Centralized Multi-Node Repair Regenerating Codes

Marwen Zorgui, Student Member, IEEE, Zhiying Wang, Member, IEEE

Abstract

In a distributed storage system, recovering from multiple failures is a critical and frequent task that is crucial for maintaining the system’s reliability and fault-tolerance. In this work, we focus on the problem of repairing multiple failures in a centralized way, which can be desirable in many data storage configurations, and we show that a significant repair traffic reduction is possible. The fundamental functional tradeoff between the repair bandwidth and the storage size for functional repair is established. Using a graph-theoretic formulation, the optimal tradeoff is identified as the solution to an integer optimization problem, for which a closed-form expression is derived. Expressions of the extreme points, namely the minimum storage multi-node repair (MSMR) and minimum bandwidth multi-node repair (MBMR) points, are obtained. We describe a general framework for converting single erasure minimum storage regenerating codes to MSMR codes. The repair strategy for e failures is similar to that for single failure, however certain extra requirements need to be satisfied by the repairing functions for single failure. For illustration, the framework is applied to product-matrix codes and interference alignment codes. Furthermore, we prove that functional MBMR point is not achievable for linear exact repair codes. We also show that exact-repair minimum bandwidth cooperative repair (MBCR) codes achieve an interior point, that lies near the MBMR point, when \( k \equiv 1 \mod e \), \( k \) being the minimum number of nodes needed to reconstruct the entire data. Finally, for \( k > 2e \), \( e \mid k \) and \( e \mid d \), where \( d \) is the number of helper nodes during repair, we show that the functional repair tradeoff is not achievable under exact repair, except for maybe a small portion near the MSMR point, which parallels the results for single erasure repair by Shah et al.

I. INTRODUCTION

Ensuring data reliability is of paramount importance in modern storage systems. Reliability is typically achieved through the introduction of redundancy. Traditionally, simple replication of data has been adopted in many systems. For instance, Google file systems opted for a triple replication policy [12]. However, for the same redundancy factor, replication systems fall short on providing the highest level of reliability. On the other hand, erasure codes can be optimal in terms of the redundancy-reliability tradeoff. In erasure codes, a file of size \( M \) is divided into \( k \) fragments, each of size \( \frac{M}{k} \). The \( k \) fragments are then encoded into \( n \) fragments using an \((n, k)\) maximum distance separable (MDS) code and then stored at \( n \) different nodes. Using such a scheme, the data is guaranteed to be recovered from any \( n - k \) node erasures, providing the highest level of worst-case data reliability for the given redundancy. However, traditional erasure codes suffer from high repair bandwidth. In the case of a single node erasure, they require downloading the entire data of size \( M \) to repair a single node storing a fragment of size \( \frac{M}{k} \). This expansion factor made erasure codes impractical in some applications using distributed storage systems. In the last decade, the repair problem has gained increasing interest and motivated the research for a new class of erasure codes with better repair capabilities. The seminal work in [10] proposed a new class of erasure codes, called regenerating codes, that optimally solve the repair bandwidth problem. Interestingly, the authors in [10] proved that one can significantly reduce the amount of bandwidth required for repair and the bandwidth decreases as each node stores more information. Formally, suppose any \( k \) out of \( n \) nodes are sufficient to recover the entire file of size \( M \). Assuming that \( d \) nodes, termed helpers, are participating in the repair process, denoting the storage capacity of each node by \( \alpha \) and the amount of information downloaded from each helper by \( \beta \), then, an optimal \((M, n, k, d, \alpha, \beta)\) regenerating code satisfies

\[
M = \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}.
\]

The equation describes the fundamental tradeoff between the storage capacity \( \alpha \) and the bandwidth \( \beta \). Two extreme points can be obtained from the tradeoff. Minimum storage regenerating (MSR) codes correspond to the best storage efficiency with \( \alpha = \frac{M}{k} \), while minimum bandwidth regenerating (MBR) codes achieve the lowest possible bandwidth at the expense of extra storage per node.

If we recover the exact same information as the failed node, we call it exact repair, otherwise we call it functional repair. Using network coding [2], [16], it is possible to construct functional regenerating codes satisfying (1). Following the seminal work in [10], there has been a flurry of interest in designing exact regenerating codes achieving the optimal tradeoff, focusing mainly on the extreme MSR and MBR points, e.g., [5], [14], [22], [23], [30], [33], [35], [37], [39], [42]. For interior

M. Zorgui and Z. Wang are with the Center for Pervasive Communications and Computing, University of California at Irvine, Irvine, CA 92697 USA (e-mail: mzorgui@uci.edu, zhiying@uci.edu). Part of this work has been presented in the 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2016 [43] and IEEE International Symposium on Information Theory, 2017 [44].

Index Terms

Regenerating codes, distributed storage, multi-node repair, minimum storage, minimum bandwidth.
points that are between the MBR and MSR points in the tradeoff of [1]. [29] showed that most points are not achievable for exact repair.

The aforementioned references, as most of the studies on regenerating codes in the literature, focus on the single erasure repair problem. However, in many practical scenarios, such as in large scale storage systems, multiple failures are more frequent than a single failure. Moreover, many systems (e.g., [4]) apply a lazy repair strategy, which seeks to limit the repair cost of erasure codes: instead of immediately repairing every single failure, one waits until \( e \) erasures occur, \( e \leq n - k \), then, the repair is done by downloading the equivalent of the total information in the system to regenerate the erased nodes. However, a natural question of interest is, whether we can reduce the amount of download in such scenarios.

In this work, we consider the repair problem of multiple erasures in a centralized manner. The framework requires the content of any \( k \) out of \( n \) nodes in the system to be sufficient to reconstruct the entire data. Upon failure of \( e \) nodes in the system, the repair is carried out by contacting any \( d \) nodes (helpers) out of the \( n - e \) available nodes, \( d \leq n - e \), and downloading \( \beta \) amount of information from each of the \( d \) helpers. Our objective is to characterize the functional repair tradeoff between the storage per node \( \alpha \) and the repair bandwidth \( \beta \) under the centralized multiple failure repair framework. Under functional repair, the repaired nodes are not necessarily the same as the failed nodes. Exact-repair however requires that the replacement nodes recover exactly the content of the failed nodes. We also seek to investigate the achievability of the functional tradeoff under exact repair.

The centralized repair framework is applicable to many practical situations. Indeed, there are situations in which, due to architectural constraints, it is more desirable to regenerate the lost nodes at a central server before dispatching the regenerated content to the replacement nodes [4]. For instance, one can think of a rack-based node placement architecture [27] in which failures frequently occur to nodes corresponding to a particular rack. In this scenario, a centralized repair of the entire rack is favorable to repairing the rack on the per-node basis. Furthermore, [27] showed that a centralized repair framework can have interesting applications to communication-efficient secret sharing. Finally, centralized repair can be used in a broadcast network, where the repair information is transmitted to all replacement nodes (e.g. [17]). For the above reasons, characterizing the repair-bandwidth tradeoff under the centralized repair framework is important from both an information-theoretic and also a practical perspective.

A. Related work

Cooperative regenerating codes (also known as coordinated regenerating codes) have been studied to address the repair of multiple erasures [13, 31] in a distributed manner. In this framework, each replacement node downloads information from \( d \) helpers in the first stage. Then, the replacement nodes exchange information between themselves before regenerating the lost nodes. Cooperative regenerating codes achieving the extreme points on the cooperative tradeoff have been developed: minimum storage cooperative regenerating (MSCR) codes [20, 31] and minimum bandwidth cooperative regeneration (MBCR) codes [36]. In [20], the authors showed that, given an instance of linear exact MSR codes, it is possible to construct an instance of exact linear MSCR codes for 2 erasures.

The number of nodes involved in the repair of a single node, known as locality, is another important measure of node repair efficiency [13]. Various bounds and code constructions have been proposed in the literature [13, 34]. Recent works have investigated the problem of multiple node repair under locality constraints [24, 32].

The problem of centralized repair has been considered in [6], in which the authors restricted themselves to MDS codes, corresponding to the point of minimum required storage per node. [6] showed the existence of MDS codes with optimal repair bandwidth in the asymptotic regime where the storage per node (as well as the entire information) tends to infinity. In [38], the authors proved that Zigzag codes, which are MDS codes designed initially for repairing optimally single erasures [35], can also be used to optimally repair multiple erasures in a centralized manner. In [27], the authors independently proved that multiple failures can be repaired in Zigzag codes with optimal bandwidth. Moreover, [27] defines the minimum bandwidth multi-node repair codes as codes satisfying the property of having the downloaded information \( d\beta \) matching the entropy of \( e \) nodes. Based on that, the authors derived lower bound on \( \beta \) for systems having a certain entropy accumulation property and then showed achievability of the minimum bandwidth using MBCR codes. However, the optimal storage size per node \( \alpha \) is not known under these codes. In [41], the authors presented an explicit MDS code construction that provide optimal repair for all \( e \leq n - k \) and \( k \leq d \leq n - e \) simultaneously. The authors in [17] studied the problem of broadcast repair for wireless distributed storage which is equivalent to the model we study in this paper. It is worth pointing out that the previous constructions are for high-rate codes, with large subpacketization \( \alpha \). For scalar MDS codes, i.e., \( \beta = 1 \), it is shown that exact-repair cannot be achieved when \( \frac{\alpha}{\beta} > \frac{1}{2} + \frac{2}{\gamma} \). In [27], the authors presented an approach that enables single erasure MSR codes to recover from multiple failures simultaneously with optimal bandwidth. Based on simulation, [21] showed that their approach can provide efficient recovery of most of the failure patterns, but not all of them. The repair problem of Reed Solomon codes has been investigated in [15]. Repairing multiples failures in Reed Solomon codes has been investigated in [8, 29]. In [7], the authors proved that the interference alignment MSR construction of [33], originally designed for repairing any single node failure, can recover from multiple failures in a cooperative way. Specifically, it is shown that any set of systematic nodes, any set of parity-check nodes, or any pair of nodes can be repaired cooperatively with optimal bandwidth.
TABLE I: Summary of achievability results of functional repair tradeoff under exact repair for an $(n, k, d, e, \alpha, \beta)$ distributed storage system. The symbol ✓ denotes achievability while ✗ denotes non-achievability, both of which are under exact repair.

| $e = 1$ | MBR point | MBMR point | Interior points |
| --- | --- | --- | --- |
| ✓ | ✓ | ✗, except maybe for a small portion near the MBR point [29]. |

| $1 < e < k$ | ✓ [6], [41], [Sections IV-B, IV-C, IV-D] | ✗ (for linear codes) [Section IV] |
| --- | --- | --- |
| • if $k \equiv 1 \mod e$: an interior point near the MBMR point is achievable [Section V-D]. |
| • if $e \mid k, e \mid d, k > 2e$: ✗, except maybe for a small portion near the MBR point [Section VI]. |

| $e \geq k$ | ✓ Section [IV-A] | ✓ Section [IV-A] |
| --- | --- | --- |
| ✓ Section [IV-A] |

B. Contributions of the paper

The main contributions of this paper are summarized as follows.

- We first establish the explicit tradeoff between the repair bandwidth and the storage size for functional repair. We obtain the tradeoff using information flow graphs. From the functional tradeoff, we characterize the minimum storage multi-node repair (MSMR) point, and the minimum bandwidth multi-node repair (MBMR) point.
- When the number of erasures $e$ satisfies $e \geq k$, $k$ being the minimum number of nodes needed to reconstruct the entire data, the tradeoff reduces to a single point, for which we provide an explicit code construction.
- We formalize a construction for exact-repair MSMR codes. Given an instance of an exact linear MSR code, we present a framework to construct an instance of an exact linear MSMR regenerating code. We note here that [21] and [20] used similar approach for their numerical results and MSCR codes, respectively. Based on this framework, we study the product-matrix (PM) MSR codes [25] and the interference alignment (IA) construction in [33]. We prove the existence of PM and IA MSMR codes for any number of failures $e$, $e \leq n - k$. Moreover, for the IA code, we prove that one can always efficiently recover from any set of $e \leq n - k$ node failures as long as the failed nodes are either all systematic nodes or all parity nodes; for failures including both systematic and parity nodes, we derive explicit design conditions under which exact recovery is ensured, for some particular system parameters. We note here that unlike previous constructions, our codes are applicable when the code rate is low and use small subpacketization size of $\alpha = k - 1$ or $k$.
- We prove that, to our surprise, functional minimum bandwidth multi-node repair point is not achievable for linear exact repair codes, while linear codes achieve such point for single erasure [25].
- We show that exact-repair minimum bandwidth cooperative repair (MBCR) codes achieve an interior point, that lies near the MBMR point, when $k \equiv 1 \mod e$.
- We show that the functional repair tradeoff is not achievable under exact repair for interior points between MBMR and MSMR points, except for maybe a small portion near the minimum storage multi-node repair point, which parallels the results for single erasure repair [29], for $e \mid k, e \mid d, k > 2e$. The achievability of the functional tradeoff under exact repair is summarized in Table I.
- Finally, we study the repair problem of multiple erasures in MBR regenerating codes and present an MBR construction with optimal repair, simultaneously for varying number of helpers and varying number of erasures.

C. Organization of the paper

The remainder of the paper is organized as follows. A description of the system model is provided in Section II. The analysis of the functional tradeoff is detailed in Section III. Section IV-A describes our code construction for the case $e \geq k$. In Section IV-B, we study the MSMR codes framework and its application to the product-matrix and the interference alignment codes. We prove the non-achievability of MBRM codes under linear exact repair in Section V. The non-achievability of the interior points under exact repair is investigated in Section VI. The repair of multiple erasures for an MBR code is presented in Section VII. Section VIII draws conclusions.

Notation. The superscript $t$ is used to denote the transpose of a matrix. For a matrix $A$, $|A|$ denotes its determinant and $A_{i,j}$ refers to its entry at position $(i,j)$. $e_i$ denotes the standard basis vector whose dimension is clear from the context. For a set $A$, $A \setminus \{i\}$ denotes the resultant set after removing item $i$, while $|A|$ denotes the size of $A$. $I_n$ denotes the identity matrix of size $n$ and $\text{diag}\{\lambda_1, \ldots, \lambda_n\}$ denotes the diagonal matrix with the corresponding elements. $[n]$ denotes the set of elements $\{1, \ldots, n\}$. The symbol $\delta_{(j=l)}$ is 1 if $j = l$ and 0 otherwise. The notations $e \mid k$ and $e \nmid k$ are used to denote whether $k$ is a multiple of $e$, or not, respectively. $u = [u_1, \ldots, u_m]$ denotes a vector of length $m$. 
II. System model and main results

The centralized multi-node repair problem is characterized by parameters $(\mathcal{M}, n, k, d, e, \alpha, \beta)$. We consider a distributed storage system with $n$ nodes storing $\mathcal{M}$ amount of information. The data elements are distributed across the $n$ storage nodes such that each node can store up to $\alpha$ amount of information. The system should satisfy the following two properties:

- Reconstruction property: a data collector (DC) connecting to any $k \leq n$ nodes should be able to reconstruct the entire data.
- Regeneration property: upon failure of $e$ nodes, a central node is assumed to contact $d$ helpers, $d \geq k$, and download $\beta$ amount of information from each of them. New replacement nodes join the system and the content of each is determined by the central node. $\beta$ is called the repair bandwidth. The total bandwidth is denoted $\gamma = d\beta$.

We consider functional repair and exact repair. In the former case, the replacement nodes are not required to be exact copies of the failed nodes, but the repaired code should satisfy again the above two properties. Our objective is to characterize the tradeoff between the storage per node $\alpha$ and the repair bandwidth $\beta$ under the centralized multiple failure repair framework. On the optimal tradeoff, the minimum bandwidth multi-node repair (MBMR) point has the minimum possible $\beta$, and the minimum storage multi-node repair (MSMR) point has the minimum possible $\alpha$.

In the paper, we will use the notation $k = ae + r$, such that $a = \lfloor \frac{k}{e} \rfloor$ and $r = k \mod e$.

We state our main theorems that will be proved in the sequel of the paper. The first result is the explicit functional repair tradeoff of $\alpha$ and $\beta$.

Theorem 2 For fixed system parameters $(\mathcal{M}, n, k, d, e, \alpha, \beta)$, functional regenerating codes satisfying the centralized multi-node repair condition exist if and only if

$$\mathcal{M} \leq f(u^*) = \sum_{i=1}^{\lfloor k/e \rfloor} \min(u_i^\alpha, (d - \sum_{j=1}^{i-1} u_j^\beta)\beta)$$ (2)

where

$$u^* = \begin{cases} [k], & \text{if } k \leq e, \\ [e, \ldots, e]_a, & \text{else if } k = ae, \\ [r, e, \ldots, e]_a, & \text{else if } k = ae + r \text{ and } \alpha \leq \frac{d+ar-ae}{r} \beta, \\ [e, \ldots, e, r]_a, & \text{otherwise}, \end{cases}$$ (3)

where $0 < r < e$.

The next result combines Theorem 5 and 9, that give constructions of MSMR codes for exact repair.

Theorems 5 and 9. There exists interference alignment MSMR codes and product-matrix MSMR codes, defined over large enough finite field, such that any $e \leq n - k$ erasures can be optimally repaired.

The next statement is a combination of Theorem 10 and Theorem 11 which states that MBMR point is not achievable for linear exact repair codes.

Theorems 10 and 11. Exact linear regenerating MBMR codes do not exist when $1 < e < k$.

The following result is a combination of Theorem 13 and Theorem 14 shows the non-feasibility of most interior points for exact repair.

Theorems 13 and 14. For $e|d, e|k, k > 2e$ and any given values of $\mathcal{M}$, exact-repair regenerating codes do not exist for points lying in the interior of the storage-bandwidth tradeoff, except for maybe a small portion near the MSMR point.

III. Functional storage-bandwidth tradeoff

In this section, we study the fundamental tradeoff between the storage size $\alpha$ and the repair bandwidth $\beta$ for $e$ erasures under functional repair. We use the technique of evaluating the minimum cut of a multi-cast information flow graph similar to the single erasure codes [10] and the cooperative regenerating codes [15].

A. Information flow graphs

The performance of a storage system can be characterized by the concept of information flow graphs (IFGs). Our constructed IFG depicts the amount of information transferred, processed and stored during repair. We design our IFG with the following different kinds of nodes (see Figure 1). It contains a single source node $s$ that represents the source of the data object. Each storage node $i$ of the IFG is represented by two distinct nodes: an input storage node $x_{in}^i$, and an output storage node $x_{out}^i$. Each node $x_{out}^i$ is connected to its input node $x_{in}^i$ with an edge of capacity $\alpha$, reflecting the storage constraint of each individual node. The information flow graph is formed with $n$ initial nodes, each with storage size $\alpha$ connected to the source node with
Nodes 1 and 2 are repaired in the first stage and nodes 3 and 4 are repaired in the second stage. A data collector connecting to any 3 nodes should be able to recover the entire information.

edges of capacity $\infty$. The IFG evolves with time. Upon failure of $e$ nodes, $e$ new nodes simultaneously join the system. Each of the replacement nodes $x^j$ is similarly represented by an input node $x^j_{\text{in}}$ and an output node $x^j_{\text{out}}$, linked with an edge of capacity $\alpha$. To model the centralized repair nature of the system, we add a virtual node $x^i_{\text{virt}}$ that links the $d$ helpers to the new storage nodes. Likewise, the virtual node consists of an input node $x^i_{\text{virt,in}}$ and an output node $x^i_{\text{virt,out}}$. The input node $x^i_{\text{virt,in}}$ is connected to the $d$ helpers with edges each of capacity $\beta$. The output node $x^i_{\text{virt,out}}$ is connected to the input node $x^i_{\text{virt,in}}$ with an edge of capacity $e\alpha$, reflecting to the overall size of the data to be stored in the new replacement nodes. The output node $x^i_{\text{virt,out}}$ is then connected to the input nodes $x^j_{\text{in}}$ of the replacement nodes, with edges of capacity $\infty$.

Each IFG represents one particular history of the failure patterns. The ensemble of IFGs is denoted by $G(n,k,d,e,\alpha,\beta)$. For convenience, we drop the parameters whenever it is clear from the context. Given an IFG $G \in \mathcal{G}$, there are $\binom{n}{k}$ different data collectors connecting to $k$ nodes in $G$. The set of all data collector nodes in a graph $G$ is denoted by $\text{DC}(G)$. For an IFG $G \in \mathcal{G}$ and a data collector $t \in \text{DC}(G)$, the minimum cut (min-cut) value separating the source node $s$ and the data collector $t$ is denoted by $\text{mincut}_G(s,t)$.

