Exact-Repair MDS Code Construction Using Interference Alignment

Changho Suh, *Student Member, IEEE*, and Kannan Ramchandran, *Fellow, IEEE*

***Abstract*—The high repair cost of (n, k) Maximum Distance Separable (MDS) erasure codes has recently motivated a new class of MDS codes, called Repair MDS codes, that can significantly reduce repair bandwidth over conventional MDS codes. In this paper, we describe (n, k, d) Exact-Repair MDS codes, which allow for any failed node to be repaired exactly with access to d survivor nodes, where $k \leq d \leq n-1$. We construct Exact-Repair MDS codes that are optimal in repair bandwidth for the cases of: (a) $k/n \leq 1/2$ and $d \geq 2k-1$[1]; (b) $k \leq 3$. Our codes are deterministic and require a finite-field size of at most $2(n-k)$. Our constructive codes are based on interference alignment techniques.**

***Index Terms*—Distributed storage, exact-repair MDS codes, interference alignment, network codes.**

I. Introduction

In distributed storage systems, maximum distance separable (MDS) erasure codes are well-known coding schemes that can offer maximum reliability for a given storage overhead. For an (n, k) MDS code for storage, a source file of size $\mathcal{M}$ bits is divided equally into k units (of size $\frac{\mathcal{M}}{k}$ bits each), and these k data units are expanded into n encoded units, and stored at n nodes. The code guarantees that a user or data collector (DC) can reconstruct the source file by connecting to any arbitrary k nodes. In other words, any $(n-k)$ node failures can be tolerated with a minimum storage cost of $\frac{\mathcal{M}}{k}$ at each of n nodes. While MDS codes are optimal in terms of reliability versus storage overhead, they come with a significant maintenance overhead when it comes to repairing failed encoded nodes to restore the MDS system-wide property. Specifically, consider failure of a single encoded node and the cost needed to restore this node. It can be shown that this repair incurs an aggregate cost of $\mathcal{M}$ bits of information from k nodes. Since each encoded unit contains only $\frac{\mathcal{M}}{k}$ bits of information, this represents a k-fold inefficiency with respect to the repair bandwidth.

Manuscript received April 18, 2010; revised August 17, 2010; accepted August 23, 2010. Date of current version February 18, 2011. This research was supported (in part) by the AFOSR under Grant FA9550-09-1-0120, by the DTRA under Grant HDTRA1-09-1-0032, and by the NSF under Grant (CCF-0830788).

C. Suh is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94704 USA (e-mail: chsuh@eecs.berkeley.edu).

K. Ramchandran is with the Wireless Foundation, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94704 USA (e-mail: kannanr@eecs.berkeley.edu).

Communicated by M. Effros, Guest Associate Editor for the Special Issue: Facets of Coding Theory: From Algorithms to Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2011.2105003

[1]In this paper, we assume that all of the survivor systematic nodes participate in the repair.

This challenge has motivated a new class of coding schemes, called Regenerating Codes [1], [2], which target the information-theoretic optimal tradeoff between storage cost and repair bandwidth. Dimakis-Godfrey-Wu-Wainwright-Ramchandran [1], [2] have translated the regenerating-codes problem into a multicast network problem. Employing the network code results in [3]–[5] that well address the multicast network, they have shown that random network coding schemes achieve the optimal repair bandwidth for a given storage cost. On one end of this spectrum of Regenerating Codes are Minimum Storage Regenerating (MSR) codes that can match the minimum storage cost of MDS codes while also significantly reducing repair bandwidth. As shown in [1], [2], the fundamental tradeoff between bandwidth and storage depends on the number of nodes that are connected to repair a failed node, simply called the degee d where $k \leq d \leq n-1$. The optimal tradeoff is characterized by

$$(\alpha, \gamma) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}}{k} \cdot \frac{d}{d-k+1} \right) \tag{1}$$

where α and γ denote the optimal storage cost and repair bandwidth, respectively for repairing a single failed node, while retaining the MDS-code property for the user. Note that this code requires the same minimal storage cost (of size $\frac{\mathcal{M}}{k}$) as that of conventional MDS codes, while substantially reducing repair bandwidth by a factor of $\frac{k(d-k+1)}{d}$ (e.g., for $(n, k, d) = (31, 6, 30)$, there is a 5x bandwidth reduction). This (n, k, d) MSR code can be considered as a Repair MDS code (to be specifically defined in Section II-A) that (a) have an (n, k) MDS-code property; and (b) can repair single-node failures with minimum repair bandwidth given a repair-degree of d. In this paper, we assume that each repair link has the equal bandwidth and its bandwidth $(\frac{\gamma}{d})$ is normalized to 1, making $\mathcal{M} = k(d-k+1)$. One can partition a whole file into smaller chunks so that each has a size of $k(d-k+1)$.

While Repair MDS codes enjoy substantial benefits over conventional MDS codes, they come with some limitations in construction. Specifically, the achievable schemes in [1], [2] that meet the optimal tradeoff bound of (1) restore failed nodes in a *functional* manner only, using a random-network-coding based framework. This means that the replacement nodes maintain the MDS-code property (that any k out of n nodes can allow for the data to be reconstructed) but do not *exactly* replicate the information content of the failed nodes.

Mere functional repair can be limiting. First, in many applications of interest, there is a need to maintain the code in systematic form, i.e., where the user data in the form of k information

units are exactly stored at k nodes and parity information (mixtures of k information units) are stored at the remaining $(n-k)$ nodes. Secondly, under functional repair, additional overhead information needs to be exchanged for continually updating repairing-and-decoding rules whenever a failure occurs. This can significantly increase system overhead. A third problem is that the random-network-coding based solution of [1] can require a huge finite-field size, which can significantly increase the computational complexity of encoding-and-decoding[2]. Lastly, functional repair is undesirable in storage security applications in the face of eavesdroppers. In this case, information leakage occurs continually due to the dynamics of repairing-and-decoding rules that can be potentially observed by eavesdroppers [6].

These drawbacks motivate the need for *exact* repair of failed nodes. This leads to the following question: is there a price for attaining the optimal tradeoff of (1) with the extra constraint of exact repair: i.e., is there an overhead cost in terms of rate needed? Unlike functional repair, this exact-repair problem can be translated into a *nonmulticast* network problem (to be specifically shown in Section II-B) where the cutset bound might not be achievable [7] and linear network codes might not suffice [8]. Due to this nature, the problem has been open in general. The work in [9] sheds some light on this exact-repair problem: specifically, it was shown that under scalar linear codes[3], the optimal tradeoff cannot be achieved when $\frac{k}{n} > \frac{1}{2} + \frac{2}{n}$. For large n, this case boils down to $\frac{k}{n} > \frac{1}{2}$, i.e., redundancy less than two. Now what about for $\frac{k}{n} \leq \frac{1}{2}$? This paper resolves this open problem and shows that it is indeed possible to attain the optimal tradeoff of (1) for the case of $\frac{k}{n} \leq \frac{1}{2}$ and $d \geq 2k-1$, while also guaranteeing exact repair. Here, we assume that all of the survivor systematic nodes participate in the repair. Furthermore, we show that for the special case of $k \leq 3$, there is no price for exact repair, regardless of the value of n. The interesting special case in this class is the (5, 3) code[4], which is not covered by the first case of $\frac{k}{n} \leq \frac{1}{2}$.

Our achievable scheme builds on the concept of interference alignment, which was introduced in the context of wireless communication networks [11], [12]. The idea of interference alignment is to align multiple interference signals in a signal subspace whose dimension is smaller than the number of interferers. Specifically, consider the following setup where a decoder has to decode one desired signal which is linearly interfered with by two separate undesired signals. How many linear equations (relating to the number of channel uses) does the decoder need to recover its desired input signal? As the aggregate signal dimension spanned by desired and undesired signals is at most three, the decoder can naively recover its signal of interest with access to three linearly independent equations in the three unknown signals. However, as the decoder is interested in only one of the three signals, it can decode its desired unknown signal even if it has access to only two equations, provided the two undesired signals are judiciously aligned in a 1-D subspace. See [11]–[13] for details.

We will describe in the sequel how this concept relates intimately to our repair problem. At a high level, the connection comes from our repair problem involving recovery of a subset (related to the subspace spanned by a failed node) of the overall aggregate signal space (related to the entire user data dimension). There are, however, significant differences some beneficial and some detrimental. On the positive side, while in the wireless problem, the equations are provided by nature (in the form of channel gain coefficients), in our repair problem, the coefficients of the equations are man-made choices, representing a part of the overall design space. On the flip side, however, the MDS requirement of our repair code and the multiple failure configurations that need to be simultaneously addressed with a single code design generate multiple interference alignment constraints that need to be simultaneously satisfied. This is particularly acute for a large value of k, as the number of possible failure configurations increases with n (which increases with k). Finally, another difference comes from the finite-field constraint of our repair problem.

We propose a *common-eigenvector* based constructive design framework (to be explained in Section IV) that covers all possible failure configurations. Based on this framework, we develop an interference alignment technique for exact repair. We also propose another interference alignment scheme for a (5, 3) code[5], which in turn shows the optimality of the cutset bound (1) for the case $k \leq 3$. Our coding schemes are deterministic and require a field size of at most $2(n-k)$. This is in stark contrast to the random-network-coding based solutions [1].

II. Problem Statement

A. Definition of Repair MDS Codes Through Code-Design Space Characterization

While conventional MDS erasure codes are completely characterized by their encoding (generator) matrix, Repair MDS codes need more. They require not only the MDS property (as in the classical case), but have the additional repair constraints corresponding to all single-node failure patterns. This makes the code design problem considerably more challenging. We discuss this here by defining Repair MDS codes through their complete code-design space characterization. In the interests of keeping the notation simple without sacrificing the conceptual insights behind this characterization, we will consciously avoid the formalism associated with a general setting, and instead use illuminating examples to illustrate our results while reserving the detailed formal proofs to the appendices.

Consider a simple example of a systematic $(n, k, d) = (4, 2, 3)$ code in Fig. 1. Note that the degree d indicates the number of nodes that are connected to repair a failed node. We introduce matrix notation for illustrative purpose. This code has $k(=2)$ information units. Let $\mathbf{a} = (a_1, \ldots, a_\alpha)^t$ and $\mathbf{b} = (b_1, \ldots, b_\alpha)^t$ be α-dimensional information-unit vectors,

[2]Recall that the regenerating-codes problem can be translated into a multicast communication problem where random-network-coding-based schemes require a huge field size especially for large networks. In storage problems, the field size issue is further aggravated by the need to support a dynamically expanding network size due to the need for continual repair.

[3]In scalar linear codes, symbols are not allowed to be split into arbitrarily small sub-symbols as with vector linear codes. This vector linear code is equivalent to having large block-lengths in the classical setting.

[4]Independently, Cullina-Dimakis-Ho in [10] found (5, 3) Exact-Repair MDS codes defined over $\mathsf{GF}(3)$, based on a search algorithm.

[5]The finite-field nature of the problem makes this challenging.

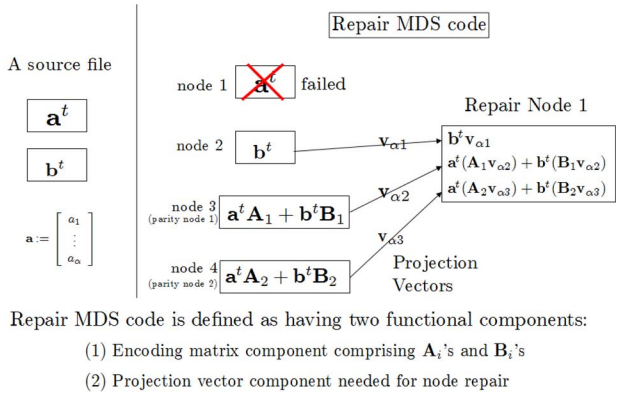


Fig. 1. Definition of a Repair MDS code through the complete characterization of the code design space using the example of a systematic $(n, k, d) = (4, 2, 3)$ code. This is illustrated for the case when systematic node 1 fails, and a unit per-link repair-bandwidth cost is assumed. Let $\mathbf{a} = (a_1, \ldots, a_\alpha)^t$ and $\mathbf{b} = (b_1, \ldots, b_\alpha)^t$ be α-dimensional information-unit vectors, where α denotes the storage cost per node. Systematic node 1 and 2 store uncoded information in the form of row vectors, i.e., $\mathbf{a}^t$ and $\mathbf{b}^t$, respectively. Let $\mathbf{A}_i$ and $\mathbf{B}_i$ be α-by-α encoding submatrices (i.e., $[\mathbf{A}_i; \mathbf{B}_i]$ corresponds to generator submatrices) for parity node $i (i = 1, 2)$. A failed node is repaired through the specification of α-dimensional projection vectors associated with each surviving node that participates in the repair. In the example, $\mathbf{v}_{\alpha i} (i = 1, 2, 3)$ are defined as the projection vectors needed for repair of systematic node 1. A Repair MDS code is thus defined as having two functional components that have to be designed *jointly*: 1) the encoding (generator) matrix associated with the storage nodes; and 2) the projection vectors needed for node repair. Note that in this example, the repair code involves 4 encoding submatrices and 12 projection vectors (3 projection vectors for each of 4 possible failure configurations) that need to be designed jointly.

where α denotes storage cost and $(\cdot)^t$ indicates a transpose. Systematic node 1 and 2 store uncoded information in the form of row vectors, i.e., $\mathbf{a}^t$ and $\mathbf{b}^t$, respectively. Let $\mathbf{A}_i$ and $\mathbf{B}_i$ be α-by-α encoding submatrices (i.e., $[\mathbf{A}_i; \mathbf{B}_i]$ corresponds to generator submatrices) for parity node $i (i = 1, 2)$. For example, parity node 1 stores information in the form of $\mathbf{a}^t\mathbf{A}_1 + \mathbf{b}^t\mathbf{B}_1$. The encoding submatrices for systematic nodes are not explicitly defined, since those are trivially inferred.

A failed node is repaired through the specification of α-dimensional projection vectors associated with each survivor node that participates in the repair. As we assume a unit per-link repair-bandwidth cost $\left(\frac{\gamma}{d} = 1\right)$, each survivor node projects its data into a scalar. In the example, $\mathbf{v}_{\alpha i} (i = 1, 2, 3)$ are defined as the projection vectors needed for repair of systematic node 1. A Repair MDS code is thus defined as having two functional components that have to be designed *jointly*:

1) the encoding (generator) matrix associated with the storage nodes;
2) the projection vectors needed for node repair.

Note that in this example, the repair code involves 4 α-by-α encoding submatrices and 12 projection vectors (3 projection vectors for each of 4 possible failure configurations) that need to be jointly designed.

We categorize the Repair MDS code depending on whether or not the failed nodes are exactly repaired. The code is called a *functional*-repair code if the repaired system maintains the MDS-code property (the repaired node can however be different from that of the failed node). The code is called an *exact*-repair code if the failed nodes are exactly repaired, thus restoring lost encoded fragments with their exact replicas. The code is called a *partial exact*-repair code if only the systematic nodes are repaired exactly, while parity nodes are repaired only functionally. Finally, the code is also called the MSR code that achieves the optimal tradeoff of (1).

The repair problem is to construct the repair code. For instance, the exact-repair problem is to jointly design: 1) the encoding (generator) matrix and 2) the projection vectors such that the failed nodes are exactly repaired.

B. Translation Into a Nonmulticast Network Problem

Unlike functional repair which is equivalent to a multicast network problem [1], [2], the exact-repair problem we study here is a more complicated nonmulticast network problem which in general is an open problem in network coding today. It is known that in general nonmulticast networks, the cutset bound might not be achievable [7] and linear codes might not suffice [8]. In this section, we explicitly show this translation to highlight the difficulty of our exact-repair problem. As we will show in the sequel, we show that exploiting the special structure of our nonmulticast problem due to the exact repair constraints, we can solve the problem for the case of $\frac{k}{n} \leq \frac{1}{2}$ and $d \geq 2k - 1$.

