ABSTRACT

Title of Dissertation: CODES FOR DATA RETRIEVAL AND NODE REPAIR IN GRAPHICAL NETWORKS

Adway Patra
Doctor of Philosophy, 2025

Dissertation Directed by: Professor Alexander Barg
Department of Electrical & Computer Engineering
Institute for Systems Research

This dissertation investigates the problem of efficient data recovery in distributed storage systems relying on erasure codes, when the connections between individual nodes of the system are constrained by a connected graph. In this model, when a node fails, i.e., the data stored in it becomes lost or unavailable, the other surviving nodes in the network send information, which is a function of their local stored data, along the edges of the graph to repair the failed node. We show that savings in communication complexity can be attained if the intermediate vertices along the path process the information rather than simply relay it toward the failed node.

We derive information-theoretic bounds on the amount of information communicated between the nodes in the course of the repair. Moreover, we show that the lower bound on the information exchange is achievable by modifying codes from the class of Minimum Storage Regenerating (MSR) codes to perform intermediate processing. Our analysis extends to both deterministic connected graphs and random graphs, where we derive conditions on the system parameters that support recovery of the failed node with complexity lower than relaying.

In the second part of the thesis, we extend our study to general regenerating codes. We derive a

lower bound on the repair bandwidth and formulate repair procedures with intermediate processing for several algebraic families of regenerating codes. We also address the problem of data retrieval in the communication-constrained setting, deriving lower bounds and optimal protocols.

In the final part, we consider regenerating codes with nonuniform contribution for node repair on graphs. We begin with deriving information-theoretic lower bounds for communication complexity of repair and propose code constructions and repair schemes that attain these bounds. As the main conclusion, we show that a combination of nonuniform contributions and intermediate processing can further reduce the communication complexity. Additionally, for repair on graphs in the presence of adversarial nodes that can introduce errors during repair, we construct codes that support simultaneous intermediate processing and error correction.

Codes for Data Retrieval and Node Repair in Graphical Networks

by

Adway Patra

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Alexander Barg, Chair/Advisor
Professor Prakash Narayan
Professor Behtash Babadi
Professor Sanghamitra Dutta
Professor Mohammad Taghi Hajiaghayi, Dean's Representative

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I'd like to thank my advisor, Professor Alexander Barg for giving me an invaluable opportunity to work on challenging and extremely interesting projects over the past years. He has always made himself available for help and advice and there has never been an occasion when I've knocked on his door and he hasn't given me time. It has been a pleasure to work with and learn from such an extraordinary individual.

My colleagues and fellow graduate students at the Dept. of Electrical and Computer Engineering have enriched my graduate life in many ways and deserve a special mention. I want to thank the outstanding faculty at the University of Maryland who played an integral part in my education. Additionally, I would also like to acknowledge help and support from some of the staff members.

I owe my deepest thanks to my family - my mother, father and sister who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times. Words cannot express the gratitude I owe them. My friends, at my community of residence, who have been the family away from home, along with my friends back in India, have been a source of constant support.

It is impossible to remember all, and I apologize to those I've inadvertently left out.

Table of Contents

List of Figures

List of Abbreviations

DSS	Distributed Storage System
RC	Regenerating Code
GRC	Generalized Regenerating Code
MSR	Minimum Storage Regenerating (codes)
MBR	Minimum Bandwidth Regenerating (codes)
MDS	Maximum Distance Separable
AF	Accumulate and Forward
IP	Intermediate Processing
PM	Product Matrix (codes)

Chapter 1: Introduction

1.1 Motivation

In today's era of social media, cloud computing, and the Internet of Things, distributed storage systems (DSS) form the critical backbone of global web infrastructure. The performance and reliability of these services hinge on the robustness of their underlying distributed architecture. Modern data centers now manage multiple exabytes of data, incurring substantial costs—not just in hardware, software, and maintenance—but also in energy and water consumption. As such, it is vital to design systems that can withstand local faults or partial failures without causing major service disruptions or degrading user experience. This necessitates the design of resilient redundancy mechanisms at the system level, maintained over extended periods. However, the traditional method of ensuring reliability by storing multiple replicas of data across various nodes is resource-intensive and inefficient.

Erasure codes are a popular alternative method of introducing redundancy in the storage system reducing the massive overhead of replication [31,66,84]. Storing data using an $[n, k]$ erasure code involves fragmenting the data in k parts and encoding into n parts so that the system can tolerate up to $n-k$ failures. However, due to node failures, fragments must be periodically replaced as nodes fail, and a key question is how to regenerate the lost fragments in a distributed way while transferring as little data as possible across the network. This motivated the construction of a new family of codes, known collectively as *regenerating codes*, proposed in [16] and studied in several works thereafter. The authors of [16] derived a lower bound on the minimum amount of information acquired from the surviving nodes for the purposes of repair.

However, a crucial underlying assumption of their analysis was that every node in the network can directly communicate to every other node, i.e., full connectivity is assumed.

In this thesis, we address communication complexity of node repair under a more realistic communication constrained setting. This group of problems is motivated by the assumption that the links between the nodes are established based on physical proximity and the associated energy constraints, limitations of the system architecture, or other features with the same effect. In the network environment such as low-power wide-area networks (LP-WAN), e.g., path loss in narrow-band lower-power IoT, the mentioned limitations arise naturally as a part of the functioning of the system. In cloud storage platforms, such limitations might arise due to geographical positioning, architectural designs, and varying operational costs across storage clusters. Hence, from the systems perspective, it is a reasonable assumption to make that in the event of a node failure, the replacement node might not have direct access to all of the other nodes selected to participate in the repair process. Moreover, in a dynamic network, where nodes or links may be added or removed from the network, the connectivity constraints may change over time. Note that we can restrict ourselves to point-to-point rather than broadcast communication. For distributed storage systems this is a natural restriction, while for IoT applications this assumption may be imposed because of energy or privacy considerations. This naturally leads to modeling the distributed system by an arbitrary, undirected and connected graph whose vertices function as individual data centers and whose edges serve as the allowed bidirectional communication links between these centers. It is this model that we adopt in this thesis, with the goals of characterizing how the structure of the graph affects the communication complexity of repair, and we explore several distinct ways of reducing this complexity.

The organization of the thesis follows a sequence of increasing generality. In Chapter 2, we introduce the system model and study the communication complexity of repair for a family of regenerating codes called Minimum Storage Regenerating (MSR) codes, designed to minimize storage overhead while also optimizing the repair bandwidth. In Chapter 3, we lift this restriction, generalizing to any class of

regenerating codes, focusing on specific non-MSR families of interest. In Chapter 4, we generalize our model even further, allowing more flexibility on the code parameters to study if and when further reduction in communication complexity of repair is possible. Finally, we discuss some related open questions in Chapter 5.

1.2 Preliminaries: Regenerating Codes

Let $\mathbb{F}$ be a finite field. Consider a file storage system with n nodes, where a file of size M symbols over $\mathbb{F}$ is encoded into nl symbols, viewed as an $l \times n$ matrix over $\mathbb{F}$, and each column of l symbols stored in a separate node. The encoding should be such that the data collector, accessing any $k \leq n$ nodes and downloading their contents, should be able to retrieve the original file of size M. A node failure corresponds to the loss of l symbols stored on a node. Note that it is always possible to recover the lost data by contacting k nodes, downloading their contents to recover the original file of size M, and re-encoding it to find the lost symbols. However, this method incurs significant download bandwidth of kl symbols. The framework of regenerating codes provides an alternative to this, allowing one to contact d nodes, with $k \leq d \leq n-1$, and downloading β symbols from each, with $\beta \leq l$, and recovering from the failure. The key observation here is that the download bandwidth $d\beta$ in this case can be made much smaller than kl, making the repair much more bandwidth efficient, see Fig. 1.1 for a visual depiction.

Formally, a regenerating code $\mathcal{C}$ is a *vector code* of length n, i.e., an $\mathbb{F}$-linear subspace of $(\mathbb{F}^l)^n$ whose codewords can be thought of as $l \times n$ matrices. In the context of storage codes, elements of $\mathcal{C}$ are often referred to as n-words whose coordinates are l-vectors over $\mathbb{F}$, and each vector is stored in a different node. The information contents of the codeword is M symbols of $\mathbb{F}$, implying that $|\mathcal{C}| = q^M$. We further assume that any k coordinates suffice to recover the entire encoded information block. A node failure corresponds to an erasure of the code coordinate that is stored on it. The repair property requires that in the event of a failure, the contents of the replacement node be a function of any d other coordinates of the

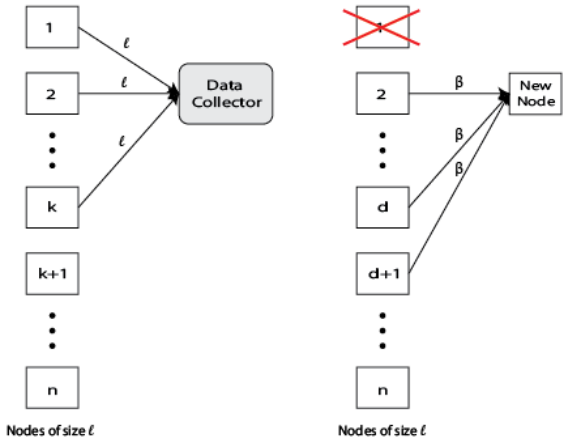


Figure 1.1: Data collection and node repair of an $[n, k, d, l, \beta, M]$ regenerating code

codeword, each of which provides β symbols of $\mathbb{F}$.

Definition 1.2.1. *An $[n, k, d, l, \beta, M]$ regenerating code (RC) over a finite field $\mathbb{F}$ is a subspace of $(\mathbb{F}^l)^n$ and has the following properties:*

- *Accessing any k out of n coordinates suffices to recover the original stored file of size M.*
- *If a coordinate of the code is erased, its contents can be "repaired" by accessing d other surviving coordinates and downloading β symbols from each.*

Two notions of repair have been identified in the literature:

1. If the contents of the newly repaired node are the same that were erased due to the failure, it is called *exact repair*.
2. If the contents of the newly repaired node are not necessarily the same as the ones of the erased node but still the two properties of regenerating code listed above continue to hold, then it is called functional repair.

Clearly, exact repair is a special case of functional repair. However, since for functional repair, the contents of the replacement node may be different than that of the failed node, every time a failure happens, the data

retrieval and repair algorithms have to be adjusted to this change. This adds extra complexity to the system maintenance. For this reason, exact repair is preferred for practical purposes as it simplifies management of the storage system. In this dissertation, we primarily focus on the more stringent notion of exact repair, adding remarks whenever the results are extendable to functional repair.

The relationship between the parameters for functional repair was investigated in [16], which established the following inequality:

$$M \leq \sum_{i=1}^{k} \min\{l, (d-i+1)\beta\}. \tag{1.1}$$

The parameter l determines the storage overhead ($\frac{nl}{M}$), whereas the parameter β is an indicator of the normalized repair bandwidth ($\frac{d\beta}{M}$). For a fixed M, k and d, different pairs of (l, β) satisfying Eq. (1.1) with equality give rise to a discrete collection of *optimal* points. Connecting these points via straight lines, which physically corresponds to a space-sharing solution, gives rise to the storage-bandwidth *trade-off* curve. The two corner points of the curve are the following.

- The Minimum Storage Regenerating (MSR) point: This point is found by first minimizing the per node storage parameter l and then finding the minimum value of the per node download parameter β. At this point, l is chosen to be the least possible value according to Eq. (1.1), resulting in $l \leq (d-k+1)\beta$. Setting the above Eq. (1.1) to equality, we get that $l = \frac{M}{k}$. Given this value of l, the minimum value of β is found to be $\beta = \frac{l}{d-k+1}$. Hence, the MSR point is characterized by the pair

  $$\Big(l = \frac{M}{k}, \beta = \frac{M}{k(d-k+1)}\Big).$$

  It also follows that MSR codes achieve the Singleton bound on the code size ($|\mathcal{C}| = (\mathbb{F}^l)^k$) and hence are Maximum Distance Separable (MDS) codes.

- The Minimum Bandwidth Regenerating (MBR) point: The other corner point is found by first minimizing the per-node download parameter β and then finding the minimum value of the storage

parameter l. The smallest value of β for which Eq. (1.1) can hold satisfies $d\beta \leq l$. Setting to equality, we get $M = \sum_{i=1}^{k}(d-i+1)\beta = (kd - \binom{k}{2})\beta$. For this value of β, the minimum value of l is clearly $l = d\beta$ and so the MBR point is characterized by the pair

$$\Big(l = \frac{Md}{kd - \binom{k}{2}}, \beta = \frac{M}{kd - \binom{k}{2}}\Big).$$

The pairs satisfying $d\beta > l > (d-k+1)\beta$ are called *interior points*. It is clear that exact repair is a more narrow requirement than functional repair, so the trade-off curve for exact repair will always be above that of functional repair. It was shown that at the two extremes, the two curves coincide and explicit constructions are known for them. For interior points, there exists a gap between the two curves. For a more detailed overview, refer to the survey in [58].

1.3 Related Work

As noted above, the concept of RCs was first proposed in [16], whose authors described the trade-off between storage vs bandwidth for this class of codes and defined the two extreme points: MBR and MSR. Subsequent papers [10, 59, 60, 69, 70, 76, 86] suggested several constructions, some of which are optimal with respect to the bound (1.1). Several variants such as Cooperative Repair for multiple failures [71, 87], Secure Repair for protection against eavesdroppers [61], Rack-aware repair with different local and global transmission costs [12, 33], error correction during repair to tackle active adversarial scenarios [73] etc. have been explored. Such constructions have been mostly restricted to the MSR point while some extensions have been proposed for the MBR point as well.

It was first shown in [78] that there is a gap between the achievability of functional and exact repair for interior points of the trade-off curve (1.1). This observation was later extended in [49] and [65]. Some popular code constructions for interior point codes include Determinant Codes [23], Cascade Codes [22],

Codes using multi-linear algebra [19] (also known as Moulin Codes). We again refer the reader to the detailed and very readable overview in [58].

Prior to this work, repair on graphs using MSR codes was considered in [28, 44] for particular network topologies, such as tandem networks, grids, and unidirectional rings. A somewhat similar setting arises when it is assumed that transmitting the data from a subset of nodes incurs a larger cost than for the remaining nodes [2, 74], resulting in the consideration of two different values of β for two class of nodes (*nonuniform download*). Another related setting was considered in [41] where the authors assumed that the links between the nodes (in a fully connected graph) are assigned weights that translate into the cost of sending symbols over them [41], resulting in a solution with different values of β for different helper nodes. Our assumptions and results are more general in the sense that these papers either relied only on relaying of repair data and did not afford the option of incorporating data processing at the intermediate nodes or when data processing was included, it was heavily dependent on the specific network topology. Another difference arises because the heterogeneity in the network in these works is fixed irrespective of the location of the failed nodes. At the same time, our setting implies that the cost of transmission from the node may be high or low depending on whether it is far from the failed node or is among its immediate neighbors. Arguably, this accounts for a more uniform treatment of the nodes in the network.

Another related communication problem is that of network coding [91] wherein (in its simplest version) the data is transmitted from a single fixed source to multiple destinations, and where it is assumed that the intermediate nodes combine the chunks of data on their incoming edges. While intermediate processing is a shared feature between node repair on graphs and network codes, they address different objectives. Specifically, while in network coding the message of the source needs to be reproduced at the receiver, the repair task is to compute a function of the cumulative contents of the helpers at the failed node. Nevertheless, some tools from network codes prove useful in the problem of node repair with errors.

To conclude this introduction, we note that the general problem of information processing or re-

covery under communication constraints represented by a graph has recently been studied in a number of specific settings. Among them, locally recoverable codes on graphs [47, 48] (and the associated problems of guessing games on graphs [27] and index coding [5]), their extension to recoverable systems [21], private information retrieval on graphs [63], and others. The problem of node repair under communication constraints introduced here is another instantiation of this broadly defined theme.

1.4 Summary of Contributions

The central topic of this thesis is node repair in graphical storage networks based on efficient encoding of the stored information using specially designed families of erasure-correcting codes. The main novel idea that the research summarized in the thesis contributes is related to communication savings in the repair process related to intermediate processing of the information transmitted over the network for the purpose of recovering the contents of the failed node. The general idea of information processing in networks is by no means new: it suffices to recall the relay channel [15,32,79], various decode-and-forward schemes [11, 34, 37], or even network coding [1, 25, 40] (on which we commented earlier), however, the particular kind of processing that we introduce and develop, yields both new bounds on the communication complexity and new procedures of designing and using erasure codes in graphical networks.

A related idea, that we also introduce here, is balancing the amount of information sourced from the helper nodes based on how far they are removed from the failed node. To an extent, we advance the understanding of code design for optimal communication in this setting, which depends on whether the system can adjust the number of helper nodes (repair degree) used for repair from one round to another. We also identify code constructions that should be used to support node recovery for systems with flexible and fixed repair degree.

The thesis contains 3 groups of main results, presented in 3 chapters, which we briefly describe in this section. A more detailed discussion of the results is postponed to the respective chapters.

1.4.1 Repair of MSR Codes on Connected Graphs

We develop a comprehensive graph-theoretic framework for node repair under communication constraints, focusing on Minimum Storage Regenerating (MSR) codes. By establishing general information-theoretic bounds, we quantify the minimum amount of information that the helper nodes must contribute to repair single and multiple failures, and derive corresponding lower bounds on communication complexity for any connected graph. We demonstrate that linear MSR codes support a repair procedure based on Intermediate Processing (IP) that achieves these bounds and illustrate the protocol with two popular code families: the product-matrix codes of [59] and diagonal-matrix codes of [88]. Building on the Cooperative Repair model, we incorporate intermediate processing to effectively handle multiple node failures. Further, we explore a setting inspired by Rack-Aware storage [12], where helper nodes are allowed to communicate among themselves prior to repair. We derive lower bounds on overall communication complexity using a linear programming formulation and also provide matching repair schemes for certain scenarios. Finally, for repair on random graphs, we determine a range of parameters for the standard Erdös-Rényi ensemble under which repair procedures supporting IP are advantageous over those based on simple relaying.

1.4.2 Extension to Codes beyond the MSR point

Exploring the diverse landscape of regenerating codes beyond the MSR point, we generalize the fundamental lower bound on repair bandwidth for arbitrary code families at non-MSR operating points. We then move on to develop explicit repair procedures for a broad class of Evaluation Codes—codes defined via algebraic evaluations of linear functionals—which includes product-matrix MSR codes, now reinterpreted under a unifying framework. This algebraic perspective enables us to extend the intermediate processing technique to more general code families, including those introduced in [19], which provide exact repair evaluation codes at the interior points of the tradeoff curve with the best known parameters. While many regenerating codes conform to the evaluation structure, we also introduce repair protocols for a family of

interior-point codes that fall outside this model. Additionally, we address the challenge of data retrieval under graph constraints, demonstrating how data processing at intermediate nodes can be adapted to improve retrieval efficiency, supported by both lower bounds and matching protocols.

1.4.3 Repair on Graphs using Generalized Regenerating Codes

Until this point, we focused on regenerating codes where each helper nodes contribute uniformly towards the repair of the failed node. Motivated by our graph-constrained environments where proximity to the failed node influences repair efficiency, we consider a new class of regenerating codes designed to accommodate nonuniform contributions from helper nodes. By allowing closer nodes to contribute more and distant nodes less, we provide a simple proof for a generalized version of the cut-set bound that accounts for nonuniform participation. We propose a simple stacking technique that yields optimal code constructions at the MSR point, capable of supporting nonuniform helper contributions. We then extend the intermediate processing bound to cover non-uniform settings, demonstrating that our construction satisfies the generalized bound at the MSR point, confirming its optimality from the IP perspective. Through detailed comparisons with earlier bounds, we identify specific parameter regimes—spanning both deterministic graphs and random graph ensembles—where non-uniform contributions lead to tangible reductions in communication complexity. Lastly, we explore the adversarial setting where some helper nodes may provide erroneous data. By leveraging techniques from network coding, we establish a robust variations of the IP bound and propose constructions that preserve the communication advantages of IP while mitigating the impact of malicious behavior.

The results of this thesis appear in [52, 53, 56], and [50, 51, 54, 55].

Chapter 2: Repair of MSR Codes on Connected Graphs

2.1 Introduction

In this chapter we focus our attention on the repair of erasures under the communication-constrained setting when the data is encoded using *Minimum Storage Regenerating* (MSR) codes, which are arguably the most interesting and well-studied subclass of Regenerating Codes. We assume that communication between the nodes is constrained by a (connected) graph $G(V, E)$, where V is an n-set of vertices and the cost of sending a unit of information from v_i to v_j is determined by the graph distance $\rho(v_i, v_j)$ in G. Under a naive approach to this problem, it is still possible to use the known methods of node repair whereby the chosen group of the helper nodes communicates some functions of their contents to the failed node. Note however that the data from the helper nodes not directly connected to the failed node will have to be relayed along some path to the failed node, increasing the bandwidth utilized for the repair. Thus, a natural question to study is whether there are more economical ways of accomplishing this goal given the structure of the graph G, under which the data from the far-off helper nodes is processed along the way and combined with the contents of the intermediate nodes, saving on the overall communication and it is this question that we answer in this chapter.

2.1.1 Review of MSR codes

Let $\mathbb{F}$ be a finite field. A *vector code* $\mathcal{C}$ of length n is an $\mathbb{F}$-linear subspace of $(\mathbb{F}^l)^n$ whose codewords can be thought of as $l \times n$ matrices. In the context of storage codes, elements of $\mathcal{C}$ are often referred to as

n-words whose coordinates are l-vectors over $\mathbb{F}$. We further assume that the information contents of the codeword is kl symbols of $\mathbb{F}$, in other words, that $|\mathcal{C}| = q^{kl}$, and that any k coordinates suffice to recover the entire codeword. Thus, the code has the MDS property, and any $n-k$ erased coordinates can be found from the remaining k ones, accounting for the optimal erasure correction capacity.

Suppose that the coordinates of the codeword are placed on n different storage nodes, and refer to the coordinates themselves as nodes. The defining property of MSR codes is related to recovering the value of an erased coordinate of the codeword, or repairing a single failed node. According to the above description, we can accomplish this by using the information from k functional nodes and downloading a total of kl symbols of the field $\mathbb{F}$. At the same time, this operation supports recovery of the entire codeword, accomplishing more than we actually need. An important finding of the work [16] was to point out that we can save on the amount of downloaded information by performing the repair based on the contents of $d > k$ helper nodes. To achieve the saving, each of the helper nodes provides a function of its contents, and [16] showed that to accomplish the repair it is necessary to download at least $\frac{dl}{d-k+1}$ field symbols. This is smaller than kl for all $k < d \leq n-1$. A code $\mathcal{C}$ with the parameters (n, k, d, l) is called MSR if it supports node recovery with *repair bandwidth* meeting the lower bound for the chosen number d of *helper nodes*. It is easy to show that for a code to have this property, each of the helper nodes necessarily provides $l/(d-k+1)$ field symbols for the recovery of the failed node (the so-called *uniform download* property).

Formally, an (n, k, d, l) linear MDS vector code $\mathcal{C}$ over $\mathbb{F}$ is called MSR if there are linear functions $h_i : \mathbb{F}^l \to \mathbb{F}^{l/(d-k+1)}, i = 1, \ldots, n$ such that for any $j \in [n]$ and any subset $\{i_1, \ldots, i_d\} \subset [n] \backslash \{j\}$ there exists a linear function $g_j : \mathbb{F}^{d(l/(d-k+1))} \to \mathbb{F}^l$ such that for any codeword $C = (C_1, \ldots, C_n) \in \mathcal{C}$ the value C_j (the contents of the failed node) is found as

$$C_j = g_j(h_{i_1}(C_{i_1}), \ldots, h_{i_d}(C_{i_d})).$$

Slightly more generally, the functions h_i could also depend on j, but this will not be important below. A number of families of MSR codes are known in the literature, among them constructions of [29, 59, 62, 75, 86, 88], see also a recent survey in [58]. In this thesis, we use two such families to exemplify our approach to node repair on graphs, namely product matrix codes [59] and diagonal-matrix codes [88]. It will become apparent toward the end of Sec. 2.3 that any family of $\mathbb{F}$-linear MSR codes can be incorporated in our repair scheme.

2.1.2 Overview of the results

The contributions of this chapter are summarized as follows:

- We begin by establishing fundamental bounds on the complexity of repair of single and multiple failures in the graph-constrained setting in Sec. 2.2. In Sec. 2.3, we provide explicit schemes for two popular MSR code families to attain the said lower bound. In doing so, we give an affirmative answer to the question of whether there exist more efficient ways of repair than simple relaying in the graph-constrained scenario, showing that if the data is encoded using an MSR code, then under some conditions it is possible to save on the communication cost of node repair compared to simple relaying of the information.

- Since intermediate data processing is already an essential component of a version of the node repair problem known as *cooperative repair* [71], it is therefore of interest to examine possible applications of cooperative MSR codes to the problem at hand, aiming again at reduced communication complexity of repair. We show one application of this idea in Sec. 2.4, using a family of cooperative codes to design a scheme with reduced repair bandwidth in the case of multiple failed nodes.

- In Sec. 2.5 we consider the problem of node repair in the situation when the helper nodes can exchange (and process) information before communicating with the failed nodes. We derive a framework to bound below the complexity of repair under this relaxation and use it to compute lower

bounds on the repair bandwidth in several examples. One of these examples also affords a matching code construction, again inspired by cooperative codes.

- In Sec. 2.6 we also address the same question for random graphs from the standard Erdös-Rényi ensemble $\mathcal{G}_{n,p}$ and determine a range of parameters under which the communication cost of repair with intermediate processing is advantageous over the repair scheme based on the relaying.

2.2 A lower bound on the repair bandwidth

Let $\mathcal{C}$ be an (n, k, d, l) MSR code and suppose that each coordinate of a codeword $C \in \mathcal{C}$ is written on a vertex of a graph $G(V, E)$ with $|V| = n$. Suppose further that the coordinate C_f, $f \in [n]$ is erased, or, as we will say, that the node v_f has failed. Let $D \subset V \backslash \{v_f\}, |D| = d$ be a set of helper nodes. To repair the failed node, the helper nodes provide information which is communicated to v_f over the edges in E. If one discounts the connectivity constraints, then to accomplish the repair, each of the helper nodes sends the information to the failed node over the shortest path in G, and the intermediate nodes simply relay this information further, possibly supplementing it with their own data. We call this repair strategy *accumulate and forward* (AF). To examine options for more economical repair including *intermediate processing* (IP) of the information, we begin with deriving a lower bound on the repair bandwidth.

Before proceeding, let us further specify our assumptions. We assume that for the failed node v_f, the helper nodes D are chosen to be the d *closest* nodes to v_f in terms of the graph distance[1]. These nodes can be found by a simple breadth-first search on G starting at v_f. Denote by $G_{f,D} = (V_{f,D}, E_{f,D})$ the subgraph spanned by $\{v_f\} \cup D$. Let $t = \max_{v \in D} \rho(v, v_f)$. We will use the following notation for spheres and balls around v_f in $G_{f,D}$:

$$\Gamma_j(v_f) = \{v \in V_{f,D} : \rho(v, v_f) = j\}, \; N_i(v_f) = \cup_{j=1}^{i} \Gamma_j(v_f),$$

[1]This assumption is not restrictive because, whenever the set D spans a connected subgraph, our bounds on communication complexity apply for the information processing within that subgraph.

and we refer to the vertices in $\Gamma_j(v_f)$ as the helper nodes in *layer* j. The case $t = 1$ corresponds to the much-studied repair scenario for complete graphs [16], and therefore we exclude it from consideration. Observe that the graph $G_{f,D}$ is not necessarily unique; in particular, there may be multiple possible choices for the helper nodes in the t-th layer.

In the next lemma, we derive lower bounds on the amount of information contributed by a group of helper nodes for the purposes of repair. The lemma is phrased in information-theoretic terms. We assume that the information stored at the vertices is given by random variables $W_i, i \in [n]$ that have some joint distribution on $(\mathbb{F}^l)^n$ and satisfy $H(W_i) = l$ for all i, where $H(\cdot)$ is the entropy. For a subset $A \subset V$ we write $W_A = \{W_i, i \in A\}$. For any $B \subset [n], |B| = k$, we have that $H(\mathcal{F}|W_B) = 0$ where $\mathcal{F}$ is the original stored file of size $M = kl$, i.e., the file $\mathcal{F}$ can be retrieved from the contents of any k nodes. Let S_i^f be the information provided to v_f by the ith helper node in the traditional, fully connected repair setting, and let $S_D^f = \{S_i^f, i \in D\}$. The RV S_i^f is a function of the contents of the node v_i, and the RVs $S_i^f, i \in D$ determine the contents of v_f, or formally,

$$(i)\ H(S_i^f|W_i) = 0,$$

$$(ii)\ H(W_f|S_D^f) = 0.$$

From the cut-set bound [16] it follows that $H(S_i^f) \geq l/(d-k+1)$, and we assume that this is achieved with equality, i.e., the codes we use have the MSR property. In the next lemma we bound below the amount of information sent by a subset of helper nodes in an MSR code. The proof that we give is close to the arguments that have previously appeared in the literature, see for instance [68].

Lemma 2.2.1. *Let $v_f, f \in [n]$ be the failed node. For a subset of the helper nodes $A \subset D$ let R_A^f be a function of W_A such that*

$$H(W_f|R_A^f, S_{D\setminus A}^f) = 0. \tag{2.1}$$

1) If $|A| \geq d - k + 1$, *then*

$$H(R_A^f) \geq l.$$

2) If $|A| \leq d - k$, *then*

$$H(R_A^f) \geq \frac{|A|l}{d - k + 1}.$$

Proof. Part (1): By the assumption (2.1), given the contents of all the nodes in $D\backslash A$, the information contained in R_A^f is sufficient to repair v_f, i.e.,

$$H(W_f|R_A^f, W_{D\backslash A}) = 0. \tag{2.2}$$

We have $|D\backslash A| \leq k - 1$. Consider a set $B \subset A$ with $|B| = k - 1 - |D\backslash A|$. Now,

$$H(R_A^f, W_{D\backslash A}, W_B) = H(R_A^f, W_{D\backslash A}, W_f, W_B) \geq kl, \tag{2.3}$$

where the equality in (2.3) follows from (2.2) and the chain rule, and the inequality follows from the MDS property of MSR codes because $|D\backslash A| + |B| + 1 = k$. Next observe that

$$\begin{aligned} H(R_A^f, W_{D\backslash A}, W_B) &\leq H(R_A^f) + H(W_{D\backslash A}, W_B) \\ &= H(R_A^f) + (k-1)l, \end{aligned} \tag{2.4}$$

where the equality again uses the independence of any $k - 1$ coordinates in an MDS code. Combining (2.3) and (2.4), we obtain the claimed inequality.

