}_3^{TSAVK}$, and $\mathcal{C}^M$, respectively. For $i = 2, 3$, let $(\alpha_i^{TSAVK}, \beta_i^{TSAVK}, \mathcal{M}_i^{TSAVK})$ be the parameters of $\mathcal{C}_i^{TSAVK}$, and let $(\alpha^M, \beta^M, \mathcal{M}^M)$ be the parameters of $\mathcal{C}^M$. By (12) and $\mathcal{M} = (w-1)b$, we have

$$(\alpha_2^{TSAVK}, \beta_2^{TSAVK}, \mathcal{M}_2^{TSAVK}) = (11, 2, 88),$$
$$(\alpha_3^{TSAVK}, \beta_3^{TSAVK}, \mathcal{M}_3^{TSAVK}) = (11, 3, 99).$$

By Definition 1-(iii), we have $rn = b_1 w_1 + b_2 w_2$, which together with Theorem 4 implies $\alpha^M = \frac{b_1 w_1 + b_2 w_2}{n} = 4$, $\beta^M = 1$, and $\mathcal{M}^M = 35$. Clearly, the normalized storage-bandwidth pair of $\mathcal{C}^M$ is $(\bar{\alpha}^M, \bar{\beta}^M) = (\frac{4}{35}, \frac{1}{35})$. By space-sharing between the codes $\mathcal{C}_2^{TSAVK}$ and $\mathcal{C}_3^{TSAVK}$, we can obtain a new $(n = 12, k = 11, d = 11)$ storage code $\mathcal{C}^{SS-TSAVK}$ with parameters

$$\alpha^{SS-TSAVK} x\alpha_2^{TSAVK} + y\alpha_3^{TSAVK},$$
$$\beta^{SS-TSAVK} x\beta_2^{TSAVK} + y\beta_3^{TSAVK},$$
$$\mathcal{M}^{SS-TSAVK} x\mathcal{M}_2^{TSAVK} + y\mathcal{M}_3^{TSAVK}.$$

By choosing appropriate values for x and y, the normalized storage-bandwidth pair of $\mathcal{C}_M$ can also be achieved by the code $\mathcal{C}^{SS-TSAVK}$. In this case, we have

$$\bar{\alpha}^M \frac{\alpha^{SS-TSAVK}}{\mathcal{M}^{SS-TSAVK}} = \frac{x\alpha_2^{TSAVK} + y\alpha_3^{TSAVK}}{x\mathcal{M}_2^{TSAVK} + y\mathcal{M}_3^{TSAVK}},$$

TABLE 1. A comparison of the sub-packetization level α among the new $(n = 6, k = 5, d = 5)$ storage code, the codes obtained by space-sharing EM codes and by space-sharing TSAVK codes with the normalized storage-bandwidth pair $(\frac{4}{35}, \frac{1}{35})$.

	New code	SS-EM [40]	SS-TSAVK [38]
α	4	220	44

$$\bar{\beta}^M \frac{\beta^{SS-TSAVK}}{\mathcal{M}^{SS-TSAVK}} = \frac{x\beta_2^{TSAVK} + y\beta_3^{TSAVK}}{x\mathcal{M}_2^{TSAVK} + y\mathcal{M}_3^{TSAVK}},$$

This results in $y = 3x$. Obviously, if we choose $x = 1$ and $y = 3$, we obtain the minimum value of $\alpha^{SS-TSAVK}$. Then the corresponding parameters of $\mathcal{C}^{SS-TSAVK}$ are

$$\left(\alpha^{SS-TSAVK}, \beta^{SS-TSAVK}, \mathcal{M}^{SS-TSAVK}\right) = (44, 11, 385).$$

Clearly, the sub-packetization level of $\mathcal{C}^{SS-TSAVK}$ is 11 times larger than that of $\mathcal{C}^M$.

Similarly, to achieve the same normalized storage-bandwidth pair of $\mathcal{C}^M$ by EM codes, one needs to space-sharing two EM codes with parameters

$$(\alpha_2^{EM}, \beta_2^{EM}, \mathcal{M}_2^{EM}) = (55, 10, 440)$$

and

$$(\alpha_3^{EM}, \beta_3^{EM}, \mathcal{M}_3^{EM}) = (165, 45, 1485).$$

This leads a storage code $\mathcal{C}^{SS-EM}$ which has the same normalized storage-bandwidth tradeoff as $\mathcal{C}^M$, but with a sub-packetization level of $\alpha^{SS-EM} = 220$, which is 55 times larger than that of $\mathcal{C}^M$. Table 1 includes the comparison.