Fig. 2 shows the translation of the (4, 2, 3) Exact-Repair MDS code into a nonmulticast network where destination nodes have asymmetric traffic demands. A source has $k(= 2)$ information units $\mathbf{a}$ and $\mathbf{b}$, each having α symbols. We have $n(= 4)$ storage nodes. The two systematic nodes store $\mathbf{a}^t$ and $\mathbf{b}^t$, respectively, while the two parity nodes store mixtures of $\mathbf{a}$ and $\mathbf{b}$. Here, we consider linear combination mixtures, although the mixtures can also be arbitrary nonlinear functions of the information. We have 4 repair nodes. When node 1 fails, repair node 1 (denoted by R_1) needs to decode $\hat{\mathbf{a}}$ by connecting to $d(= 3)$ survivor nodes. Similarly we have the other three repair nodes. In addition to this, due to the MDS-code constraint, there are $\binom{n}{k} = \binom{4}{2} = 6$ destination nodes which need to decode all of the information units. Clearly the resulting network is a nonmulticast network which contains two types of destination nodes: 1) 4 destination nodes want the individual traffic corresponding to the storage node content; 2) 6 destination nodes have the multicast demand. Therefore, the exact-repair problem is to design a network code which satisfies all of these 10 constraints. Specifically, designing the first component of the repair code corresponds to designing local encoding submatrices for the storage nodes, i.e., $\mathbf{A}_i$'s and $\mathbf{B}_i$'s. The second component corresponds to designing coding coefficients for the links between the storage nodes and repair nodes. Notice that as code parameters (n, k, d) get large, the number of constraints grows exponentially, thereby making the problem harder.

C. Related Work

As stated earlier, Regenerating Codes, which cover an entire spectrum of optimal tradeoffs between repair bandwidth and storage cost, were introduced in [1], [2]. As discussed, Repair MDS codes (also called MSR codes) occupy one end of this spectrum corresponding to minimum storage. At the other end of the spectrum live Minimum Bandwidth Regenerating

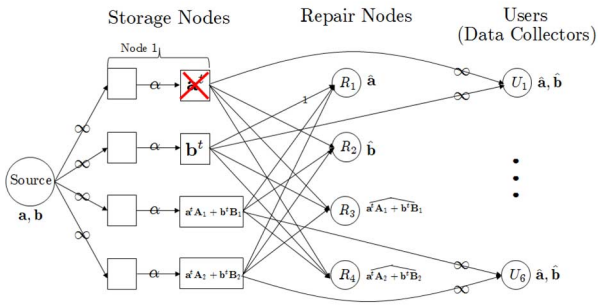


Fig. 2. Translation of the (4, 2, 3) Exact-Repair MDS code into a nonmulticast network problem. A source has $k(=2)$ information units $\mathbf{a}$ and $\mathbf{b}$, each having α symbols. We have $n(=4)$ storage nodes. The two systematic nodes store $\mathbf{a}^t$ and $\mathbf{b}^t$, respectively, while the two parity nodes store mixtures of $\mathbf{a}$ and $\mathbf{b}$. When node 1 fails, repair node 1 (denoted by R_1) needs to decode $\hat{\mathbf{a}}$ by connecting to $d(=3)$ survivor nodes. Similarly we have the other three repair nodes. In addition to this, due to the MDS-code constraint, there are $\binom{n}{k} = \binom{4}{2} = 6$ destination nodes which need to decode all of the information units.

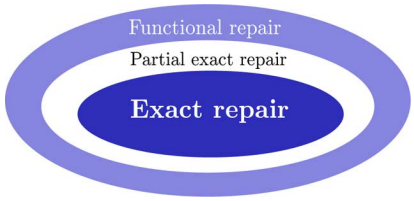


Fig. 3. Repair models for distributed storage systems. In exact-repair, the failed nodes are exactly regenerated, thus restoring lost encoded fragments with their exact replicas. In functional-repair, the requirement is relaxed: the newly generated node can contain different data from that of the failed node as long as the repaired system maintains the MDS-code property. In partial exact-repair, only systematic nodes are repaired exactly, while parity nodes are repaired only functionally.

(MBR) repair codes corresponding to minimum repair bandwidth. The optimal tradeoffs described in [1], [2] are based on random-network-coding based approaches, which guarantee only functional repair.

The topic of exact-repair codes has received attention in the recent literature [9], [10], [14]–[16]. Wu and Dimakis in [14] showed that the MSR point (1) can be attained for the cases of: $k = 2$ and $k = n - 1$. Rashmi-Shah-Kumar-Ramchandran in [15] showed that for $d = n - 1$, the optimal MBR point can be achieved with a deterministic scheme requiring a small finite-field size and zero repair-coding-cost. Subsequently, Shah-Rashmi-Kumar-Ramchandran in [9] developed partial exact-repair codes for the MSR point corresponding to $\frac{k}{n} \leq \frac{1}{2} + \frac{2}{n}$, where exact repair is limited to the systematic component of the code. See Fig. 3. Finding the fundamental limits under exact repair of *all* nodes (including parity) remained an open problem. A key contribution of this paper is to resolve this open problem by constructing Exact-Repair MDS codes that attain the optimal tradeoff of (1) for the case of $\frac{k}{n} \leq \frac{1}{2}$ and $d \geq 2k - 1$. Here, we assume that the d helper nodes participating in the repair contain all of the survivor systematic nodes.Our result covers an important operating point $d = n - 1$ where the minimum repair bandwidth can be achieved. See (1). For the general case (e.g., $\frac{k}{n} > \frac{1}{2}$ or $k + 1 \leq d < 2k - 1$), developing constructive codes remains an open problem.

The constructive framework proposed in [9] forms the inspiration for our proposed solution in this paper. Indeed, we show that the partial exact-repair code introduced in [4] (meant for exact repair of the systematic nodes only) can also be used to repair the nonsystematic (parity) node failures exactly provided the second component of the repair code (i.e., the projection vectors needed for node repair) are appropriately designed. Designing the projection-vectors of exact repair codes is challenging and had remained an open problem: resolving this for the case of $\frac{k}{n} \leq \frac{1}{2}$ and $d \geq 2k - 1$ is a key contribution of this work. Another contribution of our work is the systematic development of a family of code structures. This family of codes provides conceptual insights into the structure of solutions for the exact repair problem, while also offering a new large constructive design space of solutions.

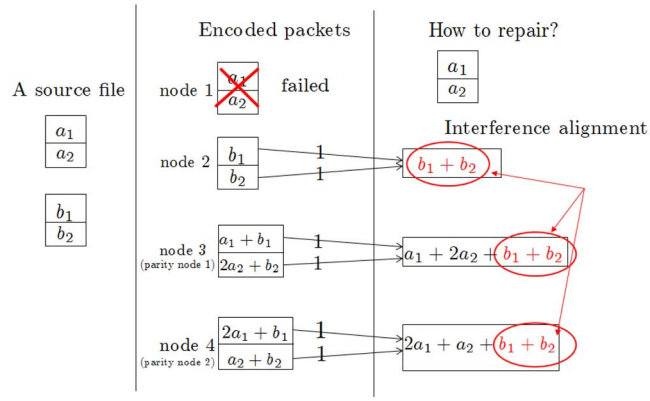


Fig. 4. Interference alignment for a (4,2,3) Exact-Repair MDS code defined over $\mathsf{GF}(5)$ [14]. Designing appropriate projection vectors, we can align interference space of (b_1, b_2) into 1-D linear space spanned by $[1, 1]^t$. As a result, we can successfully decode 2 desired unknowns (a_1, a_2) from 3 equations containing 4 unknowns (a_1, a_2, b_1, b_2).

III. Interference Alignment for Distributed Storage Repair

Network coding [3]–[5] (that allows multiple messages to be combined at network nodes) has been established recently as a useful tool for addressing interference issues even in wireline networks where all the communication links are orthogonal and noninterfering. This attribute was first observed in [14], where it was shown that interference alignment could be exploited for storage networks, specifically for Exact-Repair MDS codes having small $k(k = 2)$. However, generalizing interference alignment to large values of k (even $k = 3$) proves to be challenging, as we describe in the sequel. In order to appreciate this better, let us first review the scheme of [14] that was applied to the exact repair problem. We will then address the difficulty of extending interference alignment for larger systems and describe how to address this in Section IV.

A. Review of (4, 2) Exact-Repair MDS Codes [14]

Fig. 4 illustrates an interference alignment scheme for a (4, 2, 3) Exact-Repair MDS code defined over $\mathsf{GF}(5)$. First, one can easily check the MDS property of the code, i.e., all the source files can be reconstructed from any $k(=2)$ nodes out of $n(=4)$ nodes. Let us see how failed node 1 (storing (a_1, a_2)) can be exactly repaired. Assume a source file size $\mathcal{M}$ is 4 and repair-bandwidth-per-link $\frac{\gamma}{d} = 1$. The cutset bound (1) then gives the fundamental limits of storage cost $\alpha = 2$.

The example illustrated in Fig. 4 shows that the parameter set described above is achievable using interference alignment.

Here is a summary of the scheme. First notice that since the bandwidth-per-link is 1, each survivor node uses a projection vector to project its data into a scalar. Choosing appropriate projection vectors, we get the equations: $(b_1 + b_2)$; $a_1 + 2a_2 + (b_1 + b_2)$; $2a_1 + a_2 + (b_1 + b_2)$. Observe that the undesired signals (b_1, b_2) (interference) are aligned onto an 1-D linear subspace, thereby achieving interference alignment. Therefore, we can successfully decode (a_1, a_2) with three equations although there are four unknowns. Similarly, we can repair (b_1, b_2) when it has failed.

For parity node repair, a remapping technique is introduced. The idea is to define parity node symbols with new variables as follows:

$$\text{Node 3: } a_1' := a_1 + b_1; a_2' := 2a_2 + b_2$$
$$\text{Node 4: } b_1' := 2a_1 + b_1; b_2' := a_2 + b_2.$$

We can then rewrite (a_1, a_2) and (b_1, b_2) with respect to (a_1', a_2') and (b_1', b_2'). In terms of prime notation, parity nodes turn into systematic nodes and vice versa. With this remapping, one can easily design projection vectors for exact repair of parity nodes.

B. Geometric Interpretation

Using matrix notation, we provide geometric interpretation of interference alignment for the same example in Fig. 4. Let $\mathbf{a} = (a_1, a_2)^t$ and $\mathbf{b} = (b_1, b_2)^t$ be 2-D information-unit vectors. Let $\mathbf{A}_i$ and $\mathbf{B}_i$ be 2-by-2 encoding submatrices for parity node $i (i = 1, 2)$. Define 2-D projection vectors $\mathbf{v}_{\alpha i}$'s $(i = 1, 2, 3)$.

Let us consider exact repair of systematic node 1. By connecting to three nodes, we get: $\mathbf{b}^t\mathbf{v}_{\alpha 1}$; $\mathbf{a}^t(\mathbf{A}_1\mathbf{v}_{\alpha 2}) + \mathbf{b}^t(\mathbf{B}_1\mathbf{v}_{\alpha 2})$; $\mathbf{a}^t(\mathbf{A}_2\mathbf{v}_{\alpha 3}) + \mathbf{b}^t(\mathbf{B}_2\mathbf{v}_{\alpha 3})$. Recall the goal of decoding 2 desired unknowns out of 3 equations including 4 unknowns. To achieve this goal, we need

$$\text{rank}\left(\begin{bmatrix}(\mathbf{A}_1\mathbf{v}_{\alpha 2})^t \\ (\mathbf{A}_2\mathbf{v}_{\alpha 3})^t\end{bmatrix}\right) = 2;\ \text{rank}\left(\begin{bmatrix}\mathbf{v}_{\alpha 1}^t \\ (\mathbf{B}_1\mathbf{v}_{\alpha 2})^t \\ (\mathbf{B}_2\mathbf{v}_{\alpha 3})^t\end{bmatrix}\right) = 1. \quad (2)$$

The second condition can be met by setting $\mathbf{v}_{\alpha 2} = \mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$ and $\mathbf{v}_{\alpha 3} = \mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}$. This choice forces the interference space to be collapsed into a 1-D linear subspace, thereby achieving interference alignment. With this setting, the first condition now becomes

$$\text{rank}\left([\,\mathbf{A}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1} \quad \mathbf{A}_2\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}\,]\right) = 2. \quad (3)$$

It can be easily verified that the choice of $\mathbf{A}_i$'s and $\mathbf{B}_i$'s given in Figs. 4 and 5 guarantees the above condition. When the node 2 fails, we get a similar condition

$$\text{rank}\left([\mathbf{B}_1\mathbf{A}_1^{-1}\mathbf{v}_{\beta 1} \quad \mathbf{B}_2\mathbf{A}_2^{-1}\mathbf{v}_{\beta 1}\,]\right) = 2 \quad (4)$$

where $\mathbf{v}_{\beta i}$'s denote projection vectors for node 2 repair. This condition also holds under the given choice of encoding matrices. With this remapping, one can easily design projection vectors for exact repair of parity nodes.

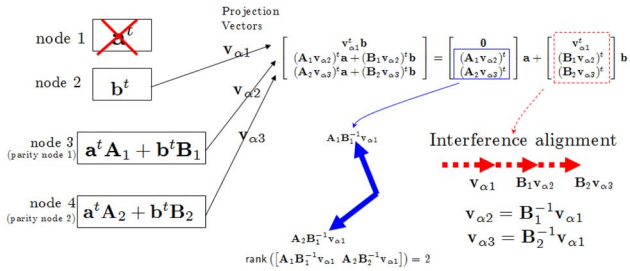


Fig. 5. Geometric interpretation of interference alignment. The blue solid-line and red dashed-line vectors indicate linear subspaces with respect to "**a**" and "**b**", respectively. The choice of $\mathbf{v}_{\alpha 2} = \mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$ and $\mathbf{v}_{\alpha 3} = \mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}$ enables interference alignment. For the specific example of Fig. 4, the corresponding encoding submatrices are $\mathbf{A}_1 = [1, 0; 0, 2]$, $\mathbf{B}_1 = [1, 0; 0, 1]$. $\mathbf{A}_2 = [2, 0; 0, 1]$, $\mathbf{B}_2 = [1, 0; 0, 1]$.

C. Connection With Interference Channels in Communication Problems

Observe the three equations shown in Fig. 5, as follows:

$$\underbrace{\begin{bmatrix}\mathbf{0} \\ (\mathbf{A}_1\mathbf{v}_{\alpha 2})^t \\ (\mathbf{A}_2\mathbf{v}_{\alpha 3})^t\end{bmatrix}\mathbf{a}}_{desired\ signals} + \underbrace{\begin{bmatrix}\mathbf{v}_{\alpha 1}^t \\ (\mathbf{B}_1\mathbf{v}_{\alpha 2})^t \\ (\mathbf{B}_2\mathbf{v}_{\alpha 3})^t\end{bmatrix}\mathbf{b}}_{interference}.$$

Separating into two parts, we can view this problem as a wireless communication problem, wherein a subset of the information is desired to be decoded in the presence of interference. Note that for each term (e.g., $\mathbf{A}_1\mathbf{v}_{\alpha 2}$), the matrix $\mathbf{A}_1$ and vector $\mathbf{v}_{\alpha 2}$ correspond to channel matrix and transmission vector in wireless communication problems, respectively.

There are, however, significant differences. In the wireless communication problem, the channel matrices are provided by nature and therefore not controllable. The transmission strategy alone (vector variables) can be controlled for achieving interference alignment. On the other hand, in our storage repair problems, both matrices and vectors are controllable, i.e., projection vectors and encoding submatrices can be arbitrarily designed, resulting in more flexibility. However, our storage repair problem comes with unparalleled challenges due to the MDS requirement and the multiple failure configurations. These induce multiple interference alignment constraints that need to be simultaneously satisfied. What makes this difficult is that the encoding submatrices, once designed, must be the same for all repair configurations. This is particularly acute for large values of k (even $k = 3$), as the number of possible failure configurations increases with n (which increases with k).