For Part (2), let $C \subseteq D\backslash A$ such that $|C| = k - 1$ and let $I = D\backslash\{A \cup C\}$. By the assumption (2.1), we have

$$H(W_f|R_A^f, W_C, S_I^f) = 0. \tag{2.5}$$

Now,

$$H(R_A^f, W_C, S_I^f) = H(R_A^f, W_f, W_C, S_I^f) \geq kl, \tag{2.6}$$

where the equality in (2.6) follows from (2.5) and the chain rule, and the inequality follows from the MDS property and the fact that $|C| = k - 1$. Next observe that

$$\begin{aligned} & H(R_A^f, W_C, S_I^f) \\ & \leq H(R_A^f) + H(W_C) + H(S_I^f) \\ & \leq H(R_A^f) + H(W_C) + \sum_{i \in D \setminus \{A \cup C\}} H(S_i^f) \\ & = H(R_A^f) + (k-1)l + \frac{(d - (k-1) - |A|)l}{d - k + 1} \end{aligned} \tag{2.7}$$

where we again use the independence of any $k - 1$ coordinates in an MDS code. Combining (2.6) and (2.7), we obtain the claimed inequality. □

Rephrasing this lemma, we obtain a lower bound on the amount of information transmitted between the layers in $G_{f,D}$.

Proposition 2.2.2. *Let R_j^f be the random variable denoting the information flow from the j-th layer to the $(j-1)$-th layer. Then*

$$H(R_j^f) \geq \min\left\{l, \frac{|\cup_{i=j}^{t} \Gamma_i(v_f)| \cdot l}{d - k + 1}\right\}$$

Proof. Follows from Lemma 2.2.1 by taking $A = \cup_{i=j}^{t} \Gamma_i(v_f)$. □

Note that R_j^f in the above proposition represents the joint information transmitted by all the nodes in layer j to layer $j - 1$ and hence it does not account for any other communication occurring among the helper nodes. If $G_{f,D}$ is a rooted tree, such communication does not occur, and this will be the main (but not the only) case studied below. In this case we can make our arguments more precise.

Let T_f be a rooted spanning tree of $G_{f,D}$ with root v_f (see Fig.1), then it defines the set of de-

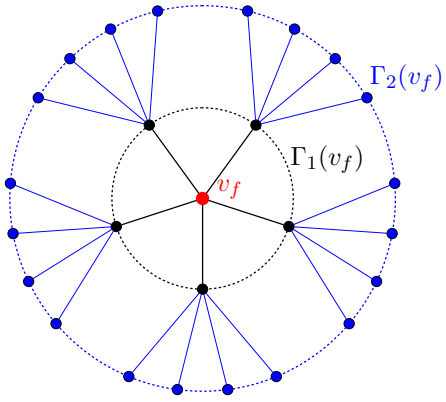


Figure 2.1: Repair tree of the node v_f

scendants of each node in T_f. Let $D(v_i)$ be the set of descendants of v_i, and let $D^*(v_i) = D(v_i) \cup \{v_i\}$. The total communication complexity of node repair using the tree T_f is bounded below in the following proposition.

Proposition 2.2.3. *Let $J_f = \{v \in V(T_f)\backslash\{v_f\} : |D^*(v)| \geq d - k + 2\}$. The total communication complexity β for the repair of node v_f on the repair tree T_f is bounded as*

$$\beta \geq |J_f|l + \sum_{v \in V(T_f)\backslash(\{v_f\} \cup J_f)} \frac{|D^*(v)|l}{d - k + 1}. \tag{2.8}$$

Proof. For every non-root node $v \notin J_f$, we have $|D^*(v)| \leq d - k$. Since T_f is a tree, any outflow of information out of the subtree spanned by $D^*(v)$ passes through the node v, so it needs to transmit at least $|D^*(v)| \cdot l/(d - k + 1)$ symbols to its immediate parent in T_f by Lemma 2.2.1. Similarly, every node $v \in J_f$ needs to transmit at least l symbols to its immediate parent by virtue of Lemma 2.2.1. □

For comparison purposes we also write out an expression for the AF repair procedure of MSR codes,

described in the beginning of this section. Its repair bandwidth can be found as

$$\beta_{\text{AF}} = \Big(t(d - |N_{t-1}(v_f)|) + \sum_{i=1}^{t-1} i|\Gamma_i(v_f)|\Big)\frac{l}{d-k+1}. \tag{2.9}$$

Note that for any node $v \notin J_f$, the AF strategy is trivially optimal. At the same time, for nodes $v \in J_f$ a better communication strategy is not a priori ruled out. This problem is addressed in the next section.

2.2.1 A bound for repair of multiple nodes

Before proceeding further, let us note a simple extension of Lemma 2.2.1 to the case of multiple failed nodes, which is often studied for regenerating codes under full node connectivity [88]. The repair of multiple nodes in a graph depends on their mutual placement and their connections to the helpers, and gives rise to several options. Denote by $F \subset V$ the set of failed nodes, and let $|F| = h \geq 1$. To keep the argument manageable, we assume that recovery of all the nodes in F relies on a *common* set D of helper nodes. With this assumption Lemma 2.2.1 affords the following extension.

Lemma 2.2.4. *Let $F \subset [n], |F| = h, 1 \leq h \leq n - d$ be the set of failed nodes. For a subset of the helper nodes $E \subset D$ let R_E^F be a function of S_E^F such that*

$$H(W_F|R_E^F, S_{D\setminus E}^F) = 0. \tag{2.10}$$

1) If $|E| \geq d - k + h$, then

$$H(R_E^F) \geq hl.$$

2) If $|E| \leq d - k + h - 1$, then

$$H(R_E^F) \geq \frac{h|E|l}{d-k+h}.$$

The proof follows closely the proof of Lemma 2.2.1 and is included in the Appendix for complete-

ness. As above, in Lemma 2.2.4 we sidestepped the specific way of communicating the information from the helpers to the failed nodes, limiting ourselves to the lower bounds on the information provided by the helpers. The communication complexity of implementing the repair depends on the topology of the graph and on the relative location of the failed nodes and the helpers. In Sec. 2.4 we present a construction that, under certain assumptions, attains the bounds of this lemma, performing the intermediate processing instead of relaying and gaining in communication complexity over the AF protocol.

2.3 MSR constructions for repair of graph vertices

In this section we show that linear MSR codes support a repair procedure that attains the lower bound (2.8) on the communication complexity. While this procedure is general, we begin with illustrating it for product-matrix codes of [59]. Then we consider several examples of graphs, estimating the savings of repair complexity compared to the AF repair. After that, we show that conceptually the same procedure applies to the diagonal-matrix codes of [88], and then briefly discuss a general version of this repair protocol as it applies to all families of $\mathbb{F}$-linear MSR codes.

2.3.1 Product-matrix (PM) codes

We begin with briefly recalling the code construction. Fix the code length n and the dimension parameter k, and take $d = 2k-2, l = k-1$. The code $\mathcal{C} : \mathbb{F}^{k(k-1)} \to \mathbb{F}^{ln}$ encodes $k(k-1)$ symbols of $\mathbb{F}$ into a codeword of length n with each coordinate formed of l symbols. To define this mapping, form a matrix $M = [S_1 \,|\, S_2]^T$, where S_1, S_2 are symmetric matrices of order l. The number of unique symbols in M equals $2\binom{l+1}{2} = k(k-1)$. Next let $x_i, i = 1, \dots, n$ be distinct elements of $\mathbb{F}$, let

$$\Phi = [\phi_1, \dots, \phi_n]^T$$

be a Vandermonde matrix with rows of the form $\phi_i = (1, x_i, \ldots, x_i^{l-1})$ and take $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with $\lambda_i = x_i^l, i = 1, \ldots, n$. Now form an $n \times 2l$ matrix $\Psi = [\Phi, \Lambda\Phi]$ The encoding mapping $\mathcal{C}$ sends the matrix M to $C = \Psi M$, which is an $n \times l$ matrix, and thus the contents of the node $v_i, i = 1, \ldots, n$ is given by the product

$$C_i := [\phi_i, x_i^l \phi_i] M = \phi_i S_1 + \lambda_i \phi_i S_2. \tag{2.11}$$

To describe the repair procedure from [59], suppose without loss of generality that the helper nodes form the set $D = \{1, \ldots, d\}$ and that the failed node's index is $f \in [n] \backslash [D]$. The original node repair (erasure correction) procedure proposed in [59] proceeds as follows. The information downloaded by the failed node v_f from the helper node $i \in D$ is given by $(\phi_i S_1 + \lambda_i \phi_i S_2)\phi_f^T$, i.e., each helper node provides one symbol of $\mathbb{F}$. Thus, the failed node downloads a d-dimensional vector $y = y_{f,D}$ given by

$$y = \Psi_D M \phi_f^T = \Psi_D \begin{bmatrix} S_1 \phi_f^T \\ \\ S_2 \phi_f^T \end{bmatrix}, \tag{2.12}$$

where Ψ_D is the submatrix of Ψ formed of the first $d = 2l$ rows. The matrix Ψ_D is square $d \times d$ and it is invertible by construction, so we can compute the vectors $(S_1 \phi_f^T)^T = \phi_f S_1$ and $(S_2 \phi_f^T)^T = \phi_f S_2$. By (2.11) the sum $\phi_f S_1 + \lambda_f \phi_f S_2$ equals C_f, and this completes the repair process.

Now we will modify the repair procedure in a way that supports processing the information received by the nodes in the repair tree as it is passed to the failed node v_f. Note that by (2.12)

$$\phi_f M^T = y^T (\Psi_D^T)^{-1}. \tag{2.13}$$

Using (2.11), (2.13), the contents of the node v_f can be written as

$$C_f = \phi_f M^T \begin{bmatrix} I_l \\ \lambda_f I_l \end{bmatrix} = y^T (\Psi_D^T)^{-1} \begin{bmatrix} I_l \\ \lambda_f I_l \end{bmatrix}.$$

Introduce a $d \times l$ matrix $U := (\Psi_D^T)^{-1} \begin{bmatrix} I_l \\ \lambda_f I_l \end{bmatrix}$ and denote its rows by $U_i, i = 1, \ldots, d$, then we have

$$C_f = \sum_{i=1}^{d} y_i U_i. \tag{2.14}$$

Note that the matrix U does not depend on the codeword, and can be precomputed. Overall this rewriting of the repair process (2.12) enables us to separate the contributions of the helper nodes, and offers savings in the communication cost of repair. Recalling our notation $D^*(v_i)$, suppose that, instead of transmitting the symbol y_i to its parent, the node transmits the sum $\sum_{j \in D^*(v_i)} y_j U_j$. Since we are now moving vectors rather than individual symbols along the edges of T_f, this may seem wasteful; however remember that the symbols are relayed many times, and that from some point on, the repair process has to move at least l symbols along the edge by Lemma 2.2.1. To justify the savings, suppose that $|D^*(v_i)| \geq d - k + 2 = k$, then forwarding the symbols $(y_j, j \in D^*(v_i))$ from v_i to its predecessor in T_f amounts to sending at least k symbols, whereas transmitting the sum $\sum_{j \in D^*(v_i)} y_j U_j$ requires $l = k - 1$ transmissions.

Therefore, the communication for repair can be summarized as follows. First, the leaf nodes in T_f send their symbols y_i one level up, then the nodes that received these symbols send them together with their symbols y_i, etc. If at any stage a node v_i has $d - k + 1$ or more descendants, then it switches to transmitting

$$\sum_{j \in D^*(v_i)} y_j U_j. \tag{2.15}$$

Finally if a node v_i received a vector $\sum_{j \in D(v_i)} y_j U_j$ from its immediate descendant, it adds to it the vector

$y_i U_i$ and forwards it to its parent in T_f.

In summary, we have shown that, for every node $v_i \in T_f$ with $|D(v_i)| \geq d - k + 1$ descendants in T_f there exists a repair procedure under which v_i transmits exactly l symbols of $\mathbb{F}$ to its parent in T_f. This proves the following theorem.

Theorem 2.3.1. *Suppose a codeword of a PM code $\mathcal{C}$ is written on the vertices of a graph G, and let T_f be the repair tree of a failed node v_f. There exists an explicit repair procedure that achieves the lower bound in (2.8) with equality.*

To match the above procedure to the bound (2.8), recall that each helper node in the PM code construction provides one symbol of $\mathbb{F}$ for repair.

2.3.2 Examples of graphs

Let us give a few examples in which the proposed repair procedure gains in communication complexity over the AF repair. For simplicity we will assume that each helper node provides one symbol of $\mathbb{F}$ for the repair of v_f.

1. Suppose that the repair tree T_f is a *star* with d rays in which v_f is one of the leaves and the remaining d vertices serve as the helper nodes. Using the AF repair, each of the nonerased leaves sends its symbol to the center, which then sends d symbols to v_f, so $\beta_{\text{AF}} = 2d - 1 = 4k - 5$. At the same time, $\beta_{\text{IP}} = 3k - 4$ because the symbols of the helpers other than the center are aggregated using (2.15) before relaying to v_f. Another elementary example, which also shows improvement, arises when the repair tree T_f is a *path* on $d + 1$ vertices.

2. *Regular tree*. Suppose that G is an $(r + 1)$-regular graph, and the repair tree T_f of every node is $(r + 1)$-regular as shown in Fig.1. We need to take the depth t of the tree to satisfy $(r + 1)\sum_{i=0}^{t-1} r^i \geq d$; suppose for simplicity that this holds with equality. The communication complexity of the AF repair is

$$\beta_{\text{AF}} = td - (r+1)\sum_{i=0}^{t-2}(t-i-1)r^i.$$

Suppose that $r > d-k+1$, then from the next to last layer we can switch to uploading the linear combination of the form (2.15), resulting in the repair bandwidth $\beta_{\text{IP}} = d + (d-k)(r+1)\sum_{i=0}^{t-2} r^i$. The difference

$$\beta_{\text{AF}} - \beta_{\text{IP}} = (t-1)d - (r+1)\sum_{i=0}^{t-2}((d-k)+(t-i-1))r^i$$

is positive if $\frac{d-k}{d}$ is small, i.e., if $d \geq k$ is close to k. Note that the regime of small $d-k$ arises also as a sufficient condition of repair bandwidth savings for random graphs in Sec. 2.6.

3. *Galton-Watson tree.* Having in mind a scenario in which the helper nodes are chosen randomly and independently by the nodes already included in the repair tree T_f, suppose that it is constructed following a branching process with the root v_f, resulting in a Galton-Watson ensemble of random trees $\mathcal{T}_f$. In this example we choose a simple "offspring pmf" under which a node has 1 or 2 descendants with probability p and $1-p$, respectively. Let $Z_i = |\Gamma_i(v_f)|$ be the total number of vertices in layer i of $\mathcal{T}_f$. Thus, $\Pr(Z_1 = 1) = p = 1 - \Pr(Z_1 = 2)$ where $p \in (0,1)$ is chosen to satisfy $m := \mathbb{E}(Z_1) = 2 - p > 1$ so that we are operating in the *supercritical regime.* Assuming that a tree of depth t suffices for repair, we have

$$\beta_{AF} = td - \sum_{i=1}^{t-1}(t-i)Z_i; \;\; \mathbb{E}[\beta_{\text{AF}}] = td - \sum_{i=1}^{t-1}(t-i)m^i.$$

If we assume that the intermediate processing technique can be applied to layers $i, 1 \leq i \leq s$, then an easy calculation yields

$$\mathbb{E}[\beta_{\text{IP}}] = (t-s)d + (d-k+1-t+s)\sum_{i=1}^{s} m^i - \sum_{i=s+1}^{t-1}(t-i)m^i$$

and so

$$\mathbb{E}[\beta_{\text{AF}} - \beta_{\text{IP}}] = sd - \sum_{i=1}^{s}(2-p)^i(d-k+1+s-i)$$

which is positive for small values of $d-k$ and large d.

2.3.3 Diagonal-matrix MSR codes

While the product-matrix codes are limited by the code rate $k/n < 1/2$, the construction of [88] removes this limitation, providing explicit families of exact-repair MSR codes for all possible values of $n-1 \geq d \geq k$.

The codes in [88] are defined in terms of the parity-check matrix which has a block diagonal structure. Below we assume that the parameters of the (n,k,l) array code $\mathcal{C}$ are fixed, and that $d=n-1, l=r^n$, where $r := n-k$. The code is defined over a finite field $\mathbb{F}$ of size at least rn. Let $\{\lambda_{i,j}\}_{i\in[n],j=0,1,\ldots,r-1}$ be rn distinct elements of $\mathbb{F}$. For an integer $a \in \{0,1,\ldots,l-1\}$ let a_i be the i-th digit of its r-ary expansion. For $i=1,2,\ldots,n$ define the matrix $A_i = \mathrm{diag}(\lambda_{i,a_i}, a=0,\ldots,l-1)$. The code $\mathcal{C}$ is formed of the codewords $C=(C_1,\ldots,C_n) \in (\mathbb{F}^l)^n$ that satisfy the following set of r parity-check equations:

$$\sum_{i=1}^{n} A_i^{t-1} C_i = 0, \quad t=1,\ldots,r. \tag{2.16}$$

Let $C_i = (c_{i,a}, a=0,\ldots,l-1)^T$. Since the matrices A_i are diagonal, the parity check equations (2.16) take the form

$$\sum_{i=1}^{n} \lambda_{i,a_i}^{t-1} c_{i,a} = 0, \quad t=1,\ldots,r,\ a=0,1,\ldots,l-1. \tag{2.17}$$

The node repair with no communication constraints proceeds as follows. Assume that the node $i \in [n]$ has failed. We partition the set of coordinates $(c_{i,a})$ into groups of size r whose indices differ only in the ith entry. Namely, start with some $a \in \{0,\ldots,l-1\}$ and consider the set of indices $a(i,u) = (a_n,\ldots,a_{i+1},u,a_{i-1},\ldots,a_1)$, $u=0,1,\ldots,r-1$. The information downloaded from the helper node

$j \in [n]\backslash\{i\}$ is given by $\mu_{j,i}^{(a)} = \sum_{u=0}^{r-1} c_{j,a(i,u)}$. Writing (2.17) for each of the indices $a(i,u)$, we obtain

$$\lambda_{i,u}^t c_{i,a(i,u)} + \sum_{j \neq i} \lambda_{j,a_j}^t c_{j,a(i,u)} = 0, \quad t = 0, 1, \ldots, r-1.$$

Summing these equations on u and writing the result in matrix form, we obtain the relation

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_{i,0} & \lambda_{i,1} & \ldots & \lambda_{i,r-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{i,0}^{r-1} & \lambda_{i,1}^{r-1} & \ldots & \lambda_{i,r-1}^{r-1} \end{bmatrix} \begin{bmatrix} c_{i,a(i,0)} \\ c_{i,a(i,1)} \\ \vdots \\ c_{i,a(i,r-1)} \end{bmatrix} = - \begin{bmatrix} \sum_{j \neq i} \mu_{j,i}^{(a)} \\ \sum_{j \neq i} \lambda_{j,a_j} \mu_{j,i}^{(a)} \\ \vdots \\ \sum_{j \neq i} \lambda_{j,a_j}^{r-1} \mu_{j,i}^{(a)} \end{bmatrix}. \tag{2.18}$$

This equation permits recovery of the symbols $c_{i,a(i,u)}, 0 \leq u \leq r-1$ of the failed coordinate, and varying a, we recover the other groups of coordinates in the same manner.

To adapt this procedure to repair on graphs, assume that the failed node is $i = n$ and write the vector on the right-hand side of (2.18) as $[\mu_{1,n}^{(a)}, \mu_{2,n}^{(a)}, \ldots, \mu_{n-1,n}^{(a)}]V_1^T$, where

$$V_1 := \text{Vandermonde}(\lambda_{1,a_1}, \lambda_{2,a_2}, \ldots, \lambda_{n-1,a_{n-1}})$$

is an $r \times (n-1)$ Vandermonde matrix with columns defined by the arguments. The matrix on the left in (2.18) is also Vandermonde, denote it by V_2. With these notations, (2.18) can be rewritten as

$$[c_{n,a(n,0)}, c_{n,a(n,1)}, \ldots, c_{n,a(n,r-1)}]V_2^T = -[\mu_{1,n}^{(a)}, \mu_{2,n}^{(a)}, \ldots, \mu_{n-1,n}^{(a)}]V_1^T$$

or

$$[c_{n,a(n,0)}, c_{n,a(n,1)}, \ldots, c_{n,a(n,r-1)}] = [\mu_{1,n}^{(a)}, \mu_{2,n}^{(a)}, \ldots, \mu_{n-1,n}^{(a)}]U$$

$$= \sum_{j=1}^{n-1} \mu_{j,n}^{(a)} U_j \tag{2.19}$$

where we denoted $U := -V_1^T (V_2^T)^{-1}$ and U_j is the jth row of U. This representation is essentially the same as (2.14), and hence the generic distributed repair scheme described in Sec. 2.3 applies to the codes considered in this section. Specifically, the matrix U is independent of the codeword, and can be computed in advance, and once a node v in the repair tree has $d - k + 1$ or more descendants, it switches to transmitting $\sum_{j \in D^*(v)} \mu_{j,n}^{(a)} U_j$. This procedure supports repair bandwidth gains over the AF strategy for each of the groups of the node components mentioned above.

2.3.4 Node repair for general linear MSR codes

From the examples in the previous sections it is clear that the graph-based repair procedure defined in (2.15) applies to any $\mathbb{F}$-linear MSR code for which the information downloaded from the helper nodes is an $\mathbb{F}$-linear function of their contents (all the known MSR codes are such). Indeed, the download operation can be written as $C(D)U$, where $C(D)$ is the contents of the helper nodes and U represents the linear transformation of the form (2.15). Once we reach the helper nodes in T_f with at least $d - k + 1$ descendants, then we can switch to relaying linear combinations rather than the contents of the helper nodes. The savings in repair bandwidth will be the same as for the two constructions considered above in this section.

Remark (MBR codes): For the other extremal point of the storage-bandwidth trade-off [16], i.e., the Minimum Bandwidth Regenerating codes, the AF repair strategy is optimal in terms of the repair bandwidth because the amount of downloaded information is minimized by the code design.

2.4 Node repair for multiple failures

In this section we present a code construction for the repair of multiple nodes that attains the lower bound of Lemma 2.2.4. We begin with specifying our assumptions. Suppose that the data is stored on a connected graph $G(V, E)$, and $F \subset V$ is a set of failed vertices of size h. Further, let $D, |D| = d$ be the subset of helper nodes. The data is encoded using an (n, k, d, l) MSR code, where $n = |V|$ is the number of vertices. The encoding scheme that we present below further assumes that the communication from D to F passes through some fixed node $w \in D$ as shown in Fig. 2 for $h = 2$ and $F = \{v_1, v_2\}$. This assumption, taken to fit the structure behind Lemma 2.2.4, suggests that we perform simple relaying along the path(s) from w to the failed vertices. The repair process becomes more complicated if the failed vertices have different access points to D, and we do not consider it here. We further assume that the set D spans a connected subgraph $G_D \subset G$ and denote by T_w a (rooted) spanning tree of G_D with root w. Finally, denote by $D_w(v)$ the set of descendants of $v \in V(T_w)$ in the tree T_w and let $D^*_w(v) = D_w(v) \cup \{v\}$.

Under these assumptions it is possible to write out a bound on the communication complexity of repair within the set of the helper nodes until the data reaches the node w (after that the data is no longer processed until it reaches the nodes in F). The following proposition is an obvious extension of the bound (2.8).

Proposition 2.4.1. *Let $J_{w,h} = \{v \in V(T_w) : D_w(v) \geq d - k + h\}$. The total communication along the edges of T_w for repair of the nodes in F is bounded below as*

$$\beta(D) \geq |J_{w,h}|l + \sum_{v \in V(T_w) \setminus J_{w,h}} \frac{|D^*_w(v)|l}{d - k + h}.$$

Below we present a construction of codes and a repair scheme that meets this bound with equality, attaining the minimum possible communication complexity of repair of the nodes in F under the assumptions discussed above (it is possible that removing these assumptions enables one to further lower the

communication cost). The scheme relies on the idea of *cooperative repair* [71]. In this setting, under the full connectivity assumption, two or more failed nodes connect directly to the same set of helpers, evaluate partial information about their contents, and then exchange the results to complete the repair.

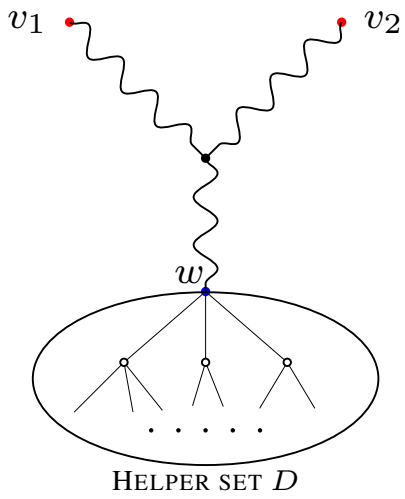


Figure 2.2: Graph topology for repair of multiple nodes

We use this idea for repair on graphs wherein the information from helpers is transmitted along some path to the failed node, relying on a family of cooperative codes constructed recently in [87]. The savings come from the fact that in the course of this transmission we can perform intermediate processing rather than simple relaying. In our presentation for simplicity we assume that $d = k + 1, h = 2$ as in Fig. 2.2. At the same time it will be obvious that the technique applies to all other feasible parameter regimes.

Let $\mathcal{C}$ be the $[n, k, d = k+1, l = 3\times 2^n]$ *cooperative repair* MSR code from the family constructed in [87][2]. Every coordinate C_i of a codeword $C = (C_1, \ldots, C_n) \in (\mathbb{F}^l)^n$ is a vector $\{c_{i,b,a} : b \in \{1, 2, 3\}, a \in \{0, 1, \ldots 2^n - 1\}\}$. For $2n$ distinct field elements $\{\lambda_{i,j} : i \in [n], j \in \{0, 1\}\}$, the parity check equations

[2]We could use other code families, for instance, the codes from [89].

that define the code are

$$\sum_{i=1}^{n} \lambda_{i,a_i}^{t} c_{i,b,a} = 0 \quad \forall \quad t \in \{0,1,\ldots,n-k-1\}, \;\; a \in \{0,1,\ldots,2^n-1\}, \;\; b \in \{1,2,3\}, \tag{2.20}$$

where a_i is the i-th coordinate in the binary representation of a. Below we use the notation $a(i, a_i \oplus 1)$ to denote the number obtained from a by flipping the ith bit in its binary expansion. Assume that the failed nodes correspond to coordinates 1 and 2 and fix a value of $a \in \{0,1,\ldots,2^n-1\}$. The standard cooperative repair under direct connectivity (on a complete graph) proceeds in two steps. In step 1, helper node i sends $\{c_{i,1,a}+c_{i,2,a(1,a_1\oplus 1)} : a \in \{0,1,\ldots,2^n-1\}\}$ to node 1 and $\{c_{i,1,a}+c_{i,3,a(2,a_2\oplus 1)} : a \in \{0,1,\ldots,2^n-1\}\}$ to node 2. Using a with $b = 1$ and $a(1, a_1 \oplus 1)$ with $b = 2$ in (2.20) and summing the corresponding equations, we obtain

$$\begin{aligned} \lambda_{1,a_1}^{t} c_{1,1,a} + \lambda_{1,a_1\oplus 1}^{t} c_{1,2,a(1,a_1\oplus 1)} + \lambda_{2,a_2}^{t}(c_{2,1,a} + c_{2,2,a(1,a_1\oplus 1)}) \\ + \sum_{i=3}^{n} \lambda_{i,a_i}^{t}(c_{i,1,a} + c_{i,2,a(1,a_1\oplus 1)}) = 0 \end{aligned} \tag{2.21}$$

for all $t \in \{0,1,\ldots,n-k-1\}$. Equations (2.21) form a set of parity checks of an $(n+1, k+1)$ Reed-Solomon code, and hence knowing $c_{i,1,a} + c_{i,2,a(1,a_1\oplus 1)}$ at $k+1$ positions allows node 1 to recover $c_{1,1,a}, c_{1,2,a(1,a_1\oplus 1)}$ and $(c_{2,1,a} + c_{2,2,a(1,a_1\oplus 1)})$. A similar argument shows that node 2 can recover $c_{2,1,a}, c_{2,3,a(2,a_2\oplus 1)}$ and $(c_{1,1,a}+c_{1,3,a(2,a_2\oplus 1)})$. In step 2 of the repair, node 1 sends $(c_{2,1,a}+c_{2,2,a(1,a_1\oplus 1)})$ to node 2 and node 2 sends $(c_{1,1,a} + c_{1,3,a(2,a_2\oplus 1)})$ to node 1, which completes the repair of both node 1 and 2; for details see [87].

To see how intermediate processing at the nodes of the tree T_w can simplify repair on a graph of the type shown in Fig. 2.2, observe that the first step above, node 1 seeks to learn three code symbols (namely $c_{1,1,a}, c_{1,2,a(1,a_1\oplus 1)}$ and $(c_{2,1,a} + c_{2,2,a(1,a_1\oplus 1)})$) of the $(n+1, k+1)$ RS codeword, and it does so by collecting $k+1$ symbols from $k+1$ helper nodes. In an RS code, once we know any $k+1$ coordinates,

all the other coordinates of the codeword can be computed via Lagrange interpolation and subsequent evaluation. This can be expressed in matrix form as follows:

$$\begin{bmatrix} c_{1,1,a} \\ c_{1,2,a(1,a_1\oplus 1)} \\ (c_{2,1,a} + c_{2,2,a(1,a_1\oplus 1)}) \end{bmatrix} = [U_1\ U_2\ \ldots\ U_{k+1}] \begin{bmatrix} (c_{i_1,1,a} + c_{i_1,2,a(1,a_1\oplus 1)} \\ (c_{i_2,1,a} + c_{i_2,2,a(1,a_1\oplus 1)} \\ \vdots \\ (c_{i_{k+1},1,a} + c_{ik+1,2,a(1,a_1\oplus 1)} \end{bmatrix}$$

where $i_1, i_2, \ldots, i_{k+1}$ are the helper nodes and the matrix $U = [U_1\ U_2\ \ldots\ U_{k+1}]$ is obtained by multiplying an inverse Vandemonde matrix (Lagrange interpolation) and a matrix corresponding to evaluating the obtained polynomial at the three coordinates being sought. Since the matrix U can again be pre-computed, a node that has collected the values $(c_{i,1,a} + c_{i,2,a(1,a_1\oplus 1)})$ from three or more helper nodes, can start transmitting the corresponding linear combinations, much in the same way as was done in Section 2.3. The above procedure is repeated for node 2 with appropriate adjustments to the subscripts in the last displayed equation. Step 2 of the repair process is unchanged from that of the standard cooperative repair, and it yields no communication savings. Exactly as in the case of a single failed node, viz., Theorem 2.3.1, we can show that this procedure meets the bound of Lemma 2.2.4.

2.5 Repair with information exchange among the helpers

The bounds and constructions presented earlier in this chapter are focused on communication from the helper nodes to the failed node. In this section we consider a more general problem (and potential savings in the repair cost) when the helper nodes may communicate with each other before transmitting the information to the failed node. Recall that a variant of this problem was considered earlier in the literature under very specific assumptions: The nodes in the storage cluster are organized in subsets, called *racks*, and communication between the nodes in the rack does not count toward the repair bandwidth. This model

enables one to derive tighter bounds on the cost of node repair [33], and there are families of codes that attain these bounds [12].