B. Network coding analysis

The key idea behind representing the repair problem by an IFG lies in the observation that the repair problem can be cast as a multicast network coding problem [10]. Celebrated results from network coding [2], [16] are then invoked to establish the fundamental limits of the repair problem.

According to the max-flow bound of network coding [2], for a data collector to be able to reconstruct the data, the min-cut separating the source to the data collector should be larger or equal to the data object size $M$. Considering all possible data collectors and all possible failure patterns, the following condition is necessary and sufficient for the existence of regenerating codes satisfying the reliability constraint:

$$\min_{G \in \mathcal{G}} \min_{t \in \text{DC}(G)} \text{mincut}_G(s,t) \geq M. \quad (4)$$

Analyzing the minimum cut of all IFGs result in the following theorem.

**Theorem 1.** For fixed system parameters $(\mathcal{M}, n, k, d, e, \alpha, \beta)$, regenerating codes satisfying the centralized multi-node repair condition exist if and only if

$$\mathcal{M} \leq \min_{u \in \mathcal{P}} \left( \sum_{i=1}^{g} \min(u_i \alpha, (d - \sum_{j=1}^{i-1} u_j)\beta) \right) \triangleq \min_{u \in \mathcal{P}} f(u), \quad (5)$$

where

$$f(u) = \sum_{i=1}^{g} \min(u_i \alpha, (d - \sum_{j=1}^{i-1} u_j)\beta), \quad (6)$$

$$\mathcal{P} = \{ u : 1 \leq u_i \leq e \text{ such that } \sum_{i=1}^{g} u_i = k, g \leq k \}. \quad (7)$$
We note that \( k \) was also independently developed in \([27]\).

**Proof:** Define a recovery scenario \( u \in \mathcal{P} \) as follows. A data collector \( DC \) connects to a subset of \( k \) nodes \( \{x^i_{\text{out}} : i \in I\} \), where \( I \) is the subset of contacted nodes. The size of the support of \( u \) corresponds to the number of repair groups of size \( e \) taking part in the reconstruction process, while \( u_i \) corresponds to the number of nodes contacted from repair group \( i \).

As all incoming edges of \( DC \) have infinite capacity, we only examine cuts \( (U, \bar{U}) \) with \( S \in U \) and \( \{x^i_{\text{out}} : i \in I\} \subseteq \bar{U} \). Every directed acyclic graph has a topological sorting, which is an ordering of its vertices such that the existence of an edge \( x \rightarrow y \) implies \( x < y \). We recall that nodes within the same repair group are repaired simultaneously. Since nodes are sorted, nodes considered at the \( i \)-th step cannot depend on nodes considered at \( j \)-th step with \( j > i \).

Considering the \( i \)-th group, consider the case \( |\{x^i_{\text{in}} \in U\}| = m \) and the remaining nodes are such that \( x^i_{\text{in}} \in \bar{U} \).

- if \( x^i_{\text{in}} \in U \), then the contribution of each node is \( m \alpha \). The overall contribution of these nodes is \( m \alpha \).
- else: \( x^i_{\text{virt,out}} \in U \), then the contribution of this node is \( \infty \). Thus, we only consider the case \( x^i_{\text{virt,out}} \in \bar{U} \). Then, we discuss two cases
  - if \( x^i_{\text{virt,in}} \in U \), the contribution to the cut is \( e \alpha \).
  - else, since the \( i \)-th group is the topologically \( i \)-th repair group, at most \( \sum_{j=1}^{i-1} u_j \) edge come from output nodes in \( \bar{U} \). Thus, the contribution is \( (d - \sum_{j=1}^{i-1} u_j) \beta \). Thus, the contribution of this node is \( \min(e \alpha, (d - \sum_{j=1}^{i-1} u_j) \beta) \). Note that \( x^i_{\text{virt,out}} \in \bar{U} \), we do not need to account for other similar nodes.

Thus, if \( m = u_i \), the contribution of the \( i \)-th repair group is \( u_i \alpha \). If \( m < u_i \), the contribution is \( m \alpha + \min(e \alpha, (d - \sum_{j=1}^{i-1} u_j) \beta) \), which can be reduced to \( \min(e \alpha, (d - \sum_{j=1}^{i-1} u_j) \beta) \) if \( m = 0 \). Thus, to lower the cut, either \( m = u_i \) in the case of \( (d - \sum_{j=1}^{i-1} u_j) \beta > u_i \alpha \) or \( m = 0 \) otherwise. Thus, the total contribution of the \( i \)-th repair group is

\[
\min(u_i \alpha, (d - \sum_{j=1}^{i-1} u_j) \beta).
\]

Finally, summing all contributions from different repair groups and considering the worst case for \( u \in \mathcal{P} \) implies that

\[
\min_G \min_{t \in DC(G)} \mincut_G(s, t) = \min_{u \in \mathcal{P}} \left( \sum_{i=1}^{[k/e]} \min(u_i \alpha, (d - \sum_{j=1}^{i-1} u_j) \beta) \right),
\]

with \( \mathcal{P} \) defined as in (7). Therefore, the existence of regenerating codes is guaranteed by (\ref{eq:optimal}) as long as \( \mathcal{M} \leq \min_{G \in \mathcal{G}} \min_{t \in DC(G)} \mincut_G(s, t) \).

**C. Solving the minimum cut problem**

In this section, we derive the structure of the optimal scenario \( u \) in (5) for any set of parameters \( (\alpha, \beta) \). For instance, we show that for \( ae < k < (a + 1)e \), the number of optimal repair groups \( g^* \) (the support of \( u \)) is equal to \( a + 1 \). The result is formalized in the following theorem. Recall that we denote \( a = [k/e], r = k \mod e \).

**Theorem 2.** For fixed system parameters \( (\mathcal{M}, n, k, d, e, \alpha, \beta) \), functional regenerating codes satisfying the centralized multi-node repair condition exist if and only if

\[
\mathcal{M} \leq f(u^*) \leq \sum_{i=1}^{[k/e]} \min(u^*_i \alpha, (d - \sum_{j=1}^{i-1} u^*_j) \beta)
\]

where

\[
u^* = \begin{cases} [k], & \text{if } k \leq e, \\ [e, \ldots, e], & \text{else if } k = ae, \\ [r, e, \ldots, e], & \text{else if } k = ae + r \text{ and } \alpha \leq \frac{d + ar - ae}{r} \beta, \\ \end{cases}
\]

\[\text{a times}\]

\[\text{a times}\]

\[\text{a times}\]

1Strictly speaking, this is only valid when the number of failures/repairs is bounded. A rigorous proof is required to drop the boundedness assumption as \(\ref{eq:optimal}, \ref{eq:optimal}\).
where \( 0 < r < e \).

We denote by \( [v, u, w] \) the vector that is the concatenation of the vectors \( v, u, w \). The next lemma shows that the minimum cut can be obtained by optimizing any subsequence of \( u \) first. The proof follows directly from the definition of \( f() \) in (5) and is omitted.

**Lemma 1.** Consider vectors \( v, w, u, u' \) such that \( \sum_i u_i = \sum_i u'_i \). If

\[
f(u) \geq f(u'),
\]

then,

\[
f([v, u, w]) \geq f([v, u', w]).
\]

In proving the result of Theorem 2 we first characterize the optimal solution in the case of \( k \leq e \). Insight and intuition gained from the first case are used to motivate and derive the general solution. We first state the following lemma, which represents a key step towards proving our result.

**Lemma 2.** Let \( \alpha, \beta, u_1, u_2, d, e, l \) be non-negative reals such that \( u_1 + u_2 = s \leq e \), then the following inequality holds

\[
f([u_1, e, \ldots, e, u_2]) \geq \min_l\{f([s, e, \ldots, e]), f([e, \ldots, e, s])\},
\]

where \( f(u) \) is defined as in (6).

**Proof:** To prove the result, we cast it as an optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \min_{u=[u_1, u_2]} \min(u_1 \alpha, d \beta) + \sum_{i=0}^{l-1} \min(e \alpha, (d - ie - u_1) \beta) + \min(u_2 \alpha, (d - (l + 1)e - u_1) \beta) \\
\text{subject to} & \quad 0 \leq u_1 \leq s, \\
& \quad 0 \leq u_2 \leq e, \\
& \quad u_1 + u_2 = s.
\end{align*}
\]

Substituting \( u_2 \) by \( s - u_1 \) in (12), using the identity \( \min(a, b) = \frac{a + b - |a - b|}{2} \) and after eliminating constant terms, (12) becomes equivalent to

\[
\begin{align*}
\text{minimize} & \quad -u_1 l \beta - |u_1 \alpha - d \beta| - \sum_{i=0}^{l-1} e \alpha - d \beta + ie \beta + u_1 \beta | - |s \alpha - u_1 (\alpha - \beta) - (d - le) \beta| \\
\text{subject to} & \quad 0 \leq u_1 \leq s.
\end{align*}
\]

The objective function in (13), as a function of \( u_1 \), is concave on the interval \([0, s]\). The concavity is due to the convexity of \( x \to |x| \). Therefore, the minimum is achieved at one of the extreme values. Equivalently, \( u_1^* = s \) or \( u_1^* = 0 \).

1) Case \( k \leq e \): In this scenario, connecting to \( k \) nodes from the same repair group yields the worst case scenario from an information flow perspective. Given a particular repair scenario characterized by a vector \( u \), for any two adjacent repair groups (i.e., two adjacent entries in \( u \)) with \( u_1 \) and \( u_2 \) nodes respectively, we have \( u_1 + u_2 \leq e \). One can combine these two groups into a single repair group to achieve a lower cut value. Indeed, from the cut expression in (5), the contribution of the initial set \([u_1, u_2]\) to the cut is \( \min(u_1 \alpha, l \beta) + \min(u_2 \alpha, (l - u_1) \beta) \), for some \( l \). After combining the groups into a single repair group, the contribution of the newly formed repair group is \( \min((u_1 + u_2) \alpha, l \beta) \), which is lower than the initial contribution by virtue of Lemma 2 thus achieving a lower cut. This means that starting from an IFG, we construct a new IFG that has one less repair group and lower min-cut value. This process can be repeated until we end up with a single repair group consisting of \( k \leq e \) nodes, which corresponds to the minimum cut over all graphs in this case.

Therefore, the tradeoff in (5) is simply characterized by \( M \leq \min(k \alpha, d \beta) \). Moreover, \( \alpha_{MSMR} = \alpha_{MBMR} = \frac{M}{k} \) and \( \beta_{MSMR} = \beta_{MBMR} = \frac{M}{d} \). Equivalently, the functional storage bandwidth tradeoff reduces to a single point given by \( (\alpha_{MSMR}, \beta_{MSMR}) = (\frac{M}{k}, \frac{M}{d}) \).

2) Case \( e < k \): Motivated by the previous case, the intuition is that, given a scenario \( u \), one should form a new scenario which exhibits as many groups of size \( e \) as possible. Subsequently, one constructs a scenario \( \tilde{u} \) such that all its entries, except maybe one entry, are equal to \( e \). Lemma 2 addresses the case \( u_1 + u_2 \leq e \). Generalizing it to the case where \( e \leq u_1 + u_2 \leq 2e \) follows the same approach.

**Lemma 3.** Assume that \( u_1 + u_2 = e + s \) and \( 0 \leq u_1, u_2, s \leq e \). Then, the following inequality holds
\[ f([u_1, e, \ldots, e, u_2]) \geq \min(f([s, e, \ldots, e]), f([e, \ldots, e, s])), \]  
(14)

where \( f(u) \) is defined as in (6).

**Proof:** First, we notice that \( u_1 = e + s - u_2 \geq s \) as \( u_2 \leq e \). Then, the proof follows along similar lines as that of Lemma 2 by replacing the constraint in (13) by \( s \leq u_1 \leq e \).

For a fixed \( \beta \), we denote the min-cut corresponding to \( u = [e, \ldots, e, r, e, \ldots, e] \), as a function of \( \alpha \), by \( C_j(\alpha) \), \( j = 0, \ldots, a \).

As will be shown later in the proof of Theorem 2, a careful analysis of the behavior of the \( a + 1 \) different scenarios \( C_j(\alpha) \) is needed to determine the overall optimal scenario leading the lowest minimum cut. We state the result in the following lemma, whose proof is relegated to Appendix A.

**Lemma 4.** There exists a point \( \alpha_c(a) \in [\frac{1}{3} \beta, \frac{2}{3} \beta] \) such that, for any \( 0 \leq j \leq a \),

\[
C_j(\alpha) \begin{cases} 
\geq C_0(\alpha), & \text{if } \alpha \leq \alpha_c(a), \\
\geq C_a(\alpha), & \text{if } \alpha \geq \alpha_c(a), 
\end{cases}
\]
(15)

with

\[
\alpha_c(a) = \frac{d + ar - ae}{r} \beta.
\]
(16)

**Proof of Theorem 2.** Now that we have the necessary machinery, we proceed as follows: given any scenario \( u \), we keep combining and/or changing repair groups by means of successive applications of Lemma 2 and Lemma 3 on subsequences of \( u \) until we can no longer reduce the minimum cut. By Lemma 1 we reduced the overall minimum cut. The algorithm converges because at each step, either the number of repair groups in \( u \) is reduced by one, or the number of repair groups of full size \( e \) is increased by one. As the number of repair groups is lower bounded by \( a + 1 \), and as the number of repair groups of full size \( e \) is upper bounded by \( a \), the algorithm must converge after a finite number of steps. It can be seen then that the above reduction procedure has a finite number of outcomes, given by

- \( u = [e, \ldots, e] \) if \( k = ae \),
- \( u = [e, \ldots, e, r, e, \ldots, e] \) when \( k = ae + r \),

with \( 0 < r < e \) and \( j \in \{0, \ldots, a\} \).

Therefore, if \( e \mid k \), then the optimal scenario corresponds to considering exactly \( a \) repair groups. On the other hand, if \( e \nmid k \), then, it is optimal to consider exactly \( a + 1 \) repair groups. However, the optimal position of the repair group with \( r \) nodes needs to be determined. Then, using Lemma 4 the result in Theorem 2 follows.

**Example 1.** Let \( u = [1, 3, 2, 3, 2] \) with \( e = 3 \). Then, one can start by reducing the first three repair groups \([1, 3, 2] \). This leads to \( u = [3, 3, 3, 2] \). Another approach would be to consider the set \([2, 3, 2] \). Reducing this set leads to either \( u = [1, 3, 3, 3, 1] \) or \( u = [1, 3, 1, 3, 3] \). Reducing further \( u = [1, 3, 3, 3, 1] \) leads to \( u = [2, 3, 3, 3] \) or \( u = [3, 3, 3, 2] \). Reducing \( u = [1, 3, 1, 3, 3] \) leads to \( u = [3, 2, 3, 3] \) or \( u = [2, 3, 3, 3] \). It remains to compare the cuts given by \( u = [3, 3, 3, 2] \), \( u = [3, 3, 2, 3] \), \( u = [3, 2, 3, 3] \) and \( u = [2, 3, 3, 3] \). Following Theorem 2 either \( u = [2, 3, 3, 3] \) or \( u = [3, 3, 3, 2] \) gives the lowest min-cut.

**D. Explicit expression of the tradeoff**

Having characterized the optimal scenario generating the minimum cut in the last section, we are now ready to state the admissible storage-repair bandwidth region for the centralized multi-node repair problem, the proof of which is in Appendix B.

**Theorem 3.** For an \((\mathcal{M}, n, k, d, e, \alpha, \beta)\) storage system, there exists a threshold function \( \alpha^*(\mathcal{M}, n, k, d, e, \alpha, \beta) \) such that for any \( \alpha \geq \alpha^*(\mathcal{M}, n, k, d, e, \alpha, \beta) \), regenerating codes exist. For any \( \alpha < \alpha^*(\mathcal{M}, n, k, d, e, \alpha, \beta) \), it is impossible to construct codes achieving the target parameters. The threshold function \( \alpha^*(\mathcal{M}, n, k, d, \gamma, e) \) is defined as follows:

- if \( k \leq e \), then: \( \alpha^* = \frac{M}{k} \), \( \gamma \in [\mathcal{M}, +\infty) \),
- else if \( k = ae \), then:

\[
\alpha^* = \begin{cases} 
\frac{M}{k}, & \gamma \in [f_0(a-1), +\infty), \\
\frac{\mathcal{M} - \gamma q_0(i)}{\gamma e}, & \gamma \in [f_0(i-1), f_0(i)], \ i = a-1, \ldots, 1,
\end{cases}
\]
(17)

**Proof:** See Appendix B.
Fig. 2: Multi-node repair tradeoff: $k = 8, d = 10, M = 1, e \in \{1, 2, 3, 4, 8\}$.

- else: $k = ae + r$ with $1 \leq r \leq e - 1$, then:

$$\alpha^* = \begin{cases} M, & \gamma \in [f_r(a - 1), +\infty), \\ M - \gamma g_r(i), & \gamma \in [f_r(i - 1), f_r(i)], i = a - 1, \ldots, 1, \\ M - \gamma g_r(0), & \gamma \in [d - k + (a - 1)^2, f_r(0)], \end{cases}$$

where

$$f_r(i) = \frac{2edM}{-k^2 - r^2 + e(k - r) + 2kd - e^2(i^2 + i) - 2ier^i}$$

$$g_r(i) = \frac{(a - i)(-2r + e + 2d - ae - ei)}{2d}.$$  

The functional repair tradeoff is illustrated in Figure 2 for multiple values of $e \in \{1, 2, 3, 4, 8\}$ for $k = 8, d = 10$ and $M = 1$.

**Remark 1.** In the case of $e|k, e|d$, the following equality holds for all points on the tradeoff

$$M = \sum_{i=0}^{a-1} \min(e\alpha, (d - ie)\beta) \iff \frac{M}{e} = \sum_{i=0}^{a-1} \min(\alpha, \frac{d}{e} - i)\beta).$$

Therefore, the tradeoff between $\alpha$ and $\beta$ is the same as the single erasure tradeoff of a system with reduced parameters given by $\frac{M}{e}, \frac{k}{e} = a$ and $\frac{d}{e}$. The expression of the tradeoff in this case can be recovered from [10] with the appropriate parameters.

We now have the expressions of the two extremal points on the optimal tradeoff. We focus on the case $e < k$, as otherwise the optimal tradeoff reduces to a single point.

**MSMR.** The MSMR point is the same irrespective of the relation between $k$ and $e$, and it is given by

$$\alpha_{\text{MSMR}} = \frac{M}{k}, \quad \gamma_{\text{MSMR}} = \frac{ed}{k - d - k + e}. $$

**MBMR.** Interestingly, the MBMR point depends on whether $e$ divides $k$ or not.

- If $k = ae$, we obtain

$$\gamma_{\text{MBMR}} = \frac{2edM}{-k^2 + ek + 2kd} = \frac{dM}{da - e(a^2/2)},$$

$$\alpha_{\text{MBMR}} = \frac{\gamma_{\text{MBMR}}}{e}. $$

The amount of information downloaded for repair is equal to the amount of information stored at the $e$ replacement nodes. This property of the MBMR point is similar to the minimum bandwidth point in the single erasure case [10] and also the minimum bandwidth cooperative repair point [31].
If \( k = ae + r \), we obtain

\[
\gamma_{MBMR} = \frac{2edM}{(k - r + e)(2d - k + r)} = \frac{dM}{d(a + 1) - e\binom{a+1}{2}},
\]

(24)

\[
\alpha_{MBMR} = \gamma_{MBMR} \frac{d + ar - ea}{rd}.
\]

(25)

This situation is novel for multiple erasures as the \( e \) nodes need to store more than the overall downloaded information. This is an extra cost in order to achieve the low value of the repair bandwidth. However, later we will see that for both \( e|k \) and \( e \nmid k \), the total bandwidth at MBMR is equal to the entropy of the failed nodes (see Lemma \ref{lem:MBMR_entropy} and Lemma \ref{lem:MSMR_entropy}):

\[
H(W_{[e]}) = d\beta,
\]

(26)

where \( W_{[e]} \) is the information stored in \( e \) nodes.

**Remark 2.** From the statement of Theorem \ref{thm:tradeoff} we note that if we only consider points between the MSMR and the MBMR points, then the scenario \( u = [r, e, \ldots, e] \) always generates the lowest cut.