IV. Proposed Framework

We propose a *common-eigenvector* based constructive design framework to address the exact repair problem. This framework draws its inspiration from the work in [9] which guarantees the exact repair of systematic nodes, while satisfying the MDS code property, but which does not provide exact repair of failed parity nodes. In providing a constructive solution for the exact repair of *all* nodes, we use geometric insights to propose a large family

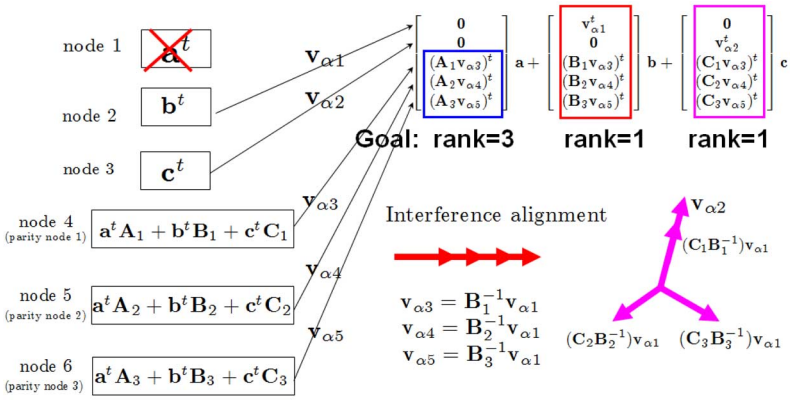


Fig. 6. Difficulty of achieving interference alignment simultaneously.

of repair codes. This both provides insights into the structure of codes for exact repair of all nodes (particularly the projection-vectors code component), as well as opens up a rich and large design space for constructive solutions. Specifically, we propose a common-eigenvector based approach building on a certain *elementary matrix* property [17], [18]. This structure provides the key geometric insights needed to facilitate the design of the key projection-vectors code component of exact repair codes. Moreover, our proposed coding schemes are deterministic and constructive, requiring a symbol alphabet-size of at most $(2n-2k)$.

Our framework consists of four components: 1) developing a family of codes[6] for exact repair of systematic codes based on the common-eigenvector concept; 2) drawing a *dual* relationship between the systematic and parity node repair; 3) guaranteeing the MDS-code property; and 4) constructing codes with finite-field alphabets. Step (2) of our framework is a significant distinction from that of [9] and is needed to tackle the full exact repair problem not addressed there. The framework covers the case of $n \geq 2k$ and $d \geq 2k-1$. It turns out that the $(2k, k, 2k-1)$ code case contains the key design ingredients and the case of $n \geq 2k$ and $d \geq 2k-1$ can be derived from this (see Section VI). Hence, we first focus on the simplest example: (6, 3, 5) Exact-Repair MDS codes. Later in Section VI, we will generalize this to arbitrary (n, k, d) repair codes in the class.

A. Systematic Node Repair

For $k \geq 3$ (more-than-two interfering information units), achieving interference alignment for exact repair turns out to be significantly more complex than the $k=2$ case. Fig. 6 illustrates this difficulty through the example of repairing node 1 for a (6, 3, 5) code. By the optimal tradeoff (1), the choice of $\mathcal{M}=9$ and $\frac{\gamma}{d}=1$ gives $\alpha=3$. Let $\mathbf{a}=(a_1,a_2,a_3)^t$, $\mathbf{b}=(b_1,b_2,b_3)^t$ and $\mathbf{c}=(c_1,c_2,c_3)^t$. We define 3-by-3 encoding submatrices of $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{C}_i$ (for $i=1,2,3$); and 3-D projection vectors $\mathbf{v}_{\alpha i}$'s.

[6]Recall that our repair code consists of two components: 1) the encoding (generator) matrix; 2) the projection vectors needed for node repair. Interestingly, the encoding matrix component of the code in [9] turns out to work for the exact repair of both systematic and parity nodes provided the second component of the repair code (projection vectors needed for repair) are appropriately designed.

Consider the $5(=d)$ equations downloaded from the nodes

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ (\mathbf{A}_1\mathbf{v}_{\alpha 3})^t \\ (\mathbf{A}_2\mathbf{v}_{\alpha 4})^t \\ (\mathbf{A}_3\mathbf{v}_{\alpha 5})^t \end{bmatrix} \mathbf{a} + \begin{bmatrix} \mathbf{v}_{\alpha 1}^t \\ \mathbf{0} \\ (\mathbf{B}_1\mathbf{v}_{\alpha 3})^t \\ (\mathbf{B}_2\mathbf{v}_{\alpha 4})^t \\ (\mathbf{B}_3\mathbf{v}_{\alpha 5})^t \end{bmatrix} \mathbf{b} + \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_{\alpha 2}^t \\ (\mathbf{C}_1\mathbf{v}_{\alpha 3})^t \\ (\mathbf{C}_2\mathbf{v}_{\alpha 4})^t \\ (\mathbf{C}_3\mathbf{v}_{\alpha 5})^t \end{bmatrix} \mathbf{c}.$$

In order to successfully recover the desired signal components of "$\mathbf{a}$", the matrices associated with $\mathbf{b}$ and $\mathbf{c}$ should have rank 1, respectively, while the matrix associated with $\mathbf{a}$ should have full rank of 3. In accordance with the (4, 2, 3) code example in Fig. 5, if one were to set $\mathbf{v}_{\alpha 3}=\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$, $\mathbf{v}_{\alpha 4}=\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}$ and $\mathbf{v}_{\alpha 5}=\mathbf{B}_3^{-1}\mathbf{v}_{\alpha 1}$, then it is possible to achieve interference alignment with respect to $\mathbf{b}$. However, this choice also specifies the interference space of $\mathbf{c}$. If the $\mathbf{B}_i$'s and $\mathbf{C}_i$'s are not designed judiciously, interference alignment is not guaranteed for $\mathbf{c}$. Hence, it is not evident how to achieve interference alignment at the same time.

In order to address the challenge of simultaneous interference alignment, we invoke a common eigenvector concept. The idea consists of two parts: (i) designing the $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$'s such that $\mathbf{v}_1$ is a common eigenvector of the $\mathbf{B}_i$'s and $\mathbf{C}_i$'s, but not of $\mathbf{A}_i$'s[7]; (ii) repairing by having survivor nodes *project* their data onto a linear subspace spanned by this common eigenvector $\mathbf{v}_1$. We can then achieve interference alignment for $\mathbf{b}$ and $\mathbf{c}$ at the same time, by setting $\mathbf{v}_{\alpha i}=\mathbf{v}_1$, $\forall i$. As long as $[\mathbf{A}_1\mathbf{v}_1, \mathbf{A}_2\mathbf{v}_1, \mathbf{A}_3\mathbf{v}_1]$ is invertible, we can also guarantee the decodability of $\mathbf{a}$. See Fig. 7.

The challenge is now to design encoding submatrices to guarantee the existence of a common eigenvector while also satisfying the decodability of desired signals. The difficulty comes from the fact that in our (6, 3, 5) repair code example, these constraints need to be satisfied for *all* six possible failure configurations. The structure of elementary matrices [17], [18] (generalized matrices of Householder and Gauss matrices) gives insights into this. To see this, consider a 3-by-3 elementary matrix $\mathbf{A}$

$$\mathbf{A}=\mathbf{u}\mathbf{v}^t+\alpha\mathbf{I} \tag{5}$$

[7]Of course, five additional constraints also need to be satisfied for the other five failure configurations for this (6,3,5) code example.

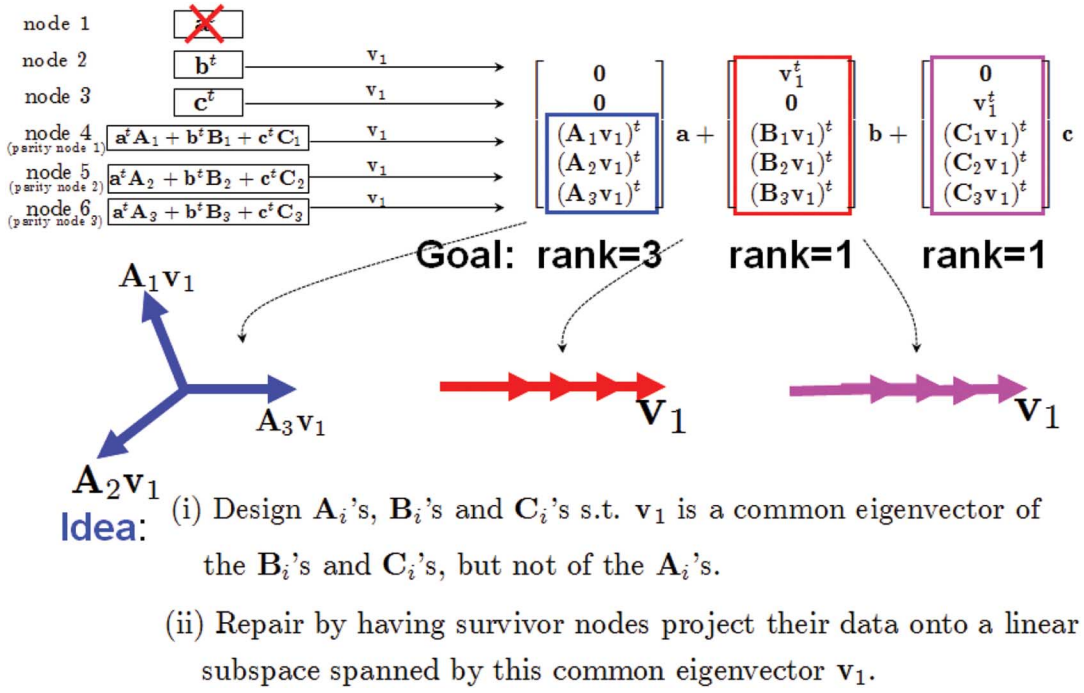


Fig. 7. Illustration of exact repair of systematic node 1 for (6,3,5) exact-repair MDS codes. The idea consists of two parts: (i) designing $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$'s such that $\mathbf{v}_1$ is a common eigenvector of the $\mathbf{B}_i$'s and $\mathbf{C}_i$'s, but not of $\mathbf{A}_i$'s; (ii) repairing by having survivor nodes project their data onto a linear subspace spanned by this common eigenvector $\mathbf{v}_1$.

where $\mathbf{u}$ and $\mathbf{v}$ are 3-D vectors. Here is an observation that motivates our proposed structure: the dimension of the null space of $\mathbf{v}$ is 2 and the null vector $\mathbf{v}^\perp$ is an eigenvector of $\mathbf{A}$, i.e., $\mathbf{A}\mathbf{v}^\perp = \alpha\mathbf{v}^\perp$. This motivates the following structure:

$$\begin{aligned}
\mathbf{A}_1 &= \mathbf{u}_1\mathbf{v}_1^t + \alpha_1\mathbf{I}; \mathbf{B}_1 = \mathbf{u}_1\mathbf{v}_2^t + \beta_1\mathbf{I} \\
\mathbf{C}_1 &= \mathbf{u}_1\mathbf{v}_3^t + \gamma_1\mathbf{I} \\
\mathbf{A}_2 &= \mathbf{u}_2\mathbf{v}_1^t + \alpha_2\mathbf{I}; \mathbf{B}_2 = \mathbf{u}_2\mathbf{v}_2^t + \beta_2\mathbf{I} \\
\mathbf{C}_2 &= \mathbf{u}_2\mathbf{v}_3^t + \gamma_2\mathbf{I} \\
\mathbf{A}_3 &= \mathbf{u}_3\mathbf{v}_1^t + \alpha_3\mathbf{I}; \mathbf{B}_3 = \mathbf{u}_3\mathbf{v}_2^t + \beta_3\mathbf{I} \\
\mathbf{C}_3 &= \mathbf{u}_3\mathbf{v}_3^t + \gamma_3\mathbf{I}
\end{aligned} \tag{6}$$

where $\mathbf{v}_i$'s are 3-D linearly independent vectors and so are $\mathbf{u}_i$'s. The values of the α_i's, β_i's and γ_i's can be arbitrary nonzero values. First consider the simple design where the $\mathbf{v}_i$'s are *orthonormal*. This is for conceptual simplicity. Later we will generalize to the case where the $\mathbf{v}_i$'s need not be orthogonal but only linearly independent. We see that for $i = 1, 2, 3$,

$$\begin{aligned}
\mathbf{A}_i\mathbf{v}_1 &= \alpha_i\mathbf{v}_1 + \mathbf{u}_i \\
\mathbf{B}_i\mathbf{v}_1 &= \beta_i\mathbf{v}_1 \\
\mathbf{C}_i\mathbf{v}_1 &= \gamma_i\mathbf{v}_1.
\end{aligned} \tag{7}$$

Importantly, notice that $\mathbf{v}_1$ is a common eigenvector of the $\mathbf{B}_i$'s and $\mathbf{C}_i$'s, while simultaneously ensuring that the vectors of $\mathbf{A}_i\mathbf{v}_1$ are linearly independent. Hence, setting $\mathbf{v}_{\alpha i} = \mathbf{v}_1$ for all i, it is possible to achieve simultaneous interference alignment while also guaranteeing the decodability of the desired signals. See Fig. 7. On the other hand, this structure also guarantees exact repair for $\mathbf{b}$ and $\mathbf{c}$. We use $\mathbf{v}_2$ for exact repair of $\mathbf{b}$. It is a common eigenvector of the $\mathbf{C}_i$'s and $\mathbf{A}_i$'s, while ensuring $[\mathbf{B}_1\mathbf{v}_2, \mathbf{B}_2\mathbf{v}_2, \mathbf{B}_3\mathbf{v}_2]$ invertible. Similarly, $\mathbf{v}_3$ is used for exact repair of $\mathbf{c}$.

We will see that a *dual basis* property gives insights into the general case where $\{\mathbf{v}\} := (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is not orthogonal but only linearly independent. In this case, defining a dual basis $\{\mathbf{v}'\} := (\mathbf{v}_1', \mathbf{v}_2', \mathbf{v}_3')$ gives the solution

$$\begin{bmatrix} \mathbf{v}_1'^t \\ \hline \mathbf{v}_2'^t \\ \hline \mathbf{v}_3'^t \end{bmatrix} := [\,\mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3\,]^{-1}.$$

The definition gives the following property: $\mathbf{v}_i'^t\mathbf{v}_j = \delta(i-j)$, $\forall i, j$. Using this property, one can see that $\mathbf{v}_1'$ is a common eigenvector of the $\mathbf{B}_i$'s and $\mathbf{C}_i$'s while ensuring the invertibility of the desired signals $\mathbf{a}$

$$\begin{aligned}
\mathbf{A}_i\mathbf{v}_1' &= \alpha_i\mathbf{v}_1' + \mathbf{u}_i \\
\mathbf{B}_i\mathbf{v}_1' &= \beta_i\mathbf{v}_1' \\
\mathbf{C}_i\mathbf{v}_1' &= \gamma_i\mathbf{v}_1'.
\end{aligned} \tag{8}$$

So it can be used as a projection vector for exact repair of $\mathbf{a}$. Similarly, we can use $\mathbf{v}_2'$ and $\mathbf{v}_3'$ for exact repair of $\mathbf{b}$ and $\mathbf{c}$, respectively.

B. Dual Relationship Between Systematic and Parity Node Repair

We have seen so far how to ensure exact repair of the systematic nodes. We have known that if $\{\mathbf{v}\}$ is linearly independent and so $\{\mathbf{u}\}$ is, then using the structure of (6) together with projection vectors enables repair, for arbitrary values of $(\alpha_i, \beta_i, \gamma_i)$'s. A natural question is now: will this structure also guarantee exact repair of parity nodes? It turns out that for exact

repair of all nodes, we need a special relationship between $\{\mathbf{v}\}$ and $\{\mathbf{u}\}$ through the correct choice of the $(\alpha_i, \beta_i, \gamma_i)$'s.