Another version of information exchange in the context of erasure recovery appeared earlier in the problem of *cooperative repair*, already mentioned in the previous section. In this setting (assuming full connectivity) several failed nodes contact the same set of helpers and process the received information, gaining some knowledge about their contents and about the contents of the other failed nodes. They then exchange information to complete the repair. This problem, introduced in [71], is vaguely reminiscent of repair on graphs because different nodes of the encoding acquire partial information about the contents of other nodes. Below we make this link more precise by presenting an example of node repair on graphs motivated by cooperative repair (albeit in a rather restricted setting).

We begin with establishing a framework for finding a lower bound on the total communication complexity of repair for general graphs. Let us define some additional notation. Let an (n, k, d, l) MSR code be defined on a connected graph $G = (V, E)$. Assume that the subgraph $G_{f,D} = (V_{f,D}, E_{f,D})$ spanned by the failed node v_f and the set of helper nodes D is connected. Construct a directed graph $\bar{G}_{f,D} = (V_{f,D}, \bar{E}_{f,D})$ as follows:

- For every edge $(u, v) \in E_{f,D}$ with $u, v \in D$, add the two directed edges (u, v) and (v, u) to $\bar{E}_{f,D}$.
- For every edge $(u, v_f) \in E_{f,D}$, add the directed edge (u, v_f) to $\bar{E}_{f,D}$.

For an arbitrary communication protocol that repairs the failed node v_f, let $X_{u,v}$, for $(u, v) \in \bar{E}_{f,D}$, be the total number of symbols sent along the edge (u, v) during the complete protocol. Fix an order of the edges in $\bar{E} = \bar{E}_{f,D}$ and let $\bar{X}$ be the vector of $X_{u,v}$'s. Let $\mathcal{P}^*(D)$ be the set of all non-empty subsets of D. Define a binary matrix M of size $(2^d - 1) \times |\bar{E}|$ by setting $M_{S,(u,v)} = \mathbb{1}(u \in S \wedge v \in S^c)$, where $S^c = V_{f,D} \backslash S$. Let $\bar{b} \in \mathbb{R}^{|2^d - 1|}$ with $\bar{b}_S = \beta \cdot \min\{d - k + 1, |S|\}$ for all $S \in \mathcal{P}^*(D)$.

Proposition 2.5.1. *For the failed node v_f and helper nodes D, the total communication complexity of*

repair is bounded below by the solution to the following linear program with $|\bar{E}|$ *variables and* $2^d - 1$ *constraints:*

$$\begin{aligned} &\textit{minimize} && \mathbf{1}^T \bar{X} \\ &\textit{subject to} && M\bar{X} \geq \bar{b}, \\ & && \bar{X} \geq 0. \end{aligned}$$

Proof. We only need to justify the inequality $M\bar{X} \geq \bar{b}$. For any set $S \in \mathcal{P}^*(D)$, Lemma 2.2.1 implies

$$\sum_{\substack{(u,v)\in\bar{E}_{f,D} \\ u\in S, v\in S^c}} X_{u,v} \geq R_S^f \geq \min\{d-k+1, |S|\}\beta.$$

Collecting these inequalities for all $S \in \mathcal{P}^*(D)$, we obtain the claimed set of constraints. □

The key observation here is that the quantity R_A^f in Lemma 2.2.1 represents the total outflow of information transmitted from the set of nodes A for the repair, and hence the bounds still hold irrespective of the communication among the nodes in set A.

In the remainder of this section we consider two settings in which the bound of this proposition enables one to prove optimality of communication for recovery while allowing communication between the helper nodes, namely when the failed node has the largest and the smallest possible number of helpers, respectively, as immediate neighbors. In both cases we allow arbitrary communication among the helper set.

2.5.1 The case of the complete graph

This case corresponds to the original repair problem of [16], and the cut-set bound provides the minimum required download per helper node for the repair of a failed node. In this model, the transmitted data of each helper node is a function of its own stored content only. Can communication complexity be reduced if the helper nodes are allowed to exchange information before communicating with the failed node? An

easy corollary of Proposition 2.5.1 and Lemma 2.2.1 implies that in case of MSR codes the answer is negative.

Proposition 2.5.2. *For the complete graph K_n, the communication complexity is bounded below by $d\beta$ and is achieved by having all the helper nodes directly transmit β symbols to the failed node.*

Proof. Consider the assignment of variables of the LP problem X^* with $X^*_{u,v} = \beta \mathbb{1}(v = v_f)$. It is clearly feasible because it corresponds to all the helper nodes transmitting β symbols to the failed node. Indeed, this assignment satisfies the bounds of Lemma 2.2.1 and thus also the inequality constraints of Proposition 2.5.1. Next we show that X^* is optimal by considering the dual LP problem, which has the form

$$\begin{aligned} \text{maximize} \quad & \bar{b}^T \bar{Y} \\ \text{subject to} \quad & M^T \bar{Y} \leq \mathbf{1}, \\ & \bar{Y} \geq 0 \ . \end{aligned}$$

Take the assignment of variables Y^* with $Y^*_S = \mathbb{1}(|S| = 1)$ for all $S \subset D$. Since for two different $S_1 = \{v_1\}$ and $S_2 = \{v_2\}$ any edge $(u, v) \in \bar{E}$ can belong to at most one of the cuts (S_1, S_1^c) or (S_2, S_2^c), we have that $M^T Y^* \leq \mathbf{1}$. Since $\mathbf{1}^T X^* = \bar{b}^T Y^* = d\beta$, we conclude that X^* is indeed optimal. □

2.5.2 The case of two neighbors

Assume that the information is encoded with an $[n, k, d = k + 1, l]$ MSR code and stored on a graph $G(V, E)$ with $|V| = n$. Consider the repair graph (no longer a tree) shown in Fig. 2.3 with the failed node v_f connected to two helper nodes which connect to the remaining subset of the helper set.

We will prove that for this graph the minimum required communication for repair equals $(d+1)\beta = (k+2)\beta$. To show this, assume that the failed node v_f relies on a set D of $k+1$ helpers for repair, and that it is connected to two of them, denoted v_1 and v_2. Assume further that the $k + 1$ helpers span a complete graph K_{k+1}, where k is the dimension of the MSR code used for the encoding of the data. To link this

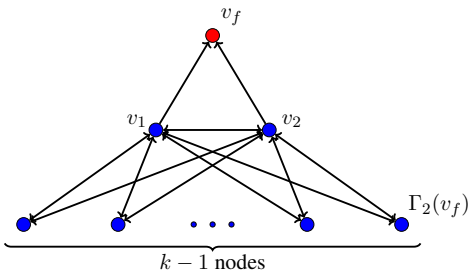


Figure 2.3: Repair graph of the node v_f that attains the LP lower bound

graph to the LP problem of Prop. 2.5.1, construct a directed graph by replacing every edge between a pair of helpers with a pair of opposing directed edges, and make a directed edge from each of v_1, v_2 to v_f. Thus, the new set of *directed* edges is

$$\bar{E} = \{((v_i, v_j), v_i, v_j \in D), \text{ and } (v_1, v_f), (v_2, v_f)\}.$$

To construct a primal LP program, assign

$$X^*_{(u,v)} = \begin{cases} \beta & \text{if } u \in \Gamma_2(v_f) \cup \{v_2\}, v = v_1 \\ 2\beta & \text{if } u = v_1, v = v_f \\ 0 & \text{otherwise.} \end{cases}$$

This assignment defines a valid repair protocol, so it's a feasible solution of the LP problem which gives the value of the objective function to be

$$\mathbf{1}^T X^* = (d-1)\beta + 2\beta = (d+1)\beta \tag{2.22}$$

Construct a dual program $Y^* = (Y^*_S)_S$ by setting

$$Y^*_S = \begin{cases} \frac{1}{d-2} & \text{if } |S| = 2, S \neq \{v_1, v_2\} \\ 0 & \text{otherwise.} \end{cases}$$

The vector Y^* is a feasible assignment of the dual program. To show this, consider an edge (u, v) with $u \in D$. Our argument depends on whether $v \in D$ or $v = v_f$. In the first case, the row of M^T contains $d-2$ ones which correspond to the cuts in $G_{f,D}$ that contain the edge (u, v) (there are exactly $d-2$ such cuts), so this row times Y^* equals one. If $v = v_f$, then u is either v_1 or v_2. Say it is v_1, then the row (v_1, v_f) contains $d-1$ ones which correspond to the cuts that contain the edge (v_1, v_f). Further, $Y^* = 0$ in the coordinate $S = \{v_1, v_2\}$, so the nonzeros in the vector Y^* and the $(M^T)_{v_1,v_f}$ overlap in $d-2$ places, again satisfying the constraints of the dual program.

To compute the value of the dual problem, note that $Y^* \neq 0$ in $\binom{d}{2} - 1 = \frac{(d-2)(d+1)}{2}$ coordinates, and the corresponding entries in $\bar{b}$ are set to 2β. Thus, $\bar{b}^T Y^* = \frac{(d-2)(d+1)}{2} \cdot \frac{1}{d-2} \cdot 2\beta = (d+1)\beta$, which equals the value of the primal problem, proving that the repair protocol defined by it yields the minimum possible communication complexity. Finally, we argue that if there does not exist a repair protocol that performs better in terms of complexity when the helper nodes form the complete graph, then there cannot exist a repair protocol that performs better for any sub-graph of the complete graph.

The repair bandwidth $(d+1)\beta$ can be attained by sending the data from all the helper nodes but v_1 to the node v_1, combining them and passing the result to v_f (which is the IP repair discussed earlier). This repair protocol does not involve two-way communication between the neighboring helper nodes. We now construct another protocol that does involve it, while still having the same communication complexity of repair.

2.5.2.1 An alternative optimal protocol with two way communication

We present an encoding/repair scheme for the example in Sec. 2.5.2, Fig. 2.3 that enables recovery of the contents of the node v_f which assumes that the nodes in $\Gamma_1(v_f)$ exchange information before passing the repair data to the node v_f. The construction shares some features of cooperative repair, and it relies on a code family constructed earlier for the case of the complete graph K_n [88, Sec.IV]. Namely, suppose that the information is encoded with an $[n, k, d = k + 1, l = 2^n]$ MSR array code $\mathcal{C}$ and the codeword coordinates are placed on the vertices of a graph $G(V, E)$ with $|V| = n$. Suppose further that the repair graph of the failed node v_f is as shown in the figure. In accordance with [88], Construction 2, we will assume that each helper node provides $\beta = 2^{n-1}$ symbols for the repair of v_f.

Let $C = (c_1, c_2, \ldots, c_n) \in \mathcal{C}$ be a codeword with $c_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,l-1}) \in \mathbb{F}^l$. The code is defined by the following parity-check equations:

$$\sum_{i=1}^{n} \lambda_{i,a_i}^t c_{i,a} = 0 \quad \text{for all } a \in \{0, 1, \ldots, l-1\},\ t \in \{0, 1, \ldots, n-k-1\}$$

where $(a_1, a_2, \ldots, a_n)$ is the binary representation of a. Below we assume that $v_f = 1$, that $\Gamma_1(1) = \{2, 3\}$. and that $\Gamma_2(1) = \{4, 5, \ldots, k+2\}$. For a string $s = (s_1, s_2, \ldots, s_{n-k-2}) \in \{0, 1\}^{n-k-2}$, consider the set of 2^{k+2} values of a for which $(a_{k+3}, a_{k+4}, \ldots, a_n) = s$. Isolate a subset of this set by fixing a string $\hat{s} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{k-1}) \in \{0, 1\}^{k-1}$ and collecting only those values of a for which $(a_4, a_5, \ldots, a_{k+2}) = \hat{s}$. Having fixed s and $\hat{s}$, we are left with 8 parity check equations which can be labeled by a binary vector $\tilde{s} \in \{0, 1\}^3$:

$$\lambda_{1,\tilde{s}_1}^t c_{1,(\tilde{s},\hat{s},s)} + \lambda_{2,\tilde{s}_2}^t c_{2,(\tilde{s},\hat{s},s)} + \lambda_{3,\tilde{s}_3}^t c_{3,(\tilde{s},\hat{s},s)} + \sum_{i=4}^{k+2} \lambda_{i,\hat{s}_{i-3}}^t c_{i,(\tilde{s},\hat{s},s)} + \sum_{i=k+3}^{n} \lambda_{i,s_{i-k-2}}^t c_{i,(\tilde{s},\hat{s},s)} = 0$$

$$\forall\ \tilde{s} \in \{0,1\}^3,\ \ t \in \{0,1,\ldots,n-k-1\} \quad (2.23)$$

For fixed $\hat{s}$ and s, the λ's in the last two sums the same in all the equations. Define

$$\mu_{2,1,i}^{(\hat{s},s)} = c_{i,(000,\hat{s},s)} + c_{i,(010,\hat{s},s)} + c_{i,(100,\hat{s},s)}$$

$$\mu_{3,1,i}^{(\hat{s},s)} = c_{i,(000,\hat{s},s)} + c_{i,(100,\hat{s},s)} + c_{i,(101,\hat{s},s)}$$

$$\mu_{2,2,i}^{(\hat{s},s)} = c_{i,(001,\hat{s},s)} + c_{i,(011,\hat{s},s)} + c_{i,(111,\hat{s},s)}$$

$$\mu_{3,2,i}^{(\hat{s},s)} = c_{i,(011,\hat{s},s)} + c_{i,(110,\hat{s},s)} + c_{i,(111,\hat{s},s)}$$

For $i \in \{4,5,\ldots,k+2\}$ the helper node i sends $\mu_{2,1,i}^{(\hat{s},s)}, \mu_{2,2,i}^{(\hat{s},s)}$ to node 2 and $\mu_{3,1,i}^{(\hat{s},s)}, \mu_{3,2,i}^{(\hat{s},s)}$ to node 3. Additionally node 3 sends $\mu_{2,1,3}^{(\hat{s},s)}, \mu_{2,2,3}^{(\hat{s},s)}$ to node 2 and node 2 sends $\mu_{3,1,2}^{(\hat{s},s)}, \mu_{3,2,2}^{(\hat{s},s)}$ to node 3.

Node 2, having $\mu_{2,1,i}^{(\hat{s},s)}$ for all $i \in \{3,\ldots,k+2\}$, can recover $c_{1,(000,\hat{s},s)} + c_{1,(010,\hat{s},s)}, c_{1,(100,\hat{s},s)}$. To see this, sum Eqns. (2.23) for $\tilde{s} \in \{000, 010, 100\}$ to obtain

$$\begin{aligned}
&\lambda_{1,0}^t(c_{1,(000,\hat{s},s)} + c_{1,(010,\hat{s},s)}) + \lambda_{1,1}^t c_{1,(100,\hat{s},s)} \\
&\quad + \lambda_{2,0}^t(c_{2,(000,\hat{s},s)} + c_{2,(100,\hat{s},s)}) + \lambda_{2,1}^t c_{2,(010,\hat{s},s)} \\
&\quad + \lambda_{3,0}^t \mu_{2,1,3}^{(\hat{s},s)} + \sum_{i=4}^{k+2} \lambda_{i,\hat{s}_{i-3}}^t \mu_{2,1,i}^{(\hat{s},s)} + \sum_{i=k+3}^{n} \lambda_{i,s_{i-k-2}}^t \mu_{2,1,i}^{(\hat{s},s)} = 0
\end{aligned}$$

The multiplies of the λ's in this equation form a codeword of an $(n+2, k+2 = d+1)$ Reed-Solomon code. Node 2 collects $\mu_{2,1,i}^{(\hat{s},s)}$ for all $i \in \{3,\ldots,k+2\}$ and it already knows $(c_{2,(000,\hat{s},s)} + c_{2,(100,\hat{s},s)}), c_{2,(010,\hat{s},s)}$, and so it can recover $(c_{1,(000,\hat{s},s)} + c_{1,(010,\hat{s},s)})$ and $c_{1,(100,\hat{s},s)}$. Similarly, it can be shown that with $\mu_{2,2,i}^{(\hat{s},s)}$ for all $i \in \{3,\ldots,k+2\}$, node 2 can recover $(c_{1,(001,\hat{s},s)} + c_{1,(011,\hat{s},s)})$ and $c_{1,(111,\hat{s},s)}$.

Node 3, using $\mu_{3,1,i}^{(\hat{s},s)}$ and $\mu_{3,2,i}^{(\hat{s},s)}$ for all $i \in \{2,4,5\ldots,k+2\}$, can recover $(c_{1,(100,\hat{s},s)} + c_{1,(101,\hat{s},s)})$ and $c_{1,(000,\hat{s},s)}$ and $(c_{1,(110,\hat{s},s)} + c_{1,(111,\hat{s},s)})$ and $c_{1,(011,\hat{s},s)}$. Nodes 2 and 3 send these recovered linear

combinations to the failed node, which can recover all $c_{1,(\tilde{s},\hat{s},s)}$ for $\tilde{s} \in \{0,1\}^3$. Finally this is done for all $\hat{s}$ and s, and this recovers the entire of node 1.

Communication Complexity: Each helper node $i \in \{4, 5, \ldots, k+2\}$ sends two symbols to node 2 and 2 symbols to node 3 for each fixed $\hat{s}, s$. Hence they transmit 4 symbols each resulting in a total transmission of $4(k-1)$ for each fixed $\hat{s}, s$. Similarly node 3 transmits 2 symbols to node 2 and node 2 transmits 2 symbols to node 3 for each fixed $\hat{s}, s$. Finally node 2 and node 3 total transmit 8 symbols to node 1 for each fixed $\hat{s}, s$. This is repeated for every possible $\hat{s}$ and s. Hence the total communication complexity is

$$\begin{aligned} B &= (4(k-1)+4+8) \cdot 2^{k-1} \cdot 2^{n-k-2} \\ &= 2^{n-1}k + 2^n = \beta(d-1) + l = (d+1)\beta \end{aligned}$$

(since $l = 2\beta$). This matches the communication complexity of repair attainable with an IP protocol of Sec. 2.5.2.

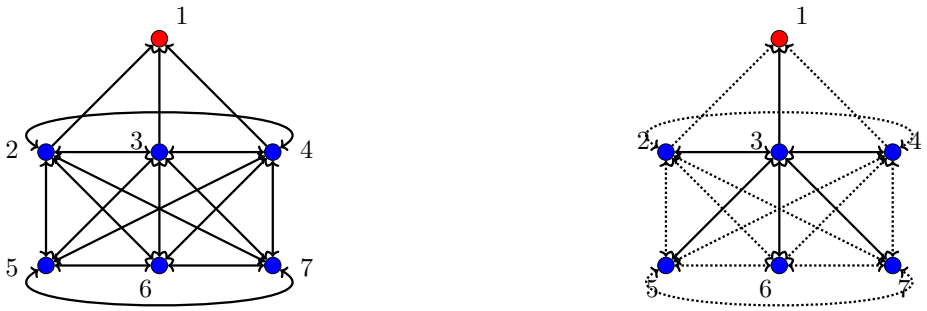
□

2.5.3 Can the repair bandwidth be lower than the IP protocol?

So far we have not identified cases in which communication among the helper nodes reduces the complexity of repair compared to the IP protocol. That this may be possible is demonstrated in the next numerical example in which the value of the linear program is below the repair bandwidth of the IP scheme. Note that we still stop short of constructing an actual node repair scheme that would have this value of the communication complexity.

Consider the graph $G_{f,D}$ in Fig. 2.4a with three direct neighbors of the failed node, and let $d = 6, k = 5$. We assume that $v_f = 1$, and it is directly connected to helper nodes 2, 3 and 4. The six helper

nodes form a complete graph K_6. The IP technique can achieve the complexity of 7 units by transmitting along the spanning tree shown in Fig. 2.4b and performing IP (combining the data) at node 3.



(a) The repair graph

(b) Repair using IP

Figure 2.4: An example with possibly smaller complexity than IP

To define the LP problem we construct a directed graph $\bar{G}_{f,D}$ as explained in the beginning of this section. The linear program of Prop. 2.5.1 in this case has value 6.75 and the assignments of variables are: the primal program

$$X^*_{(u,v)} = \begin{cases} 1 & \text{if } u \in \{2,3,4\}, v = 1, \\ 0.5 & \text{if } u, v \in \{5,6,7\}, u \neq v, \\ 0.25 & \text{if } u \in \{5,6,7\}, v \in \{2,3,4\}, \\ 0 & \text{otherwise;} \end{cases}$$

the dual program:

$$Y^*_S = \begin{cases} 0.125 & \text{if } |S| = 2, S \subset \{2,3,4\}, \\ 0.25 & \text{if } |S| = 2, S \not\subset \{2,3,4\}, \\ 0 & \text{otherwise} \end{cases}$$

Many more similar examples can be constructed for small-size graphs.

2.6 Node repair on random graphs

In this section we analyze the distributed repair procedure in the case when the underlying graph $G(V, E)$ is sampled from the $\mathcal{G}_{n,p}$ ensemble, where $0 < p < 1$. We denote such a random element from the ensemble as $\mathbb{G}_{n,p}$. As before, we assume that the coordinates $C_1, \ldots, C_n$ of a codeword of an (n, k, d) MSR code are placed on the vertices $v_1, \ldots, v_n$. The main question that we address is finding relations between the parameters p, n, k, d such that graph-based repair of the failed node with high probability results in lower repair bandwidth than the AF strategy. Throughout this section we assume that each helper node provides one field symbol for the repair of the (single) failed node.

We will assume that $p \gg \frac{\log n}{n}$ because if $\mathbb{G}_{n,p}$ is not connected, then with positive probability the node v_f is isolated, and repair is not possible (the notation $f(n) \gg g(n)$ means that $g(n) = o(f(n))$). Furthermore, $\mathbb{P}_{\mathcal{G}_{n,p}}(\deg(v_f) \geq d) = \sum_{i=d}^{n} \binom{n}{i} p^i (1-p)^{n-i}$, which goes to zero for $n \to \infty$ if $d \gg np$. Thus, overall this is the parameter regime that may make the graph-based repair (possible and) advantageous over the agnostic AF repair procedure.

Throughout we will assume that $d = \Theta(n)$. For simplicity (without loss of generality) we also assume that each helper node provides only one symbol of $\mathbb{F}$ for the repair of the failed node.

We will use the following two results regarding the random Erdös-Rényi graphs (below $\mathbb{P} = \mathbb{P}_{\mathcal{G}_{n,p}}$).

Lemma 2.6.1 ([8], p. 50; [26], Sec.7.1)**.** *(i) If* $p^2 n - 2\log n \to \infty$, *and* $n^2(1-p) \to \infty$, *then*

$$\mathbb{P}(\mathsf{diam}(\mathbb{G}_{n,p}) = 2) \to 1.$$

(ii) Suppose that the functions $t = t(n) \geq 3$ *and* $0 < p = p(n) < 1$ *satisfy*

$$(\log n)/t - 3\log\log n \to \infty, \quad p^t n^{t-1} - 2\log n \to \infty,$$

$$p^{t-1} n^{t-2} - 2\log n \to -\infty,$$

then $\mathbb{P}(\mathsf{diam}(\mathbb{G}_{n,p}) = t) \to 1$.

Lemma 2.6.2 ([14], Lemma 3)**.** *Suppose that* $p \geq \frac{\log n}{n}$. *For any* $\epsilon > 0$ *and all* $i = 1, \ldots, \lfloor \log n \rfloor$

$$\mathbb{P}(|\Gamma_i(x)| \leq (1+\epsilon)(np)^i) \geq 1 - 1/\log^2 n \tag{2.24}$$

$$\mathbb{P}(|N_i(x)| \leq (1+2\epsilon)(np)^i) \geq 1 - 1/\log^2 n. \tag{2.25}$$

2.6.1 Repair threshold

Let t be a fixed integer. We say that t*-layer repair* of the failed node v *is possible* if

$$\mathbb{P}(|N_t(v)| \geq d) \to 1 \text{ as } n \to \infty$$

and call the minimum t for which this holds the *threshold depth* for repair. Note that such a t is a function of n and p. The next proposition establishes a threshold for t-layer repair in terms of p.

Proposition 2.6.3. *If*

$$(np)^{t-1} = o(n), \quad p^t n^{t-1} - 2\log n \to \infty, \tag{2.26}$$

then t *is the threshold depth for repair.*

Proof. To show that t-layer repair is possible, we observe that from Lemma 2.6.1, $\mathbb{P}(\mathsf{diam}(\mathbb{G}_{n,p}) = t) \to 1$ for all $t \geq 2$. This implies that for any failed node v, all the other nodes in the graph are reachable in at most t steps, and in particular, $|N_t(v)| = n > d$. To show that t is the smallest radius that supports repair, observe that by (2.25) for any $\epsilon > 0$

$$\mathbb{P}(|N_{t-1}(v)| \leq (1+2\epsilon)(np)^{t-1}) \to 1. \tag{2.27}$$

Since d is a linear function of n, we have the inclusion

$$\{|N_{t-1}(v)| \geq d\} \subset \{|N_{t-1}(v)|/n > 0\} \quad (n \to \infty).$$

Together with (2.27) this implies that $\mathbb{P}(|N_{t-1}(v)|/n \geq \gamma) \to 0$ for any $\gamma > 0$. □

Remark: Given t, the conditions (2.26) are satisfied if

$$n^{-(t-1)/t} g(n) \ll p(n) \ll n^{-(t-2)/(t-1)}, \tag{2.28}$$

where $g(n) \gg (2 \log n)^{1/t}$. Rephrasing Prop. 2.6.3, we could say that for a given repair depth t the probability $p(n)$ that satisfies conditions (2.28) is a threshold for repair of depth t in the ensemble $\mathcal{G}_{n,p}$.

2.6.2 Repair bandwidth

In this section we estimate the communication complexity of node recovery on a random graph. Throughout this section we assume that t is the threshold for repair, i.e., conditions (2.26) hold for t, n, and p. We also assume $l = d - k + 1$

Proposition 2.6.4. *The repair bandwidth β_{AF} satisfies*

$$\mathbb{P}(\beta_{\mathrm{AF}} \geq td - o(n)) \to 1$$

where t is the threshold for repair as given by (2.26).

Proof. Rewriting the expression for β_{AF} in (2.9), we obtain

$$\beta_{\mathrm{AF}} = td - \sum_{i=1}^{t-1} (t-i) |\Gamma_i(f)|.$$

Let $E_i = \{|\Gamma_i(f)| \leq (1+\epsilon)(np)^i\}$ and notice that $E := \cap_{i=1}^{t-1} E_i \subseteq \{\beta_{\text{AF}} \geq (td - o(n))\}$. From Lemma 2.6.2 we know that $\mathbb{P}(E_i^c) \leq 1/\log^2 n$ for all i, and thus

$$Pr(\cup_{i=1}^{t-1} E_i^c) \leq \sum_{i=1}^{t-1} Pr(E_i^c) \leq t/\log^2 n.$$

Finally, $\mathbb{P}(\beta_{\text{AF}} \geq td - o(n)) \geq Pr(E) \geq 1 - \frac{t}{\log^2 n} \to 1$. □

This proposition implies that for large n, most of the helper nodes are at distance t from the failed node. Note that, assuming (2.26), Lemma 2.6.2 along with Lemma 8 in [14] imply that the size of the neighborhood $\Gamma_t(v)$ with high probability grows as $c(np)^t$ for some constant $c < 1$. This provides an intuitive explanation of the claim of Prop. 2.6.4 for $d = \Theta(n)$ and $(np)^{t-1} = o(n)$, implying that the AF repair strategy results in a t-fold increase of repair bandwidth compared to full connectivity.

The next proposition gives further insights into the relationship between β_{AF} and t.

Proposition 2.6.5. *Let $d = \delta n, 0 < \delta < 1$, let $\kappa(n)$ be a function of n such that $\underline{c} \leq \kappa(n)/n \leq \bar{c}$ starting with some n and let t be the threshold for repair as given by (2.26). We have*

$$\mathbb{P}(\beta_{\text{AF}} \leq \kappa(n)) \to \begin{cases} 0 & \text{if } t > \bar{c}/\delta \\ 1 & \text{if } t \leq \underline{c}/\delta. \end{cases}$$

Proof. To prove the first claim in the proposition, compute

$$\begin{aligned} \mathbb{P}(\beta_{\text{AF}} \leq \kappa(n)) &\leq \mathbb{P}(\beta_{\text{AF}} \leq \bar{c}n) \\ &= \mathbb{P}\Big(td - \sum_{i=1}^{t-1}(t-i)|\Gamma_i(v_f)| \leq \bar{c}n\Big) \\ &= \mathbb{P}\Big(\sum_{i=1}^{t-1}(t-i)|\Gamma_i(v_f)| \geq (t\delta - \bar{c})n\Big) \end{aligned}$$

$$\leq \mathbb{P}\Big(\sum_{i=1}^{t-1}(t-i)|\Gamma_i(v_f)| \geq (t\delta - \bar{c})n \;\Big|\; E\Big)\mathbb{P}(E) + \mathbb{P}(E^c),$$

where the event E is defined above in Prop. 2.6.4. Conditional on E we have $\sum_{i=1}^{t-1}(t-i)(1+\epsilon)(np)^i = \Theta((np)^{t-1})$, and (2.26) implies that the first term $\to 0$ w.h.p. To complete the proof notice that $\mathbb{P}(E^c) \leq t/\log^2 n \to 0$.

For the second claim, observe that

$$\mathbb{P}(\beta_{\text{AF}} \leq \kappa(n)) \geq \mathbb{P}(\beta_{\text{AF}} \leq \underline{c}n) \geq \mathbb{P}(\beta_{\text{AF}} \leq td) = 1$$

concluding the proof. □

Now let us show that the graph-based repair as defined in (2.15) or (2.19) with high probability has smaller repair bandwidth. Below we denote $\chi(n) = d - k$.

Theorem 2.6.6. *Let t be the threshold given in Prop. 2.6.3. Let $\chi(n)$ be such that $\chi(n)n^{s-1}p^s \to 0$ where $s \leq t-1$ is the largest integer for which this condition holds. Then $\mathbb{P}(\beta_{\text{IP}} \leq (t-s)d + o(n)) \to 1$.*

Remark: Since $pn \to \infty$, it is easy to check that the assumptions of the theorem are non-vacuous, i.e., the largest s satisfying the condition exists and is well defined.

Proof. Let T_f be the repair tree with the root v_f. By assumption, the distance from the root to the leaves is t, and we will assume that the helper nodes in $\Gamma_i(v_f), i = t, t-1, \ldots, s+1$ simply relay their information along the edges, while the nodes in $N_s(v_f)$ transmit $l = d - k + 1$ symbols given by a linear combination of the form given in (2.15).