B. COMPARISONS FOR $(n, k = n-2, d)$ STORAGE CODES

Denote our new $(n, k = n-2, d)$ code constructed in Section III by $\mathcal{C}^M$ and the $(n, k = n-2, d)$ code constructed in [38] by $\mathcal{C}^{TSAVK}$. The following theorem gives a theoretical comparison between $\mathcal{C}^M$ and $\mathcal{C}^{TSAVK}$ in terms of the repair bandwidth when $k = d = n-2$.

Theorem 6: Based on a 3-(n, w, λ) design, where $3 < w < n$, the normalized repair bandwidth of $\mathcal{C}^M$ is smaller than that of $\mathcal{C}^{TSAVK}$ under the same parameter $(n, k = n-2, d = n-2)$ and the same normalized storage.

Proof: From [38] and Theorem 2, we known that the normalized storage-bandwidth pairs of the $(n, k = n-2, d = n-2)$ codes $\mathcal{C}^{TSAVK}$ and $\mathcal{C}^M$ are respectively

$$\left(\bar{\alpha}^{TSAVK}, \bar{\beta}^{TSAVK}\right) = \left(\frac{w}{n(w-2)}, \frac{w}{n(n-2)}\right)$$

and

$$\left(\bar{\alpha}^M, \bar{\beta}^M\right) = \left(\frac{w}{n(w-2)}, \frac{w(w-1)(n+w-4)}{2n(n-1)(n-2)(w-2)}\right).$$

Therefore, we have

$$\frac{\bar{\beta}^M}{\bar{\beta}^{TSAVK}} = \frac{w(w-1)(n+w-4)}{2n(n-1)(n-2)(w-2)} \frac{n(n-2)}{w} = \frac{(w-1)(n+w-4)}{2(n-1)(w-2)}.$$

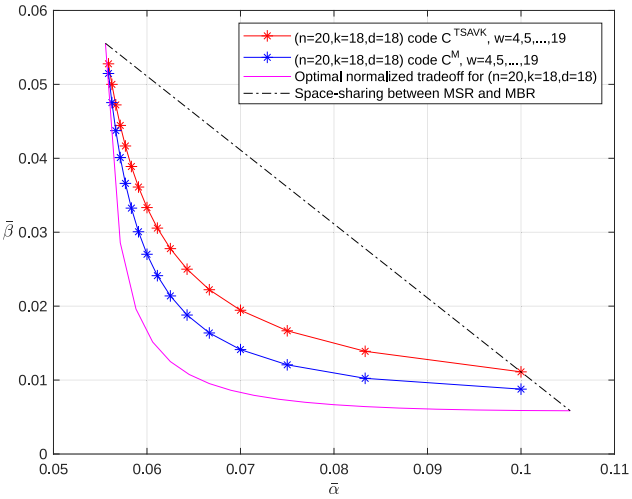

FIGURE 4. Comparisons between $\mathcal{C}^{TSAVK}$ and $\mathcal{C}^M$ under the parameters $(n = 20, k = 18, d = 18)$.

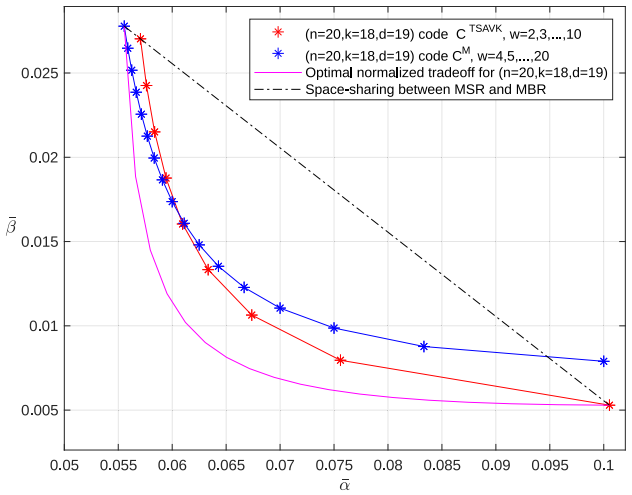

FIGURE 5. Comparisons between $\mathcal{C}^{TSAVK}$ and $\mathcal{C}^M$ under the parameters $(n = 20, k = 18, d = 19)$.