We will show that parity nodes can be repaired by drawing a *dual* relationship with systematic nodes. The procedure has two steps. The first is to remap parity nodes with $\mathbf{a}'$, $\mathbf{b}'$, and $\mathbf{c}'$, respectively:

$$\begin{bmatrix}\mathbf{a}'\\ \mathbf{b}'\\ \mathbf{c}'\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1^t & \mathbf{B}_1^t & \mathbf{C}_1^t\\ \mathbf{A}_2^t & \mathbf{B}_2^t & \mathbf{C}_2^t\\ \mathbf{A}_3^t & \mathbf{B}_3^t & \mathbf{C}_3^t\end{bmatrix}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{c}\end{bmatrix}.$$

Systematic nodes can then be rewritten in terms of the prime notations

$$\begin{aligned}\mathbf{a}^t &= \mathbf{a}'^t\mathbf{A}_1' + \mathbf{b}'^t\mathbf{B}_1' + \mathbf{c}'^t\mathbf{C}_1'\\ \mathbf{b}^t &= \mathbf{a}'^t\mathbf{A}_2' + \mathbf{b}'^t\mathbf{B}_2' + \mathbf{c}'^t\mathbf{C}_2'\\ \mathbf{c}^t &= \mathbf{a}'^t\mathbf{A}_3' + \mathbf{b}'^t\mathbf{B}_3' + \mathbf{c}'^t\mathbf{C}_3'\end{aligned} \tag{9}$$

where the newly mapped encoding submatrices $(\mathbf{A}_i', \mathbf{B}_i', \mathbf{C}_i)$'s are defined as

$$\begin{bmatrix}\mathbf{A}_1' & \mathbf{A}_2' & \mathbf{A}_3'\\ \mathbf{B}_1' & \mathbf{B}_2' & \mathbf{B}_3'\\ \mathbf{C}_1' & \mathbf{C}_2' & \mathbf{C}_3'\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3\\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3\\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3\end{bmatrix}^{-1}. \tag{10}$$

With this remapping, one can dualize the relationship between systematic and parity node repair. Specifically, if all of the $\mathbf{A}_i'$'s, $\mathbf{B}_i'$'s, and $\mathbf{C}_i'$'s are *elementary matrices* and form a similar structure as in (6), exact repair of the parity nodes becomes transparent.

The challenge is now how to guarantee the dual structure. In Lemma 1, we show that a special relationship between $\{\mathbf{u}\}$ and $\{\mathbf{v}\}$ through $(\alpha_i, \beta_i, \gamma_i)$'s can guarantee this dual relationship of (13).

Lemma 1: Suppose

$$\mathbf{P} := \left[\begin{array}{c|c|c}\alpha_1 & \alpha_2 & \alpha_3\\ \beta_1 & \beta_2 & \beta_3\\ \gamma_1 & \gamma_2 & \gamma_3\end{array}\right] \text{ is invertible.} \tag{11}$$

Also assume

$$\kappa\mathbf{U} = \mathbf{V}'\mathbf{P}. \tag{12}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$, $\mathbf{V}' = [\mathbf{v}_1', \mathbf{v}_2', \mathbf{v}_3']$, $\{\mathbf{v}'\} := \{\mathbf{v}_1', \mathbf{v}_2', \mathbf{v}_3'\}$ is the dual basis of $\{\mathbf{v}\}$, i.e., $\mathbf{v}_i'^t\mathbf{v}_j = \delta(i-j)$ and κ is an arbitrary nonzero value s.t. $1-\kappa^2 \neq 0$. Then, we can obtain the following structure dual to (6):

$$\begin{cases}\mathbf{A}_1' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_1'\mathbf{u}_1'^t - \kappa^2\alpha_1'\mathbf{I}\right)\\ \mathbf{A}_2' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_2'\mathbf{u}_1'^t - \kappa^2\beta_1'\mathbf{I}\right)\\ \mathbf{A}_3' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_3'\mathbf{u}_1'^t - \kappa^2\gamma_1'\mathbf{I}\right)\end{cases}$$
$$\begin{cases}\mathbf{B}_1' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_1'\mathbf{u}_2'^t - \kappa^2\alpha_2'\mathbf{I}\right)\\ \mathbf{B}_2' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_2'\mathbf{u}_2'^t - \kappa^2\beta_2'\mathbf{I}\right)\\ \mathbf{B}_3' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_3'\mathbf{u}_2'^t - \kappa^2\gamma_2'\mathbf{I}\right)\end{cases}$$
$$\begin{cases}\mathbf{C}_1' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_1'\mathbf{u}_3'^t - \kappa^2\alpha_3'\mathbf{I}\right)\\ \mathbf{C}_2' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_2'\mathbf{u}_3'^t - \kappa^2\beta_3'\mathbf{I}\right)\\ \mathbf{C}_3' = \frac{1}{1-\kappa^2}\left(\mathbf{v}_3'\mathbf{u}_3'^t - \kappa^2\gamma_3'\mathbf{I}\right)\end{cases} \tag{13}$$

where $\{\mathbf{u}'\}$ is the dual basis of $\{\mathbf{u}\}$, i.e., $\mathbf{u}_i'^t\mathbf{u}_j = \delta(i-j)$ and $(\alpha_i', \beta_i', \gamma_i')$'s are the dual basis vectors of $(\alpha_i, \beta_i, \gamma_i)$'s, i.e., $< (\alpha_i', \beta_i', \gamma_i'), (\alpha_j, \beta_j, \gamma_j) >= \delta(i-j)$

$$\left[\begin{array}{ccc}\alpha_1' & \beta_1' & \gamma_1'\\ \hline \alpha_2' & \beta_2' & \gamma_2'\\ \hline \alpha_3' & \beta_3' & \gamma_3'\end{array}\right] := \left[\begin{array}{c|c|c}\alpha_1 & \alpha_2 & \alpha_3\\ \beta_1 & \beta_2 & \beta_3\\ \gamma_1 & \gamma_2 & \gamma_3\end{array}\right]^{-1}. \tag{14}$$

Proof: See Appendix A. ∎

Remark 1: The dual structure (13) now gives projection vector solutions for parity node repair. For exact repair of parity node 1, we can use vector $\mathbf{u}_1$ (a common eigenvector of the $\mathbf{B}_i'$'s and $\mathbf{C}_i'$'s), since it enables simultaneous interference alignment for $\mathbf{b}'$ and $\mathbf{c}'$, while ensuring the decodability of $\mathbf{a}'$. See Fig. 8. Notice that more conditions of (11) and (12) are added to ensure exact repair of all nodes, while these conditions were unnecessary for exact repair of systematic nodes only. Also note that these are only sufficient conditions.

Remark 2: Note that the dual structure (13) is quite similar to the primary structure (6). The only difference is that in the dual structure, $\{\mathbf{u}\}$ and $\{\mathbf{v}\}$ are interchanged to form a *transpose-like* structure. This reveals insights into how to design projection vectors for exact repair of parity nodes in a transparent manner.

C. The MDS-Code Property

The third part of the framework is to guarantee the MDS-code property, which allows us to identify specific constraints on the $(\alpha_i, \beta_i, \gamma_i)$'s and/or $(\{\mathbf{v}\}, \{\mathbf{u}\})$. Consider four cases, associated in the data collector (DC) who is intended in the source file data: (a) 3 systematic nodes; (b) 3 parity nodes; (c) 2 systematic and 1 parity nodes; (d) 1 systematic and 2 parity nodes.

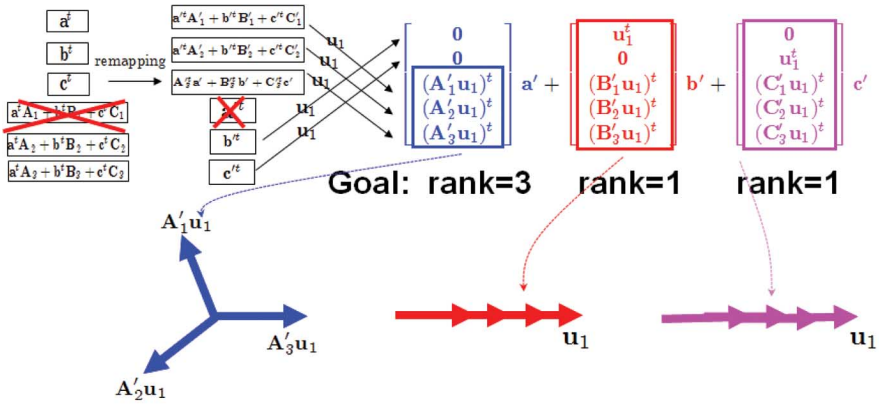
The first is a trivial case. The second case has been already verified in the process of forming the dual structure (13). The invertibility condition of (11) together with (12) suffices to ensure the invertibility of the composite matrix $[\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3; \mathbf{B}_1\mathbf{B}_2\mathbf{B}_3; \mathbf{C}_1\mathbf{C}_2\mathbf{C}_3]$. The third case requires the invertibility of all of each encoding submatrix. In this case, it is necessary that the α_i's, β_i's and γ_i's are nonzero values; otherwise, each encoding submatrix has rank 1. Also the nonzero values together with (12) guarantee the invertibility of each encoding submatrix. To see this, for example, consider

$$\begin{aligned}\mathbf{V}^t\mathbf{A}_1\mathbf{V}' &= \left(\mathbf{V}^t\mathbf{u}_1\right)\mathbf{e}_1^t + \alpha_1\mathbf{I}\\ &= \begin{bmatrix}\frac{\alpha_1}{\kappa} + \alpha_1 & 0 & 0\\ \frac{\beta_1}{\kappa} & \alpha_1 & 0\\ \frac{\gamma_1}{\kappa} & 0 & \alpha_1\end{bmatrix}\end{aligned}$$

where the second equality follows from $\mathbf{v}_1^t\mathbf{u}_1 = \frac{\alpha_1}{\kappa}$, $\mathbf{v}_2^t\mathbf{u}_1 = \frac{\beta_1}{\kappa}$ and $\mathbf{v}_3^t\mathbf{u}_1 = \frac{\gamma_1}{\kappa}$ due to (12). Here, $\mathbf{e}_1$ indicates a standard basis, i.e., $\mathbf{e}_1 = (1,0,0)^t$. Clearly, this resulting matrix is invertible. Since $\mathbf{V}$ is invertible, so is $\mathbf{A}_1$.

The last case requires some nontrivial work. Consider a specific example where the DC connects to nodes 3, 4, and 5. In this case, we first recover $\mathbf{c}$ from node 3 and subtract the terms associated with $\mathbf{c}$ from nodes 4 and 5. We then get

$$\begin{bmatrix}\mathbf{a}^t & \mathbf{b}^t\end{bmatrix}\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\\ \mathbf{B}_1 & \mathbf{B}_2\end{bmatrix}. \tag{15}$$



We provide sufficient conditions to ensure the *dual* structure:

$$(i)\ \kappa[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] = [\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3] \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix};$$

$$(ii) \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \text{ is invertible.}$$

Fig. 8. Exact repair of parity node 1 for a (6, 3, 5) exact-repair MDS code. The idea is to construct the *dual* structure of (13) by remapping parity nodes and then adding sufficient conditions of (11) and (12).

Now consider

$$\begin{bmatrix} \mathbf{V}^t & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^t \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{B}_1 & \mathbf{B}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}' & \mathbf{0} \\ \mathbf{0} & \mathbf{V}' \end{bmatrix}$$

$$= \left[\begin{array}{ccc|ccc} \frac{\alpha_1}{\kappa}+\alpha_1 & 0 & 0 & \frac{\alpha_2}{\kappa}+\alpha_2 & 0 & 0 \\ \frac{\beta_1}{\kappa} & \alpha_1 & 0 & \frac{\beta_2}{\kappa} & \alpha_2 & 0 \\ \frac{\gamma_1}{\kappa} & 0 & \alpha_1 & \frac{\gamma_2}{\kappa} & 0 & \alpha_2 \\ \hline \beta_1 & \frac{\alpha_1}{\kappa} & 0 & \beta_2 & \frac{\alpha_2}{\kappa} & 0 \\ 0 & \beta_1+\frac{\beta_1}{\kappa} & 0 & 0 & \beta_2+\frac{\beta_2}{\kappa} & 0 \\ 0 & \frac{\gamma_1}{\kappa} & \beta_1 & 0 & \frac{\gamma_2}{\kappa} & \beta_2 \end{array}\right]$$

where the equality follows from the fact that $\mathbf{V}^t\mathbf{A}_i\mathbf{V}' = (\mathbf{V}^t\mathbf{u}_i)\,\mathbf{e}_1^t + \alpha_i\mathbf{I}$ and $\mathbf{V}^t\mathbf{B}_i\mathbf{V}' = (\mathbf{V}^t\mathbf{u}_i)\,\mathbf{e}_2^t + \beta_i\mathbf{I}$, for $i = 1, 2$. Using a Gaussian elimination method, one can now easily show that this resulting matrix is invertible and so is $[\mathbf{A}_1\mathbf{A}_2; \mathbf{B}_1\mathbf{B}_2]$ if

$$\mathbf{P}_2 := \left[\begin{array}{c|c} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{array}\right] \text{ is invertible.} \tag{16}$$

Considering the above 4 cases, the following condition together with (11) and (12) suffices for guaranteeing the MDS-code property:

$$\textit{Any submatrix of } \mathbf{P} \textit{ of } (11) \textit{ is invertible.} \tag{17}$$

D. Code Construction With Finite-Field Alphabets

The last part is to design $\mathbf{P}$ of (11) and $\{\mathbf{v}\} := (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ in (6) such that $\{\mathbf{v}\}$ is linearly independent and the conditions of (12) and (17) are satisfied. As for the matrices that satisfy (17), one can think of a Cauchy matrix or a Vandermonde matrix [9], [19]. Specifically, we employ the Cauchy matrix to construct explicit codes with the guarantee on the minimum finite-field size. Notice that the Cauchy matrix is an example that guarantees (17). One may use any other matrices that satisfy (17).

Definition 1 (A Cauchy Matrix) [19]: A Cauchy matrix $\mathbf{P}$ is an $m \times n$ matrix with entries p_{ij} in the form

$$p_{ij} = \frac{1}{x_i - y_j}, \forall i = 1, \ldots, m, j = 1, \ldots n, x_i \neq y_j$$

where x_i and y_j are elements of a field and $\{x_i\}$ and $\{y_j\}$ are injective sequences, i.e., elements of the sequence are distinct.

The injective property of $\{x_i\}$ and $\{y_j\}$ requires a finite field size of $2s$ for an s-by-s Cauchy matrix. Therefore, in our (6, 3, 5) repair code example, the finite field size of 6 suffices. The field size condition for guaranteeing linear independence of $\{\mathbf{v}\}$ is more relaxed.

E. Summary

Using the structure of (6) and the conditions of (11), (12), and (17), we can now state the following theorem.

Theorem 1 ((6, 3, 5) Exact-Repair MDS Codes): Suppose $\mathbf{P}$ of (11) is a Cauchy matrix, i.e., every submatrix of is invertible. Each element of $\mathbf{P}$ is in $\mathsf{GF}(q)$ and $q \geq 6$. Suppose encoding submatrices form the structure of (6), $\{\mathbf{v}\} := (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is linearly independent, and $\{\mathbf{u}\}$ satisfies the condition of (12). Then, the repair code comprising the encoding matrix and the projection vectors achieves the optimal tradeoff of (1).

V. Examples

We provide two numerical examples: (1) $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ is orthogonal, e.g., $\mathbf{V} = \mathbf{I}$; (2) $\mathbf{U}$ is orthogonal, e.g., $\mathbf{U} = \mathbf{I}$. We will also discuss the complexity of repair construction schemes for each of these examples. It turns out that the first code has significantly lower complexity for exact repair of systematic nodes, as compared to that of parity nodes. On the other hand, the second case provides much simpler parity-node repair schemes instead. Depending on applications of interest, one can choose an appropriate code among our family of codes.