Then, for the failed node v_f, we have

$$\beta_{\text{IP}} = (t-s)(d - |N_{t-1}(v_f)|) + \sum_{i=1}^{t-s-1}(t-s-i)|\Gamma_{t-i}(v_f)| + (d-k+1)\sum_{i=1}^{s}|\Gamma_i(v_f)|$$

$$= (t-s)d + \sum_{i=1}^{s} |\Gamma_i(v_f)|(d-k+1-(t-s)) - \sum_{i=s+1}^{t-1} |\Gamma_i(v_f)|(t-i)$$
$$\leq (t-s)d + \sum_{i=1}^{s} |\Gamma_i(v_f)|(d-k+1-t+s)).$$

Proceeding similarly to the proof of Prop. 2.6.4, we obtain

$$\mathbb{P}\Big(\beta_{\text{IP}} \leq (t-s)d + \sum_{i=1}^{s}(1+\epsilon)(np)^i(\chi(n)+1-t+s)\Big) \geq 1 - s/\log^2 n \to 1.$$

Now using the assumption $\chi(n)(np)^s = o(n)$ finishes the proof. □

To conclude, we have shown a strict separation between the typical communication cost of node recovery using the IP repair procedure and the graph-agnostic AF protocol when the number of helpers d is only slightly more than k. Note the following simple corollary:

Corollary 2.6.7. *Let t be the threshold given in Prop. 2.6.3. For $\chi(n) = O(1)$,*

$$\mathbb{P}(\beta_{\text{IP}} \leq d + o(n)) \to 1.$$

Proof. By (2.26), for $\chi(n) = O(1)$, the condition $\chi(n)n^{s-1}p^s \to 0$ is satisfied for $s = t-1$. □

Corollary 2.6.7 supports the following intuition. Since $\chi(n)$ is a constant, nodes in all layers but the last can do a very high amount of compression and hence the contribution of those layers to the total communication complexity becomes insignificant; the complexity is primarily determined by the number of helper nodes in the last layer which is approximately d. This implies that even in random graphs where we do not have direct connectivity among all the helper nodes and the failed node, it is possible to bring down the communication complexity to the same order as that of the case of direct connectivity using IP.

The above theorem and corollary suggest that the communication complexity is primarily controlled

by the two parameters $p(n)$ and $\chi(n)$. One can ask the question, for what values of these parameters, does the complexity become significantly higher than that of the complexity of repair under full connectivity, i.e., d. In other words, we wish to study the behavior of $\beta^* - d$ where β^* is the minimum complexity over all possible repair schemes[3]. In this regard, Corollary 2.6.7 says that for $\chi(n) = O(1)$, we have $\beta^* - d = o(n)$ with high probability. We will now show that for sparse graphs with high probability the repair becomes significantly more complex than sending d symbols. The following theorem quantifies this claim. Its proof relies on Lemma 2.2.1 together with another lemma from [14].

Lemma 2.6.8 ([14], Lemma 2)**.** *Suppose that $p > \frac{c \log n}{n}$ for a constant $c \leq 2$. Then with probability at least $1 - o(\frac{1}{n})$, we have for all $1 \leq i \leq n$*

$$|\Gamma_i(x)| \leq \frac{9}{c}(np)^i \tag{2.29}$$

$$|N_i(x)| \leq \frac{10}{c}(np)^i. \tag{2.30}$$

Theorem 2.6.9. *For $p(n) = o(\frac{\chi(n)}{n})$, we have $\mathbb{P}(\beta^* - d = \Theta(n)) \to 1$.*

Proof. Given $p(n)$, let t be the threshold for repair as given in Prop. 2.6.3. Clearly, any helper node $v \in N_t(v_f)$ needs to transmit at least one unit of information, so $\beta^* \geq d - |N_{t-1}(v_f)|$. Let $F_n := \{\Gamma_{t-1}(v_f) \leq \frac{9}{c}(np)^{t-1}\}$. Now consider a node $v \in N_{t-1}(v_f)$. From Lemma 2.6.8, the immediate neighborhood of this node satisfies

$$\mathbb{P}\Big(|\Gamma_1(v)| \leq \frac{9}{c}np\Big) \geq 1 - o\Big(\frac{1}{n}\Big)$$

for some constant $c \leq 2$. Let $D(v)$ be the *immediate* descendants of node v in the repair tree. For every $\delta > 0$, there exists an n_1 such that $|D(v)| \leq |\Gamma_1(v)| \leq \frac{9}{c}np$ with probability at least $1 - \frac{\delta}{n}$ for every $n \geq n_1$. Further, since $np = o(\chi(n))$, for every $\epsilon > 0$, there exists an n_2 such that $np \leq \epsilon\chi(n)$ for every

[3]Our arguments rely on the information-theoretic lower bounds, so they indeed apply to all possible repair schemes.

$n \geq n_2$. Combining, these two statements, we claim that the event $E_{v,\epsilon,n} := \{|D(v)| \leq \epsilon\chi(n)\}$ satisfies

$$\mathbb{P}(E^c_{v,\epsilon,n}) \leq \delta/n \text{ for all } \epsilon, \delta > 0, n \geq \max(n_1, n_2).$$

Since this is true for all $v \in \Gamma_{t-1}(v_f)$, we have

$$\begin{aligned}\mathbb{P}(\cup_{v\in\Gamma_{t-1}(v_f)}E^c_{v,\epsilon,n}) &= \mathbb{P}(\cup_{v\in\Gamma_{t-1}(v_f)}E^c_{v,\epsilon,n}|F_n)\mathbb{P}(F_n) + \mathbb{P}(\cup_{v\in\Gamma_{t-1}(v_f)}E^c_{v,\epsilon,n}|F^c_n)\mathbb{P}(F^c_n)\\ &\leq \frac{9\delta}{c}\frac{(np)^{t-1}}{n} + o\Big(\frac{1}{n}\Big) \to 0,\end{aligned}$$

where the last step follows because by the definition of the threshold t, $(np)^{t-1} = o(n)$. This implies that $\mathbb{P}(\cap_{v\in\Gamma_{t-1}(v_f)}E_{v,\epsilon,n}) \to 1$ for all $\epsilon > 0$. Now by Lemma 2.2.1, for $|D(v)| \leq \chi(n)$ the outflow of communication from the set $D(v) \cup \{v\}$ has to be at least $|D(v)| + 1$ and by the above analysis this is true for all $v \in \Gamma_{t-1}(v_f)$ with high probability. This implies that with high probability

$$\beta^* \geq d - |N_{t-1}(v_f)| + \sum_{v\in\Gamma_{t-1}(v_f)}(|D(v)| + 1) \geq d - |N_{t-1}(v_f)| + \Gamma_t(v_f) = 2(d - |N_{t-1}(v_f)|).$$

Finally, noting that $\mathbb{P}(|N_{t-1}(v_f)| = o(n)) \to 1$ gives the desired claim. □

This theorem gives a sufficient condition for the separation of complexity of repair on a complete graph and a sparse random graph.

2.6.3 Random regular graphs

In this section we briefly address node repair on random regular graphs. We single out this ensemble from a multitude of other options because it is conceivable that the architecture of the storage system places the same number of servers in close proximity to any single server, and this is modeled by a regular graph. Let $\mathcal{G}_{n,r}$ be the set of all r-regular n-vertex graphs with a uniform distribution on it. We denote a random

element from this ensemble by $\mathbb{G}_{n,r}$. Assume again that the data is encoded with an (n, k, d, l) MSR code, and the coordinates $C_1, \ldots, C_n$ of the codeword are placed on the vertices $v_1, \ldots, v_n$.

We will derive conditions on the parameters $k = k(n), d = d(n)$ and $r = r(n)$ such that as $n \to \infty$, with high probability the graph-based repair process is advantageous over the AF strategy. We again assume that $d = \Theta(n)$. Denote by $\mathcal{G}_{n,m}$ the ensemble of graphs with n vertices and m edges and let $\mathbb{G}_{n,m}$ be a random graph sampled from it.

For the purposes of node repair we need the graph to be connected. In [7], Bollobás showed that $\mathbb{G}_{n,r}$ is r-connected with high probability.

Recall that a property of graphs is called *increasing* if it is inherited from a subgraph to any graph that contains it. The following equivalence between properties of $\mathbb{G}_{n,m}$ and $\mathbb{G}_{n,r}$ will be used below.

Lemma 2.6.10. *([26], Corollary 10.11) Let $\mathcal{L}$ be an increasing property of graphs such that $\mathbb{G}_{n,m}$ satisfies $\mathcal{L}$ with high probability for some $m = m(n)$, where $n \log n \ll m \ll n^2$. Then $\mathbb{G}_{n,r}$ satisfies $\mathcal{L}$ with high probability for $r = r(n) \sim \frac{2m}{n}$.*

The following proposition is a counterpart to Prop. 2.6.3 for random regular graphs. Here the definition of the threshold depth of repair is the same as in Sec. 2.6.1.

Proposition 2.6.11. *Let $d = \delta n, 0 < \delta < 1$ and let t be a fixed integer. Then t is the threshold depth for repair if*

$$r^{t-1} = o(n), \quad \frac{r^t}{n} - 2\log n \to \infty. \tag{2.31}$$

Proof. For finite t, $|N_{t-1}(v)| \leq \sum_{i=1}^{t-1} r^i = \Theta(r^{t-1}) = o(n) \ll d$ and so $(t-1)$-layer repair is not possible.

For the other direction, let $r(n)$ satisfy relations (2.31). Take $p(n) = \frac{r(n)}{n}$, then by Prop. 2.6.3 $\mathbb{G}_{n,p}$ satisfies t-layer repair with high probability. Recall a basic fact that the ensembles $\mathcal{G}_{n,p}$ and $\mathcal{G}_{n,m}$ are

equivalent for all monotone properties (i.e., graphs sampled from them either both have the property w.h.p. or they both do not). Therefore, the graph $\mathbb{G}_{n,m}$ with $m(n) \sim \frac{n^2}{2}p(n) = \frac{nr(n)}{2}$ affords t-layer repair with high probability. Now, satisfying t-layer repair is a monotone increasing graph property and so by Lemma 2.6.10 we have that $\mathbb{G}_{n,r}$ affords t-layer repair with high probability. □

With the threshold conditions established, counterparts of Prop. 2.6.4, 2.6.5 and Theorem 2.6.6 can be proved for $\mathbb{G}_{n,r}$ by simply replacing np with r and proceeding along similar arguments (which are in fact simpler because the neighborhood sizes of a vertex afford uniform bounds). In the following statements, given without proofs, t is the repair threshold as defined in (2.31).

Proposition 2.6.12. *The repair bandwidth β_{AF} satisfies*

$$\mathbb{P}(\beta_{\text{AF}} \geq td - o(n)) \to 1.$$

Proposition 2.6.13. *Let $d = \delta n, 0 < \delta < 1$ and let $\kappa(n)$ be a function of n such that $\underline{c} \leq \kappa(n)/n \leq \bar{c}$ starting with some n. Then*

$$\mathbb{P}(\beta_{\text{AF}} \leq \kappa(n)) \to \begin{cases} 0 & \text{if } t > \bar{c}/\delta \\ 1 & \text{if } t \leq \underline{c}/\delta. \end{cases}$$

for $t > \bar{c}/\delta$ we have $\mathbb{P}(\beta_{\text{AF}} \leq \kappa(n)) \to 0$.

Theorem 2.6.14. *Let $d - k = \chi(n)$ be a function of n such that $\frac{\chi(n)r(n)^s}{n} \to 0$ where $s \leq t-1$ is the largest integer for which this condition holds. Then $\mathbb{P}(\beta_{\text{IP}} \leq (t-s)d + o(n)) \to 1$.*

Similarly to the case of $\mathcal{G}_{n,p}$, this shows a strict separation between the typical communication cost of node recovery using the IP repair procedure and the graph-agnostic AF protocol under a certain assumption on $\chi(n)$. A theorem that parallels Theorem 2.6.9 can be also easily established (note that the bounds of the form given in Lemma 2.6.8 for regular graphs come for free).

2.7 Concluding remarks

In this chapter we posed and advanced the problem of erasure correction (node repair) when the elements of encoded information are placed on the nodes of a graph, adopting the total amount of communication for repair as the figure of merit in the analysis. The main difference of this problem from the standard setting of regenerating codes stems from the fact that most helpers are not directly connected to the failed node, and the information transmitted by them can be processed by the intermediate nodes or combined with the contents of these nodes. We showed that the intermediate processing scheme can be implemented by linear MSR codes, attaining the general lower bounds on complexity derived in the chapter. These results were also extended to the case of multiple failed nodes. We also established a framework for the analysis of repair schemes when the helpers communicate among themselves before contributing data toward the repair task, and gave simple examples when the arising complexity bounds are attained with equality. Finally, we studied the repair problem when the underlying graph is random, establishing bounds on the edge probability under which the intermediate processing scheme provides complexity savings compared to simple relaying.

While the analysis in this chapter was limited to the MSR point, a natural question to ask is whether the idea of IP extends to other regenerating code families at interior points of the trade-off curve. We already remarked that the case for the other extreme point, i.e., MBR point is trivial in the sense that the AF strategy is optimal. However, there are several other code families which have been constructed to support exact repair for parameters corresponding to the interior points of the trade-off curve. In the following chapter we explore this question in further detail.

The results of this chapter appear in [50, 52].

Chapter 3: Extension to Codes beyond the MSR point

3.1 Introduction

While the MSR case is the most widely studied, several interior-point code families are known in the literature, among them layered [67] and determinant codes [22, 24], and a recent construction of [19], called Moulin codes by its authors. In the previous chapter, we focused on the repair problem for MSR codes, proving a lower bound on the communication complexity (bandwidth) of node repair on graphs. We have pointed out that IP repair is possible for any linear MSR code, although the details of the procedure depend on the family and are not immediate to work out. It has later become clear that IP repair is driven less by the MSR property and more by the linearity of the constructions. In this part of our study, we present IP repair procedures explicitly for several families of regenerating codes constructed in earlier literature.

3.1.1 Review of Codes at the Interior Points

Recall from Chapter 1, that an $[n, k, d, l, \beta, M]$ regenerating code satisfies the following cutset bound:

$$M \leq \sum_{i=0}^{k-1} \min\{l, (d-i+1)\beta\}. \tag{3.1}$$

As mentioned before, different pairs of (l, β) satisfying the above with equality give rise to different optimal points on the storage-bandwidth tradeoff curve (for functional repair). Due to the added complexity

of keeping track of the decoding and repair strategies of functional repair, codes for exact repair are more lucrative for practical implementation purposes. Clearly, the upper bound on the filesize of Eq. (3.1), derived with the assumption of functional repair in [16], also serves as an outer bound for exact repair and hence the tradeoff curve for exact repair lies somehwere above that of functional repair. It was first showed in [68] and later in [78] that there exists a non-zero gap between the two curves and the two points where they coincide are the two extremal points, namely the MSR and MBR points. Afterwards, several attempts were made to quantify this gap by proposing better outer bounds for exact repair; see for example [18,38,49,65] to name a few. At the same time, several works focused on constructing new exact repair code families to approach the infeasibility bound from the other side, see [19, 22, 23] for example. This has resulted in several explicit regenerating code families whose parameters are between the two extreme points. These are collectively referred to as *interior-point* codes and in this chapter we shall study the performance of these codes under our graph-constrained repair setting.

We continue with the system model defined in Sec. 2.1, with a finite field $F = \mathbb{F}_q$, and with a code $\mathcal{C} \subset F^{nl}$ whose codewords $(C_i,\ i = 1, \ldots, n)$ are represented by $l \times n$ matrices over F and each coordinate (a vector in F^l) stored in a single storage node. The graph $G(V, E)$ with $|V| = n$, models the connectivity constraints of the network where each node has direct access only to its immediate neighbors in G. In the event of the failure of node $v_f \in [n]$, $D \subset V \backslash \{v_f\}, |D| = d$ be the set of helper nodes in the graph G that are the closest to v_f in terms of graph distance and information is communicated to v_f over the edges in $E_{f,D}$. Since, we are not restricting ourselves to the MSR point anymore, i.e., the code $\mathcal{C}$ is a general $[n, k, d, l, \beta, M]$ regenerating code, each helper node now contributes β symbols each towards the repair of the failed node, where β can be any integer between $\frac{l}{(d-k+1)}$ and l, and depends on the operating point of the trade-off curve. As noted before, the simple relaying technique (AF technique) can be wasteful in high-depth repair graphs since the same data gets transmitted multiple times. This gives rise to the problem of attaining savings by processing the information in the intermediate nodes relying on

the IP approach, an idea that has already been explored for MSR codes in Chapter 2.

3.1.2 Overview of the results

We summarize the contributions of this chapter as follows:

- We prove a general lower bound on the repair bandwidth of any regenerating code which extends the result from Sec. 2.2. This sets up a benchmark for IP repair in later sections.

- In Sec. 3.4.1,3.4.3, we design IP repair for two families of MSR codes, namely the product-matrix (PM) codes and their generalization introduced recently by Duursma and Wang [20]. In doing so, we shift the perspective, viewing them as *evaluation codes*, i.e., codes whose encoding can be phrased as evaluation of a linear functional written in a convenient algebraic form. For the PM codes, the results are already known from Sec. 2.3, but here we write the procedure in a different, more compact and transparent form.

- Then in Sec. 3.4.5 we turn to interior point codes of [19], which also fall in the family of evaluation codes. For this family, we explicitly formulate IP repair and estimate the repair bandwidth. Specializing the bound of Sec. 3.2 to these classes of codes, we compare the efficiency of IP technique and observe the gap between achievability and the bound. In Sec. 3.5 we consider the family of determinant codes [23], noting that their construction makes them a natural candidate for IP repair on graphs.

- Finally, in Sec. 3.6 we study the other basic functionality of these storage codes, namely, data retrieval. We note that while the task of data retrieval from an MSR code defined on an arbitrary graph becomes trivial due to their MDS nature, the situation changes when we lift the MSR constraint. We estimate the minimum number of symbols required for data retrieval on graphs, and design a retrieval procedure with optimal complexity at the MBR point.

For each of the code families discussed, we observe that IP repair results in communication savings compared to the AF procedure. A general statement characterizing the savings depends on the properties of the graph G and on the choice of the helper set D in relation to the failed vertex. It is possible to write it for some special graph families such as regular trees and other simple classes, as was done in Chapter 2, Sec. 2.3. These statements were formulated for simple graphs such as the regular tree or a path, and they were also extended to some ensembles of random graphs.

3.2 Bounds on the Repair Bandwidth

In this section we derive a lower bound on the minimum required transmission for a set of helper nodes for repair of the failed node. The proof technique is similar to that of Lemma 2.2.1, with additional steps needed to account for the lack of the MDS property of the code. As previously, let the information stored at the vertices be described by random variables $W_i, i \in [n]$ that have some joint distribution on $(F^l)^n$ and satisfy $H(W_i) = l$ for all i, where $H(\cdot)$ is the entropy. For a subset $A \subset V$ we write $W_A = \{W_i, i \in A\}$. Denote by S_i^f the information provided to v_f by the ith helper node in the traditional fully connected repair scheme, and let $S_D^f = \{S_i^f, i \in D\}$. By definition we have

$$H(S_i^f) = \beta, \quad H(S_i^f|W_i) = 0, \quad H(W_f|S_D^f) = 0.$$

We also assume that $H(\mathcal{F}|W_B) = 0$ for any $B \subset [n], |B| = k$, which supports the data retrieval property. Note, however, that unlike Sec. 2.2, the random variables $\{W_i : i \in [n]\}$ do not necessarily satisfy the MDS property anymore, as we are not restricting ourselves to the MSR point. Hence, we require the following general result from [68]:

Lemma 3.2.1. *For any $A \subset [n], |A| \leq d$ and $i \notin A$*

$$H(W_i|W_A) \leq \min(l, (d - |A|)\beta).$$

The above information theoretic formulation of the repair problem, although proposed for the exact repair problem, is also valid for functional repair with a slight modification: for functional repair we only focus on the repair cycle of a single node with the help of d helpers whose repair has already been completed (See Section VI of [68]). Hence the random variable W_f now denotes the data to be reconstructed at the failed node (which can be different from the data lost prior to this moment.). The next lemma forms a simple extension of Lemma II.1 in Chapter 2, generalizing it to all functional and exact regenerating codes at all points of the trade-off curve.

Lemma 3.2.2. *Let $v_f, f \in [n]$ be the failed node. For a subset of the helper nodes $E \subset D$ let R_E^f be a function of S_E^f such that*

$$H(W_f|R_E^f, S_{D\setminus E}^f) = 0. \tag{3.2}$$

If $|E| \geq d - k + 1$, then

$$H(R_E^f) \geq M - \sum_{i=1}^{k-1} \min\{l, (d - i + 1)\beta\}.$$

In particular, at the MSR point we have $H(R_E^f) \geq l$.

Proof. By the assumption (3.2), given the contents of all the nodes in $D\setminus E$, the information contained in R_E^f is sufficient to repair v_f, i.e.,

$$H(W_f|R_E^f, W_{D\setminus E}) = 0. \tag{3.3}$$

We have $|D\setminus E| \leq k - 1$. Consider a set $A \subset E$ with $|A| = k - 1 - |D\setminus E|$. Now, by (3.3)

$$H(R_E^f, W_{D\setminus E}, W_A) = H(R_E^f, W_{D\setminus E}, W_f, W_A) = M, \tag{3.4}$$

where the first equality in (3.4) follows from (3.3) and the chain rule, and the second follows from reconstruction property because $|D\backslash E| + |A| + 1 = k$. Next observe that

$$H(R_E^f, W_{D\backslash E}, W_A) \leq H(R_E^f) + H(W_{D\backslash E}, W_A),$$

and so

$$\begin{aligned} H(R_E^f) &\geq M - H(W_{D\backslash E}, W_A) \\ &\geq M - \sum_{i=1}^{k-1} \min\{l, (d-i+1)\beta\}, \end{aligned}$$

where the last inequality follows from Lemma 3.2.1. □

Corollary 3.2.3. *For an optimal functional repair code that meets the cut-set bound of Eq. (3.1) with equality, we have* $H(R_E^f) \geq (d-k+1)\beta$.

Note that at the MSR point $(d-k+1)\beta = l$ and we recover Lemma 2.2.1 from Chapter 2. In that work we also showed that $H(R_E^f) = l$ is achievable at the MSR point. At the same time for all other points of the tradeoff curve, $(d-k+1)\beta < l$. Below in this chapter we show that the value $H(R_E^f) = l$ can be achieved by some code families, and hence it might be possible to improve the bound. We note that the constructions presented below do not reach the bound proved in this section, leaving an open question of the optimal repair bandwidth for the IP repair technique and for codes other than the MSR/MBR families.

3.3 IP repair for linear regenerating codes

Referring to the notation of the previous section, our task is to recover the contents of W_f. Denote by $S_h^f \in F^\beta$ the symbols of F provided by h for repair, and suppose that $S_h^f = \mathcal{G}_{hf}(W_h)$, where $\mathcal{G}_{hf} : F^l \to F^\beta$ is a fixed F-linear map determined by the code. If the symbols S_h^f are available to v_f (say, if G is a complete

graph) then the value of W_f is found as

$$W_f = \mathcal{F}_{f,D}(\mathcal{G}_{h_1 f}(W_{h_1}), \mathcal{G}_{h_2 f}(W_{h_2}), \cdots, \mathcal{G}_{h_d f}(W_{h_d}))$$

where $\mathcal{F}_{f,D} : (F^\beta)^d \to F^l$ is another F-linear map. This description, including the linearity assumptions, fits most known constructions of regenerating codes. It implies that there exists a matrix $U_{f,D} \in F^{l \times d\beta}$ such that

$$W_f = U_{f,D} \cdot [(S^f_{h_1})^\intercal \mid (S^f_{h_2})^\intercal \mid \cdots \mid (S^f_{h_d})^\intercal)]^\intercal. \tag{3.5}$$

Writing the matrix $U_{f,D}$ in block form as $[U^f_{h_1} U^f_{h_2} \cdots U^f_{h_d}]$ with each block of dimensions $l \times \beta$, we can rewrite (3.5)

$$W_f = \sum_{h \in D} U^f_h S^f_h. \tag{3.6}$$

Thus, there exist linear maps $I_{hf} : F^\beta \xrightarrow{U^f_h} F^l, h \in D$. Since the matrices U^f_h do not depend on the codeword, the value $U^f_h S^f_h$ can be computed at any part of the network by any node with access to S^f_h. This shows that any set of helper nodes $A \subseteq D$, instead of sending $\{S^f_h : h \in A\}$, which requires $|A|\beta$ transmissions, can pass along $\sum_{h \in A} U^f_h S^f_h$, which requires l transmissions. Starting from the leaf nodes in $G_{f,D}$, the data is moved toward v_f, and as the set A of the already involved helpers increases in size, switching to the latter mode involves savings in the repair bandwidth. The following theorem follows along the lines of our earlier result from Chapter 2 for MSR codes.

Theorem 3.3.1. *Suppose a codeword of an $[n, k, d, l, \beta, M]$ regenerating code is encoded on the vertices of a graph G. For a failed node v_f let D be the set of helper nodes and let $G_{f,D}$ be the corresponding repair graph. Let $\mathcal{T}_{f,D}$ be the set of spanning trees of $G_{f,D}$ rooted at v_f and for any $T \in \mathcal{T}_{f,D}$ let $J(T) = \{v \in V_{f,D} \setminus \{v_f\} : |D(T, v)| \geq d - k + 1\}$ where $D(T, v)$ be the set of descendants of node v in the repair tree T. Then there exists an explicit scheme that has the following communication complexity of*

repair for the node v_f

$$\min_{T \in \mathcal{T}_{f,D}} \left[|J(T)|l + \sum_{v \notin J(T) \cup \{v_f\}} |D(T,v) + 1|\beta \right].$$

3.4 Intermediate Processing for Evaluation Codes

In this section, we show that F-linear regenerating codes support repair on graphs with lower communication complexity compared to the AF strategy. In Sec. 3.4.1 we give an alternative description of node repair using the IP strategy at the MSR point for product-matrix codes and in Sec. 3.4.3 we extend this procedure to their generalization due to Duursma and Wang [20]. In Sec. 3.4.5 we present the IP repair procedure for a family of interior-point evaluation codes.

3.4.1 Product-matrix (PM) codes

As our first goal, we rewrite the IP repair of PM codes originally introduced in Chapter 2, Sec.II.A to fit the evaluation code paradigm. We begin with a brief introduction to the original description of the PM framework. PM codes, constructed in [59], form a family of MSR codes with parameters $[n, k, d = 2(k-1), l = k-1, \beta = 1, M = k(k-1)]$. The data file $\mathcal{F}$ consists of M uniformly chosen symbols from a finite field F. These symbols are organized to form two symmetric matrices S_1, S_2 of order $k-1$, each consisting of $\binom{k}{2}$ independent symbols and hence accounting for a total of M symbols. The encoding matrix Ψ is taken to be an $n \times d$ matrix such that $\Psi = \left[\Phi | \Lambda\Phi\right]$ where Φ is a $n \times (k-1)$ Vandermonde matrix with rows of the form $\phi_i = (1, x_i, x_i^2, \ldots, x_i^{l-1}), i = 1, \ldots, n$ and $\Lambda = \text{Diag}(x_1^l, x_2^l, \ldots, x_n^l)$ is a diagonal matrix where $x_1, \ldots, x_n$ are distinct non-zero elements of F. The encoded message is defined as $C = \Psi(S_1|S_2)^\intercal$ and the l symbols of row i of C are stored in node i. Thus the ith node stores the l-vector $\phi_i S_1 + \lambda_i \phi_i S_2$.

The node repair process goes as follows: assuming that node $f \in [n]$ has failed, and the helper nodes are $D \subseteq [n] \setminus \{f\}, |D| = d$, helper node $i \in D$ sends the symbol of F found as $(\phi_i S_1 + \lambda_i \phi_i S_2)\phi_f^\intercal$.

Since the submatrix Ψ_D formed of the rows of Ψ indexed by D is invertible, node f can calculate $S_1\phi_f^\intercal$ and $S_2\phi_f^\intercal$ from which it can compute its contents as $\phi_f S_1 + \lambda_f \phi_f S_2$.

To phrase this differently, let $s_1(y,z)$ and $s_2(y,z)$ be two symmetric polynomials over F of degree at most $k-2$ in each of the two variables (this means, for instance, that $s_1(y,z) = s_1(z,y)$). Because of the symmetry, the total number of independent coefficients is M, so s_1, s_2 can be used to represent $\mathcal{F}$. Letting $x_1, \ldots, x_n$ be distinct points of F, we let node i store the l coefficients of the polynomial $g^{(i)}(z) = s_1(x_i, z) + x_i^{k-1} s_2(x_i, z)$ for all $i \in [n]$.

Using this description of the codes, the IP repair process of Chapter 2 can be phrased as follows. Let $f \in [n]$ be the failed node, let D be the set of d helpers, and let A be a set of helper nodes of size at least $d-k+1 = k-1$. For $h \in D$ define the polynomial

$$l^{(h)}(z) = \sum_{j=0}^{d-1} l_j^h z^j := \prod_{\substack{i \in D \\ i \neq h}} \frac{z - a_i}{a_h - a_i} \tag{3.7}$$

of degree at most $d-1$. Then the set A transmits the l-dimensional vector

$$\xi(f, A) := \sum_{h \in A} g^{(h)}(a_f) \begin{bmatrix} l_0^h + a_f^{k-1} l_{k-1}^h \\ l_1^h + a_f^{k-1} l_k^h \\ \vdots \\ l_{k-2}^h + a_f^{k-1} l_{2k-3}^h \end{bmatrix}. \tag{3.8}$$

We show that (i), the failed node can recover its value based on the vector $\xi(f, D)$, and (ii), the intermediate nodes can save on the repair bandwidth by processing the received information. To show (i) we prove

Lemma 3.4.1. *The content of the failed node f coincides with the vector $\xi(f, D)$, i.e.,*

$$g^{(f)}(z) = \sum_{i=0}^{l-1} (\xi(f, D))_i \, z^i.$$

Proof. Consider the polynomial $H(z) = s_1(a_f, z) + z^{k-1} s_2(a_f, z)$ and note that $\deg(H) \leq 2k-3 = d-1$. Thus if we write $H(z) = \sum_{j=0}^{d-1} g_j z^j$, then the polynomial $g^{(f)}$ defined above can be written as

$$g^{(f)}(z) = \sum_{j=0}^{k-2} (g_j + a_f^{k-1} g_{k-1+j}) z^j.$$

Rephrasing, the contents of the node f is

$$(g_0 + a_f^{k-1} g_{k-1}, g_1 + a_f^{k-1} g_k, \ldots, g_{k-2} + a_f^{k-1} g_{2k-3})^{\intercal}.$$

At the same time, using (3.7) we can write $H(z)$ in the Lagrange form $H(z) = \sum_{h \in D} g^{(h)}(a_f) l^{(h)}(z)$. The coefficient vector of this polynomial is nothing but $\xi(f, D)$. □

To show part (ii) we note that the polynomials $\{l_h(z)\}_{h \in D}$ do not depend on $\mathcal{F}$ and can be computed at any node in the network. So what we care to receive from the helper nodes are the multipliers $\{g^{(h)}(a_f)\}_{h \in D}$. Hence, for any set of helper nodes with $|A| < d-k+1$, it is gainful to send $\{g^{(h)}(a_f)\}_{h \in A}$ rather than the vector $\xi(f, A)$, since the former requires fewer than l transmissions. At the same time, when $|A| \geq d - k + 1$, we can transmit the vector $\xi(f, A)$ of dimension l, meeting the bound of Lemma 3.2.2 and reproducing the result from Chapter 2.