**Remark 3.** We compare the centralized repair scheme repairing \( e \) nodes to a separate strategy repairing each of the \( e \) nodes separately using single erasure regenerating codes. We fix \( k, \alpha \) and \( M \).

**Case I:** both strategies use \( d \) helpers. The separate strategy generates a total bandwidth given by \( ed\beta_1 \), while the centralized repair generates \( d\beta_e \), where the subscript indicates the number of erasures. For simplicity, we assume that \( e \mid k \). The case \( e \nmid k \) can be treated in a similar way. For points on the multi-node repair tradeoff, we have

\[
\mathcal{M} = \sum_{j=0}^{a-1} \min(e\alpha, (d - je)e\beta_e).
\]

Consider a point with the same \( \alpha \) and \( d \) on the single erasure tradeoff, we write

\[
\mathcal{M} = \sum_{j=0}^{a-1} \min(e\alpha, (d - je)e\beta_e)
= \sum_{j=0}^{k-1} \min(\alpha, (d - e)\beta_1)
= \sum_{j=0}^{a-1} \sum_{i=0}^{e-1} \min(\alpha, (d - i - je)\beta_1)
\leq \sum_{j=0}^{a-1} e \min(\alpha, (d - je)\beta_1)
= \sum_{j=0}^{a-1} \min(e\alpha, (d - je)e\beta_1).
\]

It follows that \( \beta_e \leq e\beta_1 \) with equality if and only if \( e = 1 \). Moreover, we note that \( \beta_{e,\text{min}} = \frac{2eM}{k + ek + 2ed} \leq e\beta_{1,\text{min}} = \frac{2eM}{d + ed} \). Therefore, for any storage capacity \( \alpha \), multi-node repair requires strictly less bandwidth than a separate strategy for the same number of helpers \( d \).

**Case II:** multi-node repair uses \( d - e + 1 \) helpers, and separate repair uses \( d \) helpers. In this case, the original number of available nodes that can serve as helpers is assumed to be \( d \), and \( e \geq 1 \) erasures occur within the available nodes. Then a separate strategy may generate smaller bandwidth for some values of \( \alpha \), as illustrated by Figure \ref{fig:tradeoff}. However, as \( d \) is sufficiently large, we observe that multi-node repair with \( d - e + 1 \) helpers performs better than a separate strategy for all values of \( \alpha \). Moreover, for MSMR point, the separate repair bandwidth is \( \gamma_{\text{separate}} = \frac{edM}{\alpha - k + e} \), and centralized repair bandwidth is \( \gamma_{MSMR} = \frac{\mathcal{M}}{\alpha - k + e} \). It follows that a centralized repair is always better than a lazy repair strategy, specifically, for \( e > 2 \),

\[
\frac{\gamma_{MSMR}}{\gamma_{\text{separate}}} = \frac{d - (e - 1)}{d} < 1.
\]

(27)

**IV. Exact MSMR codes constructions**

In the remaining of the paper, we study exact-repair. In this section, we analyze the case \( k \leq k \) and then construct MSMR codes when \( e < k \). Later, we study the feasibility of MBMR codes and the interior points under exact repair.
them using $(f_i)$ indexed with $H$. The superscript $\alpha$ obtained from the $E$ nodes successively while multi-node repair is plotted for $d = 7$ and $d = 9$.

A. Construction when $k \leq e$

In the case of $k \leq e$, the optimal tradeoff becomes a single point, so our MSMR construction in this section is also an MBMR code. The optimal parameters satisfy $\alpha = \frac{M}{d}$, $\beta = \frac{d}{k}$, and $\gamma = M$. We note that the overall repair bandwidth and the reconstruction bandwidth are the same. Therefore, one can achieve $\alpha$ and $\gamma$ by dividing the data into $k$ packets and encoding them using $(n, k)$ MDS code (for example, a Reed-Solomon code). The repair can be done by downloading the full content of any $k$ out of $d$ helpers while not contacting $d - k$ nodes. Such repair is asymmetric in nature. We describe one approach for achieving the repair with equal contribution from $d$ helpers.

1) Divide the original file into $kd$ symbols/packets (that is $M = kd$) and encode them using an $(nd, kd)$ MDS code.

2) Store the encoded packets at $n$ nodes, such that each node stores $\alpha = d$ encoded packets.

3) For reconstruction, from any $k$ nodes, we obtain $kd$ different symbols. By virtue of the MDS property, we can reconstruct the data.

4) For repair, each helper node transmits any $\beta = \frac{M}{d} = k$ symbols. The replacement nodes receive $dk$ different coded symbols, which are sufficient to reconstruct the whole data and thus regenerate the missing symbols.

Remark 4. The above procedure works for a specific predetermined $d$. However, it can be generalized to support any value of $d$ satisfying $k \leq d \leq n - e$. For instance, let $\delta = 1 \text{cm}(k, k + 1, k + 2, \ldots, n - e)$ ($\text{lcm}$ denotes the least common multiple). The file of size $M$ is then divided into $\delta$ packets and encoded into $n\delta$ with an MDS code. Each node then stores $\alpha = \frac{M}{\delta} = \frac{kd}{\delta}$ coded symbols. For repair with $d$ helpers, for any $k \leq d \leq n - e$, each node transmits any $\beta = \frac{M}{d} = k\frac{\delta}{d}$ coded symbols for his node. Similarly, it can be seen that reconstruction is always feasible. Note that the constraint of the field size arises from the need for an $(n\delta, k\delta)$ MDS code. The field size needs to be no less than $n\delta$, e.g. Reed Solomon codes.

B. Minimum storage codes framework

In the following sub-sections, we discuss an explicit MSMR code construction method using existing MSR codes designed for single failures. We first describe the general framework, and then present two specific codes.

The framework described in this section has been developed in [21] for numerical simulations. We present it here in a formal way. Consider an instance of an exact linear $(n, k, d, \alpha, \beta)$ MSR code, where $\beta = \frac{\alpha}{d-k+1}$. Consider $e$ nodes, indexed with $f_1, \ldots, f_e$, and other distinct $d - e + 1$ nodes, indexed with $h_1, \ldots, h_{d-e+1}$, such that $d - e + 1 \geq k$. Let $H = \{f_1, \ldots, f_e, h_1, \ldots, h_{d-e+1}\}$ and define $H_{f_j} = H \setminus \{f_j\}$. Consider the repair algorithm corresponding to failed node $f_j$ and helper nodes $H_{f_j}$. We denote by $s_{h_i}^{H_{f_j}}$ the information sent by node $h_i$ to repair node $f_j$, for helpers $h_i \in H_{f_j}$. We drop the superscript $H$ when it is clear from the context. The size of $s_{h_i}^{H_{f_j}}$ is $\beta$ symbols.

Now we construct an $(n, k, d - e + 1, \alpha, \beta, e\beta)$ MSR code. Upon failure of the $e$ nodes $f_1, \ldots, f_e$, the centralized node carrying the repair connects to the set of $d - e + 1$ helpers $h_1, \ldots, h_{d-e+1}$. Each helper node $h_i$ transmits $e\beta$ symbols given by $\cup_{j=1}^{e} \{s_{h_i}^{H_{f_j}}\}$. One can check that the parameters of an MSR code in $(21)$ are satisfied with equality.

The approach consists in using the underlying MSR repair procedure for each of the $e$ failed nodes. Note that $s_{h_i}^{H_{f_j}}$ can be obtained from the $d - e + 1$ helpers, for $i \in [d - e + 1]$. To this end, we need to reconstruct $s_{f_j}^{H_{f_j}}$ for all $\{(i, j), i, j \in [e], i \neq j\}$, which we treat as unknowns. Let $E_{i,j}(\cdot)$ denote the encoding function used to encode the information sent from node $h_i$ to
node $f_j$. Also, let $D_i(\cdot)$ denote the decoding function used by the MSR code to repair node $f_i$ given information from $d$ helpers. Then, we write
\begin{equation}
\begin{align*}
\mathbf{s}_{f_i,f_j}^H &= E_{i,j}(w_{f_i}) \\
&= E_{i,j}(D_i(s_{h,f_i}^H, h \in \mathcal{H}_f)),
\end{align*}
\end{equation}
where $w_j$ denotes the content of node $j$, $i, j \in [e], i \neq j$. Equation (28) generates $e(e-1)$ linear equations in $e(e-1)$ unknowns. Let $s$ be a vector containing the unknowns $s_{f_i,f_j}$. Then, we seek to form a system of linear equations as
\begin{equation}
\mathbf{As} = \mathbf{b},
\end{equation}
where $A$ is a known $(e(e-1) \times e(e-1))$ matrix and $b$ is a known $(e(e-1) \times 1)$ vector. If $A$ is non-singular, one can thus recover $s$. Then, the centralized node can recover the failed node $w_{f_i}$ as $w_{f_i} = D_i(s_{h,f_i}, h \in \mathcal{H}_f)$. We adopt the above approach throughout the section.

**Remark 5.** While the described framework applies to codes with arbitrary rates, we focus in the sequel on low-rate codes. For instance, for a target MSMR code with rate $\frac{1}{2}$, the construction in [25] yields a storage size $\alpha = k^{2k-1}$, while applying the above approach to IA codes or to PM codes results in a smaller storage size $\alpha = k$.

**C. Product-matrix codes**

In this section, we construct MSMR codes for any $e$ erasures based on product-matrix (PM) codes [25]. The PM framework allows design of MBR codes for any value of $d$ and design of MSR codes for $d \geq 2k-2$. Moreover, the PM construction enjoys simple encoding and decoding and ensures optimal repair of all nodes. Product-matrix MSR codes are a family of scalar MSR codes, i.e., $\beta = 1$, designed for parameters $d \geq 2k-2$. We first focus on the case $d = 2k-2$. Under this setup, $\alpha = d - k + 1 = k - 1$. The codeword is represented by an $(n \times \alpha)$ code matrix $C$ such that its $i$th row corresponds to the $\alpha$ symbols stored by the $i$th node. Each code matrix is given by
\begin{equation}
C = \Psi M, \Psi = \begin{bmatrix} \Phi & \Lambda \Phi \end{bmatrix}, M = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix},
\end{equation}
where $\Psi$ is an $(n \times d)$ encoding matrix and $M$ is an $(d \times \alpha)$ message matrix, $S_1$ and $S_2$ are $(\alpha \times \alpha)$ symmetric matrices constructed such that the $\binom{\alpha+1}{2}$ entries in the upper-triangular part of each of the two matrices are filled up by $\binom{\alpha+1}{2}$ distinct message symbols. $\Phi$ is an $(n \times \alpha)$ code matrix and $\Lambda$ is an $(n \times n)$ diagonal matrix. The elements of $\Psi$ should satisfy:

1) any $d$ rows of $\Psi$ are linearly independent;
2) any $\alpha$ rows of $\Phi$ are linearly independent;
3) the $n$ diagonal elements of $\Lambda$ are distinct.

The above conditions may be met by choosing $\Psi$ to be a Vandermonde matrix, in which case its $i$th row is given by $\psi_i^\prime = [1 \ \lambda_i \ \lambda_i^2 \ \ldots \ \lambda_i^\alpha]$. It follows that $\Lambda = \text{diag}\{\lambda_1^\alpha, \ldots, \lambda_n^\alpha\}$. In the following, we assume that $\Psi$ is a Vandermonde matrix.

**Repair of a single erasure in PM codes.** The single erasure repair algorithm [25] is reviewed below. Let $w_i^\prime$ denote the content stored at a failed node. Let $\phi_i^\prime$ be the $i$th row of $\Phi$. Then, $w_i^\prime = \psi_i^\prime M = \phi_i^\prime S_1 + \lambda_i^\alpha \phi_i^\prime S_2$. Let $\mathcal{H}_i = \{h_1, \ldots, h_d\}$ denote the set of $d$ helpers. Each helper $h$ transmits $s_{h,i} = w_i^\prime \phi_i = \psi_i^\prime M \phi_i$ to the replacement node, who obtains $\Psi_{\mathcal{H}_i} M \phi_i$, where $\Psi_{\mathcal{H}_i} = [\psi_{h_1} \ldots \psi_{h_d}]$. Note that $\Psi_{\mathcal{H}_i}$ is invertible by construction. Thus, using the symmetry of $S_1$ and $S_2$, we obtain $(M \phi_i)^\prime = [\phi_i^\prime S_1 \ \phi_i^\prime S_2]$. We can then reconstruct $w_i^\prime = \phi_i^\prime S_1 + \lambda_i^\alpha \phi_i^\prime S_2$.

**Repair of multiple erasures in PM codes.** Given the symmetry of PM codes, we can assume w.l.o.g. that nodes in $\mathcal{E} = \{1, \ldots, e\}$ have failed. Define $\tilde{\mathcal{E}} = \mathcal{E}\setminus\{i\}$. Let $\mathcal{H} = \{1, \ldots, d+1\}$. The centralized node connects to nodes $\{e+1, \ldots, d+1\}$. Each helper node $h$ transmits $\psi_i^\prime M \phi_i, j \in \mathcal{E}$.

Let $s = [s_{1,2}, s_{2,1}, \ldots, s_{e,1}, s_{e,1}, \ldots, s_{e, e-1}, s_{e, e-1}]^\prime$. Our goal is to express explicitly $A$ and $b$ as in [29].

Consider the repair of node $i \in \mathcal{E}$ by the set of helpers in $\mathcal{H}_i = \mathcal{H}\setminus\{i\}$. From the previous subsection, we write
\begin{equation}
w_i = \begin{bmatrix} I_{\alpha} & \lambda_i^\alpha I_{\alpha} \end{bmatrix} \Psi_{\mathcal{H}_i}^{-1} s_{\mathcal{H}_i}, \text{ such that}
\end{equation}
\begin{equation}
\Psi_{\mathcal{H}_i}^\prime = \begin{bmatrix} \psi_1 & \psi_{i-1} & \psi_{i+1} & \ldots & \psi_{d+1} \end{bmatrix},
\end{equation}
\begin{equation}
s_{\mathcal{H}_i}^i = \begin{bmatrix} s_{i,1} & \ldots & s_{i-1,i} & s_{i+1,i} & \ldots & s_{d+1,i} \end{bmatrix}.
\end{equation}
It follows that
\begin{equation}
\begin{align*}
s_{i,j} &= \phi_j^\prime w_i \\
&= \begin{bmatrix} \phi_j^\prime & \lambda_i^\alpha \phi_j^\prime \end{bmatrix} \Psi_{\mathcal{H}_i}^{-1} \left( \sum_{i \in \mathcal{H}_i} s_{i,j} e_{i,i} \right)
\end{align*}
\end{equation}
From (38), noting that $H$ that any two erasures can be optimally repaired.

Theorem 4.

Let

$$
\sum_{l \in E_i} \left( [\phi_j \lambda_i \phi_j] \Psi^{-1}_{H_i e_{l,i}} s_{l,i} + \sum_{l=e+1}^{d+1} \left( [\phi_j \lambda_i \phi_j] \Psi^{-1}_{H_i e_{l,i}} s_{l,i} \right) \right),
$$

(36)

Here we use the column standard basis $e_l$ and denote

$$
e_{l,i} = \begin{cases} e_l, & l < i, \\ e_{l-1}, & l > i. \end{cases}
$$

(37)

Note that the second term in (36) is known from the helpers. Moreover, to compute (36), one may use the inverse of Vandermonde’s matrix formula [19]. Let $h \in \{1, \ldots, d\}$, we have

$$
(\Psi^{-1}_{H_i e_{l,i}})_{h} = \frac{\gamma_h(l, i)}{\prod_{m \in H_i \setminus \{l\}} (\lambda_l - \lambda_m)} = \frac{\gamma_h(l, i)}{\sum_{j=1}^{d} \gamma_j(l, i) \lambda_j^{l-1}},
$$

(38)

where the subscript $h$ in $(\cdot)_h$ means the $h$-th entry, and

$$
\gamma_h(l, i) = (-1)^{d-h} \sum_{m_1 < \ldots < m_{d-h} \in H_i \setminus \{l\}} \lambda_{m_1} \ldots \lambda_{m_{d-h}}.
$$

(39)

As

$$\prod_{m \in H_i \setminus \{l\}} (\lambda - \lambda_m) = \sum_{h=1}^{d} \gamma_h(l, i) \lambda^{l-1},$$

we obtain

$$
[\phi_j \lambda_i \phi_j] \Psi^{-1}_{H_i e_{l,i}} = \frac{\sum_{h=1}^{n} (\gamma_h(l, i) + \lambda_i \lambda_{h+o}(l, i)) \lambda_i^{h-1}}{\sum_{h=1}^{d} \gamma_h(l, i) \lambda_i^{h-1}}.
$$

(40)

Therefore, one can construct $A$ and $b$ in (29) as follows:

- The entries of $b$ are indexed with $(i, j)$, corresponding to $s_{i,j}$. The entry of $b$ at index $(i, j)$ is given by

  $$
  \sum_{l=e+1}^{d+1} [\phi_j \lambda_i \phi_j] \Psi^{-1}_{H_l e_{l,i}} s_{l,i}.
  $$

- Index the column vector $e$ with (i, j) is -1.

  - for $l \in E_i$, column indexed by $(i, j)$ is -1.

For clearer presentation, we first prove the existence of product-matrix MSMR codes for 2 erasures, and then prove for general $e$.

**Theorem 4.** There exists $(n, k, 2k-3, 2, k-1, 2)$ product-matrix MSMR codes, defined over large enough finite field, such that any two erasures can be optimally repaired.

**Proof:** In this case, the matrix $A$ is given by

$$
A = \begin{bmatrix}
-1 & \phi_1^\lambda \phi_1^j \\
\phi_2^\lambda \phi_2^j & \phi_2^\lambda \phi_2^j
\end{bmatrix} \Psi^{-1}_{H_2 e_{1,2}}
$$

(41)

From (38), noting that $H_1 \setminus \{2\} = H_2 \setminus \{1\}$, we obtain

$$
|A| = 1 - [\phi_2 \lambda_i \phi_2] \Psi^{-1}_{H_2 e_{2,1}} [\phi_1 \lambda_i \phi_1] \Psi^{-1}_{H_1 e_{1,2}}
$$

(42)

$$
= 1 - \left( \sum_{h=1}^{k-2} \gamma_h(1, 2) + \lambda_i \lambda_{h+o}(1, 2) \right)
= 1 - \frac{D(\lambda_1, \ldots, \lambda_{d+1})}{N(\lambda_1, \ldots, \lambda_{d+1})},
$$

(43)

$$
|A| \leq 1 - \frac{N(\lambda_1, \ldots, \lambda_{d+1})}{D(\lambda_1, \ldots, \lambda_{d+1})}.
$$

(44)

$|A|$ can be viewed as a rational function of $(\lambda_1, \ldots, \lambda_{d+1})$, as $N$ and $D$ are polynomials in $(\lambda_1, \ldots, \lambda_{d+1})$. We want to show that the following polynomial is not zero:

$$
P(\lambda_1, \ldots, \lambda_{d+1}) \triangleq D(\lambda_1, \ldots, \lambda_{d+1})|A|
$$

(45)

$$
= D(\lambda_1, \ldots, \lambda_{d+1}) - N(\lambda_1, \ldots, \lambda_{d+1}).
$$

(46)

Let $y_0 = (-1)^\alpha \lambda_3 \cdots \lambda_{\alpha+2}$, $y_{\alpha-1} = (-1)^{\alpha-1} \lambda_3 \cdots \lambda_{\alpha+1}$. Then, it can be seen that $P$ contains the term
Let \( y_{a}y_{a-1}(\lambda_{1}^{d-1} + \lambda_{2}^{d-1} - \lambda_{1}^{\alpha}\lambda_{2}^{\alpha-1} - \lambda_{2}^{\alpha}\lambda_{1}^{\alpha-1}) \),

which is not zero. Hence, \( P(\lambda_{1}, \ldots, \lambda_{d+1}) \) is a non-zero polynomial. The PM construction, when based on a Vandermonde matrix, requires \( \lambda_{1}^{\alpha} \neq \lambda_{2}^{\alpha} \) [25], or equivalently, \( g(\lambda_{1}, \lambda_{2}) \triangleq \lambda_{1}^{\alpha} - \lambda_{2}^{\alpha} \neq 0 \). Let \( Q(\lambda_{1}, \ldots, \lambda_{n}) \) denote the polynomial obtained by sweeping across the different set of helpers and failure patterns, taking the product of all corresponding polynomials \( P \), and also multiplied by all \( g \) for all pairs of two nodes. Then, \( Q \) is not identically zero. By Combinatorial Nullstellensatz [3], we can find assignments of the variables \( \{\lambda_{1}, \ldots, \lambda_{n}\} \) over a large enough finite field, such that the polynomial is not zero. Equivalently, we can guarantee the successful efficient recovery of any two erasures among the \( n \) storage nodes.