Since $(w-3)(n-w) > 0$ holds for $3 < w < n$, we can conclude that $\bar{\beta}^M/\bar{\beta}^{TSAVK} < 1$ holds for $3 < w < n$. This completes the proof. ■

Theorem 6 is best illustrated through the example shown in Fig. 4. It is clear that our new code $\mathcal{C}^M$ outperforms $\mathcal{C}^{TSAVK}$ constructed in [38] when $d = n-2$.

Now let us move on to compare $\mathcal{C}^M$ and $\mathcal{C}^{TSAVK}$ with the same parameter $(n, k = n-2, d = n-1)$.

Theorem 7: When $n \to \infty$, the normalized repair bandwidth of $\mathcal{C}^M$ is smaller than that of $\mathcal{C}^{TSAVK}$ under the same parameter $(n, k = n-2, d = n-1)$ when the normalized storage $\bar{\alpha}$ is smaller than a threshold

$$T = \frac{\frac{72n^2-72n+13}{f(n)} + f(n) + 2(3n^2-3n+1)}{3n\left(2n^2-2n-3\right)} \tag{16}$$

where $f(n)$ is defined in (17), as shown at the top of the next page.

$$f(n)=\left(108n\left(2n^3-4n^2+7n-5\right)+35-6\left(2n^2-2n-3\right)\sqrt{3\left(108n^4-216n^3+108n^2-1\right)}\right)^{1/3}. \quad (17)$$

$$\bar{\beta}^{TSAVK}=h'(\bar{\alpha})=\frac{1}{2(n-1)}\left((n^2-n-1)\bar{\alpha}-\sqrt{(n^2-n-1)^2\bar{\alpha}^2-2(n^3-2n^2+2n-1)\bar{\alpha}+(n-1)^2}\right)-\frac{1}{2}. \quad (19)$$

Proof: Note that the normalized storage-bandwidth pair of the $(n, k=n-2, d=n-1)$ code $\mathcal{C}^{TSAVK}$ based on an (n, w, λ)- BIBD that constructed in [38] is given by

$$(\bar{\alpha}^{TSAVK}(w), \bar{\beta}^{TSAVK}(w)) = (\frac{w}{w-1}\frac{n-1}{n(n-1)-w}, \frac{w}{n(n-1)-w}), \quad (18)$$

where $w \geq 2$. By varying w, $\mathcal{C}^{TSAVK}$ can achieve different normalized storage-bandwidth pairs. By space-sharing, the normalized storage-bandwidth pairs that $\mathcal{C}^{TSAVK}$ can achieve are on a curve $\bar{\beta}^{TSAVK}=h(\bar{\alpha})$, which is piece-wise linear with the corner points being (18).

By (18), we have (19), as shown at the top of the page, when $\bar{\alpha}=\frac{w}{w-1}\frac{n-1}{n(n-1)-w}$ with $w\geq 2$, i.e., $h(\bar{\alpha})=h'(\bar{\alpha})$ when $\bar{\alpha}=\frac{w}{w-1}\frac{n-1}{n(n-1)-w}$ with $w\geq 2$. It is easy to see that $h(\bar{\alpha})=h'(\bar{\alpha})$ for all $\bar{\alpha}$ when $n\to\infty$.

While by Theorem 3 and Corollary 1, the normalized storgae bandwidth pair of the $(n, k=n-2, d=n-1)$ code $\mathcal{C}^{M}$ based on a 3-(n, w', λ') design are

$$(\bar{\alpha}^{M}=\frac{w'}{n(w'-2)}, \bar{\beta}^{M}=\frac{w'(w'-1)}{2n(n-1)(w'-2)}),$$

i.e., the normalized storage-bandwidth pair is on the curve

$$\bar{\beta^{M}}=\bar{\alpha}(n\bar{\alpha}+1)/2(n\bar{\alpha}-1)(n-1) \quad (20)$$

when $\bar{\alpha}=\frac{w'}{n(w'-2)}$ with $w'\geq 4$. Similarly, (20) holds for all $\bar{\alpha}$ when $n\to\infty$.

When $n\to\infty$, it is easy to verify that $\bar{\beta^{E}}<\bar{\beta^{TSAVK}}$ is equivalent to

$$n^2(2n^2-2n-3)\bar{\alpha}^3-n(6n^2-6n+2)\bar{\alpha}^2+(6n^2-6n+1)\bar{\alpha}-2n+2<0,$$

which results in

$$\bar{\alpha}<\frac{\frac{72\,n^2-72n+13}{f(n)}+f(n)+2(3n^2-3n+1)}{3\,n\left(2n^2-2n-3\right)},$$

where $f(n)$ is defined in (17).