Using multilinear algebra notation (more on it in the next section), we can rephrase the code description as follows. The encoding is defined as a linear functional

$$\phi \in (F^2 \otimes S^2 F^{k-1})^*,$$

where $S^2 F^{k-1}$ is the second symmetric power (this is another way of saying that the encoding relies on evaluations of symmetric polynomials). Node i stores a restriction of ϕ to $x_i \otimes y_i \otimes F^{k-1}$, where $x_i = [1, a_i^{k-1}], y_i = [1, a_i, \ldots, a_i^{k-2}]$. The contents of the failed node is a vector in the l-dimensional subspace $(x_f \otimes y_f \otimes F^{k-1})^*$, and the IP procedure recovers the coordinates of this vector in stages that correspond

to moving along the repair graph toward the failed node. A general version of this idea underlies the repair procedure in the following sections.

Example 3.4.1. Consider the $[n \geq 7, k = 4, d = 6, l = 3, \beta = 1, M = 12]$ PM MSR code, placed on the graph shown in Fig. 3.1. This graph should be thought of as a subgraph in a large storage network, formed by locating a helper set for the failed node. Suppose that the root node is erased, and the remaining

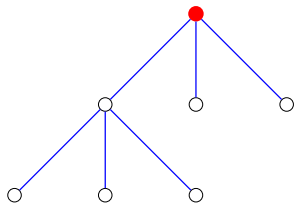


Figure 3.1: The graph for Example 3.4.1

6 nodes form the helper set. Since each of them contributes one symbol, the AF repair procedure requires transmission of 9 field symbols over the edges to complete the repair. In particular, the left most of the three vertices adjacent to v_f sends 4 symbols over the edge connecting it to v_f. At the same time, using the IP procedure described above, this node may only send $l = 3$ symbols, showing that a total of 8 transmissions are sufficient. This shows the bandwidth saving capabilities of the IP procedure.

3.4.2 Linear-algebraic notation

In this section, we introduce elements of notation used below to define code families for which we design IP procedures of node repair.

For a linear space U over F we denote by U^* its dual space; its elements are linear functionals of the form $\phi : U \to F$. The spaces U and U^* have the same dimension and $(U^*)^* \cong U$. A *restriction* of ϕ to a subspace $V \subset U$ is denoted as $\phi \restriction V$.

Let U, V be linear spaces of dimensions m and n, respectively, and let us fix bases $\{\overline{u}_i\}_{i=1}^m$ and $\{\overline{v}_j\}_{j=1}^n$. The tensor product of U and V is a linear space $U \otimes V = \{\sum_{ij} a_{ij} \overline{u}_i \otimes \overline{v}_j, a_{ij} \in F\}$ where

$a_{ij} \in F$ and the tensors $\overline{u}_i \otimes \overline{v}_j$ form a basis in $U \otimes V$ (thus $\dim(U \otimes V) = mn$). By definition, $u \otimes V = \{\sum_j a_j u \otimes \overline{v}_j, a_j \in F\}$ and $u \otimes V \subseteq U \otimes V$ The dual of a tensor product is the tensor product of duals, i.e., $(U \otimes V)^* = U^* \otimes V^*$. We denote by $T^pV := V^{\otimes p}$ the p-th tensor power of V. The dimension of T^pV is n^p.

The *symmetric power* S^pV is a linear space of symmetric tensors, i.e., the subspace of T^pV formed of the tensors invariant under transformations of the form $\overline{v}_1 \otimes \cdots \otimes \overline{v}_p \mapsto \overline{v}_{\sigma(1)} \otimes \cdots \otimes \overline{v}_{\sigma(p)}$ for any permutation σ. We write symmetric tensors as

$$\sum_{\substack{i_1,i_2,\ldots,i_p \\ 1 \le i_1 \le i_2 \le \cdots \le i_p \le n}} a_{i_1 i_2 \ldots i_p} \overline{v}_{i_1} \odot \overline{v}_{i_2} \odot \cdots \odot \overline{v}_{i_p},$$

where $\odot$ denotes the symmetric product and $a_{i_1 i_2 \ldots i_p}$ are elements of F. By definition, $\dim(S^pV) = \binom{n+p-1}{p}$. The space S^pV can be thought of as a projection

$$S : T^pV \to S^pV$$

that sends the tensor $\overline{v}_{i_1} \otimes \overline{v}_{i_2} \otimes \cdots \otimes \overline{v}_{i_p}$ to $\overline{v}_{j_1} \odot \overline{v}_{j_2} \odot \cdots \odot \overline{v}_{j_p}$ where $j_1 \le j_2 \le \cdots \le j_p$ is a sorted copy of $i_1, i_2, \ldots, i_p$.

Finally, $x \wedge y$ denotes the exterior (alternating) product of vectors, characterized by $x \wedge y = -y \wedge x$; hence $\overline{v}_{\sigma(1)} \wedge \overline{v}_{\sigma(2)} \wedge \cdots \wedge \overline{v}_{\sigma(n)} = \operatorname{sgn}(\sigma) \overline{v}_1 \wedge \overline{v}_2 \wedge \cdots \wedge \overline{v}_n$, where $\operatorname{sgn}(\sigma)$ is the signature of the permutation σ. The *exterior power* Λ^pV is a vector subspace of dimension $\binom{n}{p}$ spanned by elements of the form $\overline{v}_{i_1} \wedge \overline{v}_{i_2} \wedge \cdots \wedge \overline{v}_{i_p}, 1 \le i_1 < i_2 < \cdots < i_p \le n$, so a vector in Λ^pV has the form

$$\sum_{\substack{i_1,i_2,\ldots,i_q \\ 1 \le i_1 < i_2 < \cdots < i_q \le n}} a_{i_1 i_2 \ldots i_q} \overline{v}_{i_1} \wedge \overline{v}_{i_2} \wedge \cdots \wedge \overline{v}_{i_q}.$$

The spaces S^pV and Λ^qV are formed by the action on T^pV of the symmetric and alternating groups,

respectively.

By convention, T^0V, S^0V and Λ^0V are taken to be F.

3.4.3 Generalized PM codes

An extension of the PM construction was recently proposed in [20]. The construction of [20, Sec. 4] yields a family of MSR codes with parameters

$$n, k, d = \frac{(k-1)t}{t-1}, l = \binom{k-1}{t-1}, M = t\binom{k}{t}, \quad 2 \leq t \leq k \leq n-1.$$

In this section we follow the paradigm of evaluation codes to introduce an IP node repair procedure for this code family.

We start with a brief description of the code construction. Let $X = F^t$ and $Y = F^{k-t+1}$. Let $L := X \otimes S^tY$ and note that $\dim(L) = M$. The encoding $\phi : L \to F^{nl}$ is an F-linear map. To define a concrete encoding procedure, we fix a basis in L^* and let the coordinates of ϕ be the contents of the stored data.

To support the data reconstruction and node repair tasks, we further choose, for each $i \in [n]$, a pair of vectors $x_i \in X$ and $y_i \in Y$ such that

(i) Any t-subset of x_i's spans X.

(ii) Any $(k-t+1)$-subset of y_i's spans Y.

(iii) Any d subspaces $x_i \otimes y_i \odot S^{t-2}Y$ span $X \otimes S^{t-1}Y$.

The first two properties enable data reconstruction, while the node repair property depends on the third condition [20].

With these assumptions, the contents of node i correspond to the restriction $\phi \restriction x_i \otimes y_i \odot S^{t-1}Y \in (x_i \otimes y_i \odot S^{t-1}Y)^*$. This is consistent with the code parameters: indeed, an element in $(x_i \otimes y_i \odot S^{t-1}Y)^*$

is completely described by its evaluations on a basis of the space $x_i \otimes y_i \odot S^{t-1}Y$, which requires storing exactly $l = \binom{k-1}{t-1}$ evaluations.

As before, let $f \in [n]$ be the (index of the) failed node and let $D \subseteq [n] \setminus \{f\}$ be the helper set. Note that we wish to recover the restriction $\phi \restriction x_f \otimes y_f \odot S^{t-1}Y$. Choose a basis for $x_f \otimes y_f \odot S^{t-1}Y$ and let $x_f \otimes y_f \odot (\overline{y}_{i_1} \odot \cdots \odot \overline{y}_{i_{t-1}})$ be one of the basis vectors. Let

$$\{\underline{y}_{j_1} \odot \cdots \odot \underline{y}_{j_{t-2}}, 1 \leq j_1 \leq j_2 \cdots \leq j_{t-2} \leq k-t+1\}$$

be a basis of $S^{t-2}Y$. The helper node $i \in D$ transmits to the failed node the restriction of ϕ to the set of vectors $\{x_i \otimes y_i \odot (\underline{y}_{j_1} \odot \cdots \odot \underline{y}_{j_{t-2}}) \odot y_f\}$.

It becomes easier to think of the above construction once we connect it with PM codes described in Sec. 3.4.1. For that, take $t = 2$. In this case, the file size is

$$\dim(L) = \dim(X \otimes S^2 Y) = \dim(F^2 \otimes S^2 F^{k-1}) = k(k-1).$$

Node i stores $\phi \restriction (x_i \otimes y_i \odot Y)$, i.e, ϕ evaluated at a basis of $x_i \otimes y_i \odot Y$, which requires storing exactly $\dim(Y) = k-1$ symbols. Each node can calculate the symbol $\phi(x_i \otimes y_i \odot y_f) \in F$. Now notice that d vectors $\{x_i \otimes y_i\}$ span $X \otimes Y$, and so d values $\phi(x_i \otimes y_i \odot y_f)$ account for the evaluations of ϕ on $X \otimes Y \odot y_f$. From this set of evaluations, we can calculate ϕ on $x_f \otimes Y \odot y_f$ which by the symmetric product property is the same as $x_f \otimes y_f \odot Y$. These evaluations form the contents of the failed node.

The IP repair for this construction works as follows. By (iii) above we can write

$$\begin{aligned} x_f \otimes y_f \odot (\overline{y}_{i_1} \odot \cdots \odot \overline{y}_{i_{t-1}}) &= x_f \otimes (\overline{y}_{i_1} \odot \cdots \odot \overline{y}_{i_{t-1}}) \odot y_f \\ &= \sum_{i \in D} \sum_{j_1,\ldots,j_{t-2}} a_{i,j_1,\ldots,j_{t-2}} x_i \otimes y_i \odot \mathcal{Y}_{j_1,\ldots,j_{t-2}} \odot y_f, \end{aligned}$$

where we denoted $\mathcal{Y}_{j_1,\dots,j_{t-2}} = \underline{y}_{j_1} \odot \cdots \odot \underline{y}_{j_{t-2}}$. Again similarly to the PM codes, any set $A \subseteq D$ with $|A| \geq d - k + 1$ can transmit the following single evaluation of ϕ along the path to f:

$$\phi\Big(\sum_{i\in A}\sum_{j_1,\dots,j_{t-2}} a_{i,j_1,\dots,j_{t-2}} x_i \otimes y_i \odot \mathcal{Y}_{j_1,\dots,j_{t-2}} \odot y_f\Big)$$
$$= \sum_{i\in A}\sum_{j_1,\dots,j_{t-2}} a_{i,j_1,\dots,j_{t-2}} \phi(x_i \otimes y_i \odot \mathcal{Y}_{j_1,\dots,j_{t-2}} \odot y_f).$$

This can be done for all basis vectors of the chosen basis of $x_f \otimes y_f \odot S^{t-1}Y$, and that requires $l = \binom{k-1}{t-1}$ transmissions, which matches the lower bound of Lemma 3.2.2. Note that the AF repair would require any set A of helpers to transmit $\beta|A|$ symbols of F, which is greater than l for $|A| > d - k + 1$.

We have shown that IP repair can outperform direct relaying. Let us give an example to support this claim.

Example 3.4.2. Consider the use of $[n = 7, k = 5, d = 6, l = 6, \beta = 3, M = 30]$ generalized PM codes for the graph shown in Fig. 3.2. Suppose that $F = F_{16}$, and $X = F^3, Y = F^3$. Choose distinct $\{a_i\}_{i=1}^7$

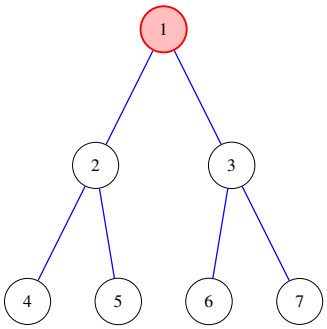


Figure 3.2: The graph for Example 3.4.2

from F and let $x_i = \begin{bmatrix} 1 & a_i^2 & a_i^6 \end{bmatrix}, y_i = \begin{bmatrix} 1 & a_i & a_i^3 \end{bmatrix}$. The file of size 30 stored by the system is defined by the evaluation of a functional ϕ on $X \otimes S^3Y$. Node i stores the restriction of ϕ to $x_i \otimes y_i \odot S^2Y$, hence storing 6 symbols per node.

For the repair of node 1, node i sends the evaluations of ϕ restricted to the tensors $\{x_i \otimes y_i \odot$

$\hat{y}_j \odot y_1\}_{j=1}^3$ where the set $\{\hat{y}_j\}$ forms a basis of Y. Since the six tensors $x_i \otimes y_i$ span $X \otimes Y$, the set $\{x_i \otimes y_i \odot \hat{y}_j \odot y_1, 1 \leq j \leq 3; 2 \leq i \leq 7\}$ spans $X \otimes Y \odot Y \odot y_1 \equiv X \otimes y_1 \odot Y \odot Y$. Hence

$$\phi(x_1 \otimes y_1 \odot \mathcal{Y}_{j_1,j_2}) = \sum_{i=2}^{7} \sum_{j=1}^{3} b_{i,j,j_1,j_2} \phi(x_i \otimes y_i \odot \hat{y}_j \odot y_1)$$

where $\{\mathcal{Y}_{j_1,j_2}\}$ is a basis of $S^2 Y$.

Consider nodes 2 and 3. Instead of relaying $\{\phi(x_i \otimes y_i \odot \hat{y}_j \odot y_1) : i = 2, 4, 5, j = 1, 2, 3\}$, node 2 can transmit $\{\sum_{i=2,4,5} \sum_{j=1}^{3} b_{i,j,j_1,j_2} \phi(x_i \otimes y_2 \odot \hat{y}_j \odot y_1)\}$ for all $\{\mathcal{Y}_{j_1,j_2}\}$. The former requires 9 symbol transmissions while the later requires only 6, and the same holds true for node 3. In total, the AF repair procedure would require transmission of $3 \cdot (1+1+1+1+3+3) = 30$ symbols, while the IP procedure requires only $3 \cdot (1+1+1+1+2+2) = 24$ symbol transmissions. This matches exactly with the bound in Lemma 3.2.2.

3.4.4 Operations on product spaces

In preparation for discussing IP repair with Moulin codes in the next section, we define (following [19]) two operations on tensor product spaces. Let $V = F^{d-k}$, $W = F^k$, and $U = V \oplus W \cong F^d$. We shall be dealing with spaces of the form $T^p V \otimes V \otimes \Lambda^q W$ and $T^p V \otimes W \otimes \Lambda^q W$ where $p + q = s - 1$ and $p, q \geq 0$. Note that

$$T^p V \otimes V \otimes \Lambda^q W \oplus T^p V \otimes W \otimes \Lambda^q W = T^p V \otimes U \otimes \Lambda^q W,$$

and hence there are natural inclusion maps from each of these spaces to their direct sum, as well as natural projection maps from the direct sum to these spaces.

Define the *co-wedge product* operator inductively as follows:

$$\nabla : T^pV \otimes \Lambda^1 W \to T^pV \otimes W$$

$$\nu \otimes w_1 \to \nu \otimes w_1$$

$$\nabla : T^pV \otimes \Lambda^2 W \to T^pV \otimes W \otimes \Lambda^1 W$$

$$\nu \otimes w_1 \wedge w_2 \to \nu \otimes w_1 \otimes w_2 - \nu \otimes w_2 \otimes w_1$$

$$\nabla : T^pV \otimes \Lambda^{q+1} W \to T^pV \otimes W \otimes \Lambda^q W$$

$$\nu \otimes \omega \wedge w_1 \to \nabla(\nu \otimes \omega) \wedge w_1 + (-1)^q \nu \otimes w_1 \otimes \omega,$$

where on the last line $\omega \in \Lambda^q W$ and $w_1 \in W$. Thus, as a result of applying ∇, the degree of the wedge product decreases by one. For tensors of higher ranks, ∇ applies term-wise, and the images are added. The operator ∇ is clearly linear. Next we define the *coboundary operators (differentials)* which increase the degree of tensors. For any $v \in V$ define the linear transformation inductively:

$$\partial_v^V : \Lambda^q W \to U \otimes \Lambda^q W$$

$$\omega \to 0$$

$$\partial_v^V : U \otimes \Lambda^q W \to T^1 V \otimes U \otimes \Lambda^q W$$

$$u \otimes \omega \to v \otimes u \otimes \omega$$

$$\partial_v^V : T^pV \otimes U \otimes \Lambda^q W \to T^{p+1} V \otimes U \otimes \Lambda^q W$$

$$\nu \otimes u \otimes \omega \to \partial_v^V(\nu) \otimes u \otimes \omega + (-1)^p \nu \otimes v \otimes u \otimes \omega$$

for all $p \geq 1, q \geq 0$. Note that when $q = 0$ we take $\Lambda^0 = F$. In the other direction, for every $w \in W$ and

$p \geq 0, q \geq 1$ define the mappings

$$\partial_w^W : T^pV \otimes U \to T^pV \otimes U \otimes \Lambda^1 W$$

$$\nu \otimes u \to (-1)^p \nu \otimes u \otimes w$$

$$\partial_w^W : T^pV \otimes U \otimes \Lambda^q W \to T^pV \otimes U \otimes \Lambda^{q+1} W$$

$$\nu \otimes u \otimes \omega \to (-1)^{p+q} \nu \otimes u \otimes \omega \wedge w..$$

Finally for $u \in U$ such that $u = v + w, v \in V, w \in W$, define

$$\partial_u^U = \partial_v^V + \partial_w^W.$$

Thus, the overall diagram has the form

$$\begin{array}{ccccc} \cdots & \xrightarrow{\partial_w^W} & T^{p+1}V \otimes U \otimes \Lambda^q W & \xrightarrow{\partial_w^W} & \cdots \\ \partial_v^V \uparrow & & \partial_v^V \uparrow & & \partial_v^V \uparrow \\ T^pV \otimes U \otimes \Lambda^{q-1} W & \xrightarrow{\partial_w^W} & T^pV \otimes U \otimes \Lambda^q W & \xrightarrow{\partial_w^W} & T^pV \otimes U \otimes \Lambda^{q+1} W \\ \partial_v^V \uparrow & & \partial_v^V \uparrow & & \partial_v^V \uparrow \\ \cdots & \xrightarrow{\partial_w^W} & T^{p-1}V \otimes U \otimes \Lambda^q W & \xrightarrow{\partial_w^W} & \cdots \end{array}.$$

Except for U, this diagram follows the standard construction of the tensor product of chain complexes [64, Sec.10.1], and the differentials satisfy the usual relations: $(\partial_v^V)^2 = 0, (\partial_w^W)^2 = 0$ for all $v \in V, w \in W$, and $\partial_v^V \partial_w^W + \partial_w^W \partial_v^V = 0$.

3.4.5 IP for Interior Point Codes

In this section, we switch attention from MSR codes to a class of intermediate-point evaluation codes introduced recently by Duursma et al. in [19] (see also [58, Sec. 7.2]). Let s be an integer such that $n - 1 \geq d \geq k \geq s - 1 \geq 1$. The family of *Moulin codes* that we discuss has parameters $[n, k, d, l, \beta, M]$

that satisfy the relations

$$\left.\begin{aligned} l &= \textstyle\sum_{p+q=s-1}(d-k)^p\binom{k}{q} \\ \beta &= \textstyle\sum_{p+q=s-2}(d-k)^p\binom{k-1}{q} \\ M &= \textstyle\sum_{p+q=s-1} d(d-k)^p\binom{k}{q} - \sum_{p+q=s}(d-k)^p\binom{k}{q}, \end{aligned}\right\} \tag{3.9}$$

where $p \geq 0, q \geq 0$ throughout. While the general idea of implementing IP for this code family is the same as before (node contents are given by restrictions of linear maps to subspaces), the detailed description relies on the operations on tensor products introduced above.

For a fixed s satisfying the constraints above, the file $\mathcal{F}$ is chosen to be an element ϕ of the dual space

$$\bigoplus_{p+q=s-1} (T^pV \otimes U \otimes \Lambda^q W)^*, \tag{3.10}$$

where V, W, U are as in the previous section. The parity checks of the code correspond to the condition of having the following diagrams commute:

$$\begin{array}{ccc} T^pV \otimes \Lambda^{q+1}W & \xrightarrow{\phi} & F \\ & {\scriptstyle\nabla}\searrow \quad \nearrow{\scriptstyle\phi} & \\ & T^pV \otimes W \otimes \Lambda^q W & \end{array}$$

for all $p \geq 1, q \geq 0$ with $p + q = s - 1$, and

$$\begin{array}{ccc} \Lambda^{q+1}W & \xrightarrow{0} & F \\ & {\scriptstyle\nabla}\searrow \quad \nearrow{\scriptstyle\phi} & \\ & W \otimes \Lambda^q W & \end{array} \qquad \begin{array}{ccc} T^pV & \xrightarrow{\phi} & F \\ & {\scriptstyle\nabla}\searrow \quad \nearrow{\scriptstyle 0} & \\ & 0 & \end{array} \tag{3.11}$$

for $p = 0$ and $q = -1$, respectively.

The file size equals the dimension of the direct sum of the vector spaces (3.10) minus the dimension of the parity check space, which is exactly M in (3.9). To each node $i \in [n]$ we associate a vector $u_i \in U$ such that any d of these vectors span U and any k vectors span U/V under the quotient map $U \to U/V$.

The i-th node stores the following restriction of the mapping ϕ:

$$\phi \restriction \bigoplus_{p+q=s-1} (T^pV \otimes u_i \otimes \Lambda^q W).$$

The size l of the node equals $\dim(T^pV \otimes u_i \otimes \Lambda^q W)$, given by l in (3.9).

Now suppose that node $f \in [n]$ fails and we are provided with a set $D \subseteq [n] \setminus \{f\}$ of d helpers. Each node $h \in D$ provides the restrictions of its contents to coboundaries:

$$\phi \restriction \partial^U_{u_f}(T^pV \otimes u_h \otimes \Lambda^q W) \tag{3.12}$$

for each pair p, q with $p + q = s - 2$. We shall need the following result.

Lemma 3.4.2 ([19], Thm. 4.1)**.** *For all possible $p, q \geq 0$, such that $p + q = s - 2$ and all $\nu \in T^pV, \omega \in \Lambda^q W$, we have*

$$\phi(\partial^U_{u_f}(\nabla(\nu \otimes \omega))) - \phi(\partial^U_{u_f}(\nu \otimes \omega)) = (-1)^p \phi(\nu \otimes u_f \otimes \omega).$$

If $p = 0$, then $\phi(\partial^U_{u_f}(\omega)) = 0$ due to (3.11).

The right-hand side of the above equation is one coordinate of the failed node, and the left-hand side can be computed from (3.12).

The statement of the next lemma appears in [19] without a proof (as a statement in the proof of [19, Thm. 4.1]). We include the proof here to set up the notation.

Lemma 3.4.3. *1) For all possible $p \geq 1, q \geq 0$ such that $p + q = s - 1$ and all $\nu \in T^pV, \omega \in \Lambda^q W$, the tensor $\nu \otimes \omega$ is contained in the linear span of the union of the spaces $\{T^{p'}V \otimes u_h \otimes \Lambda^{q'}W\}_{h \in D, p'+q'=s-2}$.*

2) For all possible $p, q \geq 0$, such that $p + q = s - 1$, for all $\nu \in T^pV, \omega \in \Lambda^q W$, $\nabla(\nu \otimes \omega)$ is contained in the linear span of the union of the spaces $\{T^{p'}V \otimes u_h \otimes \Lambda^{q'}W\}_{h \in D, p'+q'=s-2}$.

Proof. 1) Fix $p_1 \geq 1, q_1 > 0$ such that $p_1 + q_1 = s - 1$. Let $\nu \in T^{p_1}V$ and $\omega \in \Lambda^{q_1}W$. Fix a basis

$\{\overline{\nu}_i \otimes u_h \otimes \overline{\omega}_i\}_{i=1}^{(d-k)^{p_1-1}\binom{k}{q_1}}$ of $T^{p_1-1}V \otimes u_h \otimes \Lambda^q W$. Since the set $\{u_h\}_{h\in D}$ spans U, we can write

$$\nu \otimes \omega = \overbrace{(\nu_1 \otimes \cdots \otimes \nu_{p_1-1})}^{T^{p_1-1}V} \otimes \underbrace{\nu_{p_1}}_{U} \otimes \overbrace{\omega}^{\Lambda^q W},$$

and hence $\nu \otimes \omega$ is an element of $T^{p_1-1}V \otimes U \otimes \Lambda^{q_1} W$. So we can write $\nu \otimes \omega = \sum_{i,h} a_{i,h}(\overline{\nu}_i \otimes u_h \otimes \overline{\omega}_i)$.

2) Similarly, for a basis $\{\underline{\nu}_j \otimes u_h \otimes \underline{\omega}_j\}_{j=1}^{(d-k)^{p_1}\binom{k}{q_1-1}}$ of $T^{p_1}V \otimes u_h \otimes \Lambda^{q_1-1}W$, we can write $\nabla(\nu \otimes \omega) = \sum_{j,h} b_{j,h}(\underline{\nu}_j \otimes u_h \otimes \underline{\omega}_j)$. □

Our main statement in this part is the next lemma, which justifies the IP repair procedure.

Lemma 3.4.4. *Let $A \subseteq D$. For the repair of f, it is sufficient for the nodes in the set A to transmit l symbols.*

Proof. Fix $p, q \geq 0$ such that $p + q = s - 1$. Let $\nu \in T^p V, \omega \in \Lambda^q W$. If $p \geq 1$ then by parts (1) and (2) of Lemma 3.4.3, we have

$$\begin{aligned}\phi(\partial_{u_f}^U(\nabla(\nu \otimes \omega))) - \phi(\partial_{u_f}^U(\nu \otimes \omega)) \\ = \sum_{j,h} b_{j,h}\phi(\partial_{u_f}^U(\underline{\nu}_j \otimes u_h \otimes \underline{\omega}_j)) - \sum_{i,h} a_{i,h}\phi(\partial_{u_f}^U((\overline{\nu}_i \otimes u_h \otimes \overline{\omega}_i)).\end{aligned}$$

Note that if $p = 0$ then the second term on the LHS $\phi(\partial_{u_f}^U(\nu \otimes \omega))$ is already 0 by (3.11) and we simply write using Lemma 3.4.3(2)

$$\phi(\partial_{u_f}^U(\nabla(\nu \otimes \omega))) = \sum_{j,h} b_{j,h}\phi(\partial_{u_f}^U(\underline{\nu}_j \otimes u_h \otimes \underline{\omega}_j)).$$

By Lemma 3.4.2, the LHS equals $(-1)^p\phi(\nu \otimes u_f \otimes \omega)$, and we have recovered one symbol of the failed

node. For this, the set A need to transmit the element

$$\sum_{h\in A}\Big[\sum_j b_{j,h}\phi(\partial^U_{u_f}(\underline{\nu}_j\otimes u_h\otimes\underline{\omega}_j))-\sum_i a_{i,h}\phi(\partial^U_{u_f}(\overline{\nu}_i\otimes u_h\otimes\overline{\omega}_i)\Big].$$

Doing this for any fixed basis $\{\nu,\omega\}$ of $T^pV\otimes u_f\otimes\Lambda^qW$, for all values of p,q, requires the set A to transmit a total of l symbols. □

Observe that whenever $|A|\geq\lceil\frac{l}{\beta}\rceil$, the IP protocol given by this lemma results in communication savings compared to the AF repair.

Remark 1. *This, however, might not be the optimal savings obtained, as the lower bound of Lemma 3.2.2 gives the following: Let the distributed code be the $[n,k,d,l,\beta,M]$ Moulin Code as given in Eq. (3.9). Let v_f, $f\in[n]$ be the failed node. For a subset of the helper nodes $E\subset D$, let R^f_E be a function of S^f_E such that*

$$H(W_f|R^f_E,S^f_{D\setminus E})=0. \tag{3.13}$$

If $|E|\geq d-k+1$, then

$$H(R^f_E)\geq\sum_{p+q=s-1}d(d-k)^p\binom{k}{q}-\sum_{p+q=s}(d-k)^p\binom{k}{q}$$
$$-\sum_{i=1}^{k-1}\min\Big\{\sum_{p+q=s-1}(d-k)^p\binom{k}{q},(d-i+1)\sum_{p+q=s-2}(d-k)^p\binom{k-1}{q}\Big\}$$

Example 3.4.3. *We show how to perform the above IP protocol for a Moulin code with $s=4$. The other parameters we take to be the same as Example 3.4.2, i.e., $[n=7,k=5,d=6]$ and for $s=4$ we get $l=26,\beta=11,M=125$. We again use the graph in Fig. 3.2 to demonstrate the savings in required transmission bandwidth for repair. Note that this code satisfies $d\beta=66>l>(d-k+1)\beta=22$ and operates at an interior point of the storage-bandwidth trade-off curve.*

Let $V = F, W = F^5, U = V \oplus W \equiv F^6$ and let $\{\nu_i\}$ and $\{\omega_j\}$ be a basis for V and W, respectively. The file is seen as a functional in the dual space of $\oplus_{p+q=3} T^pV \otimes U \otimes \Lambda^q W$. We choose a vector u_i for each node i such that any 6 of these span U and any 5 of these span W. The i-th node stores the restriction of ϕ to the space $\oplus_{p+q=3}(T^pV \otimes u_i \otimes \Lambda^q W)$. For the repair of the failed node 1, helper node h is supposed to send the evaluations of ϕ on the restricted spaces of $\{\partial^U_{u_1}(T^pV \otimes u_h \otimes \Lambda^q W)\}_{p+q=2}$. More precisely, node h sends

$$\phi \upharpoonright \partial^U_{u_1}(u_h \otimes \omega_{j_1} \wedge \omega_{j_2}) \quad \textit{for all basis elements } \omega_{j_1} \wedge \omega_{j_2} \textit{ of } W \wedge W$$

$$\phi \upharpoonright \partial^U_{u_1}(\nu_{i_1} \otimes u_h \otimes \omega_{j_1}) \quad \textit{for all basis elements } \nu_{i_1} \textit{ of } V \textit{ and } \omega_{j_1} \textit{ for } W$$

$$\phi \upharpoonright \partial^U_{u_1}(\nu_{i_1} \otimes \nu_{i_2} \otimes u_h) \quad \textit{for all basis elements } \nu_{i_1} \otimes \nu_{i_2} \textit{ of } V \otimes V$$

From Lemma 3.4.2, to evaluate ϕ on a tensor $\nu \otimes u_1 \otimes \omega$ for $\nu \in T^pV, \omega \in \Lambda^q W, p+q = 3$, which is one coordinate of the failed node's contents, we need the evaluations of $\partial^U_{u_1}(\nu \otimes \omega)$ and $\partial^U_{u_1}(\nabla(\nu \otimes \omega))$. Choose $p = 1, q = 2$, so $\nu \in V, \omega \in W \wedge W$. Since $(\nu \otimes \omega)$ is in $U \otimes \Lambda^2 W$, we have

$$\phi(\partial^U_{u_1}(\nu \otimes \omega)) = \sum_{h \in D} \sum_{j_1, j_2} a_{h,j_1,j_2} \phi(\partial^U_{u_1}(u_h \otimes \omega_{j_1} \wedge \omega_{j_2}))$$

and since $\nabla(\nu \otimes \omega)$ is in $V \otimes U \otimes W$,

$$\phi(\partial^U_{u_1}(\nabla(\nu \otimes \omega))) = \sum_{h \in D} \sum_{i_1, j_1} b_{h,i_1,j_1} \phi(\partial^U_{u_1}(\nu_{i_1} \otimes u_h \otimes \omega_{j_1}).$$

Similar equalities can be written for all $p, q \geq 0, p+q = 3$.