A. *Example 1:* $\mathbf{V} = \mathbf{I}$

We present an example of (6, 3, 5) Exact-Repair MDS codes defined over $\mathsf{GF}(5)$ where $\mathbf{V} = \mathbf{I}$ and

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 4 \end{bmatrix},$$

$$\mathbf{U} = \kappa^{-1}\mathbf{V}'\mathbf{P} = 3\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 1 & 4 \\ 3 & 4 & 2 \end{bmatrix}$$

where $\mathbf{U}$ is set based on (12) and $\kappa = 2$. Notice that we employ a *non-Cauchy*-type matrix to construct a field-size $q = 5$ code (smaller than $q = 6$ required when using a Cauchy matrix). Remember that a Cauchy matrix provides only a sufficient condition for ensuring the invertibility of any submatrices of $\mathbf{P}$. By (6) and (13), the primary and dual structures for encoding matrices are given by

$$\mathbf{G} = \left[\begin{array}{ccc|ccc|ccc} 4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 \\ 3 & 1 & 0 & 1 & 1 & 0 & 4 & 1 & 0 \\ 3 & 0 & 1 & 4 & 0 & 1 & 2 & 0 & 1 \\ \hline 1 & 3 & 0 & 2 & 3 & 0 & 3 & 3 & 0 \\ 0 & 4 & 0 & 0 & 3 & 0 & 0 & 2 & 0 \\ 0 & 3 & 1 & 0 & 4 & 2 & 0 & 2 & 3 \\ \hline 1 & 0 & 3 & 3 & 0 & 3 & 4 & 0 & 3 \\ 0 & 1 & 3 & 0 & 3 & 1 & 0 & 4 & 4 \\ 0 & 0 & 4 & 0 & 0 & 2 & 0 & 0 & 1 \end{array}\right]$$

$$\mathbf{G}^{-1} = \left[\begin{array}{ccc|ccc|ccc} 4 & 1 & 4 & 3 & 0 & 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 1 & 4 & 4 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 0 & 3 & 1 & 1 & 1 \\ \hline 4 & 2 & 2 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 1 & 3 & 2 & 0 & 1 & 0 \\ 0 & 0 & 3 & 0 & 0 & 1 & 1 & 2 & 3 \\ \hline 1 & 2 & 4 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 4 & 3 & 4 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 & 4 & 2 & 1 \end{array}\right] \quad (18)$$

where

$$\mathbf{G} := \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3 \end{bmatrix}; \mathbf{G}^{-1} = \begin{bmatrix} \mathbf{A}'_1 & \mathbf{A}'_2 & \mathbf{A}'_3 \\ \mathbf{B}'_1 & \mathbf{B}'_2 & \mathbf{B}'_3 \\ \mathbf{C}'_1 & \mathbf{C}'_2 & \mathbf{C}'_3 \end{bmatrix}. \quad (19)$$

Fig. 9 shows an example for exact repair of (a) systematic node 1 and (b) parity node 1. Note that the projection vector solution for systematic node repair is quite simple: $\mathbf{v}_{\alpha i} = \mathbf{v}_1 = (1, 0, 0)^t$, $\forall i$. We download only the first equation from each survivor node. Notice that the downloaded five equations contain only five unknown variables of $(a_1, a_2, a_3, b_1, c_1)$ and three equations associated with $\mathbf{a}$ are linearly independent. Hence, we can successfully recover $\mathbf{a}$.

On the other hand, exact repair of parity nodes seems nonstraightforward. However, our framework provides quite a simple repair scheme: setting all of the projection vectors as $2^{-1}\mathbf{u}_1 = (1, 1, 1)^t$. This enables simultaneous interference alignment, while guaranteeing the decodability of $\mathbf{a}'$. Notice that (b'_1, b'_2, b'_3) and (c'_1, c'_2, c'_3) are aligned into $b'_1 + b'_2 + b'_3$ and $c'_1 + c'_2 + c'_3$, respectively, while three equations associated with $\mathbf{a}'$ are linearly independent.

As one can see, the complexity of systematic node repair is a little bit lower than that of parity node repair, although both repair schemes are simple. Hence, one can expect that this example is useful for the applications where the complexity of systematic node repair needs to be significantly low.

B. *Example 2:* $\mathbf{U} = \mathbf{I}$

We provide another example of (6, 3, 5) Exact-Repair MDS codes where $\mathbf{U}$ is orthogonal. We use the same field size of 5 and the same $\mathbf{P}$. Instead, we choose a nonorthogonal $\mathbf{V}$ in order to significantly reduce the complexity of parity node repair. Our framework provides a concrete guideline for accomplishing this. Remember that the projection vector solutions are $\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$ for exact repair of each parity node, respectively. For low complexity, we can first set $\mathbf{U} = \mathbf{I}$. The condition (12) then gives the following choice:

$$\mathbf{V} = \mathbf{P}^t\kappa^{-1} = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 1 & 4 \\ 3 & 4 & 2 \end{bmatrix}$$

where we use $\kappa = 2$. By (6) and (13), the primary and dual structures are given by

$$\mathbf{G} = \left[\begin{array}{ccc|ccc|ccc} 4 & 3 & 3 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 3 & 4 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 3 & 3 & 4 \\ \hline 4 & 1 & 4 & 2 & 0 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 3 & 3 & 4 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 & 3 & 1 & 2 \\ \hline 4 & 4 & 2 & 3 & 0 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 3 & 2 & 2 & 0 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 & 4 & 1 \end{array}\right]$$

$$\mathbf{G}^{-1} = \left[\begin{array}{ccc|ccc|ccc} 4 & 0 & 0 & 4 & 0 & 0 & 1 & 0 & 0 \\ 1 & 3 & 0 & 2 & 3 & 0 & 2 & 2 & 0 \\ 4 & 0 & 3 & 2 & 0 & 3 & 4 & 0 & 3 \\ \hline 3 & 1 & 0 & 1 & 1 & 0 & 1 & 4 & 0 \\ 0 & 4 & 0 & 0 & 3 & 0 & 0 & 3 & 0 \\ 0 & 4 & 3 & 0 & 2 & 1 & 0 & 4 & 1 \\ \hline 2 & 0 & 1 & 1 & 0 & 1 & 2 & 0 & 4 \\ 0 & 2 & 1 & 0 & 1 & 2 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 1 \end{array}\right] \quad (20)$$

where $\mathbf{G}$ is defined as (19). Notice that the matrices of (20) have exactly the transpose structure of the matrices of (18). Hence, this structure of (20) is a dual solution to that of (18), thereby ensuring the transfer of the lowered complexity property for parity node repair.

Fig. 10 shows an example for exact repair of (a) systematic node 1 and (b) parity node 1. In contrast to our previous case, exact repair of parity nodes is now much simpler. In this example, by downloading only the first equation from each survivor node, we can successfully recover $\mathbf{a}'$. On the contrary, systematic node repair is more involved, with all of the projection vectors being set as $2^{-1}\mathbf{v}'_1 = (1, 1, 4)^t$. Using this vector, we

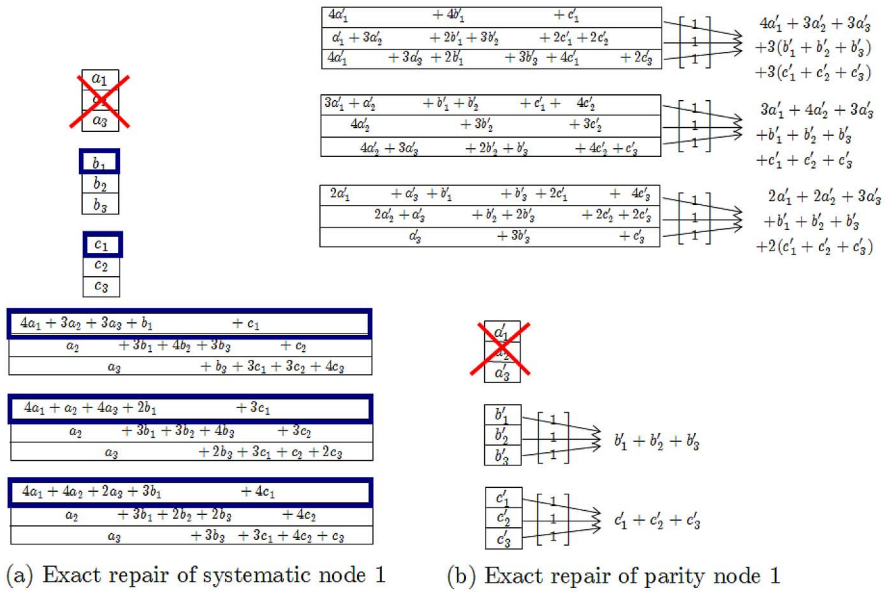


Fig. 9. Example 1: $\mathbf{V} = \mathbf{I}$. A (6, 3, 5) Exact-Repair MDS code defined over $\mathsf{GF}(5)$. The projection vector solution for systematic node repair is quite simple: $\mathbf{v}_{\alpha i} = \mathbf{v}_1 = (1,0,0)^t$, $\forall i$. This example employs the same encoding matrix and projection vectors for systematic node repair as those in [9]. We download only the first equation from each survivor node; for parity node repair, our new framework provides a simple scheme: setting all of the projection vectors as $2^{-1}\mathbf{u}_1 = (1,1,1)^t$. This enables simultaneous interference alignment, while guaranteeing the decodability of $\mathbf{a}$.

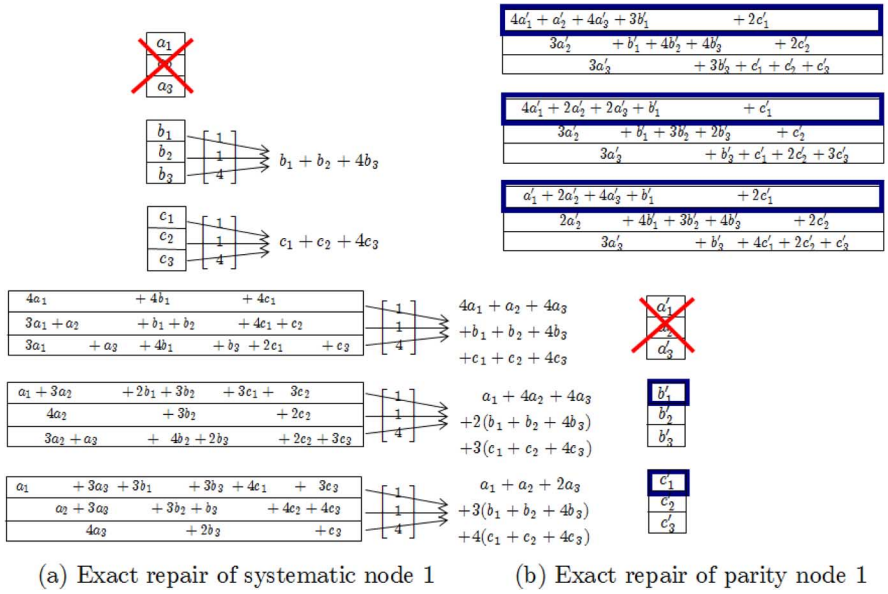


Fig. 10. Example 2: $\mathbf{U} = \mathbf{I}$. A (6, 3, 5) Exact-Repair MDS code defined over $\mathsf{GF}(5)$. Since we choose $\mathbf{U} = \mathbf{I}$, the projection vector solution for parity node repair, is much simpler. We download only the first equation from each survivor node; systematic node repair is more involved, with all of the projection vectors being set as $2^{-1}\mathbf{v}_1' = (1,1,4)^t$.

can achieve simultaneous interference alignment, thereby decoding the desired components of $\mathbf{a}$.

VI. Generalization: $n \geq 2k$; $d \geq 2k-1$

Theorem 1 gives insights into generalization to $(2k, k, 2k-1)$ Exact-Repair MDS codes. The key observation is that assuming $\mathcal{M} = k(d-k+1)$, storage cost is $\alpha = \mathcal{M}/k = d-k+1 = k$ and this number is equal to the number of systematic nodes and furthermore matches the number of parity nodes. Notice that the storage size matches the size of encoding submatrices, which determines the number of linearly independent vectors of $\{\mathbf{v}\} := \{\mathbf{v}_1, \ldots\}$. In this case, therefore, we can generate k linearly independent vectors $\{\mathbf{v}\} := \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ and corresponding $\{\mathbf{u}\} := \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ through the appropriate choice of $\mathbf{P}$. This immediately provides $(2k, k, 2k-1)$ Exact-Repair MDS codes.

A. Case: $n = 2k$

Theorem 2 ($(2k, k, 2k-1)$ Exact-Repair MDS Codes): Let $\mathbf{P}$ be a Cauchy matrix:

$$\mathbf{P} = \begin{bmatrix} p_1^{(1)} & p_1^{(2)} & \cdots & p_1^{(k)} \\ p_2^{(1)} & p_2^{(2)} & \cdots & p_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ p_k^{(1)} & p_k^{(2)} & \cdots & p_k^{(k)} \end{bmatrix}$$

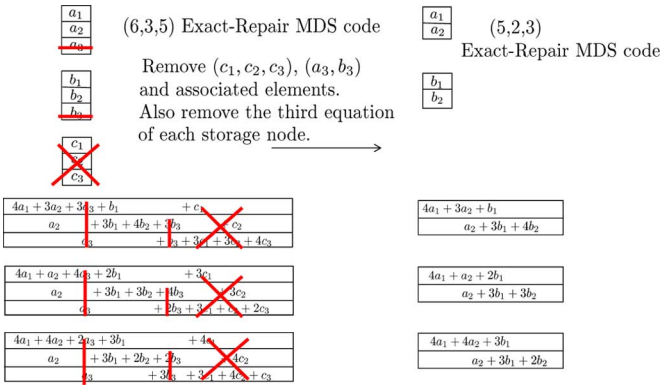


Fig. 11. Illustration of the construction of a (5, 2, 3) Exact-Repair MDS code from a (6, 3, 5) Exact-Repair MDS code defined over $\mathsf{GF}(5)$. For a larger code, we adopt the (6, 3, 5) code in Fig. 9. First, we remove all the elements associated with the last $(n-2k)=1$ information unit ("c"). Next, we prune symbols (a_3, b_3) and associated elements. Also we remove the last equation of each storage node. Finally we obtain the $(n, k, d) = (5, 2, 3)$ target code.

where each element $p_j^{(i)} \in \mathsf{GF}(q)$, where $q \geq 2k$. Suppose

$$\begin{aligned} &\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \text{ is invertible and} \\ &\mathbf{U} = \kappa^{-1}\mathbf{V}'\mathbf{P} \end{aligned} \tag{21}$$

where $\mathbf{V}' = (\mathbf{V}^t)^{-1}$ and κ is an arbitrary nonzero value $\in \mathbb{F}_q$ such that $1-\kappa^2 \neq 0$. Also assume that encoding submatrices are given by

$$\begin{aligned} \mathbf{G}_1^{(1)} &= \mathbf{u}_1\mathbf{v}_1^t + p_1^{(1)}\mathbf{I}, \ldots, \mathbf{G}_k^{(1)} = \mathbf{u}_1\mathbf{v}_k^t + p_k^{(1)}\mathbf{I} \\ &\vdots \qquad\qquad \ddots \qquad\qquad \vdots \\ \mathbf{G}_1^{(k)} &= \mathbf{u}_k\mathbf{v}_1^t + p_1^{(k)}\mathbf{I}, \ldots, \mathbf{G}_k^{(k)} = \mathbf{u}_k\mathbf{v}_k^t + p_k^{(k)}\mathbf{I} \end{aligned} \tag{22}$$

where $\mathbf{G}_l^{(i)}$ indicates an encoding submatrix for parity node i, associated with information unit l. Then, the repair code achieves the optimal tradeoff of (1).