This completes the proof. ∎

Remark 3: The normalized storage-bandwidth pairs that $\mathcal{C}^{TSAVK}$ and $\mathcal{C}^{M}$ can achieve are two complicated piece-wise linear functions, which make it difficult to characterize the formula of the exact threshold T for general n. Nevertheless, we can still use T in (16) to give an estimation.

As a concrete example, we show that under the parameters $(n=20, k=18, d=19)$, $\mathcal{C}^{M}$ has better performance than $\mathcal{C}^{TSAVK}$ when $\bar{\alpha}$ less than a given threshold, as shown in Fig. 5.

V. CONCLUSION

In this paper, we proposed codes between the MSR and MBR points using block designs. Specifically, we obtained an $(n, k=n-1, d=n-1)$ storage code based on regular MRDs, and showed that it achieves a point on the optimal storage-bandwidth trade-off curve that is distinct from the corner points, provided that the underlying regular MRD consists solely of blocks of size w and $w+1$. Additionally, our code exhibits a smaller sub-packetization level compared to the codes proposed in [38] and [40], while achieving the same normalized storage-bandwidth pairs for certain cases. We also obtained an $(n, k=n-2, d\geq n-2)$ storage code based on 3-designs and showed that the new code has a smaller normalized repair bandwidth than the one in [38] for all regions when $d=n-2$ and for some regions when $d=n-1$. The proposed construction subsumes the one in [38] as more general block designs can be employed. Though the storage bandwidth pair is not as good as the one in [40], the new codes have a simpler structure, and the new $(n, k=n-2, d\geq n-2)$ storage code supports two repair degrees.

Generalizing the new construction to any $k<n-2$ is possible, however, P2 is not easy to satisfy, and the analysis is more sophisticated, which will be left for future research.

ACKNOWLEDGMENT

The authors would like to thank an Associate Editor Dr. Adnan Abid and the two anonymous reviewers for their valuable suggestions and comments, which have greatly improved the presentation and quality of this article.

REFERENCES

[1] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proc. 19th ACM Symp. Operating Syst. Princ.*, Oct. 2003, pp. 29–43.

[2] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," *Proc. VLDB Endowment*, vol. 6, no. 5, pp. 325–336, Mar. 2013.

[3] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows Azure storage," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2012, pp. 15–26.

[4] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *Proc. NSDI*, vol. 4, 2004, pp. 85–98.

[5] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-free global data storage," *IEEE Internet Comput.*, vol. 5, no. 5, pp. 40–49, Sep./Oct. 2001.

[6] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total recall: System support for automated availability management," in *Proc. NSDI*, vol. 4, 2004, p. 25.

[7] A. G. Dimakis, P. 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.

[8] C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1425–1442, Mar. 2011.

[9] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[10] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2011, pp. 1374–1381.

[11] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.

[12] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through Hadamard designs," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3021–3037, May 2013.

[13] J. Li, X. Tang, and U. Parampalli, "A framework of constructions of minimal storage regenerating codes with the optimal access/update property," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1920–1932, Apr. 2015.

[14] Y. S. Han, H.-T. Pai, R. Zheng, and P. K. Varshney, "Update-efficient error-correcting product-matrix codes," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 1925–1938, Jun. 2015.

[15] X. Tang, B. Yang, J. Li, and H. D. L. Hollmann, "A new repair strategy for the Hadamard minimum storage regenerating codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5271–5279, Oct. 2015.

[16] Z. Wang, I. Tamo, and J. Bruck, "Explicit minimum storage regenerating codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4466–4480, Aug. 2016.

[17] B. Sasidharan, M. Vajha, and P. V. Kumar, "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair," 2016, *arXiv:1607.07335*.

[18] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2001–2014, Apr. 2017.

[19] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6307–6317, Oct. 2017.

[20] J. Li, X. Tang, and C. Tian, "A generic transformation for optimal repair bandwidth and rebuilding access in MDS codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1623–1627.

[21] J. Li, X. Tang, and C. Tian, "A generic transformation to enable optimal repair in MDS codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6257–6267, Sep. 2018.