**Theorem 5.** There exists \((n, k, 2k - e - 1, e, k - 1, e)\) product-matrix MSMR codes, defined over large enough finite field, such that any \( e \) erasures can be optimally repaired.

**Proof:** Entries in each column indexed by \( s_{i,j} \) in \( A \) is either \(-1\) or some other \((e - 1)\) non-zero entries whose denominator is the same and given by \( \prod_{m \in H \setminus \{i\}} (\lambda_{i} - \lambda_{m}) \). We multiply this common denominator to all entries in the column \( s_{i,j} \), for all pairs \( i \neq j \). When \( \lambda_{i} \)'s are chosen to be distinct, this does not change the singularity of \( A \). Denote this transformed matrix by \( B \). Using (40), the entry of \( B \) in row \((i, j)\) and column \((l, m)\) is a polynomial in \( \lambda_{1}, \ldots, \lambda_{d+1} \):

\[
B_{(i,j),(l,m)} = \begin{cases} 
- \sum_{h=1}^{e} \gamma_{h}(i,j)\lambda_{h}^{b_{h}-1}, & l = i, m = j, \\
\sum_{h=1}^{e} (\gamma_{h}(l,i)\lambda_{h}^{b_{h}-1} + \gamma_{h+\alpha}(l,i)\lambda_{h}^{b_{h}-1}), & m = i , \text{ otherwise.}
\end{cases}
\]

Notice that \( e + \alpha - 1 \leq k - 1 + \alpha - 1 = d - 1 \). Let \( y = (-1)^{\alpha-1}\lambda_{e+1} \cdots \lambda_{e+\alpha-1} \), which is a term in \( \gamma_{\alpha+1}(i,j) \) for all \((i,j)\) by (39). We observe that there is a single term \( \pm y\lambda_{i}^{\alpha} \) in the polynomial \( B_{(i,j),(l,m)} \) for the non-zero entries of \( B \).

Recall that the Leibniz formula for determinant of a \( m \times m \) matrix \( B \) is given by

\[
|B| = \sum_{\sigma} \text{sgn}(\sigma) \prod_{i} b_{\sigma(i),i},
\]

where \( \sigma \) is a permutation from the permutation group \( S_{m} \), \( \text{sgn} \) is the sign function of permutations, and \( b_{i,j} \) is the entry \((i, j)\) of \( B \).

**Claim 1.** The term \( T = \prod_{i=1}^{e} (y\lambda_{i}^{\alpha})^{e-1} \) in \(|B|\) has a non-zero coefficient.

Claim 1 implies that \(|B|\) is not a zero polynomial. Then, proceeding as in the proof in Theorem 4 by Combinatorial Nullstellensatz [3], we can find assignments of the variables \( \{\lambda_{1}, \ldots, \lambda_{n}\} \) over a large enough finite field, such that the code guarantees optimal recovery of any set of \( e \) erasures.

Next, we prove Claim 1. Note that the term \( T \) can be created if and only if we take the single term \( \pm y\lambda_{i} \) in the non-zero entries of \( B \) (depending on the permutation \( \sigma \)). Therefore, it is easy to see that the coefficient of term \( T \) in \(|B|\) is the determinant of the following matrix \( C \in \mathbb{R}^{(e-1) \times (e-1)} \):

\[
C_{(i,j),(l,m)} = \begin{cases} 
-1, & l = i, m = j, \\
1, & m = i, \text{ otherwise.}
\end{cases}
\]

One can verify that \( C \) is diagonalizable, and the eigenvalues are:

- Eigenvalue \(-2\) has multiplicity \( e - 1 \), and corresponding eigenspace \( \{(x_{1,2}, \ldots, x_{e,e-1})^{t} : x_{i,j} = x_{i,1}, \forall j \in [e], \sum_{i=1}^{e} x_{i,1} = 0\} \) of dimension \( e - 1 \).
- Eigenvalue \( 1 \) has multiplicity \( e(e-2) \), and corresponding eigenspace \( \{(x_{1,2}, \ldots, x_{e,e-1})^{t} : \sum_{1 \leq i \leq e, i \neq j} x_{i,j} = 0, \forall j \in [e]\} \) of dimension \( e(e-2) \).

When \( e > 2 \), the eigenvalues of \( C \) are non-zero, and \(|C| \neq 0 \). Therefore, Claim 1 is proved and the theorem statement follows.

**Remark 6.** There exist product-matrix MSMR codes, defined over large enough finite field, that simultaneously repair any \( e \in [n-k] \) erasures with optimal bandwidth. Indeed, let \( Q = \prod_{e=2}^{\infty} Q_{e} \), where \( Q_{e} \) is the polynomial corresponding to the code constraints for \( e \) erasures. Recall that the reconstruction process for PM codes requires that \( \alpha_{i}^{\alpha} - \alpha_{j}^{\alpha} \neq 0 \) for \( i \neq j \). Let \( g(\lambda_{1}, \ldots, \lambda_{n}) = \prod_{1 \leq i < j \leq n} (\lambda_{i}^{\alpha} - \lambda_{j}^{\alpha}) \). Let \( P(\lambda_{1}, \ldots, \lambda_{n}) = \prod_{1 \leq i < j \leq n} (\lambda_{i}^{\alpha} - \lambda_{j}^{\alpha})Q(\lambda_{1}, \ldots, \lambda_{n}) \). By Theorem 4 and Theorem 5, \( P \) is not zero and the result follows by Combinatorial Nullstellensatz.

**Example 2.** Consider the product-matrix code with \( n = 11, k = 6, d = 10, \alpha = 5 \). The code is defined over \( \mathbb{F}_{2^{6}} \) with \( \Lambda = \text{diag}\{\lambda_{1}^{\alpha}, \ldots, \lambda_{11}^{\alpha}\} \) and \( \lambda_{i} = g^{i-1} \) with \( g \) being the generator of the multiplicative group of \( \mathbb{F}_{2^{6}} \). Recall that with the
above choice of \( \lambda_i \), any field of size at least \( n\alpha = 55 \) is sufficient to meet the PM code requirements [25]. We first consider repair of \( e = 2 \) erasures. One can check that out of the \( \binom{11}{2} = 55 \) possible 2 failure patterns, 2 patterns are not recoverable according to [29]. \( \mathcal{E} \in \{\{1, 2\}, \{10, 11\}\} \). Considering the same code structure, for \( e = 3 \) erasures, one observes that out of the \( \binom{11}{3} = 165 \) possible 3 failure patterns, 5 patterns are not recoverable: \( \mathcal{E} \in \{\{1, 2, 11\}, \{2, 3, 7\}, \{2, 4, 8\}, \{3, 4, 7\}, \{5, 9, 10\}\} \). It is worth noting that a lazy repair strategy can be beneficial in the following way: if nodes 10 and 11 failed, i.e., \( \mathcal{E} = \{10, 11\} \), then one can optimally repair any 3 erasures \( \mathcal{E} \in \{\{i, 10, 11\}, i \neq 10, i \neq 11\} \). Finally, as suggested by Theorem 4 and Theorem 5 we find that increasing the underlying field size to \( F_{2^8} \) suffices to ensure optimal repair of all two and three erasure patterns in this scenario.

**Remark 7.** Following the code shortening procedure described in [25], we construct an \( (n, k, d - e + 1, e, k - 1, e) \) product-matrix MSR codes with optimal repair for any \( e \in [n - k] \) erasures such that \( 2k - 2 \leq d \leq n - 1 \). First, as described in Remark 4 we consider an \( (n + (d - 2k - 2), k + (d - 2k - 2), d + (d - 2k - 2) - e + 1, d - k + 1, 1) \) product-matrix MSR code \( C' \) in systematic form. The top \( (d - 2k - 2) \) rows (corresponding to \( \alpha(d - 2k - 2) \) information elements over \( F_q \), or \( d - 2k - 2 \) systematic nodes) of \( C' \) are set to all zero. Then, the target code \( C \) is formed by deleting the first \( (d - 2k - 2) \) rows in each code matrix of \( C' \). It can be seen that the repair procedure for \( e \) erasures in \( C \) can be done by invoking that of the original code \( C' \), which leads to the result.

**D. Interference Alignment codes**

In this section, we show the existence of MSMR codes from Interference alignment (IA) codes for \( e = 2 \) erasures. Moreover, we show that for any \( e \leq k \) erasures from only the systematic (or only the parity) nodes, optimal repair bandwidth can be achieved.

The scalar MSR IA code construction in [33] is based on interference alignment techniques. The code is systematic and defined over a finite field \( F_q \) with optimal repair bandwidth for the cases \( \frac{\alpha}{2} \leq d \leq \frac{n}{2} \) and \( d \geq 2k - 1 \). We focus on the case \( n = 2k, d = 2k - 1 \). In this scenario, the storage size is \( \alpha = d - k + 1 = k \).

**Notation.** For an invertible matrix \( B \), we define \( B^T \triangleq (B^{-1})^T \); columns of \( B^T \) constitute the dual basis of the column vectors of \( B \). We use the following symbols to denote the flow of information during repair operations.

- \( s_{i,j}^T \): from systematic node \( i \) to parity node \( j \).
- \( r_{i,j} \): from systematic node \( i \) to systematic node \( j \).
- \( s_{i,j}^T \): from parity node \( i \) to systematic node \( j \).
- \( r_{i,j} \): from parity node \( i \) to parity node \( j \).

The IA code is constructed as below. Consider \( k \) linearly independent vectors \( \{v_1, \ldots, v_k\} \), \( v_i \in \mathbb{F}_q^k, i \in [k] \). Let

\[
V = [v_1, \ldots, v_k], \quad U = \kappa^{-1}V'P,
\]

where every submatrix of the \( (k \times k) \) matrix \( P \) is invertible and \( \kappa \) is an arbitrary non-zero constant in \( \mathbb{F}_q \) satisfying \( \kappa^2 - 1 \neq 0 \). Let \( u_{l,i}, l \in [k] \) denote the content of systematic node \( l \) and \( u_{i}^T \) the content of parity node \( i \), \( i \in [k] \). Let \( u_i, v_i, u_i', v_i' \) be the \( i \)-th column of \( U, V, U', V' \), respectively. Then, we have

\[
w_i'^T = \sum_{j=1}^{k} w_j^T G_j^{(i)}, \quad G_j^{(i)} = u_i v_j^T + P_{j,i}I,
\]

such that the matrix \( G_j^{(i)} \) indicates the encoding submatrix for parity node \( i \), associated with information unit \( j \).

**Repair of a systematic node.** Assume systematic node \( l \) fails. The general repair procedure is described in [33]. In this section, we explicitly develop the exact expression of \( w_l \) as it would be needed later in repairing multiple erasures. Each systematic node \( j \in [k] \setminus \{l\} \) transmits \( r_{j,l} = v_j v_l^T \). Each parity node \( i \in [k] \) transmits \( s_{i,l} = u_i' v_l' \). Noting that \( G_j^{(i)} v_l' = \delta_{(j,l)} u_i + P_{j,i}v_l' \), it follows that

\[
s_{i,l}' = u_i'^T(u_i + P_{i,l}v_l') + \sum_{j \neq \ell} P_{j,i}r_{j,l}.
\]

Canceling the interference from systematic nodes, and arranging the contributions of parity nodes in matrix form, we write

\[
\begin{bmatrix}
  s_{1,l}' - \sum_{j \neq 1} P_{j,1}r_{j,l} \\
  \vdots \\
  s_{k,l}' - \sum_{j \neq k} P_{j,k}r_{j,l}
\end{bmatrix} = \begin{bmatrix}
  u_1'^T + P_{1,1}v_1'^T \\
  \vdots \\
  u_k'^T + P_{1,k}v_1'^T
\end{bmatrix} w_l
\]

(52)
where the last equality is obtained by substituting $U$ by its expression in (49). Using the Sherman-Morrison formula, for an invertible square matrix $A$ of size $k \times k$ and vectors $u, v$ of length $k$, 

$$(A + uv^t)^{-1} = A^{-1} - \frac{A^{-1}uv^tA^{-1}}{1 + v^tA^{-1}u},$$

we obtain that $(\frac{1}{\kappa}P(t + \kappa e_ie_i^t)V^{-1})^{-1} = U' - \frac{\kappa^2}{1 + \kappa V e_ie_i^t}P'$, it follows that

$$w_l = (U' - \frac{\kappa^2}{1 + \kappa V e_ie_i^t}P') \left[ \begin{array}{c} s_{1,l} - \sum_{j \neq l} P_{j,1}r_{j,l} \\ \vdots \\ s_{k,l} - \sum_{j \neq l} P_{j,k}r_{j,l} \end{array} \right].$$

### Repair of a parity node

The repair of a parity node is optimally achieved through the duality property of IA codes that results in a structure that is also conducive to interference alignment. Indeed, inverting the roles of parity and systematic nodes, it follows from (55) that

$$w_i = \sum_{j=1}^k w_j G_j'' G_j'' = \frac{1}{1 - \kappa^2}(v_i u_j' - \kappa^2 P_{i,j} I).$$

Assume parity node $l$ fails, then systematic node $i$ transmits $s_{i,l} = w_i u_l$ and parity node $j$ sends $r_{j,l} = w_j u_l$. Note that $G_j'' u_1 = \frac{1}{1 - \kappa^2}(-\kappa^2 u_j + \delta_{l,j} v_i')$. It follows that

$$s_{i,l} = w_i' \frac{1}{1 - \kappa^2}(v_i - \kappa^2 P_{i,l} u_1) + \sum_{j \neq l} -\kappa^2 \frac{1}{1 - \kappa^2} P_{i,l} r_{j,l}. \tag{57}$$

Combining information from different helpers and after simplification, we obtain

$$\begin{bmatrix} s_{1,l} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq l} P_{1,j} r_{j,l} \\ \vdots \\ s_{k,l} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq l} P_{k,j} r_{j,l} \end{bmatrix} = \frac{1}{1 - \kappa^2} \begin{bmatrix} v_1' - \kappa^2 P_{1,l} u_1' \\ \vdots \\ v_k' - \kappa^2 P_{k,l} u_1' \end{bmatrix} w_l' \tag{58}$$

$$= \frac{\kappa}{1 - \kappa^2} P'(I - \kappa e_ie_i^t)U' w_l', \tag{59}$$

where the last equality is obtained by replacing $V^{-1} = \kappa P' U'$. Inverting the system of equations and using the Sherman-Morrison formula, we obtain

$$w_l' = ((1 - \kappa^2)V + (1 + \kappa)U' e_ie_i^t P') \begin{bmatrix} s_{1,l} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq l} P_{1,j} r_{j,l} \\ \vdots \\ s_{k,l} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq l} P_{k,j} r_{j,l} \end{bmatrix}. \tag{60}$$

### Repair of multiple erasures

We start by deriving the equations relating the different unknowns as in (29). From (55), we write

$$s_{l,m} = u_{m,l} w_l \tag{61}$$

$$= (u_m u' - \frac{\kappa^2}{1 + \kappa} u_m V e_ie_i^t P') \begin{bmatrix} s_{1,l} - \sum_{j \neq l} P_{j,1}r_{j,l} \\ \vdots \\ s_{k,l} - \sum_{j \neq l} P_{j,k}r_{j,l} \end{bmatrix} \tag{62}$$
\begin{align}
  \left( e_{m}^{t} - \frac{\kappa}{1 + \kappa} P_{l,m} e_{l}^{t} P' \right) \begin{bmatrix}
    s_{1,l} - \sum_{j \neq l} P_{l,j} t_{j,l} \\
    \vdots \\
    s_{k,l} - \sum_{j \neq l} P_{j,k} t_{j,l}
  \end{bmatrix},
\end{align}
\tag{63}

where the equality is obtained by noting that \(U'V = \frac{1}{2} P'\). To illustrate the equations as in \(68\), let us consider the information transferred among 3 systematic \(l_1, l_2, l_3 \in [k]\) and 3 parity nodes \(m_1, m_2, m_3 \in [k]\), and denote by all other transferred information as “rest”. It follows that,

\begin{align}
  s_{l_1,m_1} &= (e_{m_1}^{t} - \frac{\kappa}{1 + \kappa} P_{l_1,m_1} e_{l_1}^{t} P') (e_{m_1} s_{m_1,l_1}^{'} + e_{m_2} s_{m_2,l_1}^{'} - P' e_{l_2} t_{l_2,l_1} - P' e_{l_3} t_{l_3,l_1}) + \text{rest} \\
  &= (1 - \frac{\kappa}{1 + \kappa} P_{l_1,m_1} P_{l_1,m_1}) s_{m_1,l_1}^{'} - \frac{\kappa}{1 + \kappa} P_{l_1,m_1} P_{l_1,m_2} s_{m_2,l_1}^{'} - P_{l_2,m_1} t_{l_2,l_1} - P_{l_3,m_1} t_{l_3,l_1} + \text{rest}. \tag{64}
\end{align}

Similarly, starting from \(65\), we write

\begin{align}
  r_{l_1,l_2} &= v_{l_2}^{t} w_{l_1} \tag{65} \\
  &= v_{l_2}^{t} (U' - \frac{\kappa^2}{1 + \kappa} V e_{l_1} e_{l_1}^{t} P') \begin{bmatrix}
    s_{1,l_1}^{'} - \sum_{j \neq l_1} P_{j,1} t_{j,l_1} \\
    \vdots \\
    s_{k,l_1}^{'} - \sum_{j \neq l_1} P_{j,k} t_{j,l_1}
  \end{bmatrix} \tag{66} \\
  &= \frac{\kappa}{1 + \kappa} e_{l_2}^{t} P' \begin{bmatrix}
    s_{1,l_1}^{'} - \sum_{j \neq l_1} P_{j,1} t_{j,l_1} \\
    \vdots \\
    s_{k,l_1}^{'} - \sum_{j \neq l_1} P_{j,k} t_{j,l_1}
  \end{bmatrix}, \tag{67}
\end{align}

where the equality is obtained by noting that \(V^{t} U' = \kappa P'\). It follows that

\begin{align}
  r_{l_1,l_2} &= \frac{\kappa}{1 + \kappa} e_{l_2}^{t} P' (s_{m_1,l_1}^{'} e_{m_1} + s_{m_2,l_1}^{'} e_{m_2} - P' e_{l_2} t_{l_2,l_1} - P' e_{l_3} t_{l_3,l_1}) + \text{rest} \\
  &= (\kappa P_{l_2,m_1}^{'} s_{m_1,l_1} + (\kappa P_{l_2,m_2}^{'} s_{m_2,l_1}^{'} - \kappa r_{l_2,l_1}) + \text{rest}. \tag{68}
\end{align}

Proceeding in a similar way, starting from \(60\), we obtain

\begin{align}
  s_{m_1,l_1}^{'} &= v_{l_1}^{t} w_{m}^{t} \tag{70} \\
  &= v_{l_1}^{t} ((1 - \kappa^2) V + (1 + \kappa) U' e_{m} e_{m}^{t} P') \begin{bmatrix}
    s_{1,m} + \frac{\kappa^2}{1 - \kappa} \sum_{j \neq m} P_{1,j} t_{j,m} \\
    \vdots \\
    s_{k,m} + \frac{\kappa^2}{1 - \kappa} \sum_{j \neq m} P_{k,j} t_{j,m}
  \end{bmatrix} \tag{71} \\
  &= ((1 - \kappa^2) e_{l_1}^{t} + (1 + \kappa) \kappa P_{l,m}^{t} e_{m}^{t} P') \begin{bmatrix}
    s_{1,m} + \frac{\kappa^2}{1 - \kappa} \sum_{j \neq m} P_{1,j} t_{j,m} \\
    \vdots \\
    s_{k,m} + \frac{\kappa^2}{1 - \kappa} \sum_{j \neq m} P_{k,j} t_{j,m}
  \end{bmatrix}, \tag{72}
\end{align}