Proof: See Appendix B. ∎

Remark 3: Note that the minimum required alphabet size is $2k$. As mentioned earlier, this is because we employ a Cauchy matrix for ensuring the invertibility of any submatrices of $\mathbf{P}$. One may customize codes to find smaller alphabet-size codes.

B. Case: $n \geq 2k$; $d \geq 2k-1$

Now what if k is less than the size $(=\alpha = d-k+1)$ of encoding submatrices, i.e., $d \geq 2k-1$? Note that this case automatically implies that $n \geq 2k$, since $n \geq d+1$. The key observation in this case is that the encoding submatrix size is bigger than k, and therefore we have more degrees of freedom (a larger number of linearly independent vectors) than the number of constraints. Hence, exact repair of systematic nodes becomes transparent. This was observed in [9], where it was shown that for $d = n-1 \geq 2k-1$, exact repair of systematic nodes only can be guaranteed by carefully manipulating $(2k, k, 2k-1)$ codes through a pruning operation.

In this paper, we propose a generalized pruning algorithm that ensures exact repair of *all* nodes for $n \geq 2k$ and $d \geq 2k-1$. The recipe for this has two parts:

1) Constructing a target code from a larger code.
2) Showing that the resulting target code ensures exact repair of all nodes as well as the MDS-code property.

We provide detailed procedures[8] of the first part.

1(a) Using Theorem 2, construct a larger $(2n-2k, n-k, 2n-2k-1)$ code with a finite field size of $q \geq 2n-2k$.

1(b) Remove all the elements associated with the $(n-2k)$ information units (e.g., from the $(k+1)$th to the $(n-k)$th information unit). The number of nodes is then reduced by $(n-2k)$ and so are the number of information units and the number of degrees. Hence, we obtain the $(n, k, n-1)$ code.

1(c) Prune the last $(n-1-d)$ equations in each storage node and also the last $(n-1-d)$ symbols of each information unit, while keeping the number of information units and storage nodes. We can then get the (n, k, d) target code.

Indeed, based on our framework of Section IV, it can be shown that the resulting code described above guarantees exact repair of all nodes while retaining the MDS-code property.

Theorem 3 $\left(\frac{k}{n} \leq \frac{1}{2}, d \geq 2k-1\right)$*:* Suppose that all of the survivor systematic nodes participate in the repair. Then, under exact repair constraints of all nodes, the optimal tradeoff of (1) can be attained with a deterministic scheme requiring a field size of at most $2(n-k)$.

Proof: See Appendix C. ∎

Example 1: Fig. 11 illustrates how to construct an $(n, k, d) = (5, 2, 3)$ target code based on the above recipe. First construct the $(2n-2k, n-k, 2n-2k-1) = (6, 3, 5)$ code, which is larger than the (5, 2, 3) target code, but which belongs to the category of $n = 2k$. For this code, we employ the example in Fig. 9. We now remove all the elements associated with the last $(n-2k) = 1$ information unit, which corresponds to (c_1, c_2, c_3). Next, prune the last symbol (a_3, b_3) of each information unit and associated elements to shrink the storage size into 2. We can then obtain the (5, 2, 3) target code. Based on the proposed framework in Section IV, it can be shown that the resulting code guarantees exact repair of all nodes and the MDS-code property.

VII. Generalization: $k \leq 3$

As a side generalization, we consider the case of $k \leq 3$ (and $d \geq 2k-1$). The interesting special case of the (5, 3, 4) code[9] will be focused on, since it is not covered by the above case of $\frac{k}{n} \leq \frac{1}{2}$. For this case, we propose another interference alignment technique.

Theorem 4 $(k \leq 3, d \geq 2k-1)$*:* The optimal tradeoff of (1) can be attained with a deterministic scheme requiring a finite-field size of at most $2n-2k$.

Proof: The case of $k=1$ is trivial. By Theorems 2 and 3, we prove the case of $k=2$. However, additional effort is needed to prove the case of $k=3$. By Theorems 2 and 3, $(n, 3, d)$ for $n \geq 6$ and $d \geq 5$ can be proved. But (5, 3, 4) codes are not in the class. In Section VII-A, we will address this case to complete the proof. ∎

[8]While Steps *(1a)* and *(1b)* come from the pruning technique in [9], Step *(1c)* is a significant distinction from that of [9].

[9]Independently, the authors in [10] found (5, 3, 4) codes defined over $\mathsf{GF}(3)$, based on a search algorithm.

Remark 4: In order to cover general n, we provide a looser bound on the required finite-field size: $q \geq 2n - 2k$. In fact, for the (5, 3, 4) repair code (that will be shown in Lemma 2), a smaller finite-field size of $q = 3 (< 4 = 2n - 2k)$ is enough for code construction. We have taken the maximum of the required field sizes of all the cases.

A. (5, 3, 4) Exact-Repair MDS Codes

Assume $\mathcal{M} = 6$ and repair-bandwidth-per-link $= 1$. The cutset bound (1) then gives the fundamental limits of storage cost $\alpha = 2$; hence, the dimension of encoding submatrices is 2-by-2. Note that the size is *less* than the number of systematic nodes. Therefore, our earlier framework does not cover this category.

We propose an eigenvector-based interference alignment technique that guarantees exact repair of *all* nodes. Let $\mathbf{a} = (a_1, a_2)^t$, $\mathbf{b} = (b_1, b_2)^t$ and $\mathbf{c} = (c_1, c_2)^t$. For exact repair, we connect to $d (= 4)$ nodes to download a 1-D scalar value from each survivor node. Fig. 12 illustrates exact repair of node 1. We download four equations from survivor nodes: $\mathbf{b}^t\mathbf{v}_{\alpha 1}$; $\mathbf{c}^t\mathbf{v}_{\alpha 2}$; $\mathbf{a}^t(\mathbf{A}_1\mathbf{v}_{\alpha 3}) + \mathbf{b}^t(\mathbf{B}_1\mathbf{v}_{\alpha 3}) + \mathbf{c}^t(\mathbf{C}_1\mathbf{v}_{\alpha 3})$; $\mathbf{a}^t(\mathbf{A}_2\mathbf{v}_{\alpha 4}) + \mathbf{b}^t(\mathbf{B}_2\mathbf{v}_{\alpha 4}) + \mathbf{c}^t(\mathbf{C}_2\mathbf{v}_{\alpha 4})$. The approach is different from that of our earlier proposed framework. Instead an idea here consists of three steps: 1) choosing projection vectors for achieving interference alignment; 2) gathering all the alignment constraints and the MDS-code constraint; 3) designing the encoding submatrices that satisfy all the constraints. Notice the design of encoding submatrices is the last part.

Here are details. Note that there are 6 unknown variables: 2 desired unknowns (a_1, a_2) and 4 undesired unknowns (b_1, b_2, c_1, c_2). Therefore, it is required to align (b_1, b_2, c_1, c_2) onto at least 2-D linear space. We face the challenge that appeared in the (6, 3, 5) code example in Fig. 6. Projection vectors $\mathbf{v}_{\alpha 3}$ and $\mathbf{v}_{\alpha 4}$ affect interference alignment $\mathbf{b}$ and $\mathbf{c}$ simultaneously. Therefore, we need simultaneous interference alignment. To solve this problem, we introduce an eigenvector-based interference alignment scheme.

First choose $\mathbf{v}_{\alpha 3}$ and $\mathbf{v}_{\alpha 4}$ such that $\mathbf{v}_{\alpha 3} = \mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$ and $\mathbf{v}_{\alpha 4} = \mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}$, thus achieving interference alignment for "$\mathbf{b}$". Observe the interfering vectors associated with "$\mathbf{c}$"

$$\mathbf{v}_{\alpha 2}; \mathbf{C}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}; \mathbf{C}_2\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}.$$

The first and second vectors can be aligned by setting $\mathbf{v}_{\alpha 2} = \mathbf{C}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$. Now what about for the following two vectors: $\mathbf{C}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$ and $\mathbf{C}_2\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}$? Suppose that the associated matrices ($\mathbf{C}_1\mathbf{B}_1^{-1}$ and $\mathbf{C}_2\mathbf{B}_2^{-1}$) and the projection vector $\mathbf{v}_{\alpha 1}$ are randomly chosen. Then, these two vectors are not guaranteed to be aligned. However, a judicious choice of $\mathbf{v}_{\alpha 1}$ makes it possible to align them. The idea is to choose $\mathbf{v}_{\alpha 1}$ as an eigenvector of $\mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{C}_1\mathbf{B}_1^{-1}$. Since $\mathbf{v}_{\alpha 1}$ can be chosen arbitrarily, this can be easily done. Lastly, consider the condition for ensuring the decodability of desired signals $\mathsf{rank}\left(\left[\mathbf{A}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}\ \mathbf{A}_2\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}\right]\right) = 2$.

We repeat the procedure for exact repair of "$\mathbf{b}$" and "$\mathbf{c}$". For parity nodes, we employ the remapping technique described earlier

$$\begin{bmatrix}\mathbf{a}'\\ \mathbf{b}'\\ \mathbf{c}'\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1^t & \mathbf{B}_1^t & \mathbf{C}_1^t\\ \mathbf{A}_2^t & \mathbf{B}_2^t & \mathbf{C}_2^t\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{c}\end{bmatrix}$$
$$\begin{bmatrix}\mathbf{A}_1' & \mathbf{A}_2' & \mathbf{0}\\ \mathbf{B}_1' & \mathbf{B}_2' & \mathbf{0}\\ \mathbf{C}_1' & \mathbf{C}_2' & \mathbf{I}\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0}\\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{0}\\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{I}\end{bmatrix}^{-1}. \quad (23)$$

We gather all the conditions that need to be guaranteed for exact repair of all nodes

$$\begin{aligned}
\mathsf{rank}\left(\left[\mathbf{A}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}\ \mathbf{A}_2\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}\right]\right) &= 2\\
\mathsf{rank}\left(\left[\mathbf{B}_1\mathbf{C}_1^{-1}\mathbf{v}_{\beta 1}\ \mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{v}_{\beta 1}\right]\right) &= 2\\
\mathsf{rank}\left(\left[\mathbf{C}_1\mathbf{A}_1^{-1}\mathbf{v}_{\gamma 1}\ \mathbf{C}_2\mathbf{A}_2^{-1}\mathbf{v}_{\gamma 1}\right]\right) &= 2\\
\mathsf{rank}\left(\left[\mathbf{A}_1'\mathbf{B}_1'^{-1}\mathbf{v}_{\alpha' 1}\ \mathbf{A}_2'\mathbf{B}_2'^{-1}\mathbf{v}_{\alpha' 1}\right]\right) &= 2\\
\mathsf{rank}\left(\left[\mathbf{B}_1'\mathbf{C}_1'^{-1}\mathbf{v}_{\beta' 1}\ \mathbf{B}_2'\mathbf{C}_2'^{-1}\mathbf{v}_{\beta' 1}\right]\right) &= 2
\end{aligned} \quad (24)$$

where

$$\begin{aligned}
\mathbf{v}_{\alpha 1}&\text{: an eigenvector of } \mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{C}_1\mathbf{B}_1^{-1}\\
\mathbf{v}_{\beta 1}&\text{: an eigenvector of } \mathbf{C}_2\mathbf{A}_2^{-1}\mathbf{A}_1\mathbf{C}_1^{-1}\\
\mathbf{v}_{\gamma 1}&\text{: an eigenvector of } \mathbf{A}_2\mathbf{B}_2^{-1}\mathbf{B}_1\mathbf{A}_1^{-1}\\
\mathbf{v}_{\alpha' 1}&\text{: an eigenvector of } \mathbf{B}_2'\mathbf{C}_2'^{-1}\mathbf{C}_1'\mathbf{B}_1'^{-1}\\
\mathbf{v}_{\beta' 1}&\text{: an eigenvector of } \mathbf{C}_2'\mathbf{A}_2'^{-1}\mathbf{A}_1'\mathbf{C}_1'^{-1}.
\end{aligned} \quad (25)$$

Note that eigenvectors may not exist for the finite Galois field. However, the existence is guaranteed by carefully choosing the encoding submatrices. We provide an explicit code in the following lemma.

Lemma 2 ((5, 3, 4) Exact-Repair MDS Codes): Let $\alpha, \beta \in \mathsf{GF}(3)$ and be nonzero. Suppose encoding submatrices are given by

$$\begin{aligned}
\mathbf{A}_1 &= \begin{bmatrix}2\alpha & 0\\ 2\beta & \beta\end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix}\alpha & 2\alpha\\ 0 & 2\beta\end{bmatrix}\\
\mathbf{C}_1 &= \begin{bmatrix}2\alpha & 0\\ \beta & 2\beta\end{bmatrix}\\
\mathbf{A}_2 &= \begin{bmatrix}2\alpha & 0\\ \beta & 2\beta\end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix}\alpha & 2\alpha\\ 0 & \beta\end{bmatrix}\\
\mathbf{C}_2 &= \begin{bmatrix}\alpha & 0\\ 2\beta & 2\beta\end{bmatrix}.
\end{aligned} \quad (26)$$

The projection vectors for exact repair are chosen through the above procedures. Then, the code achieves the optimal tradeoff of (1).

Proof: See Appendix D. ∎

Remark 5: Note that encoding submatrices are lower-triangular or upper-triangular. This structure has important properties. Not only does this structure guarantee invertibility, it can in fact guarantee the existence of eigenvectors. It turns out the structure as above satisfies all of the conditions needed for the MDS property and exact repair.

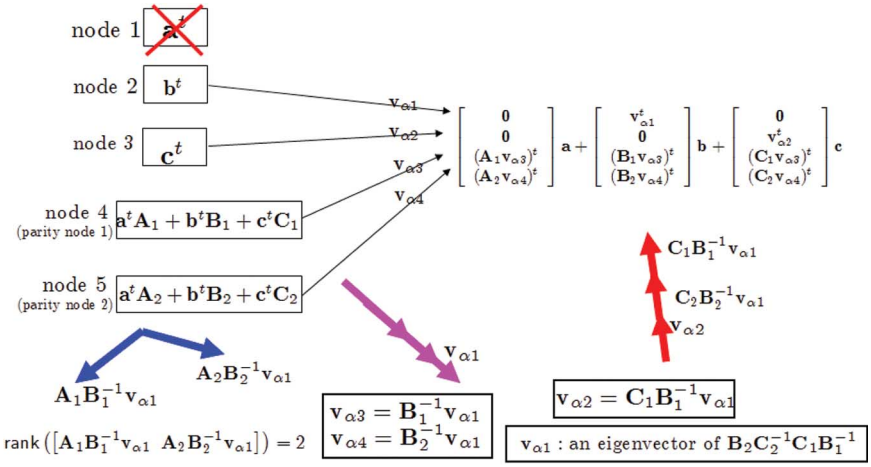


Fig. 12. Eigenvector-based interference alignment for (5, 3, 4) Exact-Repair MDS codes. First we align interference "b" by setting $\mathbf{v}_{\alpha 3}=\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$ and $\mathbf{v}_{\alpha 4}=\mathbf{B}_2^{-1}\mathbf{v}_{\alpha 1}$. Next, partially align interference of "c" by setting $\mathbf{v}_{\alpha 2}=\mathbf{C}_1\mathbf{B}_1^{-1}\mathbf{v}_{\alpha 1}$. Finally, choosing $\mathbf{v}_{\alpha 1}$ as an eigenvector of $\mathbf{B}_2\mathbf{C}_2^{-1}\mathbf{C}_1\mathbf{B}_1^{-1}$, we can achieve interference alignment for $\mathbf{c}$.