With reference to the graph in Fig. 3.2, we see that node 2 in the traditional AF scheme sends the

evaluations

$$\phi \restriction \partial^U_{u_1}(\{u_h \otimes W \wedge W\}_{h=2,4,5}), \partial^U_{u_1}(\{V \otimes u_h \otimes W\}_{h=2,4,5}), \partial^U_{u_1}(\{V \otimes V \otimes u_h\}_{h=2,4,5}),$$

which may require it to transmit up to 33 symbols according to the bound in Lemma 8 of [19]. At the same time, if node 2 sends only

$$\sum_{h=2,4,5} [\sum_{i_1,j_1} b_{h,i_1,j_1} \phi(\partial^U_{u_1}(\nu_{i_1} \otimes u_h \otimes \omega_{j_1}) - \sum_{j_1,j_2} a_{h,j_1,j_2} \phi(\partial^U_{u_1}(u_h \otimes \omega_{j_1} \wedge \omega_{j_2})]$$

for every $\nu \otimes \omega$, its communication comprises 26 symbols. The same is true for node 3. Note that by the lower bound of Lemma 3.2.2, the number of symbols that either of these nodes contributes for repair is at least 21.

3.5 Determinant codes

Determinant codes [23,24] represent another well-known family of intermediate-point regenerating codes. Of several versions of the construction presented by the authors, we follow the one appearing in [24]. To remind ourselves of the general context, let n, k be fixed, and let $d = k$. Recall that the tradeoff curve (3.1) isolates a polygon on the bandwidth-storage plane called the *exact repair region*. In particular, as shown in [23], for $d = k$ the exact repair region is a convex hull of k points given by $l_m = \binom{k}{m}, \beta_m = \binom{k-1}{m-1}, M_m = m\binom{k+1}{m+1}$, for $m = 1, 2, \ldots, k$ and these points are achieved by the determinant code construction. Moreover, the intermediate points of the bound (3.1) can be achieved by space sharing. In this section, we observe that determinant codes can be easily adapted to support the IP technique.

Let us begin with a brief description of the code construction (see the original paper [24] for more details), noting that linearity of the codes is again at the root of this application. Fix some $m \in [k]$. The symbols of the data file $\mathcal{F}$ are arranged in two matrices, denoted below by V and W, of dimensions $l_m \times d$

and $l_{m+1} \times d$, respectively. The rows are of V are indexed by the m-subsets of the set $[d] := \{1, 2, \ldots, d\}$, the rows of W are indexed by the $(m+1)$-subsets, and the columns of either matrix are indexed by the elements of $[d]$. Accordingly we label the data symbols with two subscripts j and A, where $j \in [d]$ and $A \subset [d]$. Write these symbols as

$$\mathcal{V} = \{v_{A,j} \in F \mid A \subset [d], |A| = m, j \in A\}$$

$$\mathcal{W} = \{w_{S,j} \in F \mid S \subset [d], |S| = m+1, j \in S, \tau_S(j) \leq m\},$$

where $\tau_S(j) = |\{i \in S : i \leq j\}|$, i.e., in $\mathcal{W}$ we do not assign a data element to the largest index within each of the subsets S. Instead, the largest location within each S is assigned the value that fulfills the parity check equation

$$\sum_{j \in S} (-1)^{\tau_S(j)} w_{S,j} = 0,$$

yielding a total of $\binom{d}{m+1}$ parity symbols. Now assign the data symbols (and in the case of W also the parity symbols) to the corresponding places in the matrices V and W, writing them in the locations indexed by the elements of the subsets, and fill the remaining empty places in the matrices with zeros.

In the next step the matrices V and W are used to construct an $l_m \times d$ data matrix D, whose rows are again indexed by the sets $A \subset [d]$ and columns by $[d]$, as follows

$$d_{A,j} = \begin{cases} v_{A,j} & \text{if } j \in A \\ w_{A \cup \{j\},j} & \text{if } j \notin A \end{cases}.$$

Note that $|\mathcal{V}| + |\mathcal{W}| = M_m$ and $M_m + \binom{d}{m+1} = l_m d$, the number of matrix elements in D. Finally, to obtain a codeword that corresponds to the data file, we multiply D by a $d \times n$ matrix Φ such that each k-subset of its rows has full rank over F, for instance a Vandermonde matrix. This yields an $l_m \times n$

codeword matrix C over F.

Next we describe the node repair procedure suggested in [24]. For a matrix G denote its ith row by $G_{i,:}$ and ith column by $G_{:,i}$. Thus, the contents of the ith node (the ith coordinate of the codeword C) is given by $C_{:,i} = D\Phi_{:,i}$. Without loss of generality assume that node 1 has failed and nodes in the set $H = \{2, 3, \ldots, d+1\}$ are used as the helper nodes. Define the $l_{m-1} \times l_m$ matrix R, whose rows and columns are indexed by $(m-1)$- and m-subsets of $[d]$, as follows:

$$R_{B,A} = \begin{cases} (-1)^{\tau_A(j)}\phi_{j,1} & \text{if } \exists y \text{ s.t. } A = B \cup \{j\} \\ 0 & \text{otherwise,} \end{cases}$$

where $\phi_{j,1}$ is an element of the matrix Φ. We note that the matrix R depends only on the index of the failed node and can be pre-computed independently at each helper node. To perform repair, the failed node downloads from helper node $i \in H$ the vector $RC_{:,i}$. The dimension of this vector is l_{m-1}, so on the face of it, the required size of the download exceeds the allotted repair bandwidth β_m. However, by [22, Prop. 1] the rank of the matrix R is at most β_m, so as many symbols suffice to communicate the vector $RC_{:,i}$ from the ith (helper) node to the failed node.

At the failed node, the vectors $RC_{:,i}, i \in H$ are written as columns of a $l_{m-1} \times d$ matrix T. Since $C_{:,i} = D\Phi_{:,i}$, we can write $T = RD\Phi_{:,H}$, where $\Phi_{:,H}$ is the submatrix of Φ formed of the columns $2, 3, \ldots, d+1$. By construction, $\Phi_{:,H}$ is invertible, and the failed node can find the $l_{m-1} \times d$ matrix $R\,D$. These elements suffice to recover the contents of the failed node as shown in the following lemma due to [22], Prop. 2. Since our modification of the repair procedure depends on this statement, we include a proof in the appendix.

Lemma 3.5.1. *For any $A \subset [d], |A| = m$,*

$$C_{A,1} = \sum_{i \in A} (-1)^{\tau_A(i)} R_{A\setminus\{i\},:} D_{:,i}, \tag{3.14}$$

where $R_{A\setminus\{i\},:}$ is the row of R with index $A\setminus\{i\}$. Thus the contents of the failed node can be recovered from the matrix RD.

Note that $R_{A\setminus\{i\},:} D_{:,i}$ is an element in the product $R \cdot D$, which is exactly the information available to the failed node. The point that we wish to make is that the described repair procedure can be modified to support IP repair for determinant codes used on a graph. To formulate it, we need some notation. Let $\overline{C}_H = \left[C_{:,2}^{\mathsf{T}} C_{:,3}^{\mathsf{T}} \dots C_{:,d+1}^{\mathsf{T}} \right]^{\mathsf{T}}$ be a dl_m-dimensional column vector obtained by concatenating columns $2, 3, \dots, d+1$ of C. Define $l_m \times l_m$ matrices $W^{(i)}, i = 1, \dots, d$, whose rows are indexed by m-subsets of $[d]$. For a given m-subset $A \subset [d]$ the A-th row of $W^{(i)}$ is defined as:

$$W^{(i)}_{A,:} = \begin{cases} (-1)^{\tau_A(i)} R_{A\setminus\{i\},:} & \text{if } i \in A \\ \mathbf{0} & \text{otherwise} \end{cases}.$$

Proposition 3.5.2. *The contents of the failed node can be found as*

$$C_{:,1} = U\overline{C}_H = \left[U^{(1)} U^{(2)} \dots U^{(d)} \right] \overline{C}_H, \tag{3.15}$$

where U is an $l_m \times dl_m$ matrix determined by the contents of the helper set $\Phi_{:,H}$.

Proof. As before, let $C_{:,H}$ and $\Phi_{:,H}$ be the submatrices of the matrices C and Φ with columns indexed by the set H, so $C_{:,H} = D\,\Phi_{:,H}$, or

$$D = C_{:,H}\,\Phi_{:,H}^{-1}.$$

Similarly to $\overline{C}_H$, let $\overline{D} = \left[D_{:,1}^\intercal D_{:,2}^\intercal \dots D_{:,d}^\intercal\right]^\intercal$ be the flattened matrix D, written as a column vector of length dl_m. Let $\overline{\Phi}_H = (\Phi_{:,H}^{-1} \otimes I_{l_m})^\intercal$ be the $dl_m \times dl_m$ block matrix. Then

$$\overline{D} = \overline{\Phi}_H \overline{C}_H.$$

Now, according to this relation and (3.14),

$$C_{:,1} = W\overline{D} = U\overline{C}_H,$$

where $W = \left[W^{(1)}W^{(2)}\dots W^{(d)}\right]$ and $U = \left[W^{(1)}W^{(2)}\dots W^{(d)}\right]\overline{\Phi}_H$, proving (3.15). Moreover, the matrix U depends only on $\Phi_{:,H}$, and the proof is complete. □

As before, representation (3.15) supports "pipeline" repair of the contents of $C_{:,1}$, which can be spread across the nodes of the helper set. Specifically, instead of transmitting $|E|\beta_m$ symbols, any set E of helper nodes can only transmit the vector

$$\sum_{i\in E} U^{(i)} C_{:,i},$$

which requires sending a total of l_m symbols over the edges leaving E along the shortest path toward the failed node. Hence whenever $|E| > \frac{l_m}{\beta_m}$, this procedure accounts for savings in the repair bandwidth over simple forwarding (the AF repair).

Remark 2. *For Determinant Codes, Lemma 3.2.2 does not give us any improvement over the trivial bound of $H(R_E^f) \geq \beta_m$ for any set E. This is because*

$$M_m - \sum_{i=1}^{k-1} \min\{l_m, (k-i+1)\beta_m\} - \beta_m = M_m - \underbrace{\sum_{i=1}^{k} \min\{l_m, (k-i+1)\beta_m\}}_{\textit{value of the cutset bound for } d=k} \leq 0$$

and hence

$$M_m - \sum_{i=1}^{k-1} \min\{l_m, (k-i+1)\beta_m\} \leq \beta_m.$$

Example 3.5.1. *Going back to our running example in Fig. 3.2, choose* $m = 3$*, then the code parameters are* $n = 7, k = d = 6, M = 105$*, and we obtain an interior-point code operating at the point* $(l_m = 20, \beta_m = 10)$ *of the trade-off curve. As before, suppose our goal is to repair the root node, while all the remaining 6 nodes serve as helpers. The AF repair procedure would require transmission of* $4 \cdot 10 + 2 \cdot 30 = 100$ *symbols while performing intermediate processing at the nodes adjacent to the root node results in a total of* $4 \cdot 10 + 2 \cdot 20 = 80$ *symbol transmissions, saving* 10 *transmissions at each of the two nodes. While this example affirms that IP repair entails communication savings, as per Remark 2, we cannot say how far they are from optimality.*

Remark 3. *A family of regenerating codes based on determinant codes, termed* cascade codes*, was introduced in [22]. Their parameters coincide with the parameters of Moulin codes* (3.9)*. A cascade code is formed by stacking together several determinant codes with different values of the parameter* μ*, called the mode of the component codes. Cascading together determinant codes of varying modes enables the authors of [22] to obtain codes for all values* $d \geq k$ *as opposed to* $d = k$ *in the previous section. Repair of the failed node is performed by concatenating the repair data obtained independently from the constituent determinant codes. For this reason, the node repair procedure for them can be implemented relying on the IP repair of determinant codes.*

3.6 Data retrieval for codes on graphs

Until now, we have only focused on the node repair aspect of regenerating codes on graphs, circumventing the more frequently occurring task of data retrieval. The reason behind this is that at the MSR point, which was the main focus of Chapter 2, the task of data retrieval from a regenerating code defined on an

arbitrary graph becomes trivial. Since MSR codes are Maximum Distance Separable (MDS) by definition, any set A of k or fewer nodes has to transmit $|A| \cdot l$ symbols and there is no hope of compressing this any further. This implies that in the restricted connectivity setting, when the Data Collector (DC) does not have direct access to k nodes, standard relaying of data is optimal. The situation changes when we lift the MSR constraint.

Example 3.6.1. The following example shows that for MBR code families the task of data recovery can be accomplished by downloading fewer than kl symbols from the chosen subset of k nodes. Consider the family of *polygonal codes* [68], which closely follows the definition of MBR codes. The parameters of the family are $n, k, d = n-1, l = n-1, \beta = 1, M = k(n-1) - k(k-1)/2$. To construct the code, fix n and k and choose an MDS code of length $N = n(n-1)/2$ and dimension M over a field F_q of size $q \geq N$. The encoding mapping of the polygonal code $\mathcal{C}_n$ is formed of two steps. In the first step we encode M symbols of the file $\mathcal{F}$ into a codeword of the MDS code. The length N is chosen to support a bijection between the coordinates of the codeword and edges of a complete graph K_n, so we place each encoded symbol on an edge of the graph. Each vertex of K_n models a storage node. To complete the data placement in the system, we assign to each node the symbols written on the edges incident to it. Thereby, every node carries $n-1$ symbols of the encoding, which matches the parameters of the code $\mathcal{C}_n$.

To reconstruct the file $\mathcal{F}$, the DC accesses an arbitrary subset K of k nodes of the graph, which in total contain $k(n-1)$ symbols of the codeword. Since each pair of nodes shares one common symbol, the DC downloads $l, l-1, \ldots, 1$ symbols from the nodes in K (taken in some fixed order). This yields a total of M stored symbols, so the DC is able to recover the MDS codeword and therefore also the file. Note a saving of $\binom{k}{2}$ symbols compared to downloading the entire contents of the k nodes.

In this section we elaborate on this example in two ways. First, in Lemma 3.6.1 below we derive a lower bound on the number of symbols required to complete the data collection task. The bound applies to all sets of parameters on the storage-bandwidth tradeoff curve (3.1), with a caveat that for the intermediate

points, we have to allow codes with functional rather than exact repair. At the MBR point the bound is attainable, as shown by the above example as well as by another example that we consider in this section, namely the MBR Product-Matrix codes.

3.6.1 Lower bound for the data retrieval bandwidth

For non-MSR regenerating codes, the possibility of reducing the number of downloaded field symbols motivates us to seek a lower bound on the communication complexity. Let us formally introduce our model. Like before, an $[n, k, d, l, \beta, M]$ regenerating code is defined on a connected graph $G = (V, E)$. A set of k nodes, denoted by K, wish to send their data to the DC for the purpose of recovering the original file of size M. We assume DC to be an external node (if DC is a node in the graph G itself then it needs to contact $k-1$ other nodes but the analysis remains the same.). To formalize this model, suppose that DC has direct access only to a subset $\bar{K} \subset K$ with $|\bar{K}| < k$ and let $G_{\bar{K},K}$ be the graph with $V_{\bar{K},K} = \{DC\} \cup K$ and $E_{\bar{K},K} = \{(DC, v) : v \in \bar{K}\} \cup \{(u, v) : u, v \in K, (u, v) \in E\}$. We will assume that this graph is connected and all communication for the data retrieval process will be done on this new graph $G_{\bar{K},K}$.

Lemma 3.6.1. *For an $[n, k, d, l, \beta, M]$ regenerating code and any set $A \subseteq K$ of size a, let R_A be the data derived as a function W_A such that $H(\mathcal{F}|R_A, W_{K\setminus A}) = 0$. Then*

$$H(R_A) \geq \sum_{i=k-a}^{k-1} \min\{l, (d-i)\beta\}. \tag{3.16}$$

Proof. From Lemma 3.2.1, we know that for any set $B = \{b_1, b_2, \ldots, b_{|B|}\} \subset K$,

$$H(W_B) = \sum_{i=1}^{|B|} H(W_{b_i}|W_{b_{i-1}}, \ldots, W_{b_1}) \leq \sum_{i=1}^{|B|} \min\{l, (d-i+1)\beta\} = \sum_{i=0}^{|B|-1} \min\{l, (d-i)\beta\}.$$

From the data retrieval property of the code, we have

$$H(R_A, W_{K\setminus A}) \geq \sum_{i=0}^{k-1} \min\{l, (d-i)\beta\},$$

which implies

$$\begin{aligned} H(R_A) &\geq \sum_{i=0}^{k-1} \min\{l, (d-i)\beta\} - H(W_{K\setminus A}) \\ &\geq \sum_{i=0}^{k-1} \min\{l, (d-i)\beta\} - \sum_{i=0}^{k-a-1} \min\{l, (d-i)\beta\} \\ &= \sum_{i=k-a}^{k-1} \min\{l, (d-i)\beta\}. \end{aligned}$$

□

Specializing bound (3.16) for the MSR and MBR points, we obtain

Corollary 3.6.2. *(1) For an $[n, k, d, l, \beta, M]$ MSR code and any subset $A \subseteq K$*

$$H(R_A) \geq |A| \cdot l.$$

(2) For an $[n, k, d, l, \beta, M]$ MBR code and any subset $A \subseteq K$ of size a

$$H(R_A) \geq \sum_{i=k-a}^{k-1} (d-i)\beta. \tag{3.17}$$

Remark: Part (1) of this corollary gives a formal proof of our earlier claim as to why standard relaying is optimal for data retrieval with MSR codes.

Remark: Note that at the MBR point, even for the fully connected setting when the DC has direct access to k nodes, data retrieval is performed by downloading full contents of the k nodes. At the same time,

Cor. 3.6.2(2) shows that it might be possible to retrieve the file by downloading fewer symbols. In the next section we show that this is indeed the case and that bound (3.17) is achievable with PM MBR codes; thus this bound is in fact tight.

3.6.2 Data retrieval with communication constraints

In this section we describe a data collection procedure on a graph with communication complexity attaining the bound (3.17), using the PM code family as an example. Let us first recall the standard PM MBR construction of Rashmi et al. [59]. The parameters of the codes are $[n, k, d, l = d, \beta = 1, M = kd - \binom{k}{2}]$. The data file $\mathcal{F}$ is formed of M symbols of the field F, and it is represented by a $d \times d$ matrix B that has the following structure:

$$B = \begin{bmatrix} S & T \\ T^{\intercal} & 0 \end{bmatrix}.$$

Here S is a $k \times k$ symmetric matrix and T is a $k \times (d-k)$ matrix. Together these two matrices contain $\binom{k}{2} + k(d-k) = dk - \binom{k}{2}$ message symbols. To encode the message, choose n distinct nonzero elements $x_1, \ldots, x_n$ of F and use them to construct an $n \times d$ Vandermonde matrix Ψ with each row formed of consecutive powers of one of the x_i's. The $n \times d$ codeword matrix is found as

$$C = \Psi B.$$

The data retrieval proceeds as follows. Assume that the DC aims at retrieving $\mathcal{F}$ by accessing the stored contents of nodes $C_1, \ldots, C_k$ (or any other k-tuple of the nodes). Denote by Ψ_k the submatrix of Ψ formed by the first k rows of Ψ, and write it as $\Psi_k = [\Psi_{k,1} | \Psi_{k,2}]$, where $\Psi_{k,1}$ is a $k \times k$ Vandermonde matrix. Upon retrieving the information from the nodes $C_1, \ldots, C_k$, the DC has access to the $k \times d$ matrix

$$[\Psi_{k,1} S + \Psi_{k,2} T^{\intercal} \mid \Psi_{k,1} T], \tag{3.18}$$

where the left submatrix has k and the right $d-k$ columns. Since $\Psi_{k,1}$ is invertible, from the right submatrix the DC can find the matrix T. Once found, it gives access to the product $\Psi_{k,1}S$ and then to S, completing the decoding (data recovery) process.

Inspired by the polynomial description of PM MSR codes in [20], we now present a similar description of the PM MBR codes and show how this achieves the bound (3.17). Since B is a symmetric matrix, we can associate its elements with the coefficients of a symmetric polynomial $s(y,z)$ such that

$$\min(\deg_y(s), \deg_z(s)) \leq k-1 \text{ and } \max(\deg_y(s), \deg_z(s)) \leq d-1.$$

Next, we let node i store the d coefficients of the polynomial $g_i(z) = s(x_i, z)$, where $x_i \in F$ is one of the elements chosen above. Altogether this forms an equivalent description of the encoding procedure of the code. It is clear that retrieving the coefficients of any k of these polynomials results in the retrieval of the file: for instance, the coefficients of $g_i, i = 1, \ldots, k$ exactly correspond to the rows of the matrix (3.18).

The standard data retrieval scheme described above suggests acquiring all the coefficients of k polynomials. We observe that this is in fact not necessary because the file can be recovered by accessing exactly M elements of the codeword. Without loss of generality, assume that the nodes $1, 2, \ldots, k$ are contacted for data retrieval. Node i returns the $d-i+1$ symbols $\{g_i(x_j) : j = i, i+1, \ldots, d\}$. To see that this scheme achieves the bound (3.16), without loss of generality let $A = \{1, 2, \ldots, a\}$. Then the total data transmitted by the set A is $\sum_{i \in A} |\{g_i(x_j) : j = i, i+1, \ldots, d\}| = \sum_{i=k-a+1}^{k} (d-i+1)$ which matches the bound. The correctness of the scheme follows from the next lemma.

Lemma 3.6.3. *The set of symbols $\{g_i(x_j) : j = i, i+1, \ldots, d, i = 1, 2, \ldots, k\}$ are sufficient to recover the original M symbols.*

Proof. We have the following set of evaluations of the symmetric polynomial $s(y,z)$ at the points $\{(x_i, x_j) : j = i, i+1, \ldots, d, i = 1, 2, \ldots, k\}$. From node 1, the DC gets d evaluations of the polynomial $g_1(z)$ of

degree $d-1$, sufficient to find its coefficients. From node 2, the DC obtains $\{g_2(x_i) : i = 2, \ldots, d\}$; since it already knows $g_2(x_1) = s(x_2, x_1) = s(x_1, x_2) = g_1(x_2)$, altogether it has therefore access to d evaluations of $g_2(z)$ which again suffice to recover its coefficients. By induction, if the DC has recovered the coefficients of all $g_m(z)$ for $m \leq i$ for some $i < k$, then after acquiring further $d - i$ symbols from node $i+1$, it will have access to d evaluations of $g_{i+1}(z)$. This process results in recovery of all the polynomials $\{g_i(z) : 1 \leq i \leq k\}$, and this completes the data retrieval. □

Example 3.6.2. Consider again the graph from Example 3.4.1 but this time assume that codewords of a PM MBR code with parameters $[n = 7, k = 5, d = 6, l = 6, \beta = 1, M = 20]$ are placed on the nodes. Assume that DC has direct access only to node 1, i.e., $\bar{K} = \{1\}$. By the breadth-first search algorithm, the 5 nodes taking part in the data retrieval process are chosen to be nodes $1, 2, 3, 4$ and 5. Figure 3.3(a) shows the data transmission required in the traditional setting where each node sends its $l = 6$ symbols to the DC for retrieval of the file. Applying the Corollary 3.6.2(2), we have $H(R_{\{4\}}) \geq 2, H(R_{\{4,5\}}) \geq 5$ and so on. Figure 3.3(b) shows the optimal data transmission matching the bound (3.17) under the same connectivity constraints. Examining the results, we see that optimizing the communication results in moving 27 fewer symbols.

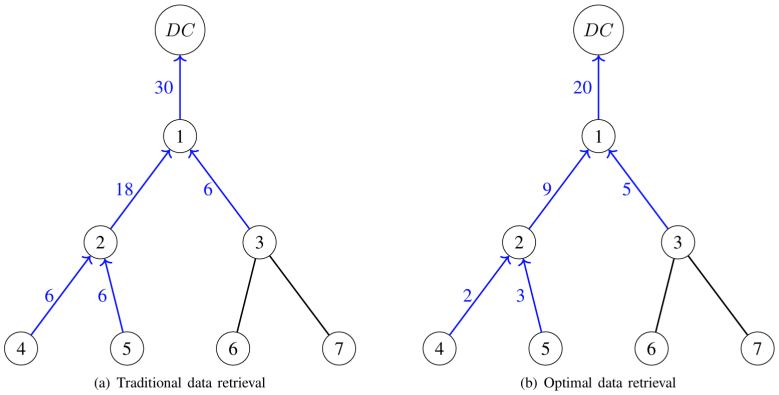


Figure 3.3: Traditional vs optimal transmission for data retrieval for PM MBR code. In part (b) the graph $G_{\bar{K},K}$ is formed of vertices 1 through 5, where $\bar{K} = \{1\}, K = \{2, 3, 4, 5\}$.

As noted above, even in the full connectivity setting when DC has direct access to every node, to perform data retrieval it suffices to download only M symbols, not $kl > M$ symbols as proposed in the original work [59]. We believe in fact that this is a general phenomenon that applies for all F-linear families of MBR codes.

3.7 Concluding remarks

In this chapter, we advanced our analysis of the problem of node repair on graphs, by explicitly constructing repair schemes for several families of regenerating codes. Although the proposed protocols of this chapter are not provably optimal, unlike their MSR counterparts from Chapter 2, they certainly provide significant savings in data transmission during node repairs. We further analyzed the other important functionality of such storage systems, namely the data retrieval task, under the present system model of graph-constrained connectivity and showed how IP can be beneficial for this task.

In the next chapter we explore yet another possibility for reducing communication complexity of repair even further: by considering a variant of regenerating codes which can support non-uniform contributions from helper nodes during repair. Furthermore, we also address security challenges posed by adversarial helper nodes, presenting protocols that preserve repair efficiency gains due to IP, despite possible data corruption by an adversary.

The results of this chapter appear in [51, 53].

Chapter 4: Repair on Graphs using Generalized Regenerating Codes

4.1 Introduction

4.1.1 Heterogeneous and graph-constrained storage systems

Most earlier works on regenerating codes, with a few exceptions mentioned below, assume that any d of the surviving $n-1$ nodes can serve as helpers, and the choice of a particular helper set is not addressed in the code design. Existing code constructions typically also assume that downloading the same amount of data from each of the helpers minimizes the communication complexity of repair, and in many cases (e.g., for MSR codes) one can show that this is indeed true. Constructions with nonuniform download are considered when the transmission cost from different helpers to the failed node is not the same, giving rise to *heterogeneous regenerating codes*. Different versions of such systems include models with two subsets of nodes assigned different communication costs [3], systems with different link capacities [83], rack-aware storage [12, 13, 30, 33, 80, 81], general models of clustered systems [57, 74], systems with varying amounts of data downloaded from the helper nodes [82], [42], and general systems with topology-aware repair [85].

The system model, so far considered in this thesis, which involves placing the nodes of the storage system on the vertices of a graph $G = (V, E)$, with the repair information flowing along the edges of the graph, has a natural connection towards such heteregeneous regenerating codes. Since the cost of sending a unit of information from v_i to v_j is determined by the graphical distance (the number of edges in a shortest path) $\rho(v_i, v_j)$ in G, there is a natural bias in the information cost of node repair in favor of the

helper nodes closer to the failed node v_f, suggesting that coding schemes should rely on heterogeneous codes. In the previous chapters, we showed how the idea of performing *intermediate processing* (IP) along the path towards the failed node may provide significant savings in terms of overall communication complexity of repair in the graph-constrained setting. The immediate follow up question then becomes of how to combine the idea of heterogeneity, i.e., non-uniform contributions from helper nodes, with the idea of intermediate processing (IP) to, potentially, reduce the repair complexity even further. In this chapter, we primarily explore this question.

4.1.2 Repair in the presence of adversarial nodes

The simplest model of errors in a graphical storage system arises from assuming unreliable links across the network. This problem has been widely analyzed in network coding literature; see [4] and references therein. For regenerating codes, a straightforward way of handling noise edges, if one assumes that a given edge does not introduce more than a fixed number of errors, is to simply encode each transmission using a local error correcting code with sufficient minimum distance and placing a decoder on every node.

The error process is more difficult to handle if the network includes adversarial nodes that try to interfere with the repair by introducing errors that can affect its outcome. If the information residing on a node is corrupted, it can spread through the network when this faulty node is chosen to be a helper. Previous works on error control during repair ([60], [86], [73] and others) all focus on the traditional model of direct connectivity. These schemes still work in the graph scenario if the nodes rely upon standard relaying of data, i.e., the AF strategy. At the same time, if the nodes perform intermediate processing, error amplification can happen, similar to what happens in network coding with errors. This is because even a single corrupted symbol can potentially affect all the linear combinations evaluated at the node. If left unchecked, data corruption can quickly spread through the network in the course of multiple failures.

Our main results for repair with errors are related to the model of a *limited-power adversary*. We

assume that the adversary may alter the contents of a subset of nodes $\Delta \subset D$ (not known to the system), but has no access to the computations performed at any of the nodes. In other words, in this model the helper nodes perform valid transformations of the data, but their own contents may be corrupted without their knowledge.

It is also possible to consider an *all-powerful adversary* that actively controls a vertex and can alter both the stored data and data that is relayed through that vertex. Coding in the presence of such an adversary appears to be a more difficult problem [35], [36], and we do not consider it here.

4.1.3 Overview of the results

The results of this chapter are summarized as follows:

- In Section 4.2 we derive a bound for the repair bandwidth (the cutset bound) for the heterogeneous setting, which includes a previously derived result in [3] as a special case. We also present a *stacking code construction*, similar to one in [42], showing that it attains the cutset bound at the MSR point. These codes support flexible download assignment mentioned above. Note that, while the results of the previous chapters show that it is possible to perform repair on graphs with download cost smaller than direct relaying, they do not support the varying download property.

- In Sec. 4.3 we turn our attention to IP repair under the GRC framework. We prove a lower bound on the amount of information sent by any subset of helpers for repair, which serves as a benchmark for IP repair in the network. This bound also extends previous results established for the case of uniform download in Chapters 2 and 3. We also show that the stacking construction proposed in Sec. 4.2 attains the mentioned lower bound with equality. Next, we present a simple linear-algebraic argument showing that any family of linear GRCs supports IP repair. The final result of Sec. 4.3 quantifies savings in repair bandwidth that result from using the IP repair scheme.

- In Sec. 4.4 we turn to the problem of optimizing the download amounts from different helpers based

on their distance to the failed node. An application of the result of [42] shows that, under both IP and AF strategies, if the number of helpers can be dynamically adjusted once we know the identity of the failed node, there always exists a uniform download scheme that minimizes the overall download. With this insight, we show that achieving the optimal download cost amounts to a combination of choosing the optimal repair degree while also carefully maximizing the benefits of IP. We further show that for codes with high rate, the best AF strategy is to involve all the remaining nodes in the repair process.