[22] H. Hou and P. P. C. Lee, "Binary MDS array codes with optimal repair," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1405–1422, Mar. 2020.

[23] J. Li, X. Tang, and C. Hollanti, "A generic transformation for optimal node repair in MDS array codes over f2," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 727–738, Feb. 2022.

[24] M. Vajha, S. B. Balaji, and P. V. Kumar, "Small-d MSR codes with optimal access, optimal sub-packetization and linear field size," 2018, *arXiv:1804.00598*.

[25] H. Hou, P. P. C. Lee, K. W. Shum, and Y. Hu, "Rack-aware regenerating codes for data centers," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 4730–4745, Aug. 2019.

[26] Z. Chen and A. Barg, "Explicit constructions of MSR codes for clustered distributed storage: The rack-aware storage model," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, pp. 886–899, Feb. 2020.

[27] Y. Liu, J. Li, and X. Tang, "A generic transformation to generate MDS array codes with δ-optimal access property," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 759–768, Feb. 2022.

[28] N. Wang, G. Li, S. Hu, and M. Ye, "Constructing MSR codes with subpacketization $2^{n/3}$ for $k+1$ helper nodes," *IEEE Trans. Inf. Theory*, vol. 69, no. 6, pp. 3775–3792, Jun. 2023.

[29] Y. Liu, J. Li, and X. Tang, "A generic transformation to enable optimal repair/access MDS array codes with multiple repair degrees," *IEEE Trans. Inf. Theory*, vol. 69, no. 7, pp. 4407–4428, Jul. 2023.

[30] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

[31] C. Tian, "Characterizing the rate region of the (4,3,3) exact-repair regenerating codes," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 967–975, May 2014.

[32] M. Elyasi, S. Mohajer, and R. Tandon, "Linear exact repair rate region of (k + 1, k, k) distributed storage systems: A new approach," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 2061–2065.

[33] N. Prakash and M. N. Krishnan, "The storage-repair-bandwidth trade-off of exact repair linear regenerating codes for the case d = k = n–1," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 859–863.

[34] I. M. Duursma, "Shortened regenerating codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1000–1007, Feb. 2019.

[35] M. Elyasi and S. Mohajer, "A probabilistic approach towards exact-repair regeneration codes," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 865–872.

[36] S. Goparaju, S. El Rouayheb, and R. Calderbank, "New codes and inner bounds for exact repair in distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 1036–1040.

[37] T. Ernvall, "Codes between MBR and MSR points with exact repair property," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6993–7005, Nov. 2014.

[38] C. Tian, B. Sasidharan, V. Aggarwal, V. A. Vaishampayan, and P. V. Kumar, "Layered exact-repair regenerating codes via embedded error correction and block designs," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1933–1947, Apr. 2015.

[39] M. Elyasi and S. Mohajer, "Determinant codes with helper-independent repair for single and multiple failures," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5469–5483, Sep. 2019.

[40] M. Elyasi and S. Mohajer, "Cascade codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7490–7527, Dec. 2020.

[41] A. S. Rawat, I. Tamo, V. Guruswami, and K. Efremenko, "MDS code constructions with small sub-packetization and near-optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6506–6525, Oct. 2018.

[42] K. G. Benerjee and M. K. Gupta, "Trade-off for heterogeneous distributed storage systems between storage and repair cost," *Problems Inf. Transmiss.*, vol. 57, no. 1, pp. 33–53, Jan. 2021.

[43] A. Patra and A. Barg, "Interior-point regenerating codes on graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 1560–1565.

[44] K. Mahdaviani, S. Mohajer, and A. Khisti, "Product matrix MSR codes with bandwidth adaptive exact repair," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3121–3135, Apr. 2018.

[45] C. J. Colbourn, *CRC Handbook of Combinatorial Designs*. Boca Raton, FL, USA: CRC Press, 2010.

XIAOFANG WANG received the M.S. degree in computer application technology from China West Normal University, Nanchong, China, in 2020. She is currently a Lecturer with the Geely University of China. She has published more than ten journal articles and patented three inventions. Her research interests include distributed storage and computer vision.

YUAN LIAO received the B.S. degree in computer science and technology from Sichuan Normal University, Chengdu, China, in 2004, and the M.S. degree in computer science and technology from Southwest Jiaotong University, Chengdu, in 2007. She is currently a Lecturer with the Geely University of China. Her research interests include distributed storage, artificial intelligence, virtual reality, and image processing.