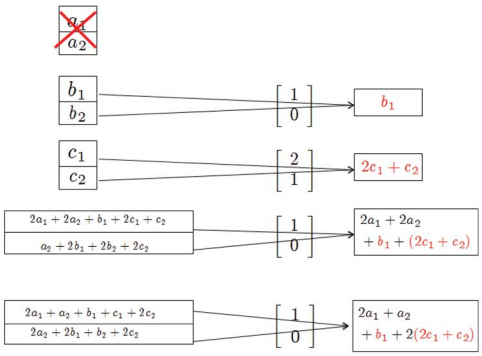


Fig. 13. Illustration of exact repair of node 1 for a (5, 3, 4) Exact-Repair MDS code defined over $\mathsf{GF}(3)$. The eigenvector-based interference alignment scheme enables to decode 2 desired unknowns (a_1, a_2) from 4 equations containing 6 unknowns. Notice that interference "b" and "c" are aligned simultaneously although the same projection vectors $\mathbf{v}_{\alpha 3}$ and $\mathbf{v}_{\alpha 4}$ are used.

Example 2: Fig. 13 illustrates exact repair of node $1(a_1, a_2)$ for a (5, 3, 4) Exact-Repair MDS code defined over $\mathsf{GF}(3)$. Notice that interference "b" and "c" are aligned simultaneously. One can check exact repair of the remaining four nodes based on our proposed method.

VIII. Conclusion

We have systematically developed interference alignment techniques that attain the cutset bound (1) under exact repair constraints of *all* nodes. Based on the proposed framework, we provided a family of codes for the cases: (a) $\frac{k}{n}\leq\frac{1}{2}$ and $d\geq 2k-1$; (b) $k\leq 3$, for arbitrary $n\geq k$ (and $d\geq 2k-1$). This family of codes provides insights into a dual relationship between the systematic and parity node repair, as well as opens up a larger constructive design space of solutions. For (5, 3, 4) codes which do not satisfy $\frac{k}{n}\leq\frac{1}{2}$, we have developed an eigenvector-based interference alignment to show the optimality of the cutset bound. Unlike wireless communication problems, our storage repair problems have more flexibility in designing encoding matrices which correspond to wireless channel coefficients (provided by nature) in communication problems. Exploiting this fact, we developed interference alignment techniques for optimal exact repair codes in distributed storage systems.

Appendix A
Proof of Lemma 1

It suffices to show that

$$\begin{bmatrix}\mathbf{A}_1' & \mathbf{A}_2' & \mathbf{A}_3'\\ \mathbf{B}_1' & \mathbf{B}_2' & \mathbf{B}_3'\\ \mathbf{C}_1' & \mathbf{C}_2' & \mathbf{C}_3'\end{bmatrix}\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3\\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3\\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}.$$

Using (6) and (13), we compute

$$\begin{aligned}&(1-\kappa^2)(\mathbf{A}_1'\mathbf{A}_1+\mathbf{A}_2'\mathbf{B}_1+\mathbf{A}_3'\mathbf{C}_1)\\&\quad=\left(\mathbf{v}_1'\mathbf{u}_1'^t-\kappa^2\alpha_1'\mathbf{I}\right)\left(\mathbf{u}_1\mathbf{v}_1^t+\alpha_1\mathbf{I}\right)\\&\qquad+\left(\mathbf{v}_2'\mathbf{u}_1'^t-\kappa^2\beta_1'\mathbf{I}\right)\left(\mathbf{u}_1\mathbf{v}_2^t+\beta_1\mathbf{I}\right)\\&\qquad+\left(\mathbf{v}_3'\mathbf{u}_1'^t-\kappa^2\gamma_1'\mathbf{I}\right)\left(\mathbf{u}_1\mathbf{v}_3^t+\gamma_1\mathbf{I}\right)\\&\quad\overset{(a)}{=}(\mathbf{v}_1'\mathbf{v}_1^t+\mathbf{v}_2'\mathbf{v}_2^t+\mathbf{v}_3'\mathbf{v}_3^t)+(\alpha_1\mathbf{v}_1'+\beta_1\mathbf{v}_2'+\gamma_1\mathbf{v}_3')\mathbf{u}_1'^t\\&\qquad-\kappa^2\mathbf{u}_1(\alpha_1'\mathbf{v}_1+\beta_1'\mathbf{v}_2+\gamma_1'\mathbf{v}_3)^t-\kappa^2\mathbf{I}\\&\quad\overset{(b)}{=}(\mathbf{v}_1'\mathbf{v}_1^t+\mathbf{v}_2'\mathbf{v}_2^t+\mathbf{v}_3'\mathbf{v}_3^t)+\kappa\mathbf{u}_1\mathbf{u}_1'^t\\&\qquad-\kappa^2\mathbf{u}_1(\alpha_1'\mathbf{v}_1+\beta_1'\mathbf{v}_2+\gamma_1'\mathbf{v}_3)^t-\kappa^2\mathbf{I}\\&\quad\overset{(c)}{=}(\mathbf{v}_1'\mathbf{v}_1^t+\mathbf{v}_2'\mathbf{v}_2^t+\mathbf{v}_3'\mathbf{v}_3^t)-\kappa^2\mathbf{I}\\&\quad\overset{(d)}{=}(1-\kappa^2)\mathbf{I}\end{aligned}$$

where (a) follows from $\alpha_1\alpha_1'+\beta_1\beta_1'+\gamma_1\gamma_1'=1$ due to (11); (b) follows from (12); (c) follows from $\mathbf{u}_1'=\kappa(\alpha_1'\mathbf{v}_1+\beta_1'\mathbf{v}_2+\gamma_1'\mathbf{v}_3)$ (See Claim 1); and (d) follows from the fact that $\mathbf{v}_1'\mathbf{v}_1^t+\mathbf{v}_2'\mathbf{v}_2^t+\mathbf{v}_3'\mathbf{v}_3^t=\mathbf{I}$, since $(\mathbf{v}_1', \mathbf{v}_2', \mathbf{v}_3')$ are dual basis vectors.

Similarly, one can check that $\mathbf{B}_1'\mathbf{A}_2+\mathbf{B}_2'\mathbf{B}_2+\mathbf{B}_3'\mathbf{C}_2=\mathbf{I}$ and $\mathbf{C}_1'\mathbf{A}_3+\mathbf{C}_2'\mathbf{B}_3+\mathbf{C}_3'\mathbf{C}_3=\mathbf{I}$. Now let us compute one of the cross terms

$$\begin{aligned}&(1-\kappa^2)(\mathbf{A}_1'\mathbf{A}_2+\mathbf{A}_2'\mathbf{B}_2+\mathbf{A}_3'\mathbf{C}_2)\\&\quad=\left(\mathbf{v}_1'\mathbf{u}_1'^t-\kappa^2\alpha_1'\mathbf{I}\right)\left(\mathbf{u}_2\mathbf{v}_1^t+\alpha_2\mathbf{I}\right)\\&\qquad+\left(\mathbf{v}_2'\mathbf{u}_1'^t-\kappa^2\beta_1'\mathbf{I}\right)\left(\mathbf{u}_2\mathbf{v}_2^t+\beta_2\mathbf{I}\right)\\&\qquad+\left(\mathbf{v}_3'\mathbf{u}_1'^t-\kappa^2\gamma_1'\mathbf{I}\right)\left(\mathbf{u}_2\mathbf{v}_3^t+\gamma_2\mathbf{I}\right)\\&\quad\overset{(a)}{=}\left(\alpha_2\mathbf{v}_1'+\beta_2\mathbf{v}_2'+\gamma_2\mathbf{v}_3'\right)\mathbf{u}_1'^t\\&\qquad-\kappa^2\mathbf{u}_2\left(\alpha_1'\mathbf{v}_1+\beta_1'\mathbf{v}_2+\gamma_1'\mathbf{v}_3\right)^t\\&\quad\overset{(b)}{=}0\end{aligned}$$

where (a) follows from $\mathbf{u}_i'^t\mathbf{u}_j = \delta(i-j)$ and $< (\alpha_1',\beta_1',\gamma_1'),(\alpha_2,\beta_2,\gamma_2) >= 0$; (b) follows from (12) and Claim 1. Similarly, we can check that the other cross terms are zero matrices. This completes the proof.

Claim 1: For all i, $\mathbf{u}_i' = \kappa\left(\alpha_i'\mathbf{v}_1 + \beta_i'\mathbf{v}_2 + \gamma_i'\mathbf{v}_3\right)$.
Proof: By (12), we can rewrite

$$[\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3] = \frac{1}{\kappa}[\mathbf{v}_1',\mathbf{v}_2',\mathbf{v}_3']\begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\\ \beta_1 & \beta_2 & \beta_3\\ \gamma_1 & \gamma_2 & \gamma_3\end{bmatrix}.$$

Using the fact that $(\mathbf{u}_1',\mathbf{u}_2',\mathbf{u}_3')$ are dual basis vectors, we get

$$\begin{bmatrix}\mathbf{u}_1'^t\\ \mathbf{u}_2'^t\\ \mathbf{u}_3'^t\end{bmatrix} = \kappa\begin{bmatrix}\alpha_1' & \beta_1' & \gamma_1'\\ \alpha_2' & \beta_2' & \gamma_2'\\ \alpha_3' & \beta_3' & \gamma_3'\end{bmatrix}\begin{bmatrix}\mathbf{v}_1^t\\ \mathbf{v}_2^t\\ \mathbf{v}_3^t\end{bmatrix}.$$

This completes the proof. ∎

Appendix B
Proof of Theorem 2

For generalization, we are forced to use some heavy notation but only for this section and the related appendices. Let $\mathbf{w}_j \in \mathbb{F}_q^k$ be a message vector for information unit j. Let $\mathbf{G}_j^{(i)} \in \mathbb{F}_q^{k\times k}$ be an encoding submatrix for parity node i, associated with the jth information unit.

1) Exact Repair of Systematic Nodes: By symmetry, we consider only systematic node 1. We have each survivor node project their data with projection vector $\mathbf{v}_1'$, which is the first column vector of $\mathbf{V}' = (\mathbf{V}^t)^{-1}$. We then get

From systematic node j: $\mathbf{w}_j^t\mathbf{v}_1'$,

From parity node i: $\mathbf{w}_1^t\left(\mathbf{u}_i + p_1^{(i)}\mathbf{v}_1'\right) + \underbrace{\sum_{j=2}^{k} p_j^{(i)}\left(\mathbf{w}_j^t\mathbf{v}_1'\right)}_{interference}$

where $2 \le j \le k$ and $1 \le i \le k$. Note that we can achieve simultaneous interference alignment for nonintended signals. Since $\mathbf{u}_i$'s are linearly independent, we can decode desired signals $\mathbf{w}_1$, thus ensuring exact repair.

2) Exact Repair of Parity Nodes: The idea is the same as that of Theorem 1. First we remap parity nodes with new variables

$$\begin{bmatrix}\mathbf{w}_1'\\ \mathbf{w}_2'\\ \vdots\\ \mathbf{w}_k'\end{bmatrix} := \begin{bmatrix}\mathbf{G}_1^{(1)t} & \mathbf{G}_2^{(1)t} & \cdots & \mathbf{G}_k^{(1)t}\\ \mathbf{G}_1^{(2)t} & \mathbf{G}_2^{(2)t} & \cdots & \mathbf{G}_k^{(2)t}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{G}_1^{(k)t} & \mathbf{G}_2^{(k)t} & \cdots & \mathbf{G}_k^{(k)t}\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ \mathbf{w}_2\\ \vdots\\ \mathbf{w}_k\end{bmatrix}.$$

Define the newly remapped encoding submatrices as

$$\begin{bmatrix}\mathbf{G}_1'^{a(1)} & \mathbf{G}_1'^{(2)} & \cdots & \mathbf{G}_1'^{(k)}\\ \mathbf{G}_2'^{(1)} & \mathbf{G}_2'^{(2)} & \cdots & \mathbf{G}_2'^{(k)}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{G}_k'^{(1)} & \mathbf{G}_k'^{(2)} & \cdots & \mathbf{G}_k'^{(k)}\end{bmatrix} := \begin{bmatrix}\mathbf{G}_1^{(1)} & \mathbf{G}_1^{(2)} & \cdots & \mathbf{G}_1^{(k)}\\ \mathbf{G}_2^{(1)} & \mathbf{G}_2^{(2)} & \cdots & \mathbf{G}_2^{(k)}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{G}_k^{(1)} & \mathbf{G}_k^{(2)} & \cdots & \mathbf{G}_k^{(k)}\end{bmatrix}^{-1}. \quad (27)$$

We can now apply the generalization of Lemma 1 to obtain the dual structure

$$\begin{cases}\mathbf{G}_1'^{(1)} = \frac{1}{1-\kappa^2}\left(\mathbf{v}_1'\mathbf{u}_1'^t - \kappa^2 p_1'^{(1)}\mathbf{I}\right)\\ \vdots\\ \mathbf{G}_k'^{(1)} = \frac{1}{1-\kappa^2}\left(\mathbf{v}_1'\mathbf{u}_k'^t - \kappa^2 p_1'^{(k)}\mathbf{I}\right)\end{cases}$$

$$\vdots$$

$$\begin{cases}\mathbf{G}_1'^{(k)} = \frac{1}{1-\kappa^2}\left(\mathbf{v}_k'\mathbf{u}_1'^t - \kappa^2 p_k'^{(1)}\mathbf{I}\right)\\ \vdots\\ \mathbf{G}_k'^{(k)} = \frac{1}{1-\kappa^2}\left(\mathbf{v}_k'\mathbf{u}_k'^t - \kappa^2 p_k'^{(k)}\mathbf{I}\right)\end{cases}$$

where the dual basis vectors are defined as

$$\begin{bmatrix}p_1'^{(1)} & p_2'^{(1)} & \cdots & p_k'^{(1)}\\ p_1'^{(2)} & p_2'^{(2)} & \cdots & p_k'^{(2)}\\ \vdots & \vdots & \ddots & \vdots\\ p_1'^{(k)} & p_2'^{(k)} & \cdots & p_k'^{(k)}\end{bmatrix} := \begin{bmatrix}p_1^{(1)} & p_2^{(2)} & \cdots & p_1^{(k)}\\ p_2^{(1)} & p_2^{(2)} & \cdots & p_2^{(k)}\\ \vdots & \vdots & \ddots & \vdots\\ p_k^{(1)} & p_k^{(2)} & \cdots & p_k^{(k)}\end{bmatrix}^{-1}.$$

By symmetry, we consider only parity node 1. Choosing the projection vector $\mathbf{u}_1$, we get

From systematic node j:
$$\frac{\mathbf{w}_1'^t}{1-\kappa^2}\left(\mathbf{u}_j' - \kappa^2 p_1'^{(j)}\mathbf{v}_1'\right) - \frac{\kappa^2}{1-\kappa^2}\sum_{i=2}^{k} p_i'^{(j)}\left(\mathbf{w}_i'^t\mathbf{u}_1\right)$$
From parity node i: $\mathbf{w}_i'^t\mathbf{u}_1$

where $1 \le j \le k$ and $2 \le i \le k$. Note that we can achieve simultaneous interference alignment for nonintended signals. Since $\mathbf{u}_i'$'s are linearly independent, we can decode desired signals $\mathbf{w}_1'$, thus ensuring exact repair of parity node 1.

3) The MDS-Code Property: We check the invertibility of a composite encoding submatrix when a Data Collector connects to i systematic nodes and $(k-i)$ parity nodes for $i = 0,\ldots,k$. The main idea is to use a Gaussian elimination method as we did in Section IV-C. The verification is tedious and therefore details are omitted.

4) Minimum Required Finite-Field Size: Note that the dimension of a Cauchy matrix $\mathbf{P}$ is k-by-k. Therefore, the minimum finite-field size required to generate the Cauchy matrix is $2k$, i.e., $q \ge 2k$.