It follows that

\begin{align}
  s_{m_1,l_1}^{'} &= ((1 - \kappa^2) e_{l_1}^{t} + (1 + \kappa) \kappa P_{l,m}^{t} e_{m}^{t} P') (s_{l_1,m} e_{l_1} + s_{l_2,m} e_{l_2} + \frac{\kappa^2}{1 - \kappa} P' e_{m_2} t_{m_2,m_1} + \frac{\kappa^2}{1 - \kappa} P' e_{m_3} t_{m_3,m_1}) + \text{rest} \\
  &= (1 - \kappa^2 + \kappa (1 + \kappa) P_{l_1,m_1} P_{l_1,m_1}) s_{l_1,m_1} + (\kappa (1 + \kappa) P_{l_1,m_1} P_{l_1,m_1}) s_{l_2,m_1} + (\kappa^2 P_{l_1,m_2}^{t} t_{m_2,m_1} \\
  &\quad + (\kappa^2 P_{l_1,m_3}^{t} t_{m_3,m_1} + \text{rest}. \tag{73}
\end{align}

Similarly, starting from \(60\), we write

\begin{align}
  s_{l_1,m_1}^{'} &= (1 - \kappa^2) e_{l_1}^{t} + (1 + \kappa) \kappa P_{l_1,m_1} P_{l_1,m_1}^{t} e_{m}^{t} P' (s_{l_1,m_1} e_{l_1} + s_{l_2,m_1} e_{l_2} + \frac{\kappa^2}{1 - \kappa} P' e_{m_2} t_{m_2,m_1} + \frac{\kappa^2}{1 - \kappa} P' e_{m_3} t_{m_3,m_1}) + \text{rest} \\
  &= (1 - \kappa^2 + \kappa (1 + \kappa) P_{l_1,m_1} P_{l_1,m_1}) s_{l_1,m_1} + (\kappa (1 + \kappa) P_{l_1,m_1} P_{l_1,m_1}) s_{l_2,m_1} + (\kappa^2 P_{l_1,m_2}^{t} t_{m_2,m_1} \\
  &\quad + (\kappa^2 P_{l_1,m_3}^{t} t_{m_3,m_1} + \text{rest}. \tag{74}
\end{align}
\[ r'_{m_1,m_2} = u'_{m_2}u'_{m_1} \]
\[ = u'_{m_2}((1 - \kappa^2)V + (1 + \kappa)U'e_m'1P') \]
\[ \begin{bmatrix} s'_{1,m_1} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq m_1} P'_{1,j}r'_{j,m_1} \\ \vdots \\ s'_{k,m_1} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq m_1} P'_{k,j}r'_{j,m_1} \end{bmatrix} \]
\[ = \frac{1 - \kappa^2}{\kappa} e'_{m_2} P' \]
\[ \begin{bmatrix} s'_{1,m_1} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq m_1} P'_{1,j}r'_{j,m_1} \\ \vdots \\ s'_{k,m_1} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq m_1} P'_{k,j}r'_{j,m_1} \end{bmatrix} \]
\[ It \ follows \ that \]
\[ r'_{m_1,m_2} = \frac{1 - \kappa^2}{\kappa} e'_{m_2} P' \begin{bmatrix} s_{1,m_1} + \frac{\kappa^2}{1 - \kappa^2} P'_{1,m_1} \sum_{j \neq m_1} r'_{j,m_1} \\ \vdots \\ s_{k,m_1} + \frac{\kappa^2}{1 - \kappa^2} \sum_{j \neq m_1} P'_{k,j}r'_{j,m_1} \end{bmatrix} \]
\[ + \text{rest} \]
\[ = \frac{1 - \kappa^2}{\kappa} P'_{1,m_1}s_{1,m_1} + \left( \frac{1 - \kappa^2}{\kappa} P'_{2,m_1} \right) s_{2,m_1} + \kappa r'_{m_2,m_1} + \text{rest}. \]

Equations (64), (69), (74) and (79) can thus be used to derive \( A \) and \( b \) as defined in (29). In the following theorem, we show that the IA code already provides optimal repair for systematic (respectively parity) failures, without the need to modify the coding matrices.

**Theorem 6.** In the interference alignment MSR code [33], it is possible to optimally repair any set of \( e \leq k \) systematic (respectively parity) failures.

**Proof:** Assume w.l.o.g that nodes \( \{1, \ldots, e\} \) have failed. Let \( s = [r'_{1,2}, r'_{2,1}, \ldots, r'_{e-1,e}, r'_{e,e-1}]^t \). Then, from (69), it follows that \( A \) is a block-diagonal matrix given by
\[ A = \begin{bmatrix} 1 & \kappa \\ \kappa & 1 \\ & & \ddots \\ & & & 1 & \kappa \\ & & & \kappa & 1 \end{bmatrix}. \]
\[ It \ follows \ that \ |A| = \frac{(1 - \kappa^2)^e}{(1 - \kappa^2)^{e-1}} \neq 0 \ as \ \kappa^2 \neq 1 \ by \ design. \ The \ same \ procedure \ applies \ to \ any \ set \ of \ e \ failures \ among \ parity \ nodes \ using \ equation \ (79). \]

**Theorem 7.** The interference alignment MSR code achieves optimal simultaneous repair of one systematic node \( l \) and one parity node \( m \) if \( P_{l,m}(P^{-1})_{m,l} \neq 1 \).

**Proof:** Assume that systematic node \( l \) and parity node \( m \) failed. From (64), we obtain
\[ s_{l,m} = \left( 1 - \frac{\kappa}{1 + \kappa} P_{l,m}P'_{l,m} \right) s'_{m,l} + \text{constant}, \]
\[ \text{Similarly, from (74), we obtain} \]
\[ s'_{m,l} = (1 - \kappa^2 + \kappa(1 + \kappa)P_{l,m}P'_{l,m}) s_{m,l} + \text{constant}. \]

Let \( s = [s_{l,m}, s'_{m,l}]^t \). Then, \( A \) is given by
\[ A = \begin{bmatrix} -1 & 0 \\ - \frac{\kappa}{1 + \kappa} P_{l,m}P'_{l,m} & 1 - \frac{\kappa^2}{1 + \kappa} P_{l,m}P'_{l,m} \end{bmatrix}. \]

It follows that, after simplification, \( |A| \neq 0 \), which implies that \( \kappa^2(P_{l,m}P'_{l,m} - 1)^2 \neq 0 \iff P_{l,m}(P^{-1})_{m,l} \neq 1, \) as \( \kappa \neq 0 \).

Combining Theorems [3] and [7] we know that \( (2k, k, 2k - 2, 2, k, 2) \) MSMR codes for 2 erasures can be constructed through IA codes. We point out that Theorems [6] and [7] have been derived in [7] for cooperative repair, using a different technique. Recall that MSCR codes are in particular MSMR codes [27]. However, their technique cannot be extended to more than two node failures including systematic and parity nodes [7].
Theorem 8. The interference alignment MSR code achieves optimal simultaneous repair of:
- two systematic failures $l_1, l_2$ and one parity failure $m$ if $1 - P_{l_1,m}(P^{-1})_{m,l_1} - P_{l_2,m}(P^{-1})_{m,l_2} \neq 0$,
- one systematic failure $l$ and two parity failures $m_1, m_2$ if $1 - P_{l,m_1}(P^{-1})_{m_1,l} - P_{l,m_2}(P^{-1})_{m_2,l} \neq 0$,
- three systematic failures $l_1, l_2, l_3$ and one parity failure $m$ if
  \[ 1 - P_{l_1,m}(P^{-1})_{m,l_1} - P_{l_2,m}(P^{-1})_{m,l_2} - P_{l_3,m}(P^{-1})_{m,l_3} \neq 0, \]
- one systematic failure $l$ and three parity failures $m_1, m_2, m_3$ if
  \[ 1 - P_{l,m_1}(P^{-1})_{m_1,l} - P_{l,m_2}(P^{-1})_{m_2,l} - P_{l,m_3}(P^{-1})_{m_3,l} \neq 0, \]
- two systematic failures $l_1, l_2$ and two parity failures $m_1, m_2$ if
  \[
  1 - P_{l_1,m_1}(P^{-1})_{m_1,l_1} - P_{l_1,m_2}(P^{-1})_{m_2,l_1} - P_{l_2,m_1}(P^{-1})_{m_1,l_2} - P_{l_2,m_2}(P^{-1})_{m_2,l_2} \\
  + P_{l_1,m_1}(P^{-1})_{m_1,l_1} P_{l_2,m_2}(P^{-1})_{m_2,l_1} P_{l_1,m_1}(P^{-1})_{m_1,l_2} P_{l_2,m_2}(P^{-1})_{m_2,l_2} \\
  - P_{l_1,m_1}(P^{-1})_{m_1,l_1} P_{l_2,m_2}(P^{-1})_{m_2,l_1} P_{l_1,m_2}(P^{-1})_{m_2,l_2} P_{l_2,m_1}(P^{-1})_{m_1,l_2} \neq 0.
  \]

Proof: The proof follows along similar lines as Theorem 7 by constructing $A$ using (64), (69), (74) and (79). The explicit expression of $|A|$ can then be obtained using the Symbolic Math Toolbox of MATLAB, from which the above conditions can be readily obtained.

Combining Theorems 6 and 8 we know that $(2k, k, 2k - e, e, k, e)$ MSMR codes for $e = 3, 4$ erasures can be constructed through IA codes.

Remark 8. Deriving an exact condition under which the recovery of multiple failures for large $e$ is not straightforward. However, we suspect that the general formula is given by the following expression

\[
|A| = k^{2sp}(1 - k^2)\binom{\frac{1}{2}}{\frac{1}{2}}(\frac{1}{2})^e \left( 1 - \sum_{I \subset S, J \subset P, |I| = |J| \leq \min(s, p)} \sum_{\sigma \in \Pi_{I,J}} \sum_{\sigma' \in \Pi_{I,J}} (\text{sgn}(\sigma) \prod_{i \in I} P_{i,\sigma(i)})(\text{sgn}(\sigma') \prod_{j \in J} P'_{j,\sigma'(j)}) \right)^e,
\]

(84)

where $\Pi_{I,J}$ is the group of permutations between the two sets $I$ and $J$ ($I$ and $J$ are ordered in increasing order), and $\text{sgn}(\sigma)$ refers to the sign of a permutation $\sigma$, counting the number of inversions in $\sigma$, and given by

\[
\text{sgn}(\sigma) = (-1)^{\sum_{1 < i < j} 1_{s_{\sigma(i)}>s_{\sigma(j)}}}.
\]

(85)

For example, if $I = \{1, 2, 3\}$, $J = \{2, 3, 4\}$ and $\sigma(1) = 3, \sigma(2) = 4, \sigma(3) = 2$. Then, $\text{sgn}(\sigma)=1$.

One can check that the formulas in Theorem 6, Theorem 7 and Theorem 8 satisfy (84). A general proof of (84) is still open.

Example 3. Consider the IA code with $n = 8, k = 4, d = 7, \alpha = 4$. The code is defined over $F_{2^8}$ with $P$ being a Vandermonde matrix

\[
P = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 2 & 4 & 8 \\
1 & 4 & 16 & 10 \\
1 & 8 & 10 & 26
\end{bmatrix}.
\]

(86)

Using Theorem 6, Theorem 7 and Theorem 8 one can check that every two, three and four erasures can be repaired optimally using our repair framework.

In the following theorem, we provide an existence proof of IA MSMR codes for multiple erasures.

Theorem 9. There exists $(2k, k, 2k - e, e, k, e)$ interference alignment MSMR codes, defined over large enough finite field, such that any $e \leq k$ erasures can be optimally repaired.

Proof: From Theorem 6 we know that if the errors are all either systematic or parity nodes, then efficient repair is possible. Thus, we only need to analyze the case of a mixture of systematic and parity nodes failures.

Consider $e \leq k$ failures consisting of $q$ systematic nodes and $p$ parity nodes, described by the sets $Q$ and $P$. W.l.o.g. assume that $Q = [q]$ and $P = [p]$. Let $s$ denote the vector of unknowns such that pairs $(r_{i,j}, r'_{i,j}), (r''_{i,j}, r'''_{i,j})$ and $(s_{i,j}, s'_{i,j})$ are grouped together. Using (64), (69), (73) and (79), we construct $A$ as in (29). Denote the determinant of $A$ as $F(\kappa, P_{i,j}, i \in Q, j \in P) \triangleq |A|$. The rows and columns of $A$ are indexed with $\{r_{i,j}, s_{i,j}, r''_{i,j}, s'_{i,j}\}$. Let $M_{i,j}$ denote the minor in $A$ corresponding
Therefore, for a non-zero rational function $F$ to a non-zero rational function $Q$

e non-zero rational polynomial in $r$
only at columns indexed with $M$
linearly dependent. Thus, the minor of $F$
A to any scalar multiples of the same standard basis vector. Thus, the minor of

Claim 2. $F$ is not identically zero for any $q,p \geq 0, q+p = e \leq k$.

If Claim 2 holds, then the theorem is proved due to the following argument. By symmetry any $e$-erasure pattern corresponds to a non-zero rational function $F$. Recall from [33], the reconstruction process requires that every submatrix of $P$ is invertible. This can be translated into a polynomial constraint given by $Q(P_{i,j}, i \in Q, j \in P) \neq 0$. Let $T \triangleq Q \prod_e \text{erases } F$. Here the product is over all possible $e$ erasures, and the rational function $F$ depend on the erasure pattern. Then, it follows that $T$ is a non-zero rational polynomial in $(\kappa, P_{i,j}, (i,j) \in [k] \times [n-k])$. By Combinatorial Nullstellensatz [3], we can find assignments of the variables $(\kappa, \{P_{i,j}\})$ over a large enough finite field, such that the code guarantees optimal recovery of any set of $e$ erasures.

Next, we prove Claim 2. We assume first that $q \leq \frac{k}{2}$. Let

$$P_{i,j} = 0 \ \forall \ (i,j) \in Q \times P.$$  \hspace{1cm} (87)

Note that one can always construct a (normalized) invertible matrix $P$ satisfying (87), so we can assume $|P| = 1$. Thus $F$ is a polynomial. We will show

$$F(\kappa, P_{i,j} = 0, P'_{i,j}, (i,j) \in Q \times P) = F(\kappa, P_{i,j} = 0, P'_{i,j} = 0, (i,j) \in Q \times P) \neq 0,$$  \hspace{1cm} (88)

which implies

$$F(\kappa, P_{i,j}, P'_{i,j}, (i,j) \in Q \times P) \neq 0.$$  \hspace{1cm} (89)

To this end, we first prove that $F(\kappa, P_{i,j} = 0, P'_{i,j}, (i,j) \in Q \times P)$, viewed as a polynomial of $(\kappa, \{P'_{i,j}\})$, does not depend on $\{P'_{i,j}\}$. From [69] and [74], one can check that $P'_{i,j}$ appears in $A$ at entries given by

- $A_{r_{l,i},s'_{l,j}}$ for $l \in Q \setminus \{i\}$,
- $A_{s'_{m,i},r'_{j,m}}$ for $m \in P \setminus \{j\}$.

For any $l \in Q \setminus \{i\}$, consider the two columns in $A$ indexed by $r_{l,i}$ and $r_{l,t}$. Both columns have non-zero entries only at rows indexed with $r_{l,i}$ and $r_{l,t}$. Then, after removing entries at row $r_{l,i}$, it follows that both columns become linearly dependent, as both columns are scalar multiples of the same standard basis vector. Thus, the minor of $A$ satisfies $M_{r_{l,i},s'_{l,j}} = 0$.

The example in [69] illustrates the case of two systematic failures, given by nodes 1 and 2, and one parity failure, given parity node 1. In this case $s = [s_{11}, s'_{11}, s_{12}, s_{12}, s_{21}, s'_{21}, s'_{12}, s_{21}]$. Setting $P_{i,j} = 0$ for all $i,j$ and looking at the submatrix of $A$ by removing row $r_{1,2}$ and column $s_{2,1}$ in (90), it can be seen that columns $r_{1,2}, r_{2,1}$ are dependent, hence its corresponding minor $M_{r_{1,2},s_{2,1}} = 0$.

Similarly, for any $m \in P \setminus \{j\}$, consider the two rows in $A$ indexed by $r'_{j,m}$ and $r'_{m,j}$. Both rows have non-zero entries only at columns indexed with $r'_{j,m}$ and $r'_{m,j}$. Then, after removing entries at column $r'_{j,m}$, it follows that both rows become linearly dependent. Thus, $M_{r'_{j,m},r'_{m,j}} = 0$.

The minors in $A$ of all terms corresponding to $P'_{i,j}$ are thus equal to zero. Therefore, w.l.o.g. one can assume that $P'_{i,j} = 0 \ \forall (i,j) \in Q \times P$. It follows that $A$ is block-diagonal matrix such that

- Row/column pairs $(r_{i,j}, r_{j,i})$ correspond to $\begin{bmatrix} -1 & -\kappa \\ -\kappa & -1 \end{bmatrix}$,
- Row/column pairs $(r'_{i,j}, r'_{j,i})$ correspond to $\begin{bmatrix} -1 & -\kappa \\ -\kappa & -1 \end{bmatrix}$,
- Row/column pairs $(s_{i,j}, s'_{j,i})$ correspond to $\begin{bmatrix} -1 & 1 \\ 1 - \kappa^2 & -1 \end{bmatrix}$.
- Other entries are 0.

Therefore, $|A| = \kappa^2 \delta p (1 - \kappa^2) (\frac{1}{2})^+ (\frac{1}{2}) \neq 0$, as $\kappa \neq 0$ and $\kappa^2 \neq 1$. 

$$A = \begin{bmatrix}
-1 & -\kappa + 1 & 0 & 0 & 0 & 0 & \frac{1}{2} \\
1 - \kappa^2 + \kappa(\kappa + 1) P_{i,j} & 1 - \kappa P_{i,j} & 0 & 0 & \kappa P_{i,j} & 0 & \frac{1}{2} \\
\kappa P_{i,j} & -1 & -\kappa & 0 & 0 & \kappa P_{i,j} & \frac{1}{2} \\
1 - \kappa^2 + \kappa(\kappa + 1) P_{i,j} & 0 & 0 & 1 - \kappa^2 + \kappa(\kappa + 1) P_{i,j} & -1 & \frac{1}{2} \\
0 & 0 & 0 & -\kappa & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}$$  \hspace{1cm} (90)
Assume now that \( q > \frac{1}{2} \). Then, \( p \leq \frac{1}{2} \). Proceeding similarly, one can show that if \( P_{ij} = 0 \forall (i, j) \in Q \times P \), then all terms \( P_{ij} \) have no impact on \( |A| \) and one obtains similarly \( |A| = \kappa^2 p(1 - \kappa^2)^{\frac{1}{2}} + \frac{1}{2} \).

**V. Non-existence of exact MBMR regenerating codes**

In this section, we explore the existence of linear exact MBMR regenerating codes for \( e < k \). Unlike the single erasure repair problem \([25]\) and the cooperative repair problem \([25]\), we prove that linear exact regenerating codes do not exist. Following \([25], [36]\), we proceed by investigating subspace properties linear exact MBMR codes should satisfy. Then, we prove that the derived properties over-constrain the system.

**A. Subspace viewpoint**

Linear exact regenerating codes for the MBMR point can be analyzed from a viewpoint based on subspaces. A linear storage code is a code in which every stored symbol is a linear combination of the source symbols. Let \( f \) denote an \( M \)-dimensional vector containing the source symbols. Then, any symbol \( x \) can be represented by a vector \( h \) satisfying \( x = f^t h \) such that \( h \in \mathcal{F}^M \), \( \mathcal{F} \) being the underlying finite field. The vectors \( h \) define the code. A node storing \( r \) symbols can be considered as storing \( r \) vectors of the code. Node \( i \) stores \( h^{(i)} \). It is easy to see that linear operations performed on the stored symbols are equivalent to the same operations performed on the these vectors: \( \sum \gamma_i f^t h_i = f^t(\sum \gamma_i h_i) \). Thus, each node is said to store a subspace of dimension at most \( r \). We denote \( W_A \) to denote the subspace stored by all nodes in the set \( A \), \( A \subseteq [n] \). For regeneration, each node passes \( \beta \) symbols. Equivalently, each node passes a subspace of dimension at most \( \beta \). We denote the subspace passed by node \( j \) to repair set \( R \) of \( e \) nodes by \( S^R_j \). The subspace passed by a set of nodes \( A \) to repair a set \( R \) of \( e \) nodes is denoted by \( S^R_A \). The notation \( \bigoplus \gamma X_j \) denotes the direct sum of subspaces \( X_j \).