- In Sec. 4.5, we consider the repair task under the possible presence of adversaries where the main goal is to counter the spreading of errors while performing IP repair on the graph. Extending the Singleton bound from network coding, we derive a bound on the repair bandwidth in the presence of errors for IP repair. We also present a construction that supports repair of systematic nodes in the presence of a limited-power adversary in the network.

4.2 Generalized regenerating codes

4.2.1 Definition

Motivated by the graph model of the system discussed above, we analyze the problems of optimizing the repair bandwidth through nonuniform download, varying the repair degree, IP repair, and repair in the presence of an adversary. In graphical networks, sending the repair data along a path from the helper to the failed node incurs the cost proportional to the path length, which naturally introduces heterogeneity based on the graphical distance. Accordingly, we extend the definition of regenerating codes as follows.

Definition 4.2.1 (Generalized regenerating codes (GRCs))**.** *Let $\mathcal{B} = \{\beta_i\}_{i=1}^{d}$ be a set of d positive integers. An $[n, k, d, l, \mathcal{B}, M]$ GRC encodes a file $\mathcal{F}$ of size M symbols over a finite field F by storing l symbols in each of the n nodes such that*

1. *(reconstruction) the original file can be recovered by accessing any k out of n nodes;*

2. *(repair) the contents of any node $f \in [n]$ can be recovered by contacting a set $D \subseteq [n]\backslash\{f\}, |D| = d$ of nodes and downloading β_i symbols from node $\tau^{-1}(i)$, for any bijective mapping $\tau : D \to [d]$.*

The parameter d is called the repair degree.

The mapping τ corresponds to the allocation of contributions for repair to the set of the helpers, and it highlights the fact that the assignments can be arbitrary as long they form the set $\mathcal{B}$. Specifically, the amount of data downloaded from a given helper node may change from one repair round to another depending on its distance to the failed node. If all the β_i's are equal, we call such a repair procedure a *uniform download* scheme.

4.2.2 The cutset bound

In this section, we present a general form of the communication complexity bounds for the nonuniform download case. Previously this question was discussed in [93] based on the information flow graph approach, but the bound we obtain here is easier to interpret and use. For a finite field $F = \mathbb{F}_q$, we consider a code $\mathcal{C} \subset F^{nl}$ whose codewords $(C_i, i = 1, \ldots, n)$ are represented by $l \times n$ matrices over F. We assume that each coordinate (a vector in F^l) is written on a single storage node, and that a failed node amounts to having its coordinate erased.

Suppose that the information stored at the nodes is described by random variables $W_i, i \in [n]$ that have some joint distribution on $(F^l)^n$ and satisfy $H(W_i) = l$ for all i, where $H(\cdot)$ is the entropy. For a subset $A \subset [n]$ we write $W_A = \{W_i, i \in A\}$. We assume that $H(\mathcal{F}|W_A) = 0$ for any $A \subset [n], |A| = k$, which supports the data retrieval property. Let $f \in [n]$ be the failed node and let $D \subseteq [n] \setminus \{f\}, |D| = d$ be the set of helper nodes. Let S_i^f be the information provided to the failed node f by the ith helper node in the traditional fully connected repair scheme, and let $S_A^f = \{S_i^f : i \in A\}$ for any $A \subseteq D$. By definition

we have $H(S_i^f) = \beta_{\tau(i)}$, and

$$\begin{aligned} &H(W_K) = M, K \subset [n], |K| = k \\ &H(S_i^f|W_i) = 0, i \in D; \;\; H(W_f|S_D^f) = 0. \end{aligned} \tag{4.1}$$

Let $\Delta_r(\mathcal{B}) = \min_{R\subseteq[d],|R|=r} \sum_{i\in R} \beta_i$ denote the sum of r smallest elements from $\mathcal{B}$.

The following statement gives a general bound for information transmission during repair, giving a simple extension to the results in [68].

Theorem 4.2.1. *For an* $[n, k, d, l, \mathcal{B}, M]$ *GRC,*

$$M \leq \sum_{i=0}^{k-1} \min\{l, \Delta_{d-i}(\mathcal{B})\}. \tag{4.2}$$

Proof. For any $f \in [n]$, any $D \subseteq [n] \setminus \{f\}, |D| = d$ and any set $A \subset D$, we have

$$H(W_f|S_A^f, S_{D\setminus A}^f) = 0,$$

which implies

$$\begin{aligned} H(W_f|S_A^f) &= I(W_f; S_{D\setminus A}^f|S_A^f) \leq H(S_{D\setminus A}^f|S_A^f) \\ &\leq H(S_{D\setminus A}^f) \leq \sum_{i\in D\setminus A} H(S_i^f) \\ &= \sum_{i\in D\setminus A} \beta_{\tau(i)}. \end{aligned}$$

Since this is true for any bijective mapping τ, we conclude that

$$H(W_f|W_A) \leq H(W_f|S_A^f) \leq \min\{l, \Delta_{d-|A|}(\mathcal{B})\}. \tag{4.3}$$

Finally,

$$M = H(W_{[k]}) = \sum_{i=1}^{k} H(W_i|W_{[i-1]}) \leq \sum_{i=0}^{k-1} \min\{l, \Delta_{d-i}(\mathcal{B})\}.$$

Remark 4. *For the case of uniform download, the bound of this theorem reduces to the inequality* $H(W_i|W_A) \leq \min(l, (d - |A|)\beta$*, where* $A \subset [n], |A| \leq d, i \notin A$*, proved in [68]. If the set* $\mathcal{B}$ *contains only two distinct values, then* (4.2) *recovers the main result of [3], obtained here with a much shorter proof.*

As in the case of homogeneous systems [16], we define the following corner points of the above trade-off curve.

Definition 4.2.2. *The* Minimum Storage *(MSR) point of the bound* (4.2)*, is defined by* $l = \Delta_{d-k+1}(\mathcal{B})$*, and any code that achieves these parameters for a given set* $\mathcal{B}$ *is called an MSR code. Similarly, the Minimum Bandwidth (MBR) point is defined by* $l = \Delta_d(\mathcal{B}) = \sum_{i=1}^{d} \beta_i$*.*

4.2.3 Code construction

In this section, we describe a construction of an $[n, k, d, l, \mathcal{B}, M]$ GRC that meets the bound of Theorem 4.2.1 at the MSR point. This construction generalizes the construction of [42], and is reminiscent of multilevel concatenated codes of [6]. We further show here that for any set $\mathcal{B} = \{\beta_j\}_{j=1}^{d}$ the constructed code family achieves the minimal possible per-node storage parameter l and hence is MSR-optimal in the sense of Definition 4.2.2. In the next section we prove that this construction is also optimal for IP repair.

Given a repair degree d and a set of integers $\mathcal{B} = \{\beta_j\}_{j=1}^{d}$, we aim to construct a regenerating code that repairs any failed node f by downloading at most β_j symbols from node $\tau^{-1}(j)$ for any subset of helper nodes $D \subseteq [n] \setminus \{f\}$ and any permutation $\tau : D \to [d]$. By relabeling the nodes, we can always

assume that the set $\{\beta_j\}$ is sorted in nondecreasing order. Let $\mu = (\mu_1, \ldots, \mu_{d-k+1})$ be the binary vector with $\mu_j = \mathbb{1}_{(\beta_j > \beta_{j-1})}, j = 1, \ldots, d-k+1$, where $\beta_0 := 0$.

Construction 4.2.1. *Suppose that $\mathcal{B} = \{\beta_j\}_{j=1}^{d}$ and μ are as defined above, and let $S = \{j : \mu_j \neq 0\}$. For every $j \in S$ take an MSR code $\mathcal{C}_j$ with parameters*

$$[n, k, d-j+1, l_j = (d-j-k+2)(\beta_j - \beta_{j-1}),$$

$$\beta_j - \beta_{j-1}, M_j = kl_j].$$

The $[n, k, d, l, \mathcal{B}, M]$ GRC code is formed by stacking the codes $\{\mathcal{C}_j\}_{j \in S}$, where $l = \sum_{j \in S} l_j$ and $M = \sum_{j \in S} M_j$.

The intuition behind the construction is as follows. Upon arranging the β_js in nondecreasing order, for every j such that $\beta_j > \beta_{j-1}$, we add to the stack an MSR code with per node download equal to the gap $\beta_j - \beta_{j-1}$ and repair degree $d - j + 1$.

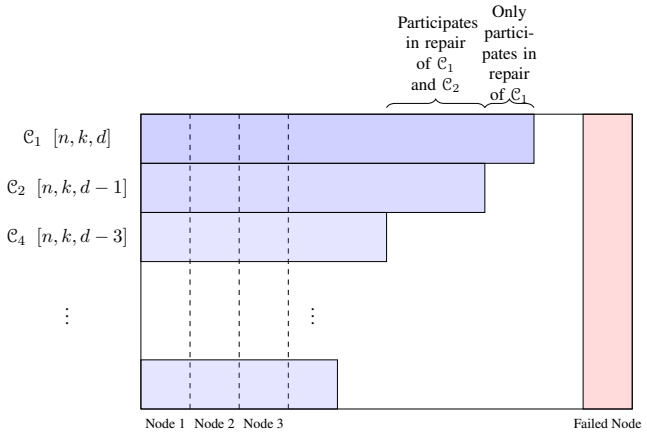


Figure 4.1: Example: Stacked MSR construction for $S = \{1, 2, 4, \cdots\}$.

Proposition 4.2.2. *The code in Construction 4.2.1 is an $[n, k, d]$ regenerating code that supports repair of any node f from a helper set $D \subseteq [n] \setminus \{f\}, |D| = d$ by downloading at most β_j symbols of F from node*

$\tau^{-1}(j)$ *for any permutation* $\tau : D \to [d]$.

Proof. The length n and the data reconstruction parameter k are the same for all the component codes and hence inherited directly. To prove the repair property, fix a helper set D and a permutation τ such that $\{\beta_j\}$ are in nondecreasing order. Node $\tau^{-1}(j)$ participates in the recovery of node f only in the component codes $\{\mathcal{C}_p : p \leq j\}$ and hence it sends a total of $\sum_{p=1}^{j}(\beta_p - \beta_{p-1}) = \beta_j$ symbols. □

The component codes can be chosen from a variety of known MSR constructions. Since the parameters of these outer codes depend on the given set $\{\beta_j\}$, a convenient choice is the product-matrix codes [59], which in their basic version work by downloading a single symbol from every helper. By stacking several codewords of these codes we can obtain any of the component codes $\mathcal{C}_j$ as in Fig. 4.1, thereby matching any set of per-node download values as required by the construction.

Next, we show that this construction attains the MSR point as per Definition 4.2.2.

Proposition 4.2.3. *Codes of Construction 4.2.1 meet the bound* (4.2) *with equality at the MSR point.*

Proof. The node size for the code of this construction equals

$$\begin{aligned} l &= \sum_{j=1}^{d-k+1} (d-j-k+2)(\beta_j - \beta_{j-1}) \\ &= \sum_{i=1}^{d-k+1} i(\beta_{d-i-k+2} - \beta_{d-i-k+1}) = \sum_{i=1}^{d-k+1} \beta_i. \end{aligned}$$

Since $\beta_j \geq \beta_{j-1}$ for all j, summing the first d of the β_js gives the minimum value over all permutations τ. Thus, the sum on the last line equals $\Delta_{d-k+1}(\mathcal{B})$, matching (4.2). □

4.3 IP repair

In the previous section we considered regenerating codes with nonuniform download in general, without associating them with repair on graphs. In the first part of this section, we continue this line of thought,

deriving a lower bound on the minimum required transmission for a set of helper nodes for the repair of a failed node. Then we turn to repair on graphs, showing that linear regenerating codes support IP repair on graphs, and that the stacking code family attains this lower bound.

4.3.1 Lower bound

In the next theorem, we prove a generalized version of the IP bound, extending Lemma 3.2.2.

Theorem 4.3.1. *Let $f \in [n]$ be the failed node. For a subset of helper nodes $E \subset D$, let R_E^f be a function of S_E^f such that $H(W_f|R_E^f, S_{D\setminus E}^f) = 0$. Then if $|E| \geq d - k + 1$,*

$$H(R_E^f) \geq M - \sum_{i=0}^{k-2} \min\{l, \Delta_{d-i}(\mathcal{B})\}. \tag{4.4}$$

Proof. Since we assumed that $H(W_f|R_E^f, S_{D\setminus E}^f) = 0$, all the more it is true that

$$H(W_f|R_E^f, W_{D\setminus E}) = 0. \tag{4.5}$$

We have $|D\setminus E| \leq k - 1$. Consider a set $A \subset E$ with $|A| = k - 1 - |D\setminus E|$. Now,

$$H(R_E^f, W_{D\setminus E}, W_A) = H(R_E^f, W_{D\setminus E}, W_f, W_A) \geq M, \tag{4.6}$$

where the equality in (4.6) follows from (4.5) and the chain rule, and the inequality follows from the reconstruction property because $|D\setminus E| + |A| + 1 = k$. Next, observe that

$$H(R_E^f, W_{D\setminus E}, W_A) \leq H(R_E^f) + H(W_{D\setminus E}, W_A),$$

and so

$$\begin{aligned} H(R_E^f) &\geq M - H(W_{D\setminus E}, W_A) \\ &\geq M - \sum_{i=0}^{k-2} \min\{l, \Delta_{d-i}(\mathcal{B})\}, \end{aligned}$$

where the last inequality follows from (4.3). □

Corollary 4.3.2. *(a) Assume that $\beta_i = \beta$ for all $i \in [d]$, then we have*

$$H(R_E^f) \geq M - \sum_{i=0}^{k-2} \min\{l, (d-i)\beta\}. \tag{4.7}$$

(b) For MSR codes, we have

$$H(R_E^f) \geq l = \Delta_{d-k+1}(\mathcal{B}) \tag{4.8}$$

(c) Assuming in addition uniform download with $\beta_i = \beta$ for all $i \in [d]$ at the MSR point, we have

$$H(R_E^f) \geq (d-k+1)\beta. \tag{4.9}$$

Proof. Part (a) follows directly from (4.4). To prove part (b), recall from (4.2) that at the MSR point, we have $l = \Delta_{d-k+1}(\mathcal{B})$ and $M = kl$, and use (4.4). Part (c) is immediate from (4.8). □

Theorem 4.3.1 bounds below the amount of data necessarily obtained from a subset $E \subset D$ irrespective of the processing performed by the nodes in E, including the IP repair on graphs, discussed in the next section. Eq. (4.9) in Corollary 4.3.2 is a restatement of Lemma 2.2.1. We already showed that $H(R_E^f) = l$ is achievable at the MSR point for the uniform download case. Below in this chapter we show that the value $H(R_E^f) = l$ can be achieved by some code families in the non-uniform download case as well. Note that, in general, for all other non-MSR points of the tradeoff curve, $\Delta_{d-k+1}(\mathcal{B}) < l$.

The above information-theoretic formulation of the repair problem is also valid for functional repair with a slight modification: for functional repair we only focus on the repair cycle of a single node from d helpers whose repair has already been completed (See Section VI of [68]). Hence the random variable W_f now denotes the data to be reconstructed at the failed node (which can be different from the data lost prior to this moment.). The next proposition forms an extension of Lemma 2.2.1, generalizing it to all functional and exact regenerating codes at all points of the trade-off curve.

Proposition 4.3.3. *For an optimal functional repair code that meets the cutset bound of Eq. (4.2) with equality, we have* $H(R_E^f) \geq \Delta_{d-k+1}(\mathcal{B})$. *Assuming in addition uniform download with* $\beta_i = \beta$ *for all* $i \in [d]$, *we have* $H(R_E^f) \geq (d-k+1)\beta$.

The proof follows straightforwardly by replacing M in (4.4) with the maximum value of M, given by the right-hand side of (4.2).

4.3.2 Nonuniform download and regenerating codes on graphs

Limited connectivity of the network is modeled as placing each node on a vertex of a graph $G(V, E)$ with $|V| = n$, where each node has direct access only to its immediate neighbors in G. Suppose further that the coordinate C_f for some $f \in [n]$ is erased, i.e., that the node $f \in [n]$ has failed. Below we denote the vertex in V that corresponds to f by v_f and use f and v_f interchangeably. Suppose further that the set $D \subset [n]$ of helper nodes is fixed and consists of the immediate neighbors of f as well as of some other nodes. To accomplish the repair, the helper nodes provide information which is communicated to f over the edges in $E_{f,D}$. Each helper node in the graph, starting from the nodes farthest from the failed node, sends data to the next node along the shortest path towards f. These nodes can be found by a simple breadth-first search on G starting at f. Denote by $G_{f,D} = (V_{f,D}, E_{f,D})$ the subgraph spanned by $\{f\} \cup D$, and let $T_{f,D}$ be a spanning tree of this subgraph with f as the root. Let $t = \max_{v \in D} \rho(v, f)$ be the height of this tree. We

will use the following notation for spheres and balls around f in $G_{f,D}$:

$$\Gamma_j(f) = \{v \in V_{f,D} : \rho(v, f) = j\},\ N_i(f) = \cup_{j=1}^{i} \Gamma_j(f),$$

and we refer to the vertices in $\Gamma_j(f)$ as the helper nodes in *layer* j. We have $V(T_{f,D}) = \sqcup_i \Gamma_i(f)$, where we denote $d_i := |\Gamma_i(f)| = d_i$, with $\sum_i d_i = d$. To accomplish the repair, the helper nodes provide information which is communicated to f over the edges in $E_{f,D}$. The overall repair bandwidth for the repair of f is the total amount of transmitted data across all edges.

Note that there can be multiple possible choices of $G_{f,D}$ and $T_{f,D}$ and the communication complexity of repair depends on this choice. The analysis can be made more concrete if we assume that G satisfies certain regularity assumptions.

Example 4.3.1. *As an example, suppose $G(V, E)$ is a connected t-regular graph. One way to guarantee the existence of a convenient repair tree is to consider graphs with girth g, in which case a ball of radius $\lfloor g/2 \rfloor - 1$ around any vertex is a tree with t immediate neighbors of the center and $t(t-1)^{i-1}$ vertices in layer i. A line of work starting with Margulis's paper [46] yielded constructions of such graph families with $g \geq C(n, t) \log_{t-1} n$, where n is the number of vertices and $C(n, t)$ is a constant. We give concrete examples of repair on these graphs below.*

4.3.3 IP repair for generalized regenerating codes

In this section we present a general framework for IP repair for any linear GRC. Define the *repair tree* of the failed node f as a spanning tree of $G_{f,D}$ with the root at f. The main result in this part is as follows.

Theorem 4.3.4. *(a)* (Existence of IP repair procedure) *Suppose that we are given an F-linear generalized regenerating code, a failed node f, and a helper node h that is on a path from a subset $B \subset D$ to f in the repair tree $T_{f,D}$. There exists an F-linear map that enables one to combine the information from the*

nodes in $B \cup \{h\}$, resulting in an l-dimensional vector sent to f to complete the repair.

*(b) (*Optimality of the stacking construction) *The IP repair procedure for codes of Construction 4.2.1 meets the lower bound (4.8) with equality.*

Proof. (a) The proof follows along the lines of the arguments given in Sec. 3.3, with slight modifications to account for the nonuniform contributions (different βs). Hence, it is not repeated.

(b) Observe that codes of Construction 4.2.1 are formed of F-linear MSR codes, so they are themselves F-linear, and therefore support IP repair. Clearly, they also minimize the amount of data sent by any subset of $d-k+1$ nodes at the MSR point.

For a given j and a code $\mathcal{C}_j$ it is possible to perform IP repair. Specifically, any subset of at least $d-j-k+2$ nodes can perform intermediate processing for $\mathcal{C}_j$ to compress their repair data to l_j symbols of F. Therefore overall, the subset of nodes of size $d-k+1$ or more can perform IP repair, compressing their data to $\sum_{j\in S} l_j = l$ symbols. □

4.3.4 Repair bandwidth gains with nonuniform download

The repair procedure for MSR codes that optimizes the overall communication complexity of repair with uniform download was presented in Chapter 2. It involves transporting the helper data towards the failed node, i.e., the root of the tree, along the edges of the tree, whereby nodes having more than $d-k+1$ children process the information and send l symbols relying on the IP technique. Let J be the set of nodes in $T_{f,D}$ with at least $d-k+1$ children and let $J_i = \Gamma_i(f) \cap J$ be the set of nodes in J at distance i from the root. For an $i \notin J$, let $\mathcal{P}(i)$ denote the nearest parent of node i in J, and if no such parent exists, then let $\mathcal{P}(i) = f$.

Clearly, the set J does not change when we switch from the uniform download model to the nonuniform one and vice versa. Furthermore, every node in J keeps transmitting l symbols by relying on the IP procedure. Indeed, if a node has $d-k+1$ or more children in the tree, they jointly must transmit at least

l symbols for repair because of the bound (4.8), irrespective of whether the β_i's are equal or different.

Assume now that nodes in layer i each contribute β_i symbols for repair with $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_t$. This can be accomplished by using an $[n, k, d, l, \mathcal{B}, M]$ MSR code from Construction 4.2.1 with the set $\mathcal{B}$ formed of β_i's, each appearing d_i times, for all $i \in [t]$. Let $\delta_i = \beta - \beta_i$ where $\beta = \frac{l}{d-k+1}$ is the uniform download value at the MSR point for the same per-node storage l. Furthermore, let

$$t' := \max\Big(s \in [t] : \sum_{i=s}^{t} d_i \geq d - k + 1\Big).$$

It can be checked that the vector μ of Construction 4.2.1 in this case is given by

$$\mu = e_1 + \sum_{i=t'}^{t} e_{x_i},$$

where $x_i = \sum_{j=i}^{t} d_j + 1$ and $e_x \in \{0, 1\}^{d-k+1}$ is a vector with a single 1 in position x.

We denote by $\Lambda(T_{f,D})$ the total communication complexity of repairing node f using the repair tree $T_{f,D}$. Additionally, we use superscripts IP and AF for repair with the IP and AF schemes, and subscripts U and NU for uniform and nonuniform download. For instance, $\Lambda_{\mathrm{U}}^{IP}$ refers to the communication complexity of repair with IP and uniform download.

The next theorem states conditions under which the nonuniform download model attains bandwidth gains over the uniform one.

Theorem 4.3.5. *For a fixed $\mathcal{B} = \{\beta_i\}$ and $\tau : D \to [d]$, suppose that a codeword of an $[n, k, d, l, \mathcal{B}, M]$ GRC at the MSR point is encoded on the vertices of a graph G. There exists an explicit scheme that has the following communication complexity of repair for the node f with repair tree $T_{f,D}$:*

$$\Lambda_{\mathrm{NU}}^{\mathrm{IP}}(T_{f,D}) = \sum_{i \in J \setminus \{f\}} l + \sum_{i=1}^{t} \sum_{j \in \Gamma_i \setminus J_i} \rho(j, \mathcal{P}(j))\beta_i. \tag{4.10}$$

Furthermore, the nonuniform contribution model attains savings over the uniform one whenever

$$\sum_{i=1}^{t} \sum_{j \in \Gamma_i \setminus J_i} \rho(j, \mathcal{P}(j))\delta_i > 0 \tag{4.11}$$

subject to

$$\sum_{i=t}^{t'+1} d_i\delta_i + \Big(d - k + 1 - \sum_{i=t}^{t'+1} d_i\Big)\delta_{t'} = 0. \tag{4.12}$$

Proof. By Theorem 4.3.4(a), every node in J may transmit only l symbols. Expression (4.10) is obtained simply by accounting for the number of symbols transmitted by each node. The condition in Eq. (4.11) is obtained by comparing the expressions for uniform and nonuniform download. To obtain (4.12), recall our notation Δ_{d-k+1} defined before Theorem 4.2.1. For the graph case considered, it has the following form:

$$\Delta_{d-k+1} = \sum_{i=t'+1}^{t} d_i\beta_i + \Big(d - k + 1 - \sum_{i=t'+1}^{t} d_i\Big)\beta_{t'} = l,$$

where the last equality follows from Theorem 4.2.1. Rewriting this using the δ_i's, we obtain Eq. (4.12). □

Remark 5. *Note that the set J may not include all the nodes capable of performing IP. Indeed, for a choice of $\mathcal{B} = \{\beta_i\}$, any node in the repair tree that accumulates the repair data of a set A such that $\sum_{i \in A} \beta_i \geq l$ can gainfully perform IP. Hence, the minimum communication complexity of repair can potentially be even lower than Eq. (4.10).*

Ways of applying Theorem 4.3.5 depend on the structure of the specific graph family. One such example is given next.

Example 4.3.2. *Consider the t-regular Cayley graphs mentioned in Example 4.3.1. Suppose that the repair tree $T_{f,D}$ is formed of a layers, where $a < g/2$, then*

$$d_i = t(t-1)^{i-1}, i \leq a-1 \text{ and } d_a = d - \sum_{i=1}^{a-1} t(t-1)^{i-1}.$$

Suppose further that $d_a + d_{a-1} \geq d - k + 1$. *To simplify the analysis, we are not including IP since it is somewhat independent of the current discussion and can be easily incorporated into it. The overall repair bandwidth for a uniform contribution repair scheme for an* $[n, k, d, l, \beta = \frac{l}{d-k+1}, M]$ *MSR code is* $\Lambda_U^{AF}(T_{f,D}) = \beta \sum_{i=1}^{a} id_i$. *Now let us switch to the nonuniform scheme with helper nodes in layer* i *contributing* β_i *symbols each, with* β_i*'s nonincreasing. From (4.12), we have that* $d_a\delta_a + (d - k + 1 - d_a)\delta_{a-1} = 0$, *with* $\delta_i = \beta - \beta_i$, *and the repair bandwidth under this scheme is* $\Lambda_{NU}^{AF}(T_{f,D}) = \sum_{i=1}^{a} id_i\beta_i$. *Note that if* $\delta_a > 0$ *then* $\delta_{a-1} < 0$ *and* $\delta_i \leq \delta_{a-1}$ *for all* $i \leq a - 2$, *so we let* $\delta_i = -\frac{d_a}{d-k+1-d_a}\delta_a$ *for all* $i \leq a - 1$ *and observe that the savings in the nonuniform setting are*

$$\begin{aligned} &\Lambda_U^{AF}(T_{f,D}) - \Lambda_{NU}^{AF}(T_{f,D}) \\ &= \sum_{i=1}^{a} id_i\delta_i = \frac{d_a\delta_a}{d - k + 1 - d_a}\Big(\sum_{i=1}^{a-1}(a - i)d_i - a(k - 1)\Big). \end{aligned} \tag{4.13}$$

In summary, using the nonuniform scheme results in savings whenever the expression in the parentheses is positive, which is possible for small k.

We end this section with a remark on the generalization of some of the results. Using other interior point exact-repair code families as component codes in the stacking construction of Section 4.2.3, one can get non-MSR GRCs. Existence of IP repair for such codes still follows from Part (a) of Theorem 4.3.4, however, optimality, i.e., Part (b) of Theorem 4.3.4 might not hold anymore. Conditions for the advantage of the nonuniform download scheme over the uniform one for such interior point GRCs can be stated similarly to Theorem 4.3.5.

4.4 Optimizing the helper data and the repair degree

In the analysis up until now, we have focused on the problem of repair on graphs with a fixed repair degree d. We showed that the framework of GRCs along with the IP technique give nontrivial ways to minimize

the overall communication complexity of node repair in the graph constrained setting by lowering the contribution of the farthest away nodes at the expense of increasing the contributions from the nearer nodes. This gives rise to the question of the limits of this exchange. In the limiting case, one might stop accessing data from the farthest nodes altogether, effectively *decreasing* the repair degree d of the repair process. Indeed, the repair degree need not be a fixed parameter and may be dynamically adjusted if necessary. Universal constructions of regenerating codes proposed in [86], [43] support the option of such dynamical adjustment.

At the same time, the original work on regenerating codes [16] showed that (in the fully connected setting), the repair bandwidth is a decreasing function of the repair degree. This implies that *increasing* the repair degree reduces the communication complexity. Based on these contrasting observations, a natural question to ask is what is the optimal choice of the repair degree under the communication constraints described by a graph.

In this section we find the minimum repair bandwidth obtained by optimizing the repair degree and the download amounts with GRCs introduced in Construction 4.2.1 above. We also show that under the AF repair scheme (i.e., with no intermediate processing), for certain parameter regimes, the overall repair bandwidth decreases as more and more helper nodes are involved and hence the optimal choice of d is $n-1$. This result is established for both deterministic and random graphs.

Example 4.4.1. *To motivate the discussion, let us return to Example 4.3.2, showing that the idea of adjusting the repair degree naturally arises from GRCs. We will show that the maximum savings can be attained by adjusting the repair degree and switching to the uniform assignment. Expression* (4.13) *implies that, as we increase* $\delta_a = \beta - \beta_a$ *above, the advantage of the nonuniform assignment* $\Lambda_U^{AF}(T_{f,D}) - \Lambda_{NU}^{AF}(T_{f,D})$ *increases, attaining the maximum when* $\delta_a = \beta$ *or* $\beta_a = 0$*. At this point,*

$$\beta_i = \beta - \delta_{a-1} = \beta + \frac{d_a \delta_a}{d - k + 1 - d_a}$$

$$= \frac{l}{d - d_a - k + 1} = \frac{l}{d' - k + 1}, 1 \leq i \leq a - 1,$$

where $d' = d - d_a$ is the new repair degree upon discarding the nodes in layer a. Note also that every helper node in layers $a-1$ and below contributes equally. In summary, the maximum savings are obtained with a uniform assignment, but with a smaller repair degree.

In the remainder of this section we address the question of the optimal choice of the repair degree using the already established framework of GRCs with IP. First, note that for a fixed repair tree, finding the minimum bandwidth can be formulated as a linear programming (LP) problem. It was shown in [42] that such a problem for GRCs always has an optimal solution that supports uniform contribution from a possibly smaller set of helper nodes. Leveraging this result, we incorporate the IP repair technique into the optimization, which becomes feasible because we need to consider only the uniform contribution schemes.

To form the optimization problem, without loss of generality, we start with $D = [n-1]$ and assume that node $i \in D$ contributes $\beta_i \geq 0$ symbols for the repair of f, where $\beta_i = 0$ accounts for the variation of the repair degree. By Theorem 4.2.1, we have that $l = \Delta_{n-k}(\mathcal{B})$, which imposes constraints on our choice of β_i's. Recalling Theorem 4.3.5, given a specific choice of the repair tree $T_{f,D}$, our goal is to minimize

$$\Lambda_{\text{NU}}^{\text{IP}}(T_{f,D}) = \sum_{i \in J \setminus \{f\}} l + \sum_{j=1}^{t} \sum_{i \in \Gamma_j \setminus J_j} \rho(i, \mathcal{P}(i))\beta_i \tag{4.14}$$

over the choices of $\mathcal{B} = \{\beta_i\}_{i \in D}$ such that $\Delta_{n-k}(\mathcal{B}) \geq l$ and $0 \leq \beta_i \leq l$ for all $i \in D$. Note that letting some β_i's to be 0 does not change the set J, since due to the constraint $l \leq \Delta_{n-k}(\mathcal{B})$, each node in J can still perform IP[1]. Since the first term on the right in (4.14) depends only on the set J, the minimization is restricted only to the second term. Setting the weights in the linear program to $b_i = \rho(i, \mathcal{P}(i))$, we obtain

[1]This observation shows that the savings from IP do not depend on the chosen value of the repair degree.

the following linear program:

$$\begin{aligned} &\min_{\{\beta_i\}} \sum_{i\in D\setminus J} b_i\beta_i \\ &\text{subject to} \sum_{i\in A}\beta_i \geq l, \ \ \forall A\subseteq D, |A| = n-k \\ &0\leq \beta_i \leq l, \ \ i\in D. \end{aligned} \tag{4.15}$$

Below we assume that the costs b_i's are in nonincreasing order relative to i, which is always possible by relabeling the nodes in D. This LP problem has been studied in [42] where the authors claimed that the optimal solution takes the form given in the next theorem. The proof does not seem to appear in the published literature, so we have included it in the Appendix.