APPENDIX C
PROOF OF THEOREM 3

According to the proposed pruning algorithm, we start with an larger $(2n-2k, n-k, 2n-2k-1)$ code which has encoding submatrices as follows:

$$\begin{cases} \mathbf{G}_1^{(1)} = \mathbf{u}_1\mathbf{v}_1^t + p_1^{(1)}\mathbf{I}, \\ \vdots \\ \mathbf{G}_{n-k}^{(1)} = \mathbf{u}_1\mathbf{v}_{n-k}^t + p_{n-k}^{(1)}\mathbf{I} \end{cases}$$
$$\vdots$$
$$\begin{cases} \mathbf{G}_1^{(n-k)} = \mathbf{u}_{n-k}\mathbf{v}_1^t + p_1^{(n-k)}\mathbf{I} \\ \vdots \\ \mathbf{G}_{n-k}^{(n-k)} = \mathbf{u}_{n-k}\mathbf{v}_{n-k}^t + p_{n-k}^{(n-k)}\mathbf{I} \end{cases} \tag{28}$$

where $\mathbf{G}_j^{(i)} \in \mathbb{F}_q^{(n-k)\times(n-k)}$ indicates an encoding submatrix for parity node i, associated with the jth information unit. We use an invertible matrix for $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{n-k}]$ and set

$$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{n-k}] = \kappa^{-1}\mathbf{V}'\mathbf{P} \tag{29}$$

where $\mathbf{V}' = (\mathbf{V}^t)^{-1}$ and $\kappa \in \mathbb{F}_q$ is an arbitrary nonzero value such that $1-\kappa^2 \neq 0$. We use a Cauchy matrix $\mathbf{P}$ and let $p_j^{(i)}$ be the (j,i) element of $\mathbf{P}$. Notice that we have $(n-k)$ information units $\mathbf{w}_j \in \mathbb{F}_q^{n-k}$, $1 \leq j \leq n-k$.

Next we remove the last $(n-2k)$ information units and associated elements to obtain the $(n, k, n-1)$ code. This code has information units $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ and encoding submatrices $\mathbf{G}_j^{(i)}$ for $1 \leq j \leq k$ and $1 \leq i \leq n-k$. Lastly, we prune the last $(n-1-d)$ equations in each storage node and also the last $(n-1-d)$ symbols of each information unit. We then obtain the (n, k, d) target code which has encoding submatrices

$$\begin{cases} \bar{\mathbf{G}}_1^{(1)} = \bar{\mathbf{u}}_1\bar{\mathbf{v}}_1^t + p_1^{(1)}\mathbf{I}, \\ \vdots \\ \bar{\mathbf{G}}_k^{(1)} = \bar{\mathbf{u}}_1\bar{\mathbf{v}}_k^t + p_k^{(1)}\mathbf{I} \end{cases}$$
$$\vdots$$
$$\begin{cases} \bar{\mathbf{G}}_1^{(n-k)} = \bar{\mathbf{u}}_{n-k}\bar{\mathbf{v}}_1^t + p_1^{(n-k)}\mathbf{I} \\ \vdots \\ \bar{\mathbf{G}}_k^{(n-k)} = \bar{\mathbf{u}}_{n-k}\bar{\mathbf{v}}_k^t + p_k^{(n-k)}\mathbf{I} \end{cases} \tag{30}$$

where $\bar{\mathbf{u}}_i, \bar{\mathbf{v}}_j \in \mathbf{F}_q^{d-k+1}$ indicate the top $(d-k+1)$ symbols of $\mathbf{u}_i$, $\mathbf{v}_j \in \mathbf{F}_q^{n-k}$, respectively. Here, the size of an identity matrix $\mathbf{I}$ is $(d-k+1)$. For simplicity, we use the same notation for a different dimension of an identity matrix. It can be easily differentiated from the context.

Let us now prove that the resulting code ensures exact repair of all nodes and MDS-code property. We will provide the detailed proof for a simple case of $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{n-k}] = \mathbf{I}$.

1) Exact Repair of Systematic Nodes: By symmetry, we consider only systematic node 1. We connect to $(k-1)$ systematic nodes and $(d-k+1)$ parity nodes. Without loss of generality, we consider parity nodes from 1 to $d-l$. As for a projection vector, we use $\mathbf{e}_1 = [1, 0, \ldots, 0]^t$. We then get

$$\text{From systematic node } j\text{: } \bar{\mathbf{w}}_j^t\mathbf{e}_1,$$
$$\text{From parity node } i\text{: } \bar{\mathbf{w}}_1^t\left(\bar{\mathbf{u}}_i + p_1^{(i)}\mathbf{e}_1\right) + \sum_{j=2}^{k} p_j^{(i)}\left(\bar{\mathbf{w}}_j^t\mathbf{e}_1\right)$$

where $2 \leq j \leq k$ and $1 \leq i \leq d-k+1$. Note that we can achieve simultaneous interference alignment for nonintended signals. The interference term can be canceled with side information obtained from systematic nodes. After cancellation, we rewrite $(d-k+1)$ equations obtained from parity nodes

$$\bar{\mathbf{w}}_1^t\left[\,\bar{\mathbf{u}}_1 + p_1^{(1)}\mathbf{e}_1, \bar{\mathbf{u}}_2 + p_1^{(2)}\mathbf{e}_1, \ldots, \bar{\mathbf{u}}_{d-k+1} + p_1^{(d-k+1)}\mathbf{e}_1\,\right].$$

By (29), $\bar{\mathbf{u}}_i = \kappa^{-1}\left(p_1^{(i)}, \ldots, p_{d-k+1}^{(i)}\right)^t$, $\forall i = 1, \ldots, n-k$. Using the fact that any submatrix of $\mathbf{P}$ is invertible, we can show that the right-hand-side matrix is invertible. This guarantees the decodability of the desired message vector $\bar{\mathbf{w}}_1$.

2) Exact Repair of Parity Nodes: By symmetry, it suffices to consider parity node 1. We connect to k systematic nodes and $(d-k)$ parity nodes. Without loss of generality, we consider parity nodes from 2 to $d-k+1$. As for a projection vector, we use $\bar{\mathbf{u}}_1 = \kappa^{-1}\left(p_1^{(1)}, \ldots, p_{d-k+1}^{(1)}\right)^t$. We then get

$$\text{From systematic node } j\text{: } \bar{\mathbf{w}}_j^t\bar{\mathbf{u}}_1,$$
$$\text{From parity node } i\text{: } \sum_{j=1}^{k}\bar{\mathbf{w}}_j^t\left(\bar{\mathbf{u}}_i\mathbf{e}_j^t + p_j^{(i)}\mathbf{I}\right)\bar{\mathbf{u}}_1$$
$$= \frac{1}{\kappa}\left(\sum_{j=1}^{k} p_j^{(1)}\bar{\mathbf{w}}_j^t\right)\bar{\mathbf{u}}_i + \sum_{j=1}^{k} p_j^{(i)}\left(\bar{\mathbf{w}}_j^t\bar{\mathbf{u}}_1\right)$$

where $1 \leq j \leq k$ and $2 \leq i \leq d-k+1$. Here, the equality follows from the fact that $\mathbf{e}_j^t\bar{\mathbf{u}}_1 = \kappa^{-1}p_j^{(1)}$. Note that the second term in the parity node equation can be canceled with side information $(\bar{\mathbf{w}}_1^t\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{w}}_k^t\bar{\mathbf{u}}_1)$ obtained from systematic nodes. After cancelation, we rewrite $(d-k)$ equations obtained from parity nodes

$$\left[\sum_{j=1}^{k} p_j^{(1)}\bar{\mathbf{w}}_j^t\right]\left[\,\tfrac{1}{\kappa}\bar{\mathbf{u}}_2, \tfrac{1}{\kappa}\bar{\mathbf{u}}_3, \ldots, \tfrac{1}{\kappa}\bar{\mathbf{u}}_{d-k+1}\,\right].$$

Since we know $\bar{\boldsymbol{w}}_j^t\bar{\boldsymbol{u}}_1$ (side-information obtained from systematic nodes), we can construct $\frac{1}{k}\sum_{j=1}^{k} p_j^{(1)}\bar{\mathbf{w}}_j^t\bar{\boldsymbol{u}}_1$. Adding this value to the above, we get:

$$\left[\sum_{j=1}^{k} p_j^{(1)}\bar{\boldsymbol{w}}_j^t\right]\left[\frac{1}{\kappa}\bar{\boldsymbol{u}}_1, \frac{1}{\kappa}\bar{\boldsymbol{u}}_2, \frac{1}{\kappa}\bar{\boldsymbol{u}}_3, \cdots, \frac{1}{\kappa}\bar{\boldsymbol{u}}_{d-k+1}\right].$$

Using the fact that any submatrix of $\mathbf{P}$ is invertible, we can show that the right-hand-side matrix is invertible. This enables to decode the left-hand-side vector, thus obtaining

$$\bar{\mathbf{w}}_1^t\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{w}}_k^t\bar{\mathbf{u}}_1, \sum_{j=1}^{k} p_j^{(1)}\bar{\mathbf{w}}_j^t. \tag{31}$$

Using this information, we can now regenerate

$$\sum_{j=1}^{k}\left(\bar{\mathbf{w}}_j^t\bar{\mathbf{u}}_1\right)\mathbf{e}_j^t+\sum_{j=1}^{k}p_j^{(1)}\bar{\mathbf{w}}_j^t = \sum_{j=1}^{k}\bar{\mathbf{w}}_j^t\left(\bar{\mathbf{u}}_1\mathbf{e}_j^t+p_j^{(1)}\mathbf{I}\right)=\sum_{j=1}^{k}\bar{\mathbf{w}}_j^t\bar{\mathbf{G}}_j^{(1)}.$$

This matches the content of parity node 1, thus ensuring exact repair of the parity node.

3) The MDS-Code Property: We check the invertibility of a composite encoding submatrix when a Data Collector connects to i systematic nodes and $(k-i)$ parity nodes for $i=0,\ldots,k$. The main idea is to use a Gaussian elimination method as we did in Section IV-C. The verification is tedious and therefore details are omitted.

4) Minimum Required Finite-Field Size: Note that the dimension of a Cauchy matrix $\mathbf{P}$ is $(n-k)$-by-$(n-k)$. Therefore, the minimum finite-field size required to generate the Cauchy matrix is $2(n-k)$, i.e., $q\geq 2(n-k)$.

Appendix D
Proof of Lemma 2

1) Exact Repair: With the Gaussian elimination method, we get

$$\mathbf{A}_1'=\begin{bmatrix}\frac{1}{\alpha} & \frac{1}{\beta}\\ \frac{1}{\alpha} & 0\end{bmatrix},\mathbf{B}_1'=\begin{bmatrix}\frac{1}{\alpha} & \frac{2}{\beta}\\ \frac{1}{\alpha} & 0\end{bmatrix},\mathbf{C}_1'=\begin{bmatrix}0 & \frac{2\alpha}{\beta}\\ \frac{2\beta}{\alpha} & 1\end{bmatrix}$$
$$\mathbf{A}_2'=\begin{bmatrix}0 & \frac{1}{\beta}\\ \frac{1}{\alpha} & \frac{1}{\beta}\end{bmatrix},\mathbf{B}_2'=\begin{bmatrix}0 & \frac{2}{\beta}\\ \frac{1}{\alpha} & \frac{2}{\beta}\end{bmatrix},\mathbf{C}_2'=\begin{bmatrix}0 & \frac{2\alpha}{\beta}\\ \frac{2\beta}{\alpha} & 1\end{bmatrix}. \quad (32)$$

Using this, we can easily check the the existence of eigenvectors (25) and decodabiity of desired signals (24). This completes the proof.

2) The MDS-Code Property: Obviously, all the encoding submatrices are invertible due to their lower-triangular or upper-triangular structure. We consider three cases where a Data Collector connects to: 1) 3 systematic nodes; 2) 2 systematic nodes and 1 parity node; and 3) 1 systematic node and 2 parity nodes. The first is a trivial case where the composite matrix associated with information units is an identity matrix. The second case is also trivial, since each encoding submatrix is invertible so that the composite matrix is invertible as well. For the last case, we consider

$$\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0}\\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{0}\\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{I}\end{bmatrix}=\left[\begin{array}{cc|cc|cc} 2\alpha & 0 & 2\alpha & 0 & 0 & 0\\ 2\beta & \beta & \beta & 2\beta & 0 & 0\\ \hline \alpha & 2\alpha & \alpha & 2\alpha & 0 & 0\\ 0 & 2\beta & 0 & \beta & 0 & 0\\ \hline 2\alpha & 0 & \alpha & 0 & 1 & 0\\ \beta & 2\beta & 2\beta & 2\beta & 0 & 1\end{array}\right]. \quad (33)$$

It is easy to check the invertibility of this matrix via the Gaussian elimination method. The invertibility for all the cases guarantees the MDS property.

Acknowledgment

The authors gratefully acknowledge Prof. P. V. Kumar (of IISc) and his students, N. B. Shah and K. V. Rashmi, for insightful discussions and fruitful collaboration related to the structure of Exact-Repair MDS codes.

References

[1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[2] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Proc. Allerton Conf. Control, Computing and Communication*, Sep. 2007.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 1204–1216, Jul. 2000.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[5] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[6] S. Pawar, S. ElRouayheb, and K. Ramchandran, "On secure distributed data storage under repair dynamics," in *Proc. IEEE Int. Symp. Information Theory*, Austin, TX, Jun. 2010.

[7] R. W. Yeung, *Information Theory and Network Coding*. New York: Springer, 2008.

[8] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear network codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[9] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *Proc. IEEE Information Theory Workshop*, Cairo, Egypt, Jan. 2010.

[10] D. Cullina, A. G. Dimakis, and T. Ho, "Searching for minimum storage regenerating codes," in *Proc. Allerton Conf. Control, Computing and Communication*, Sep. 2009.

[11] M. A. Maddah-Ali, S. A. Motahari, and A. K. Khandani, "Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3457–3470, Aug. 2008.

[12] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degree of freedom for the K user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[13] C. Suh and D. Tse, "Interference alignment for cellular networks," in *Proc. Allerton Conf. Control, Computing and Communication*, Sep. 2008.

[14] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Proc. IEEE Int. Symp. Information Theory*, Seoul, Korea, Jul. 2009.

[15] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. Allerton Conf. Control, Computing and Communication*, Sep. 2009.

[16] Y. Wu, "A construction of systematic MDS codes with minimum repair bandwidth," in *arXiv:0910.2486*, Oct. 2009.

[17] A. S. Householder, *The Theory of Matrices in Numerical Analysis*. Toronto, Canada: Dover, 1974.

[18] A. A. Bubrulle, Work Notes on Elementary Matrices Hewlett-Packard Laboratory, 1993, Tech. Rep. HPL-93-69.

[19] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas With Application to Linear Systems Theory*. Princeton, NJ: Princeton Univ. Press, 2005.

Changho Suh (S'10) received the B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2000 and 2002, respectively.

Since 2006, he has been with the Department of Electrical Engineering and Computer Science, University of California at Berkeley. Prior to that, he had been with the Communications Department, Samsung Electronics. His research interests include information theory and wireless communications.

Mr. Suh is a recipient of the Best Student Paper Award from the IEEE International Symposium on Information Theory 2009 and the Outstanding Graduate Student Instructor Award in 2010. He was awarded several fellowships, including the Vodafone U.S. Foundation Fellowship in 2006 and 2007; the Kwanjeong Educational Foundation Fellowship in 2009; and the Korea Government Fellowship from 1996 to 2002.

Kannan Ramchandran (F'05) is a Professor of Electrical Engineering and Computer Science at the University of California at Berkeley, where he has been since 1999. Previously, he was with the University of Illinois at Urbana-Champaign from 1993 to 1999, and AT&T Bell Laboratories from 1984 to 1990. His current research interests include distributed signal processing algorithms for wireless sensor and *ad hoc* networks, multimedia and peer-to-peer networking, multi-user information and communication theory, and wavelets and multi-resolution signal and image processing. He has published extensively in his field, holds eight patents, serves as an active consultant to industry, and has held various editorial and Technical Program Committee positions.

Dr. Ramchandran received the Elaihu Jury Award for the best doctoral thesis in the area of systems from Columbia University, the NSF CAREER award, the ONR and ARO Young Investigator Awards, two Best Paper awards from the IEEE Signal Processing Society, a Hank Magnuski Scholar award for excellence in junior faculty at the University of Illinois, and an Okawa Foundation Prize for excellence in research at Berkeley.