**Notation.** For a general exact regenerating code, which can be nonlinear, we use by abuse of notation \( W_A, S^R \) to represent the random variables of the stored information in nodes \( A \), and of the transmitted information from helpers \( A \) to failed nodes \( R \). Properties that hold using entropic quantities for a general code do hold when considering linear codes. For instance, consider two sets \( A \) and \( B \). Then, we note the following

\[
H(W_A) \rightarrow \text{dim}(W_A),
\]

\[
H(W_A|W_B) \rightarrow \text{dim}(W_A) - \text{dim}(W_A \cap W_B),
\]

\[
I(W_A, W_B) \rightarrow \text{dim}(W_A \cap W_B),
\]

where the symbol \( \rightarrow \) means translates to. When results hold for general codes, we only prove for the entropy properties, and the proof for the subspace properties of linear codes is omitted. All results on entropic quantities are for general codes, and all results on subspaces are for linear codes. Moreover, all results in this section refers to properties of optimal exact multi-node repair codes with \( k > e \) (constructions for \( k \leq e \) are presented in Section \([V,A]\), some of which are specific to MBMR codes and will be noted.

In this section, we focus on symmetric codes. Namely, the results do not depend on the indices of the nodes. Note that one can always construct a symmetric code form a non-symmetric code \([11]\). We now start by proving some properties that exact regenerating codes should satisfy. We note that it is also presented in \([38] \) Lemma 4).

**Lemma 5.** Let \( B \subseteq [n] \) be a subset of nodes of size \( e \), then for an arbitrary set of nodes \( A \), such that \( 0 \leq |A| \leq d, B \cap A = \emptyset \),

\[
H(W_B|W_A) \leq H(W_B|S^B_A) \leq \min(\epsilon \alpha, (d - |A|)\beta).
\]

**Proof:** If nodes \( B \) are erased, consider the case of having nodes \( A \) and nodes \( C \) as helper nodes, \( |C| = d - |A| \). Then, the exact repair condition requires

\[
0 = H(W_B|S^B_A, S^B_C) = H(W_B|S^B_A) - I(W_B, S^B_C|S^B_A) \\
\geq H(W_B|S^B_A) - H(S^B_C) \\
\geq H(W_B|S^B_A) - (d - |A|)\beta.
\]

Moreover, we have \( H(W_B|S^B_A) \leq H(W_B) \leq |B|\alpha \), \( H(W_B|W_A) \leq H(W_B|S^B_A) \), and the results follows. \( \blacksquare \)

In the next two subsections, we focus on the cases where \( e \mid k \) and \( e \nmid k \), respectively.

**B. Case \( e \mid k \)**

Note that in this case since \( e < k \), we have \( k \geq 2e \). Recall from Theorem \([2]\) that points on the optimal tradeoff satisfy

\[
\mathcal{M} = \sum_{j=0}^{a-1} \min(\epsilon \alpha, (d - je)\beta).
\]
Points between and including MSMR and MBMR satisfy
\[
\frac{d-k+e}{e} \beta \leq \alpha \leq \frac{d}{e} \beta.
\]

**Lemma 6.** (Entropy of data stored): Consider points on the optimal tradeoff. For an arbitrary set $L$ of storage nodes of size $e$, and a disjoint set $A$ such that $|A| = em < k$ for some integer $m$,

\[
H(W_L) = e\alpha,
\]
\[
H(W_L | W_A) = \min(e\alpha, (d-em)\beta).
\]

For linear codes,

\[
\dim(W_L) = e\alpha,
\]
\[
\dim(W_L) - \dim(W_L \cap W_A) = \min(e\alpha, (d-em)\beta).
\]

**Proof:** By reconstruction requirement, we write

\[
\mathcal{M} = H(W_{[k]})
\]
\[
= H(W_{[e]}) + \sum_{j=1}^{a-1} H(W_{e_{j+1},\ldots,e_{j+1}} | W_{[e]})
\]
\[
\leq \min(e\alpha, d\beta) + \sum_{j=1}^{a-1} \min(e\alpha, (d-ej)\beta)
\]
\[
= \mathcal{M},
\]

where (103) uses Lemma 5. Thus, all inequalities must be satisfied with equality.

**Remark 9.** Lemma 6 states that the contents of any group of $e$ nodes are independent. In particular, for each node $i$, we have $H(W_i) = \alpha$. For a set $A$ of nodes, $|A| \leq e$, $H(W_A) = |A|\alpha$.

**Corollary 1.** At the MBMR point, for any set $L$ of size $e$ and disjoint set $A$ of size $|A| = em < k$, we have

\[
\dim(W_L \cap W_A) = em\beta.
\]

**Proof:** By Lemma 6 and $e\alpha = d\beta$,

\[
\dim(W_L) - \dim(W_L \cap W_A) = \min(e\alpha, (d-em)\beta)
\]
\[
= (d-em)\beta.
\]

Using the fact that $\dim(W_L) = e\alpha = d\beta$, we obtain the result.

**Lemma 7.** For any set $E$ of size $e$, and a disjoint set $A$ of size $d$, the MBMR point satisfies

\[
W_E = \bigoplus_{j \in A} S^E_j, \dim(S^E_j) = \beta.
\]

The subspaces $S^E_j$ and $S^E_j$ are linearly independent. For every set $Q \subseteq A$, $\dim(S^E_Q) = |Q|\beta$. Moreover, each subspace has to be in the span of $W_E$: $S^E_j \subseteq W_E$.

**Proof:** For exact repair, we need $W_E \subseteq \sum_j S^E_j$. Thus,

\[
d\beta = e\alpha = \dim(W_E) \leq \dim(\sum_j S^E_j) \leq d\beta.
\]

Thus, every inequality has to be satisfied with equality.

**Lemma 8.** At the MBMR point, for any set $E$ of $e$ nodes and any other disjoint set $Q$ of size $|Q| \leq k-e$, we have

\[
S^Q_E = W_E \cap W_Q, \dim(W_E \cap W_Q) = \dim(S^Q_E) = |Q|\beta.
\]

**Proof:** Consider $Q$ nodes such that $|Q| \leq k-e$ helping in the repair of a set $E$ of $e$ nodes. Let $J$ contains $Q$ such that $|J| = k-e$. Denote $Q^c = J \setminus Q$. From Corollary 1 we have $\dim(W_E \cap W_J) = (k-e)\beta$. On the other hand, from Lemma 7 we have $\dim(S^E_J) = (k-e)\beta$ and $S^E_J \subseteq W_E$. Moreover, by definition, $S^E_J \subseteq W_J$. Thus, $S^E_J \subseteq W_E \cap W_J$. As the dimensions
match, it follows that $S^E_J = W_E \cap W_J$. Note that $S^E_A \subseteq W_E \cap W_A$ holds for any subset $A$ of size $|A| \leq d$. Now, we write

$$S^E_J = W_E \cap W_J = W_E \cap (W_Q + W_{Q'})$$
$$\supseteq W_E \cap W_Q + W_E \cap W_{Q'}$$
$$\supseteq S^E_Q + S^E_{Q'} = S^E_J.$$

This implies that all inclusion inequalities have to be satisfied with equality and the result follows.

The next lemma plays an important role in establishing the non-existence of exact MBMR codes. It only holds true when $e \geq 2$, which conforms with the existence of single erasure MBMR codes.

**Lemma 9.** Consider the MBMR point. When $e \geq 2$, for any set of $e + 2 \leq k$ nodes, labeled 1 through $e + 2$, it holds that

$$\dim(W_{e+2} \cap W_{[e+1]}) = \dim(W_{e+2} \cap W_{[e]}) = \beta. \quad (106)$$

**Proof:** We have

$$\dim(W_{[e+2]}) = \dim(W_{[e]}) + \dim(W_{e+1} + w_{e+2})$$
$$- \dim(W_{[e]} \cap (W_{e+1} + w_{e+2}))$$
$$= e\alpha + 2\alpha - 2\beta,$$

where the second equality follows from Lemma 8. On the other hand, we write

$$\dim(W_{[e+2]}) = \dim(W_{[e]})$$
$$+ \dim(W_{e+1}) - \dim(W_{e+1} \cap W_{[e]})$$
$$+ \dim(W_{e+2}) - \dim(W_{e+2} \cap W_{[e+1]})$$
$$= e\alpha + 2\alpha - \beta - \dim(W_{e+2} \cap W_{[e+1]}).$$

The lemma follows from equating both equations.

**Theorem 10.** Exact linear regenerating MBMR codes do not exist when $2 \leq e < k$ and $e \mid k$.

**Proof:** Assuming that there exists an exact-repair regenerating code, we consider the first $e$ nodes. Then, these nodes store linearly independent vectors. We write, for $i = 1, \ldots, e$, $W_i = (V_{i1}, V_{i2})$ where $V_{i1}$ contains $\beta$ linearly independent columns and $V_{i2}$ contains the remaining $(\alpha - \beta)$ basis vectors for node $i$. Now, consider node $e + 1$. We have $\dim(W_{e+1} \cap W_{[e]}) = \beta$ by Lemma 8. That means that node $e + 1$ contains $\beta$ columns, linearly dependent on the columns from the first $e$ nodes. Since the first $e$ nodes should be linearly independent, w.l.o.g., we can assume that the $\beta$ dependent vectors of node $e + 1$, denoted by $V_{e+1,1}$, is of the form

$$V_{e+1,1} = \sum_{i=1}^{e} V_{i1}x_i, \quad (107)$$

such that $x_i \neq 0_{\beta \times 1}$ $\forall i = 1, \ldots, e$. Now, consider node $e + 2$. From Lemma 9, node $e + 2$ contains $(\alpha - \beta)$ vectors linearly independent form vectors in nodes 1 through $e + 1$. The remaining basis vectors of node $e + 2$ (which are linearly independent of the $(\alpha - \beta)$ vectors) are denoted by $V_{e+2,1}$. Now, to repair any set of $e$ nodes from the set of first $e + 1$ nodes, node $e + 2$ can only pass $V_{e+2,1}$. Otherwise, Lemma 8 will be violated. Then, this implies that $V_{e+2,1} \subseteq W_J$, for all $J \subseteq \{1, \ldots, e + 1\}$ such that $|J| = e$. Then, it can be seen that $V_{e+2,1}$ can only be of the same form in (107)

$$V_{e+2,1} = \sum_{i=1}^{e} V_{i1}y_i, \text{ such that } y_i \neq 0_{\beta \times 1} \forall i = 1, \ldots, e.$$

Similar reasoning applies to node $i$ for $i = e + 3, \ldots, k + 1$ to conclude that $V_{i1}$ can be written as in (107).

Now, assume the first $e$ nodes fail. Then, node $i$ can only pass $V_{i1}$ for $i = e + 1, \ldots, k + 1$. We recall from Lemma 8 that $S^{[e]} = W_i \cap W_{[e]}$. The total number of vectors passed by these nodes is $(k - e + 1)\beta \geq (e + 1)\beta$. On the other hand, from (107), all $V_{i1}$ are generated by $e\beta$ nodes. Thus, the set $\{V_{i1}, i = e + 1, \ldots, k + 1\}$ must be linearly dependent, which contradicts the linear independence property of the passed subspaces passed for repair, as stated by Lemma 7.

**C. Case $e \nmid k$**

Recall that from the analysis of Theorem 7 for $e \nmid k$, at the MBMR point, two scenarios generate the same minimum cut:

$$u_1 = [r, e, \ldots, e] \text{ and } u_2 = [e, \ldots, e, r].$$
Equivalently, we have
\[ M = f(u_1) = f(u_2), \] (108)
where \( f() \) is defined as in (6).
Moreover, all points on the tradeoff satisfy
\[ M = \min(ra, d\beta) + \sum_{i=0}^{a-1} \min(e\alpha, (d - r - ie)\beta) = f(u_1). \] (109)
Points between and including MSMR and MBMR satisfy
\[ \frac{d - k + e}{e} \beta \leq \alpha \leq \frac{d + ar - ae}{r} \beta. \] (110)
Properties satisfied by exact regenerating codes developed in the previous section extend to the case \( e \nmid k \) with slight modifications. We state the properties without detailed proofs as the techniques are the same.

**Lemma 10.** Consider points on the optimal tradeoff. For an arbitrary set \( R \) of storage nodes of size \( r \), and a set \( A \) such that \( |A| = je + r < k \) for some integer \( j \leq a - 1 \), for all exact-regenerating codes operating on the functional tradeoff, it holds that
\[ H(W_R) = r\alpha, \] (111)
\[ H(W_E|W_A) = (d - je)\beta. \] (112)
For linear codes,
\[ \dim(W_R) = r\alpha, \] (113)
\[ \dim(W_E) - \dim(W_E \cap W_A) = (d - je)\beta. \] (114)

*Proof:* The result can be derived by proceeding as in Lemma 6 and using the fact that \( M = f(u_1) \) from (109). \( \blacksquare \)

**Remark 10.** In the case of \( e \nmid k \), a set of \( e \) are no longer linearly independent. This is expected as \( e\alpha > d\beta \). Instead, it can be seen from Lemma 10 that any set of \( r \) nodes are linearly independent.

**Lemma 11.** For exact-regenerating codes operating at the MBMR point, given sets \( E, A, R \), and \( B \) such that \( |E| = je \), \( E \) and \( A \) are disjoint, \( R \) and \( B \) are disjoint, \( |A| = je \) with \( j \leq a - 1 \), \( |R| = r \) and \( |B| = ae \), it holds that
\[ H(W_E) = d\beta, \] (115)
\[ H(W_E|W_A) = (d - je)\beta, \] (116)
\[ H(W_R|W_B) = (d - ae)\beta. \] (117)
For linear codes,
\[ \dim(W_E) = d\beta, \] (118)
\[ \dim(W_E) - \dim(W_E \cap W_A) = (d - je)\beta, \] (119)
\[ \dim(W_R) - \dim(W_R \cap W_B) = (d - ae)\beta. \] (120)

*Proof:* The result can be derived by proceeding as in Lemma 6 and using the fact that \( M = f(u_2) \) from (108) and \( e\alpha \geq d\beta, ra \geq (d - ae)\beta \). \( \blacksquare \)

It is easy to see that Lemma 7 and Lemma 8 hold true for the case \( e \nmid k \), and for conciseness we do not repeat these lemmas.

The following lemma is used to derive the contradiction.

**Lemma 12.** Let \( k = ae + r \), then at the MBMR point, for any set of \( r + 1 \) nodes, it holds that
\[ \dim(W_{r+1} \cap W_{[r]}) = \beta. \] (121)

*Proof:* We have
\[ d\beta = \dim(W_{[e]}) \] (122)
\[ = \sum_{i=1}^{e} \dim(W_i) - \dim(W_i \cap W_{[i-1]}) \] (123)
Namely, we should have $dβ$.

Thus, the span of all sub-spaces $S_i$ follows from Lemma 8. Combining (130) and (132), it follows
\[
\dim(W_{r+1} \cap W_{[r]}) = (e-r) \dim(W_{r+1} \cap W_{[r]}),
\]
where the last equality follows using symmetry. Then, it follows that
\[
\dim(W_{r+1} \cap W_{[r]}) \leq \alpha - aβ.
\]

Combining (125) and (129), we obtain
\[
\sum_{i=r+2}^e \dim(W_i \cap W_{[i-1]}) = (e-r-1)(\alpha - aβ).
\]

On the other hand, we have
\[
\sum_{i=r+2}^e \dim(W_i \cap W_{[i-1]}) \leq \sum_{i=r+2}^e \dim(W_i \cap W_{[i-1]}),
\]
where $E_i$ is a set of $e$ nodes containing the first $i-1$ nodes and arbitrary $e-i+1$ nodes, excluding node $i$, and the equality follows from Lemma 8. Combining (130) and (132), it follows
\[
(e-r-1)(\alpha - aβ) \leq \sum_{i=r+2}^e \dim(W_i \cap W_{[i-1]}) \leq (e-r-1)β.
\]

It follows that $\alpha - aβ = \frac{d-rα}{a} ≤ β$. The last inequality holds only when $d = k$ and $α - aβ = β$. When $d > k$, we have $α - aβ > β$. Therefore, we only consider the case $d = k$. Hence, it follows from (133) that
\[
\sum_{i=r+2}^e \dim(W_i \cap W_{[i-1]}) = (e-r-1)β.
\]

Using (125), we obtain $\dim(W_{r+1} \cap W_{[r]}) = β$.

From the proof of the above lemma, we see that exact linear MBMR codes may only be feasible when $d = k$. The next theorem states that in fact such codes do not exist.

**Theorem 11.** Exact linear regenerating MBMR codes do not exist when $e < k$ and $e \nmid k$.

**Proof:** Consider repair of the set of nodes $E$ containing nodes 1 through $e$. Consider helper node $i$. As $\dim(W_i \cap W_{[r]}) = \dim(W_i \cap W_{[r]}) = β$, it follows that $W_i \cap W_{[r]} = W_i \cap W_{[r]} = S_i^E$. Then, each helper node sends vectors in the span of $W_{[r]}$. Thus, the span of all sub-spaces $S_i^E$ is included in the span of $W_{[r]}$: $\sum_i S_i^E \subseteq W_{[r]}$. This implies that $\dim(\sum_i S_i^E) \leq \dim(W_{[r]})$.

Namely, we should have $dβ ≤ rα$: this is a contradiction as $dβ > rα$. ■
D. Minimum bandwidth cooperative regenerating codes as centralized multi-node repair codes

In [27], the authors argued that MBCR codes can be used as centralized multi-node repair regenerating codes. We recall that MBCR codes are characterized with

\[ \alpha_{\text{MBCR}} = \frac{M(2d + e - 1)}{2k(2d - k + e)} \]
\[ \gamma_{\text{MBCR}} = \frac{M2de}{k(2d - k + e)}. \]  

(135)

In the case that \( e \mid k \), it is shown that MBCR codes achieve the MBMR bandwidth, i.e., \( \gamma_{\text{MBCR}} = \gamma_{\text{MBMR}} \). On the other hand when \( e \nmid k \), by imposing a certain entropy accumulation property on the entropy of any group of \( r \) nodes, [27] showed that the bandwidth achieved by MBCR codes is optimal. It is important to note here that, from (24), it can be checked that the entropy constraint condition results in \( \gamma_{\text{MBCR}} > \gamma_{\text{MBMR}} \). Moreover, in both cases, it is not clear whether the storage size achieved by MBCR codes is optimal. The next theorem determines the cases in which MBCR codes meet the functional tradeoff.

Theorem 12. Assume \( 1 < e \leq k \), then, minimum bandwidth cooperative regenerating codes meet the centralized functional repair tradeoff if and only if \( k = ae + 1 \).

Proof: When \( e \mid k \), from (22) and (23), it follows that \( \gamma_{\text{MBMR}} = \gamma_{\text{MBCR}} \) and \( \alpha_{\text{MBCR}} = \alpha_{\text{MBMR}} + \frac{M(e-1)}{k(2d-k+e)} \). Thus, \( \alpha_{\text{MBMR}} < \alpha_{\text{MBCR}} \). When \( e \nmid k \), from (19) and (25), one can check that \( \gamma_{\text{MBMR}} < \gamma_{\text{MBCR}} < f_r(0) \). Using (18), it follows that the optimal storage size corresponding to \( \gamma_{\text{MBCR}} \) and achieving the functional repair tradeoff is given by

\[ \alpha^*(\gamma_{\text{MBCR}}) = \frac{M - \gamma_{\text{MBCR}}g_r(0)}{r} \]
\[ = \frac{M(2d + e - r)}{k(2d - k + e)} \]
\[ = \alpha_{\text{MBCR}} - \frac{M(r - 1)}{k(2d - k + e)}. \]  

(136)

(137)

(138)

Therefore, \( \alpha^*(\gamma_{\text{MBCR}}) \leq \alpha_{\text{MBCR}} \) with equality if and only if \( r = 1 \).

Figure 4 illustrates the functional tradeoff for fixed \( e = 3, M = 1 \) and multiple values of \( k \in \{7, 8, 9\} \) such that \( d = k \). As proved in Theorem 12, MBCR codes are optimal centralized repair codes only when \( r = 1 \), which corresponds to \( k = 7 \) in Figure 4. When \( e \mid k \), MBCR codes achieve the same bandwidth as MBMR codes, but have a higher storage cost. In Section VII, we investigate the achievability of MBMR codes under exact repair.

Remark 11. Theorem 12 proves that, when \( e \mid k, r = 1 \), MBCR codes achieve an interior point on the functional tradeoff that lies near the MBMR point. We note that the existence of this exact-repair interior point does not contradict the non-feasibility result in Section VII where we assume \( e/k \).

VI. NON-FEASIBILITY OF THE EXACT-REPAIR INTERIOR POINTS

In this section, we study the non-feasibility of the interior points on the optimal functional-repair tradeoff for \( e \mid k, e \mid d, 2e < k \), similarly to [29]. We note that all interior points satisfy \( (d - k + e)\beta \leq e\alpha \leq d\beta \). This can be written as
Thus, we have \(d' - a + 1)\beta \leq \alpha \leq d'\beta\), where \(d' = \frac{d}{e}\) and \(a = \frac{d}{e}\). This is similar to the single erasure case with reduced parameters.

**Parameterization of the interior points.** Let \(\alpha = (d' - p)\beta - \theta\), namely \(e\alpha = (d - ep)\beta - e\theta\) with \(p \in \{0, 1, \ldots, a - 1\}\), \(\theta \in [0, \beta)\) such that \(\theta = 0\) if \(p = a - 1\). Points at the tradeoff satisfy

\[
M = e \sum_{i=0}^{a-1} \min(\alpha, (d' - i)\beta).
\]

### A. Properties of Exact-Repair Codes

We present a set of properties that exact-repair codes, satisfying the optimal functional tradeoff, must satisfy.

**Lemma 13.** For a set \(A\) of arbitrary nodes of size \(e\), a set \(L\) of nodes of size \(e\) such that \(L \cap A = \emptyset\), we have

\[
I(W_L, W_A) = \begin{cases} 
0 & j \leq p, \\
\varepsilon((j - p)\beta - \theta) & p < j < a \\
\varepsilon\alpha & j \geq a.
\end{cases}
\] (139)

**Proof:** First, we note that when \(j \geq a\), \(I(W_L, W_A) = H(W_L) - H(W_L|A) = H(W_L) = e\alpha\). In the following, we assume \(j < a\). We write

\[
I(W_L, W_A) = H(W_L) - H(W_L|A) = e\alpha - \min(e\alpha, (d - j)e\beta) = e\alpha - \min(e\alpha, (d' - j)\beta) = e\alpha - (d' - j)\beta + = e((j - p)\beta - \theta) + .
\] (140)

Here (141) follows from Lemma 6.

**Remark 12.** Note here that only when \(e\mid k\) does (141) hold, when \(e \not\mid k\), \(H(W_L) < e\alpha\). In fact, as pointed out in Remark 11 the non-feasibility result does not hold for \(e \not\mid k\).

**Corollary 2.** For an arbitrary set \(L\) of size \(e\), and a disjoint set \(A\) such that \(|A| = em < k\) for some integer \(m\), we have

\[
H(W_L|S_{A}^{k}) = H(W_L|W_A) = \min(e\alpha, (d - em)\beta).
\] (145)

**Proof:** From Lemma 5 we have \(H(W_L|S_{A}^{k}) \leq \min(e\alpha, (d - em)\beta)\). On the other hand, from Lemma 6

\[
H(W_L|S_{A}^{k}) \geq H(W_L|W_A) = \min(e\alpha, (d - em)\beta).
\] (146)

Thus, \(H(W_L|S_{A}^{k}) = H(W_L|W_A) = \min(e\alpha, (d - em)\beta)\).

**Lemma 14.** In the situation where node \(m\) is an arbitrary helper node assisting in the repair of a second set of arbitrary nodes \(L\) of size \(e\), we have

\[
H(S_{L}^{m}) = \beta,
\] (147)

irrespective of the identity of the other \(d - 1\) helper nodes. Moreover, for set \(B\) of size \(|B| \leq d - k + e\) with \(B \cap L = \emptyset\), we have

\[
H(S_{B}^{k}) = |B|\beta.
\] (148)

**Proof:** Partition the set of \(d\) helpers into \(A\) and \(B\) such that \(|A| = k - e\) and \(|B| = d - k + e\), such that \(m \in B\). We have \(H(W_L|S_{A}^{k}) = \min(e\alpha, (d - k + e)\beta) = (d - k + e)\beta\), as \(e\alpha \geq (d - k + e)\beta\) for all points on the tradeoff. Moreover, exact repair requires \(H(W_L|S_{A}^{k}) = 0\). Thus, \(H(S_{B}^{k}) \geq (d - k + e)\beta\). This implies \(H(S_{B}^{k}) = (d - k + e)\beta\). Moreover, it must hold that \(H(S_{m}^{L}) = \beta\) in addition to \(S_{L}^{k}\) and \(S_{m}^{L}\) being independent if \(m \neq m'\). Moreover, by choosing \(M \subseteq B\), one obtains \(H(S_{M}^{L}) = e\beta\).

**Helper node pooling.** Consider a set \(F\) consisting of a collection of \(f \leq d + e\) nodes \((f\) is a multiple of \(e)\), and a subset \(R\) of the set \(F\) consisting of \(er^\prime\) nodes. A helper node pooling scenario is a scenario where on failure on any \(e\) nodes \(L \subseteq R\), the \(d\) helper nodes assisting in its repair include all the \(f - e\) remaining nodes in \(F\). The remaining helper nodes are fixed given \(L\), denoted by \(V(L)\). Consider a subset of nodes \(M \subseteq F\backslash R\). Partition the nodes in \(R\) into arbitrary but fixed sets

\[
R_{1}, R_{2}, \ldots, R_{r'},
\] (149)
each of size $e$. Denote by $S_M^{R_i} = (S_M^{R_1}, \ldots, S_M^{R_k})$ the collective transmitted information from helper nodes $M$ to repair $R_1, \ldots, R_k$, respectively.

**Lemma 15.** In the helper node pooling scenario where $\min(a, \frac{1}{e}) > p + 2 \geq r'$, for any set of $e$ arbitrary node $M \subseteq F - R$, we have

$$H(S_M^{R}) \leq e(2\beta - \theta). \quad (150)$$

**Proof:** If the statement holds true for some $f$, then it also holds true for all $f' \geq f$ and $r'' \leq r'$. Thus, for the proof, we only need to consider $F = R \cup M, |F| = f = e(p + 3), |R| = r'e = (p + 2)e, |M| = e$.

Consider repair of an arbitrary set of $e$ nodes $L \subseteq R$, where the set of helpers include $M$ and the $e(p + 1)$ remaining nodes in $R$. Then, we write

$$I(S_M^{L}; W_R) = I(S_M^{L}; W_L, W_R - L) \quad (151)$$

$$\geq I(S_M^{L}; W_L|W_R - L) \quad (152)$$

$$= H(W_L|W_R - L) - H(W_L|W_R - L, S_M^{L}) \quad (153)$$

$$\geq H(W_L|W_R - L) - H(W_L|W_R - L, S_M^{L}) \quad (154)$$

$$= \min(\alpha, (d - e(p + 1))\beta) - \min(\alpha, (d - e(p + 2))\beta) \quad (155)$$

$$= (d - e(p + 1))\beta - (d - e(p + 2))\beta = \beta. \quad (156)$$

Here (156) follows from Lemma 6 and Corollary 2

Then, we obtain

$$H(S_M^{R}|W_R) = H(S_M^{R}) - I(S_M^{R}; W_R) \leq e\beta - e\beta = 0. \quad (158)$$

Hence, $H(S_M^{R}|W_R) = 0$. Since $L$ is arbitrary, it follows that $H(S_M^{R}|W_R) = 0$.

It follows from Lemma 15 that

$$H(S_M^{R}) = I(S_M^{R}; W_R) \leq I(W_M; W_R) = e(2\beta - \theta). \quad (159)$$

**Lemma 16.** In the helper node scenario where $\min\{a, \frac{1}{e}\} > p + 1 \geq r'$, for an arbitrary set of $e$ nodes $M \subseteq F - R$, and an arbitrary pair of set of $e$ nodes $L_1, L_2$, it must be that

$$H(S_M^{L_1}|S_M^{L_2}) \leq e\theta, \quad (159)$$

and hence

$$H(S_M^{L_1}) \leq e(\beta + (r' - 1)\theta). \quad (160)$$

**Proof:** The set is $R$ assumed to consist of $|R| = er' = e(p + 1)$ nodes, and the set $F$ is such that $F = R \cup \{M\}, |M| = e$.

Similar to Lemma 15

$$I(S_M^{L}; W_R) \geq H(W_L|W_R - L) - H(W_L|S_M^{L - L}, S_M^{L}) \quad (161)$$

$$= \min(\alpha, (d - (r' - 1)e)\beta) - \min(\alpha, (d - r'e)\beta) \quad (162)$$

$$= (d - pe)\beta - e\theta - (d - (p + 1)e)\beta \quad (163)$$

$$= e(\beta - \theta). \quad (164)$$

Then, it must be that

$$H(S_M^{L}|W_R) = H(S_M^{L}) - I(S_M^{L}; W_R) \leq e\beta - e(\beta - \theta) = e\theta. \quad (165)$$

Note that the last inequality holds for any set $L \subseteq R$. Next, consider $L_1, L_2 \subseteq R$. For this, consider

$$H(S_M^{L_1}, S_M^{L_2}) \quad (166)$$

$$\leq I(W_R; W_M) + H(S_M^{L_1}, S_M^{L_2}|W_R) \quad (167)$$

$$= I(W_R; W_M) + H(S_M^{L_1}|W_R) + H(S_M^{L_2}|W_R, S_M^{L_1}) \quad (168)$$
\[\leq e(\beta - \theta) + e\theta + e\theta = e(\beta + \theta),\] (169)

where the last inequality follows from Lemma \[14\] and \[165\]. Then, we have
\[H(S_M^{L_1} | S_M^{L_2}) = H(S_M^{L_1}, S_M^{L_2}) - H(S_M^{L_2})\] (170)
\[= H(S_M^{L_1}, S_M^{L_2}) - e\beta\] (171)
\[\leq e(\beta + \theta) - e\beta = e\theta,\] (172)

where the first equality follows from Lemma \[14\].

Finally, partitioning the nodes in \(R\) into sets \(R_1, R_2, \ldots, R_{r'}\) of size \(e\), it follows
\[H(S_M^{R_1}) \leq H(S_M^{R_1}) + \sum_{i=2}^{r'} H(S_M^{R_i} | S_M^{R_{i-1}}) \leq e\beta + e(r' - 1)\theta.\] (173)

\[\blacksquare\]

**B. Non-existence proof**

For interior points, \(1 \leq p \leq a - 2\). First, we consider the interior points that are multiple of \(\beta\). That is: \(e\alpha = (d - ep)\beta, \theta = 0\).

**Theorem 13.** Exact-repair codes do not exist for the interior points with \(\theta = 0\).

**Proof:** Consider a sub-network \(F\) consisting of \(d + e\) nodes. The parameters satisfy the condition in Lemma \[16\]. Note that by the regeneration property for any set of \(e\) nodes \(L \subseteq F\), \(H(W_L | S_F^{L \setminus L}) = 0\). Moreover, for distinct \(M, L_1, L_2 \subseteq F\), with \(\theta = 0\), we have \(H(S_M^{L_1} | S_M^{L_2}) = 0\). We partition the nodes in \(F\) into groups of size \(e\), denoted \(L_i, i = 1, 2, \ldots, d' + 1\). Then, we write
\[M \leq H(W_F) \leq H(S^{L_1}_{F \setminus L_1}, \ldots, S^{L_{d'+1}}_{F \setminus L_{d'+1}})\] (174)
\[= H(S^{L_1}_{F \setminus L_1}, \ldots, S^{L_{d'+1}}_{F \setminus L_{d'+1}})\] (175)
\[\leq \sum_{i=1}^{d'+1} H(S^{L_i}_{F \setminus L_i})\] (176)
\[\leq \sum_{i=1}^{d'+1} e\beta\] (177)
\[= (d + e)\beta,\] (178)

where the inequality (177) follows from Lemma \[16\]. On the other hand,
\[M = \sum_{i=0}^{d'-1} \min(e\alpha, (d - ie)\beta)\] (179)
\[= \sum_{i=0}^{d'-1} \min((d - ep)\beta, (d - ie)\beta)\] (180)
\[= 2(d - ep)\beta + \sum_{i=2}^{d'-1} \min((d - ep)\beta, (d - ie)\beta)\] (181)
\[\geq 2(d - ep)\beta + (a - 2)e\beta\] (182)
\[\geq 2e\beta + (d - ep)\beta + (a - 2)e\beta\] (183)
\[= (d - 2e)\beta + (a - 2e - ep)\beta\] (184)
\[\geq (d - 2e)\beta,\] (185)

where we assume \(1 \leq p \leq a - 2\) (non-MSMR point). Thus, \(ep + 2e \leq k \leq d\). Both bounds are contradictory, thus proving the impossibility result in the case of \(\theta = 0\). \[\blacksquare\]

**Theorem 14.** For any given values of \(M\), exact-repair regenerating codes do not exist for the parameters lying in the interior of the storage-bandwidth tradeoff when \(\theta \neq 0\), except possibly for the case \(p + 2 = a\) and \(\theta \geq \frac{d - ep - e}{d - ep} \cdot \beta\).

**Proof:** See Appendix \[C\] \[\blacksquare\]
VII. MULTI-NODE REPAIR FOR MINIMUM BANDWIDTH REGENERATING CODES

In Section V, we proved that MBMR codes are not achievable for linear exact repair codes, when \(2 < e < k\). When \(e = 1\), exact MBMR codes are MBR codes and their existence is well established in the literature \[25\]. Adaptive regenerating codes possess the extra feature that the number of helpers involved in the repair process can be adaptively selected, which provides the storage system with robustness to the network varying conditions \[1\], \[18\]. Adaptive MSR codes have been constructed in \[22\]. We now briefly describe a construction of adaptive MBR codes that simultaneously and efficiently repair single node failures, presented in \[22\]. Then, we show how to optimally repair multiple failures in this construction.

A. Adaptive MBR construction

The construction is based on product matrix codes \[22\], \[25\]. We first review the repair of single failure, and then present our scheme for multiple failures. Let \(\alpha = \prod_{d=d_{\text{min}}}^{d_{\text{max}}} d\). Define \(z = \frac{\alpha}{d_{\text{min}}}\) and construct the \((\alpha \times \alpha)\) data matrix \(M\) as

\[
M = \begin{bmatrix}
    M_1 & O & \cdots & O \\
    O & M_2 & \cdots & O \\
    \vdots & \vdots & \ddots & \vdots \\
    O & \cdots & O & M_z
\end{bmatrix},
\]

(190)

where \(O\) is a \((d_{\text{min}} \times d_{\text{min}})\) zero matrix and each of the submatrices \(M_i\) is filled with information symbols, and is symmetric and satisfies the structural properties of a product-matrix MBR code for parameters \(k\) and \(d_{\text{min}}\). For instance, \(M_i\) is given by

\[
M_i = \begin{bmatrix}
    N_i & L_i \\
    L_i^t & O
\end{bmatrix},
\]

(191)

where \(N_i\) is a symmetric \((k \times k)\) matrix, \(L_i\) is \((k \times (d_{\text{min}} - k))\) matrix, and \(O'\) is \((d_{\text{min}} - k) \times (d_{\text{min}} - k)\) zero matrix. Let \(\Psi\) be an \((zn \times d_{\text{min}})\) Vandermonde matrix, with rows denoted by \(\psi_{ij}\), for \(1 \leq j \leq zn\). Then, storage node \(l\) is associated with

\[
w_l^i = [\psi_{(l-1)z+1}^i, \ldots, \psi_{lz}^i] M = [\psi_{(l-1)z+1}^i M_1, \ldots, \psi_{lz}^i M_z].
\]

**Single node repair.** Denote the set of helpers by \(\mathcal{H}\) such that \(|\mathcal{H}| = d\) and \(d_{\text{min}} \leq d \leq d_{\text{max}}\). Let \(\Omega\) be an \((z \times z)\) matrix such
that \( \Omega^d \) is a Vandermonde matrix. Assume that node \( f \) fails. Let \( \Omega_d \) be an \((\frac{d}{\alpha} \times z)\) containing the first \( \frac{d}{\alpha} \) rows of \( \Omega \). Moreover, let \( \Phi_i \) be an \((\alpha \times z)\) matrix

\[
\Phi_i = \begin{bmatrix}
\psi_{(i-1)z+1} & \cdots & \psi_{iz}
\end{bmatrix}.
\]

Each helper node \( i_j \in \mathcal{H} \) transmits \( s_{ij,f} = w_{ij}^f \Phi_f \Omega_j^d \). After simplification, the replacement node obtains

\[
w_{ij}^f [\Phi_i, \Omega_i^d, \ldots, \Phi_i d^d] = w_{ij}^f \Theta_{H}
\]

Noting that \( \Theta_{H} \) is invertible \([22]\), the replacement node can thus recover \( w_{ij}^f \).

**B. Multi-node repair in adaptive MBR codes**

We state our result in the following theorem.

**Theorem 16.** Adaptive MBR regenerating codes with storage per node \( \alpha \) and arbitrary number of helpers \( d_{\min} \leq d \leq d_{\max} \), presented in \([22]\), can simultaneously and optimally repair \( e \) failures with \( d \) helpers, for all \( d_{\min} \leq d \leq d_{\max} \), \( 1 \leq e \leq k \), and \( e + d \leq n \).

**Proof:** Assume w.l.o.g. that the first \( e \) nodes failed and \( d \) helpers are used, where \( d_{\min} \leq d \leq d_{\max} \), \( 1 \leq e \leq k \), \( e + d \leq n \). Denote the helpers by the set \( \mathcal{H} = \{i_1, \ldots, i_d\} \). First, the repair of node 1 is done by contacting all the \( d \) helpers and downloading \( \frac{d}{\alpha} \) symbols from each one of them, using the procedure described for single node repair. Node 2 is then repaired using only \( d_{\min} \) helpers, comprising repaired node 1 and any other \( d_{\min} - 1 \) helpers in \( \mathcal{H} \), such that each helper provides \( \frac{\alpha}{d_{\min}} \) symbols. The same procedure is then applied repeatedly until recovering the last node \( e \) by contacting any \( d_{\min} - e + 1 \) helpers in \( \mathcal{H} \) and using contributions from the \( e - 1 \) already repaired nodes. The overall repair bandwidth is given by

\[
d\frac{\alpha}{d} + \sum_{i=1}^{e-1} z(d_{\min} - i) = e\alpha - \left(\begin{array}{c}e \\ d\end{array}\right) \frac{\alpha}{d_{\min}},
\]

which matches the bound in \([188]\), establishing the optimality of the repair procedure. \(\blacksquare\)

**Remark 13.** Repairing \( e \) failures in an \((n, k, d_{\min}, \alpha, \beta)\) MBR code separately requires a bandwidth of size \( e\alpha \). However, simultaneously repairing \( e \) failures using \( d \geq d_{\min} \) reduces the bandwidth by \( \left(\begin{array}{c}e \\ d\end{array}\right) \frac{\alpha}{d_{\min}} \).

**Remark 14.** The repair procedure of multiple erasures in Theorem 16 is asymmetric. However, one can always duplicate the code a sufficient number of times to achieve a symmetric repair strategy (e.g. \([11]\)).

**VIII. Conclusion**

We studied the problem of centralized repair of multiple erasures in distributed storage systems. We explicitly characterized the optimal functional tradeoff between the repair bandwidth and the storage size per node. For instance, we obtained the expressions of the extreme points on the tradeoff, namely the minimum storage multi-node repair (MSMR) and the minimum bandwidth multi-node repair (MBMR) points. In the case of \( e \geq k \), we showed that the tradeoff reduces to a single point, for which we have provided a code construction achieving it. We described a general framework for converting single erasure minimum storage regenerating codes to MSMR codes. Then we applied the framework to product-matrix codes and interference alignment codes. Furthermore, we proved that the functional MBMR point is not achievable for linear exact repair codes. Similarly, we have shown that the functional repair tradeoff is not achievable under exact repair, except for maybe a small portion near the MSMR point. Finally, we showed how to adaptively repair multiple failures for MBR codes, with varying number of helpers and varying number of failures.

Open problems include the generalization of the non-existence proof of linear exact-repair MBMR regenerating codes to non-linear codes. It is interesting to determine the exact-bound for MBMR regenerating codes under exact-repair. Moreover, characterization of storage-bandwidth tradeoff for exact repair for the interior points is still not known. Last but not least, reducing the subpacketization size for high rate exact MSMR codes is an important direction to study for its practical implications.

**Appendix**

**A. Proof of Lemma 2**

We first state the following lemma which will be useful in the proof.


Lemma 17. The scenario \( u = [e, \ldots, e, r] \) achieves the lowest final value of minimum cut:

\[
\lim_{\alpha \to +\infty} f(u) \geq \lim_{\alpha \to +\infty} f([e, \ldots, e, r]), \forall u \in \mathcal{P},
\]

where \( f(u) \) and \( \mathcal{P} \) are defined in (6) and (7), respectively.

Proof: for a specific cut \( u \), we have

\[
\lim_{\alpha \to +\infty} f(u) = \sum_{i=1}^{g} (d - \sum_{j=1}^{i-1} u_j) \beta
\]

\[
= d \beta g - \beta \sum_{i=1}^{g} \sum_{j=1}^{i-1} u_j = gd \beta - \beta \sum_{i=1}^{g-1} u_i (g - i)
\]

\[
= \beta (dg - \sum_{i=1}^{g-1} u_i + \sum_{i=1}^{g-1} i u_i) = \beta ((d - k)g + \sum_{i=1}^{g} i u_i).
\]

To obtain the smallest minimum cut value, we need to solve the following problem

\[
\begin{align*}
\text{minimize} & \quad (d - k)g + \sum_{i=1}^{g} i u_i \\
\text{subject to} & \quad 1 \leq u_i \leq e, \\
& \quad \sum_{i=1}^{g} u_i = k.
\end{align*}
\]

It can be seen that the solution to (194) is given by \( u = [e, \ldots, e, r] \).

We now study the different functions \( C_j(\alpha) \) for \( j = 0, \ldots, a \). An example of the different functions to be analyzed is given in Figure 5 with \( k = 9, d = 10, \beta = 1 \). It is observed that \( u = [1, 3, 3, 3] \) generates the lowest cut before some threshold \( \alpha^* = 5 \), after which the lowest cut is generated by \( u = [3, 3, 3, 1] \). In the following, by analyzing \( C_j(\alpha) \) for \( j = 0, \ldots, a \), we prove that the above observation holds true in general.

a) \( j=0 \): we have

\[
C_0(\alpha) = \min(r \alpha, d \beta) + \sum_{i=0}^{a-1} \min(e \alpha, (d - r - i e) \beta)
\]

\[
= r \min(\alpha, \frac{d \beta}{r}) + \sum_{i=0}^{a-1} \min(e, \frac{(d - r - i e) \beta}{e}).
\]
$C_0(\alpha)$ is a piecewise linear function with breakpoints given by \( \{ \frac{d-r-(a-1)e}{e} \beta, \frac{d-r-(a-2)e}{e} \beta, \ldots, \frac{d-r-1e}{e} \beta, \frac{d-r}{e} \beta \} \). $C_0$ increases from 0 at a slope of $k$. Its slope is then reduced by $e$ by the successive breakpoints and then finally by $r$ until it levels off.

b) $1 \leq j \leq a$: for each $j$, we have

$$C_j(\alpha) = \sum_{i=0}^{j-1} \min(e\alpha, (d-ie)\beta) + \min(r\alpha, (d-je)\beta) + \sum_{i=j}^{a-1} \min(e\alpha, (d-r-ie)\beta)$$

$$= \sum_{i=0}^{j-1} e\min(\alpha, \frac{(d-ie)\beta}{e}) + r\min(\alpha, \frac{(d-je)\beta}{r}) + \sum_{i=j}^{a-1} e\min(\alpha, \frac{(d-r-je)\beta}{e}).$$

$C_j(\alpha)$ is also piecewise-linear function with non-increasing successive slopes. Its breakpoints are given by

\( \{ \frac{d-r-(a-1)e}{e} \beta, \ldots, \frac{d-r-je}{e} \beta, \frac{d-(j-1)e}{e} \beta, \ldots, \frac{d}{e} \beta \} \cup \{ \frac{d-je}{r} \beta \} \).

The exact relative position of the breakpoint $\frac{d-je}{r} \beta$ with respect to the other breakpoints of $C_j(\alpha)$ depends on the system’s parameters. However, we give a lower bound on $\frac{d-je}{r} \beta$.

\[
\frac{d-je}{r} - \frac{d-r-(j-1)e}{e} = \frac{ed-rd-re+r^2-j(e^2-re)}{re} \\
\geq \frac{(e-r)d-re+r^2-a(e^2-re)}{re} \\
\geq \frac{(e-r)k-re+r^2-a(e^2-re)}{re} \\
= 0,
\]

where the first inequality follows by noticing that the expression is decreasing in $j$ and letting $j = a$, and the second inequality follows as the corresponding expression is increasing $d$.

Figure 6 illustrates the relative positions of all the breakpoints of $C_0(\alpha)$ and $C_j(\alpha), j \geq 1$, where for example $\frac{d-je}{r} \in \left[ \frac{d-r-(j-1)e}{e}, \frac{d-r-(j-2)e}{e} \right]$. We denote by

\[
C_j(\infty) = \lim_{\alpha \to +\infty} C_j(\alpha).
\]

**Lemma 18.** For $1 \leq j \leq a$, there exists a point $\alpha_c(j) \in \left[ \frac{d}{e}, \frac{d}{r} \right]$ such that

\[
C_0(\alpha_c(j)) = C_j(\alpha_c(j)), \\
C_0(\alpha) \leq C_j(\alpha) \quad \text{if} \quad \alpha \leq \alpha_c(j), \\
C_0(\alpha) \geq C_j(\alpha) \quad \text{if} \quad \alpha \geq \alpha_c(j), \\
C_j(\alpha) = C_j(\infty) \quad \text{if} \quad \alpha \geq \alpha_c(j).
\]

**Proof:** W.l.o.g, assume $\beta = 1$. First, we note that

\[
C_0(\alpha) = C_j(\alpha) = k\alpha \quad \text{for} \quad \alpha \leq \frac{d-r-(j-1)e}{e}.
\]

Next, we analyze the behavior of each of the functions $C_0(\alpha)$ and $C_j(\alpha)$ over the successive intervals $I_i \equiv \left( \frac{d-r-je}{e}, \frac{d-r-(i-1)e}{e} \right]$ for $i \in \{j-1, j-2, \ldots, 1\}$. Let $x_i = \frac{d-r-je}{e}$ and define $s_j(I_i)$ as the slope of $C_j(\alpha)$ just before $\alpha = x_i$. Consider a given interval $I_i = (x_i, x_{i-1}]$, we have
\[ C_0(\alpha) \] has no breakpoint inside \( I_i \). Thus, \( C_0(\alpha) \) increases by

\[ C_0(x_{i-1}) - C_0(x_i) = s_0(I_i) - e. \]

\( C_j(\alpha) \) has either one or two breakpoints inside \( I_i \).

1. In the case of \( C_j(\alpha) \) has a single breakpoint inside \( I_i \) (at \( \alpha = \frac{d - ie}{e} \)), \( C_j(\alpha) \) increases by

\[ C_j(x_{i-1}) - C_j(x_i) = s_j(I_i) \frac{r}{e} + (s_j(I_i) - e) \frac{e - r}{e} = s_j(I_i) - e + r. \]

2. In the case of \( C_j(\alpha) \) has two breakpoints inside \( I_i \), namely at \( \alpha = \frac{d - je}{r} \) and \( \alpha = \frac{d - ie}{e} \). Let \( \Delta = \frac{d - je}{r} - \frac{d - r - ie}{e} \) (c.f. Figure 5).

Assuming \( \frac{d - je}{r} \leq \frac{d - ie}{e} \), then, \( C_j(\alpha) \) increases by

\[ C_j(x_{i-1}) - C_j(x_i) = (s_j(I_i) - r)(1 - \frac{e - r}{e}) + \frac{e - r}{e}(s_j(I_i) - r - e) + s_j(I_i) \Delta = s_j(I_i) - e + \Delta r. \]

Assuming \( \frac{d - je}{r} \geq \frac{d - ie}{e} \), then, \( C_j(\alpha) \) increases by

\[ C_j(x_{i-1}) - C_j(x_i) = \frac{r}{e} s_j(I_i) + (k - e)(\Delta - \frac{r}{e}) + (s_j(I_i) - r - e)(1 - \Delta) = s_j(I_i) - e + \Delta r, \]

which shows that the increase does not depend on the relative position of the two breakpoints.

Now that we have computed the increase increment of each \( C_j \) over \( I_i \), we proceed to compare \( C_0(\alpha) \) and \( C_j(\alpha) \) for \( 1 \leq j \leq a \).

We discuss two cases:

*Case 1:* Assume \( \frac{d - je}{r} \in I_jb \) for some \( j_0 \in [1, j - 1] \), \( j_0 \) may not exist, which will be discussed in the second case. Based on the above discussion, it can be seen that

\[ C_j(\alpha) \geq C_0(\alpha), \quad \text{for } \alpha \leq x_{j_0}. \]

This can be seen by noticing that \( s_0(I_i) = s_j(I_i) \) and that

\[ (C_j(x_{i-1}) - C_j(x_i)) - (C_0(x_{i-1}) - C_0(x_i)) = r \geq 0, \quad \forall i < j_0. \]

Over \( I_{j_0} \), \( C_j \) also dominates \( C_0 \) at every point as \( s_0(I_{j_0}) = s_j(I_{j_0}) \) and

\[ (C_j(x_{i-1}) - C_j(x_i)) - (C_0(x_{i-1}) - C_0(x_i)) = \Delta r \geq 0. \]

For \( i > j_0 \), we have \( s_0(I_i) - s_j(I_i) = r \). Moreover, over each \( I_i, i > j_0 \), we have

\[ (C_j(x_{i-1}) - C_j(x_i)) - (C_0(x_{i-1}) - C_0(x_i)) = (s_j(I_i) - e + r) - (s_0(I_i) - e) = 0. \]

Combining the last equation and the observation that \( C_j(x_{j_0-1}) \geq C_j(x_{j_0-1}) \), it follows that \( C_j \) continue to dominate \( C_0 \) over the successive intervals \( I_i, i > j_0 \). So far, we have shown that

\[ C_j(\alpha) \geq C_0(\alpha), \quad \text{for } \alpha \leq \frac{d - r}{e}. \]

For \( \alpha \geq \frac{d - r}{e} \), we observe that \( C_j \) increases with a slope of \( e \) and levels off at \( \frac{d}{e} \) while \( C_0 \) increases at smaller slope given by \( r \) and levels off at \( \frac{d}{e} \). Moreover, we know from Lemma 17 that \( C_0 \) levels off at a higher value than that of \( C_j \). Thus, there exists \( \alpha_{e}(j) \in \left[ \frac{d}{e}, \frac{d}{e} \right] \) that satisfies (195).

*Case 2:* Assume \( \frac{d - je}{r} < \frac{d - ie}{e} \leq \frac{d}{e} \), then, using similar arguments as in the first case, it follows that for \( \alpha \leq \frac{d - r}{e} \), \( C_j(\alpha) \geq C_0(\alpha) \). At \( \alpha = \frac{d - r}{e} \), \( C_j(\alpha) \) has a slope of \( r + e \), which is higher than that of \( C_0 \), given by \( r \). Thus, the slope of \( C_j \) remains higher than that of \( C_0 \) until \( C_j \) levels off. Combining these observations with the fact that \( C_0 \) levels off at a higher value, it follows that both curves will intersect only once. Moreover, the intersection at a point at which \( C_j \) has leveled off i.e., we have \( \alpha_{r,j}(\alpha) \geq \max \left( \frac{d}{e}, \frac{d - je}{e} \right) \). Therefore, (195) holds also in this case.

Using Lemma 18 and the fact that \( C_{a,i} \) achieves the smallest final value from Lemma 17 that is \( C_{a,i}(\infty) \leq C_{j,i}(\infty), j \in [0, a - 1] \), it follows that (15) holds for any \( j \in [0, a] \). Moreover, as \( \alpha_{e,a}(a) \in \left[ \frac{d}{e}, \frac{d}{e} \right], \alpha_{e,a}(a) \) satisfies

\[ r\alpha_{e,a}(a) + \sum_{i=0}^{a-1} (d - r - ie)\beta = (a + 1)\beta d - e\beta \frac{a^2 + a}{2}, \]

which implies that

\[ r\alpha_{e,a}(a) + a(d - r - \frac{ea}{2} + \frac{e}{2})\beta = (a + 1)\beta d - e\beta \frac{a^2 + a}{2}. \]

Simplifying the last equation yields (16).
B. Storage-bandwidth tradeoff expression

We start with the case $k = ae + r$. The optimization trade-off is

$$\begin{align*}
\text{minimize} & \quad \alpha \\
\text{subject to} & \quad C(\alpha) \geq M.
\end{align*}$$

(196)

The constraint is a piece-wise linear function $C(\alpha)$ is given by

$$C(\alpha) = \begin{cases}
\frac{(a + 1)\beta d - e\beta a(a + 1)/2}{r\alpha + \sum_{j=0}^{a-1} b_j}, & \alpha \geq \alpha_c, \\
\frac{r\alpha + \sum_{j=0}^{a-1} b_j}{\alpha \in [\frac{b_a}{e}, \alpha_c]}, \\
\frac{(r + i\epsilon)\alpha + \sum_{j=1}^{a-1} b_j}{\alpha \in [\frac{b_a}{e}, \frac{b_{a-1}}{e}],} \\
k\alpha, & \text{for } i = 1, \ldots, a - 1,
\end{cases}$$

(197)

with $\alpha_c = \frac{d + ar - ae}{r} \beta$, $b_i = (d - r - i\epsilon)\beta$ and

$$\sum_{j=1}^{a-1} b_j = \beta(a - i)(d - r - \frac{e(a - 1 + i)}{2}) = \gamma (a - i)(-2r + e + 2d - ae - ei) \triangleq g_r(i),$$

such that

$$g_r(i) = \frac{(a - i)(-2r + e + 2d - ae - ei)}{2d}.$$

The expression $C(\alpha)$ increases from 0 to a maximum value given by $\beta((a + 1)d - (\frac{a + 1}{2})^2)$. To solve (196), we let $\alpha^* = C^{-1}(M)$ under the condition $M \leq \beta((a + 1)d - (\frac{a + 1}{2})^2))$. Therefore, we obtain,

$$\alpha^* = \begin{cases}
\frac{M}{k}, & M \in [0, \frac{kb_a}{e - 1}]
\frac{M - \sum_{j=1}^{a-1} b_j}{r + i\epsilon}, & M \in [(r + i\epsilon)\frac{b_a}{e} + \sum_{j=1}^{a-1} b_j, (r + i\epsilon)b_{a-1}/e + \sum_{j=1}^{a-1} b_j] \\
\frac{M - \sum_{j=0}^{a-1} b_j}{r}, & M \in [\frac{b_a}{e} + \sum_{j=0}^{a-1} b_j, r\alpha_c + \sum_{j=0}^{a-1} b_j],
\end{cases}$$

(198)

with

$$\frac{rb_i}{e} + ib_i + \sum_{j=1}^{a-1} b_j = \frac{-a^2 e^2 + a e^2 - 2ae r + 2de - e^2 i^2 - e^2 i - 2ei r - 2r^2 + 2dr}{2de} \gamma$$

$$= \frac{-k^2 - r^2 + e(k - r) + 2kd - e^2(i^2 + i) - 2ie r}{2ed} \gamma$$

$$\triangleq \frac{M}{f_r(i)},$$

such that

$$f_r(i) = \frac{2edM}{-k^2 - r^2 + e(k - r) + 2kd - e^2(i^2 + i) - 2ie r}.$$ 

Therefore, fixing $M$ and varying $\gamma$, we write

$$\alpha^* = \begin{cases}
\frac{M}{k}, & M \in [0, \frac{kb_a}{e - 1}]
\frac{M - g_r(a-1)\gamma}{r + i\epsilon}, & M \in [(r + i\epsilon)\frac{b_a}{e} + \frac{M}{f_r(i)}, (r + i\epsilon)b_{a-1}/e + \frac{M}{f_r(i)}], \\
\frac{M - g_r(0)\gamma}{r}, & M \in [\frac{b_a}{e} + \frac{M}{f_r(0)}, (g_r(0) + \frac{d + ar - ae}{d})\gamma].
\end{cases}$$ 

(199)

As a function of $\gamma$, after simplifications, we obtain the expression of $\alpha^*$ as in Theorem 3. We note that there are a piece-wise
linear portions on the curve. Moreover, the minimum bandwidth point \( \gamma_{MBMR} \) is given by

\[
\gamma_{MBMR} = \frac{\mathcal{M}}{g_r(0) + \frac{d + ar - ea}{d}} = \frac{d \mathcal{M}}{d(a + 1) - e\left(\frac{a + 1}{2}\right)}.
\]

The expression of \( \alpha_{MBMR} \) is given by

\[
\alpha_{MBMR} = \frac{\mathcal{M} - \gamma_{MBMR} g_r(0)}{r} = \gamma_{MBMR} \frac{d + ar - ea}{rd}.
\]

in the case of \( e \mid k \), we have \( r = 0 \). The expression of the tradeoff is obtained from (199) by setting \( r = 0 \) and eliminating the last line. We note that in this case, there are \( a - 1 \) piece-wise linear portions on the trade-off curve.

C. Proof of Theorem 14

Proof: Take a subnetwork \( F \) of \( d + e \) nodes. Let \( L, M \subseteq F \) be two disjoint groups of \( e \) nodes. Partition the \( d - e \) remaining nodes into two sets, \( A \) of cardinality \( ep \) and \( B \) of cardinality \( d - ep - e \). Exact repair requires

\[
H(W_L|S^L_A, S^L_B, S^L_M) = 0, \quad H(W_M|S^M_A, S^M_B, S^M_L) = 0.
\]

(200)

It follows that

\[
H(W_L, W_M|W_A, S^L_B, S^L_B, S^L_M) = H(W_L|W_A, S^L_B, S^L_B, S^L_M) + H(W_M|W_L, W_A, S^L_B, S^L_B, S^L_M).
\]

(202)

Therefore, we have

\[
(S^L_B, S^M_B, S^L_M) \geq H(W_L, W_M|W_A)
\]

(204)

\[
= H(W_L|W_A) + H(W_M|W_A, W_L)
\]

(205)

\[
= H(W_L) - I(W_L; W_A) + H(W_M) - I(W_M; W_A, W_L)
\]

(206)

\[
= e\alpha - 0 + e\alpha - e(\beta - \theta)
\]

(207)

\[
= 2e\alpha - e\beta + e\theta
\]

(208)

\[
= 2((d - ep)\beta) - e\theta - e\beta + e\theta
\]

(209)

\[
= (2d - 2ep - e)\beta - e\theta.
\]

(210)

Here (207) follows from Lemma 13. We note that the lower bound does not depend on whether \( d \) is a multiple of \( e \). Next, we obtain an upper bound on the same quantity.

Partition \( B \) into sets of size \( e \), denoted by \( L_i \). We will use \( R = L \cup M, r' = 2 \), in the helper node pooling.

Case: \( p + 2 < a \): In this case, the parameters satisfy the condition in Lemma 15

\[
H(S^L_B, S^M_B, S^L_M) \leq \sum_{L_i \in B} H(S^L_{L_i}, S^L_{L_i}) + H(S^M_{M_i})
\]

(211)

\[
\leq \sum_{L_i \in B} e(\beta + \theta) + e\beta
\]

(212)

\[
= (d - ep)\beta + (d - ep - e)\theta.
\]

(213)

where the inequality (212) is obtained using Lemma 14 and Lemma 13. Equations (210) and (214) are in contradiction if \( d - ep - e > e \iff d > e(p + 2) \), which is true as \( d \geq k = ac > (p + 2)e \).

Case: \( p + 2 = a \): In this case, Lemma 16 is used to derive an upper bound on \( H(S^L_B, S^M_B, S^L_M) \). Lemma 16 does not hold if \( a = 2 \). It holds for \( a > 2 \iff k > 2e \). Thus, we consider \( k > 2e \). We have

\[
H(S^L_B, S^M_B, S^L_M) \leq \sum_{L_i \in B} H(S^L_{L_i}, S^L_{L_i}) + H(S^L_{M_i})
\]

(215)

\[
\leq \sum_{L_i \in B} e(\beta + \theta) + e\beta
\]

(216)

\[
= (d - ep)\beta + (d - ep - e)\theta.
\]

(217)

Equations (210) and (217) are in contradiction when

\[
\theta < \frac{d - ep - e}{d - ep} \beta.
\]

(218)
[43] M. Zorgui and Z. Wang, “Centralized multi-node repair in distributed storage,” in 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Sept 2016, pp. 617–624.

[44] ———, “Centralized multi-node repair for minimum storage regenerating codes,” in IEEE International Symposium on Information Theory, 2017. [Online]. Available: http://faculty.sites.uci.edu/zhiying/publications/