Theorem 4.4.1. ([42], Theorem 1]) *There exists an optimal solution of the above LP such that*

$$\beta_i^* = \begin{cases} 0 & 1\leq i\leq n-d-1 \\ \frac{l}{d-k+1} & n-d\leq i\leq n-1 \end{cases} \tag{4.16}$$

for some d in the range $k\leq d\leq n-1$.

This implies that among the uniform download schemes, there exists an assignment of β_i's that minimizes the repair bandwidth. Let $\mathcal{T}_f$ denote the set of all spanning trees of G rooted at f. Any rooted spanning tree $T_{f,D}$ with $D\subset[n]\setminus\{f\}$ is a subtree of some element $T_f\in\mathcal{T}_f$. Let $\sigma(v)$ be the number of descendants of $v\in D$ in the tree $T_{f,D}$. Then the minimum repair bandwidth is given as follows:

Corollary 4.4.2. *Using the stacking construction of codes and the repair (transmission) scheme found in Theorem 4.3.5, the minimum total communication complexity of repairing the failed node f is*

$$\min_{T_f\in\mathcal{T}_f} \ \min_{\substack{T_{f,D}\subseteq T_f \\ D:k\leq|D|\leq n-1}} \Lambda_U^{IP}(T_{f,D})$$

where

$$\Lambda_U^{IP}(T_{f,D}) = \sum_{h \in D} \min\{\sigma(h) + 1, (d - k + 1)\} \frac{l}{(d - k + 1)}.$$

This implies that for every failed node, there exists at least one *optimal* repair tree and a corresponding *optimal* set of helpers such that the uniform contribution from them, combined with IP, gives the minimum complexity of repair across all GRC-IP schemes. Furthermore, this optimal choice of helpers can be found in time polynomial in the number of vertices.

4.4.1 Optimizing the repair degree

Sometimes using IP repair may be too complicated for the storage system. In this case the nodes rely only on the AF strategy and do not perform intermediate processing. This assumption enables us to further simplify the minimization in Cor. 4.4.2, as shown below. In the following analysis, we will use the notation

$$\Gamma_j^G(f) = \{v \in V : \rho(v, f) = j\},\ N_i^G(f) = \cup_{j=1}^{i} \Gamma_j^G(f),$$

to denote spheres and balls around f in the graph G, with the superscript denoting the difference with the notation used in Section 4.3.2.

4.4.1.1 DETERMINISTIC GRAPHS

With the above assumption, the repair bandwidth does not depend on the choice of the repair tree, and the weights in the linear program are simply the distances to the failed node. We obtain the following LP

problem:

$$\begin{aligned} &\min_{\beta_1,\ldots,\beta_{n-1}} \sum_{i\neq f} \rho(i,f)\beta_i \\ &\text{subject to} \sum_{i\in A}\beta_i \geq l, \quad \forall A \subseteq \{1,\ldots,n-1\}, |A| = n-k \\ &0 \leq \beta_i \leq l, \quad i \in \{1,\ldots,n-1\}. \end{aligned} \tag{4.17}$$

By Theorem 4.4.1, for some repair degree d^*_{AF} there exists an optimal solution with uniform node contributions. For a given $d \in \{k, k+1, \ldots, n-1\}$, define

$$\Lambda^{\text{AF}}_{\text{U}}(d) = \sum_{i=1}^{t} i d_i \frac{l}{d-k+1},$$

where t is such that $|N^G_t(f)| \geq d > |N^G_{t-1}(f)|$, and $d_i = |\Gamma^G_i(f)|$ for $1 \leq i \leq t-1$, $d_t = d - |N^G_{t-1}(f)|$. Then, the optimal repair degree is given by

$$d^*_{\text{AF}} = \underset{d\in\{k,k+1,\ldots,n-1\}}{\arg\min} \Lambda^{\text{AF}}_{\text{U}}(d).$$

As mentioned before, for complete graphs, the quantity $\Lambda^{\text{AF}}_{\text{U}}(d)$ is a decreasing function of the repair degree and so $d^*_{\text{AF}} = n-1$. We will show that this is also true in general for arbitrary graphs provided that the code rate is large enough. As the first step, we prove that involving more than k nearest nodes in the repair process entails saving of the repair bandwidth.

Proposition 4.4.3. *For a repair graph, if $k \in \Gamma^G_a(f)$, then*

$$\Lambda^{AF}_U(k) \geq \Lambda^{AF}_U(|N^G_a(f)|).$$

Proof. If $k = |N^G_a(f)|$, then the claim is trivially true, so assume that $|N^G_a(f)| \geq k+1$. Let $p =$

$k - |N_{a-1}^G(f)|$. For $d \in \{k, k+1, \ldots, |N_a^G(f)|\}$, we have

$$\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d) = \sum_{i=1}^{a} i d_i \frac{l}{d-k+1}.$$

For any $m \in \{1, 2, \ldots, |N_a^G(f)| - k\}$,

$$\begin{aligned}
&\Lambda_{\mathrm{U}}^{\mathrm{AF}}(k) - \Lambda_{\mathrm{U}}^{\mathrm{AF}}(k+m) \\
&\quad = \sum_{i=1}^{a-1} i d_i l \Big(1 - \frac{1}{1+m}\Big) + al\Big(p - \frac{p+m}{1+m}\Big) > 0
\end{aligned}$$

for $p, m \geq 1$. □

This claim can be further specified if the underlying graph is regular. We will show that for codes of sufficiently high rate, the repair bandwidth decreases with the increase of the repair degree.

Theorem 4.4.4. *Consider repair on a t-regular graph. Let $m = \max_{h \in V \setminus \{f\}} \rho(f, h)$ be the maximum height of the repair tree. If*

$$k > 1 + \max_{1 \leq a \leq m} \sum_{i=1}^{a-1} t(t-1)^{i-1}(1 - i/a),$$

then $d_{AF}^* = n - 1$.

Proof. Proposition 4.4.3 implies that $\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d)$ becomes smaller as d is increased from k in the corresponding layer. We first show that if $d = \sum_{i=1}^{a-1} t(t-1)^{i-1}$ for some a, then $\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d) \geq \Lambda_{\mathrm{U}}^{\mathrm{AF}}(d+1)$, that is involving one extra node from the $(a+1)$-th layer does not increase the overall bandwidth. Let

$$K(a) = \sum_{i=1}^{a-1} it(t-1)^{i-1}, \quad C(a) = \sum_{i=1}^{a-1} t(t-1)^{i-1} + 1.$$

Indeed,

$$\begin{aligned}
&\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d) - \Lambda_{\mathrm{U}}^{\mathrm{AF}}(d+1) \\
&\quad = \sum_{i=1}^{a-1} it(t-1)^{i-1} \frac{l}{d-k+1} \\
&\qquad\qquad - \Big(\sum_{i=1}^{a-1} it(t-1)^{i-1} + a \Big) \frac{l}{d-k+2} \\
&\quad = \frac{K(a)l}{C(a)-k} - \frac{(K(a)+a)l}{C(a)-k+1} \\
&\quad = \frac{K(a)l}{(C(a)-k)(C(a)-k+1)} - \frac{al}{C(a)-k+1}
\end{aligned}$$

which is nonnegative whenever $(C(a)-k)a \leq K(a)$ or $k \geq C(a) - \frac{K(a)}{a}$. Now we show that, as we keep involving more nodes in this a-th layer, the overall bandwidth can only further decrease. For some $x \in \{1, \ldots, t(t-1)^{a-1}\}$, let $d = \sum_{i=1}^{a-1} t(t-1)^{i-1} + x$. Then

$$\begin{aligned}
\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d) &= \Big(\sum_{i=1}^{a-1} it(t-1)^{i-1} + ax \Big) \frac{l}{d-k+1} \\
&= \frac{l(K(a)+ax)}{C(a)-k+x}.
\end{aligned}$$

The function $f(x) = \frac{K(a)+ax}{C(a)-k+x}$ is a continuous function of x in the interval $[1, t(t-1)^{a-1}]$ and is decreasing on x whenever $(C(a)-k)a < K(a)$ or equivalently $k > C(a) - \frac{K(a)}{a}$. Hence the values $\{\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d)\}$ at the points $d \in \{C(a), C(a)+1, \ldots, C(a+1)-1\}$ form a decreasing sequence. The result now follows since this analysis applies to all $a \in \{1, \ldots, m\}$. □

Example 4.4.2. *For the sake of example, suppose that the network is given by the Petersen graph, a cubic graph on $n = 10$ vertices with 15 edges. Suppose that a code of length 10 is used to support node repair. The diameter of the graph is 2, so the height of the complete repair tree is $m = 2$. By Theorem 4.4.4, if the parameter k of the code is 3 or more, then $d_{AF}^* = 9$.*

Example 4.4.3. *Let i be an integer satisfying $i^2 \equiv -1 \pmod{29}$, e.g., $i = 12$. By the LPS construction [45], the Cayley graph of the group PSL$(2, 29)$ with generating set*

$$\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 1+2i & 0 \\ 0 & 1-2i \end{bmatrix},$$
$$\begin{bmatrix} 1-2i & 0 \\ 0 & 1+2i \end{bmatrix}, \begin{bmatrix} 1 & 2i \\ -2i & 1 \end{bmatrix}, \begin{bmatrix} 1 & -2i \\ 2i & 1 \end{bmatrix}$$

is a 6-regular graph with $n = 12180$ vertices and girth 10. Considering repair on this graph, we see that by Theorem 4.4.4, if $k \geq 5000$ then $d^*_{AF} = n - 1$.

Theorem 4.4.4 can be generalized for arbitrary graphs as follows.

Theorem 4.4.5. *For any node f in the graph $G = (V, E)$, let $m_f = \max_{h \in V \setminus \{f\}} \rho(f, h)$. For $a \in \{1, \ldots, m_f\}$, let $K(a) = \sum_{i=1}^{a-1} i|\Gamma_i^G(f)|, C(a) = |N_{a-1}^G(f)| + 1$. If $k > \max_{1 \leq a \leq m_f}(C(a) - \frac{K(a)}{a})$, then for the repair of node f, $d^*_{AF} = n - 1$.*

The proof is very similar to the proof of Theorem 4.4.4, and is therefore omitted.

4.4.1.2 Random graphs

In this section, we consider the case when the underlying graph $G(V, E)$ is sampled uniformly from the Erdös-Rényi ensemble of graphs $\mathcal{G}_{n,p}$ with $p \in (0, 1)$. Denote a random element from the ensemble by $\mathbb{G}_{n,p}$ and the spheres and balls around a node f accordingly by $\Gamma_i^{\mathbb{G}}(f)$ and $N_i^{\mathbb{G}}(f)$. For a fixed value of d, the repair problem was considered and analyzed in [52], where we established the parameter regimes for which IP repair is beneficial to the AF repair. Following that work, here we assume that $p \gg \frac{\log n}{n}$, which ensures that $\mathbb{G}_{n,p}$ is connected w.h.p., and that $k = \Theta(n)$ for the code rate to be asymptotically positive.

We say that t- *layer repair* of the failed node f *is possible* if

$$\mathbb{P}(|N_t^{\mathbb{G}}(f)| \geq d) \to 1 \text{ as } n \to \infty.$$

and call the minimum t for which this holds the *threshold depth* for repair. We have

$$\Lambda_{\mathrm{U}}^{\mathrm{AF}}(d) = \frac{l(td - |N_{t-1}^{\mathbb{G}}(f)|)}{d - k + 1}.$$

We will use the following two results regarding the random Erdös-Rényi graphs (below $\mathbb{P} = \mathbb{P}_{\mathcal{G}_{n,p}}$).

Lemma 4.4.6 ([8], p. 50; [26], Sec.7.1)**.** *(i) If* $p^2 n - 2\log n \to \infty$, *and* $n^2(1-p) \to \infty$, *then*

$$\mathbb{P}(\mathsf{diam}(\mathbb{G}_{n,p}) = 2) \to 1.$$

(ii) Suppose that the functions $t = t(n) \geq 3$ *and* $0 < p = p(n) < 1$ *satisfy*

$$(\log n)/t - 3\log\log n \to \infty, \quad p^t n^{t-1} - 2\log n \to \infty,$$

$$p^{t-1} n^{t-2} - 2\log n \to -\infty,$$

then $\mathbb{P}(\mathsf{diam}(\mathbb{G}_{n,p}) = t) \to 1$.

Lemma 4.4.7 ([14], Lemma 3)**.** *Suppose that* $p \geq \frac{\log n}{n}$. *For any* $\epsilon > 0$ *and all* $i = 1, \ldots, \lfloor \log n \rfloor$

$$\mathbb{P}(|\Gamma_i^{\mathbb{G}}(x)| \leq (1+\epsilon)(np)^i) \geq 1 - 1/\log^2 n$$

$$\mathbb{P}(|N_i^{\mathbb{G}}(x)| \leq (1+2\epsilon)(np)^i) \geq 1 - 1/\log^2 n.$$

We now state the result for the optimal repair degree.

Theorem 4.4.8. *If* $(np)^{t-1} = o(n)$, $\frac{(np)^t}{n} - 2\log n \to \infty$ *and* $k = \Theta(n)$, *then*

$$\mathbb{P}(d^*_{AF} = n-1) \to 1.$$

Proof. Define the two events $E = \{|N^{\mathbb{G}}_{t-1}(f)| < k\}$ and $F = \{|N^{\mathbb{G}}_t(f)| = n-1\}$. Since, $(np)^{t-1} = o(n)$ abd $k = \Theta(n)$, by Lemma 4.4.7,

$$\mathbb{P}(E) \geq \mathbb{P}(|N^{\mathbb{G}}_{t-1}(f)| \leq (1+2\epsilon)(np)^{t-1}) \to 1$$

and by Lemma 4.4.6,

$$\mathbb{P}(F) = \mathbb{P}(|N^{\mathbb{G}}_t(f)| = n-1) = \mathbb{P}(\mathsf{diam}(\mathbb{G}_{n,p}) = t) \to 1$$

and so $\mathbb{P}(E \cap F) \geq \mathbb{P}(E) + \mathbb{P}(F) - 1 \to 1$. For any element under the event $E \cap F$, by definition the k-th nearest node from the root f is at distance t from it, and the remaining $n-k-1$ nodes are also at distance t from f. Hence, by using Proposition 4.4.3, $\Lambda^{\text{AF}}_{\text{U}}(d)$ is a nonincreasing function of d for $d \in \{k, k+1, \ldots, n-1\}$. □

As a final remark, note that for codes of sufficiently high rate, the repair degree $d^*_{\text{AF}} = n-1$. It is not difficult to show that this conclusion is also true for d^*_{IP} as the set J does not change with the change of the repair degree. This is formally justified by an appropriate modification of the proof of Theorem 4.4.4.

4.5 Error Correction during Repair with GRCs on Graphs

In this section, we extend the analysis of node repair using the GRC framework to the adversarial case, assuming that parts of the network (some helper nodes participating in the repair) contribute corrupted information. As explained in Section 4.2, such nodes can have a detrimental effect on the repair process

due to the possible error amplification by IP. We prove a variant of the cutset bound of Theorem 4.2.1 for this case and propose a code construction to harness the bandwidth saving capabilities of IP while also counteracting the effects of adversarial nodes. Our proposed solutions make use of tools and ideas from network coding.

4.5.1 Network Coding preliminaries

In this subsection, we briefly describe the general problem of single source network coding and introduce the network singleton bound. A network $(\hat{G}, v_s, U, \mathcal{R})$ consists of a directed acyclic graph $\hat{G} = (\hat{V}, \hat{E})$ with a single source node $v_s \in \hat{V}$, a set of destination nodes $U \subseteq \hat{V} \setminus \{v_s\}$, and a set of non-negative integers $\mathcal{R} = \{R_{a,b} : (a,b) \in \hat{E}\}$ to denote the set of capacities of the edges (links) in the network. A link with unit capacity transmits one symbol of F per single use. To accommodate integer capacities $R_{a,b} > 1$, we simply add parallel unit-capacity edges between the nodes a and b. An error is said to occur when the output of such a unit-capacity edge is different from the input. For a partition (A, B) of the set $\hat{V}$, let $\text{cut}_{\hat{G}}(A, B) = \{(a,b) \in \hat{E} : a \in A, b \in B\}$ and let $c_{\hat{G}}(A, B) = \sum_{(a,b)\in \text{cut}_{\hat{G}}(A,B)} R_{a,b}$. Additionally, for any two nodes s and u, let

$$c_{\hat{G}}(s, u) = \min_{\substack{(S,U) \text{ is a partition of } \hat{V}, \\ s\in S, u\in U}} c_{\hat{G}}(S, U).$$

Definition 4.5.1. *A network code over the code alphabet $\mathcal{X}$ for the network $(\hat{G}, v_s, U, \mathcal{R})$ with source message set $\mathcal{Z}$ is a family of local encoding functions $\{\phi_{(a,b)} : (a,b) \in \hat{E}\}$ such that $\phi_{(v_s,b)} : \mathcal{Z} \to \mathcal{X}^{r_{v_s,b}}$ for every $(v_s, b) \in \hat{E}$ and $\phi_{(a,b)} : \prod_{(c,a)\in\hat{E}} \mathcal{X}^{r_{c,a}} \to \mathcal{X}^{r_{a,b}}$ for every $(a,b) \in \hat{E}, a \neq v_s$, where $0 \leq r_{a,b} \leq R_{a,b}$. Such a code is said to correct t errors if it recovers the source message at each of the destinations as long as at most t* unit capacity *links are subjected to errors.*

The network Singleton bound [90] is as follows:

Lemma 4.5.1. *Let $(\hat{G}, v_s, U, \mathcal{R})$ be an acyclic network and let $\hat{c} = \min_{u\in U} c_{\hat{G}}(s, u)$. If there exists a*

q-ary t error-correcting code for the network then the number of messages that can be transmitted from s to U is at most $q^{\hat{c}-2t}$.

4.5.2 A cutset bound for repair with adversarial nodes

We now extend the above framework to node repair. Recall that helper node $h \in D$ of a GRC code contributes $\beta_{\tau(h)}$ symbols for the repair of failed node f for some assignment $\tau : D \to [d]$. The following definition relies on notation from Sec. 4.3.2.

Definition 4.5.2. *For a failed node f and its repair graph $G_{f,D} = (V_{f,D}, E_{f,D})$, assumed to be a tree, define a directed acyclic graph $\tilde{G}_{f,D} = (\tilde{V}_{f,D}, \tilde{E}_{f,D})$ in the following way:*

1. *Let $\tilde{V}_{f,D} = V_{f,D}$ and for every edge in $E_{f.D}$ define an edge in $\tilde{E}_{f,D}$ whose direction is defined by the direction of the data flow in the repair process.*
2. *Next, for every $h \in V_{f,D}$, $h \neq v_f$ add another vertex $\tilde{h}$ to $\tilde{V}_{f,D}$ and add a directed edge $(\tilde{h}, h)$ to $\tilde{E}_{f,D}$.*

The purpose of adding additional nodes in the graph is to formally define the *limited-power adversary* that we are considering. Recalling the notation introduced in Sec. 4.3.3, we assume that for a node $h \in V_{f,D} \setminus \{v_f\}$, $\tilde{h} \in \tilde{V}_{f,D}$ stores W_h and computes the functions $\mathcal{G}^{\tau}_{h,f}$. The repair data then is transmitted to $h \in \tilde{V}_{f,D}$ via the directed edge $(\tilde{h}, h)$ which, after possibly receiving the repair data from other nodes, computes the function $I^{\tau}_{h,f}$ (denoted earlier in Sec. 4.3.2 by $I_{h,f}$ and forwards to the next node. The adversary can now be assumed to have control only over the nodes of the former type. The following relationship holds between the cuts of original repair graph $G_{f,D}$ and the modified graph $\tilde{G}_{f,D}$:

Proposition 4.5.2. *For any set $A \subset D$ in $G_{f,D}$, let $\tilde{A} = \{\tilde{h} : h \in A\}$ in $\tilde{G}_{f,D}$. Then*

$$\text{cut}_{\tilde{G}}(A \cup \tilde{A}, \tilde{V}_{f,D} \setminus (A \cup \tilde{A})) = \text{cut}_{G}(A, V_{f,D} \setminus A).$$

Proof. Clearly, $\text{cut}_G(A, v_f \cup \{D \setminus A\}) \subseteq \text{cut}_{\tilde{G}}(A \cup \tilde{A}, \tilde{V}_{f,D} \setminus (A \cup \tilde{A}))$ as we have not deleted any of the edges involved in the repair. Further, every newly added edge has either both ends in $\tilde{A}$ or both ends outside of it, so none of them can be a part of $\text{cut}_{\tilde{G}}(A \cup \tilde{A}, \tilde{V}_{f,D})$. □

Now we state our main result of this section which gives a variation of Theorem 4.2.1 for the adversarial case. Let $\Omega_r(\mathcal{B}) = \max_{R \subseteq [d], |R|=r} \sum_{i \in R} \beta_i$ denote the sum of r largest elements from $\mathcal{B}$.

Theorem 4.5.3. *Suppose that the data stored on the graph G is encoded using an $[n, k, d, l, \mathcal{B} = \{\beta_j\}_{j=1}^d, M]$ Regenerating code. For the repair of a failed node f, let D be the set of chosen helper nodes. Suppose the limited power adversary has control over a set $T \subset D$, $|T| \leq t$ of these helper nodes. For any subset $A \subseteq D$ of size at least $d - k + 1 + 2t$ that contains T, we have*

$$c(A, V_{f,D} \setminus A) \geq M - \sum_{i=1}^{k-1} \min\{l, \Delta_{d-i+1}(\mathcal{B})\} + 2\Omega_t(\mathcal{B}).$$

Proof. We begin with transforming the repair problem into a network coding problem $(\hat{G}, v_s, U, \mathcal{R})$, and then apply the network Singleton bound. For a selected repair protocol and transmission scheme, we set $\hat{G}$ to be the directed graph $\tilde{G}_{f,D}$, introduce a dummy source node connected to all the vertices $\tilde{A} = \{\tilde{h} : h \in A\}$ of $\tilde{V}_{f,D}$ by infinite-capacity edges, and set the sink U to be the failed node v_f. Fix some assignment τ and set the capacity of the edges to be equal to the number of symbols transmitted over them as determined by the repair protocol. Note that the capacity of an edge of the form $(\tilde{h}, h)$ for $h \in D$ is set to $\beta_{\tau(h)}$ by this assignment. Hence an adversary controlling a set of T nodes in $G_{f,D}$, can inject at most $\sum_{h \in T} \beta_{\tau(h)}$ errors. In network coding terms, this implies that the adversary can cause errors in at most $\sum_{h \in T} \beta_{\tau(h)}$ unit-capacity edges in $\tilde{G}_{f,D}$.

We phrase the rest of the proof using the notation used to prove Theorem 4.2.1. Recall that R_A^f is the random variable which is a function of the contents of the helper node set A in the original graph $G_{f,D}$ such that $H(W_f | R_A^f, S_{D \setminus A}^f) = 0$. In words, R_A^f is the random variable jointly produced from

the data stored at the set A of the graph $G_{f,D}$ that is to be sent to the failed node such that the repair process is successful. Switching to the language of network coding, the set of vertices $\tilde{A} \in \tilde{V}_{f,D}$ wants to communicate the message R_A^f to the sink node v_f via the network $\tilde{G}_{f,D}$, where at most $\sum_{h\in T} \beta_{\tau(h)}$ edges can inject errors. If the random variable R_A^f is supported on a set $\mathcal{Z}$ and if $\hat{c}$ is the mincut between the source and the destination, then by Lemma 4.5.1, $\log|\mathcal{Z}| \leq \hat{c} - 2\sum_{h\in T} \beta_{\tau(h)}$. Since $H(R_A^f) \leq \log|\mathcal{Z}|$, we obtain

$$H(R_A^f) \leq \hat{c} - 2\sum_{h\in T} \beta_{\tau(h)}.$$

By Proposition 4.5.2,

$$\begin{aligned} c_G(A, V_{f,D} \setminus A) &= c_{\tilde{G}}(A \cup \tilde{A}, \tilde{V}_{f,D} \setminus (A \cup \tilde{A})) \\ &\geq \hat{c} \geq H(R_A^f) + 2\sum_{h\in T} \beta_{\tau(h)}. \end{aligned}$$

Since this is true for all choices of τ, the tightest lower bound is obtained when, for some τ, $\sum_{h\in T} \beta_{\tau(h)} = \Omega_t(\mathcal{B})$. Now substituting the lower bound of Theorem 4.2.1 concludes the proof. □

Corollary 4.5.4. *At the MSR point, the lower bound takes the form*

$$c(E, V_{f,D} \setminus E) \geq \Delta_{d-k+1}(\mathcal{B}) + \Omega_t(\mathcal{B}),$$

which under the uniform download assumption further simplifies to

$$c(E, V_{f,D} \setminus E) \geq (d - k + 1 + 2t)\beta = l + 2t\beta.$$

4.5.3 Code Construction

In this section, we present a code construction that combines error correction with IP, and analyze its performance relative to the bound of Theorem 4.5.3. The construction supports error control of systematic nodes in the encoding and is based upon concatenating rank-metric codes with GRC codes at the MSR point, i.e., Construction 4.2.1. Such concatenation was previously used in [73] for error correction during data recovery with full connectivity between the nodes. However, our purpose is to correct errors during the repair process itself, as was done in [86], without sacrificing the benefits of IP on graphs.

Rank-metric codes: Recall that a rank metric code $\mathcal{C}$ is an F-linear subspace of the space of matrices $F^{n\times m}$ with distance $d(A,B) = \mathrm{rk}(A-B)$. By $d_{\min}(\mathcal{C})$ we denote the minimum distance of the code. For our construction we use a classic family of MRD codes, namely the *Gabidulin codes*. To define them, recall that a linearized polynomial $f(x) \in \mathbb{F}_{q^m}[x]$ of q-degree t is defined as

$$f(x) = \sum_{i=0}^{t} a_i x^{q^i}, \quad a_t \neq 0.$$

Definition 4.5.3. *Let $N \leq m$ be integers and let $g_1, \cdots, g_N$ be elements of $\mathbb{F}_{q^m}$ linearly independent over $\mathbb{F}_q$. An $[N, K, D = N-K+1]_{q^m}$ Gabidulin code maps a K-tuple $f_0, \ldots, f_{K-1}$ of elements of $\mathbb{F}_{q^m}$ to an N-tuple $\mathbf{c} = (f(g_1), f(g_2), \cdots, f(g_N))$, where $f(x) = \sum_{i=0}^{K-1} f_i x^{q^i}$ is a linearized polynomial.*

Code Construction: The code construction is a concatenation of Gabidulin codes and MSR codes. Let $\mathcal{C}_1$ be an $[N, K]_{q^m}$ Gabidulin code and let $\mathcal{C}_2$ be an F-linear systematic GRC code at the MSR point with parameters $[n, k, d, l = N, \mathcal{B} = \{\beta_j\}_{j=1}^{d}]$. The data to be encoded comprises Kkm message symbols of $F = \mathbb{F}_q$, or equivalently Kk symbols of F_{q^m}. Partition them in k blocks of K elements and encode each block using code $\mathcal{C}_1$. This gives a matrix A of dimensions $k \times N$ over $\mathbb{F}_{q^m}$.

Next fix a basis of $\mathbb{F}_{q^m}$ over F and expand each row of the matrix A, obtaining k matrices $B^{(i)}, i = 1, \ldots, k$ over F. Let $B_j^{(i)}$ be row $j = 1, \ldots, m$ of the matrix $B^{(i)}$. For a fixed j we form a $k \times N$ matrix $R_j = ((B_j^{(1)})^\intercal, \ldots, (B_j^{(k)})^\intercal)^\intercal$.

The code $\mathcal{C}_2$ defines an F-linear encoding map $F^{kN} \xrightarrow{\mathcal{C}_2} F^{nN}$. We assume that the encoding is *systematic*, i.e., there are some k nodes that contain the kN data symbols. Encoding the matrices R_j with the code $\mathcal{C}_2$, we obtain m codewords $C_j = \mathcal{C}_2(R_j), j = 1, \ldots, m$, viewed as $n \times N$ matrices over F.

Finally, for a fixed $i = 1, \ldots, n$, we take the rows $(C_j)_i$ and place them on the storage node i. Thus, each node contains m N-dimensional vectors over F, each of which is a coordinate of its codeword of the MSR code $\mathcal{C}_2$.

Proposition 4.5.5. *The contents of any of the k systematic nodes, viewed as an N dimensional vector over $\mathbb{F}_{q^m}$, forms a codeword of $\mathcal{C}_1$.*

The proof follows from the fact that the encoding for these nodes is an identity map, so mapping their contents back to $\mathbb{F}_{q^m}$ recovers the original codewords of $\mathcal{C}_1$.

Proposition 4.5.6. *The resulting code satisfies the k-node data reconstruction and d-node repair properties.*

The proof follows immediately from the properties of the MSR code $\mathcal{C}_2$.

Remark 6. *Note that for recovering a file of size kKm, the required download is kNm symbols over $\mathbb{F}$, i.e., the communication complexity increses by a factor of $\frac{1}{R_1}$ where $R_1 = \frac{K}{N}$ is the rate of $\mathcal{C}_1$. If the k nodes contacted for data reconstruction are all the systematic nodes of $\mathcal{C}_2$, then this extra download is not required, since each of the systematic nodes can locally decode its own Gabidulin codeword and send the decoded symbols, which requires kKm symbol transmissions.*

Next we show that the above code construction supports repair of systematic nodes with IP despite the presence of a limited number of adversarial nodes.

Theorem 4.5.7. *Let v_f be a systematic node. Suppose that a subset $T \subset D$ of helper nodes with $|T| \leq t$ are controlled by a limited-power adversary. If $N-K \geq 2\Omega_t(\mathcal{B})$, then the repair procedure of v_f recovers the original data.*

Proof. Recall that the adversary corrupts the data of at most t nodes $\{W_h : h \in T\}$, but it does not affect the evaluation of the functions $\{\mathcal{G}^{\tau}_{h,f}\}$ and $\{I^{\tau}_{h,f}\}$ defined in Sec. 4.3.3. Since the MSR code is defined over F, these functions are F-linear. W.l.o.g. suppose that the adversary changes W_h to $W_h + Z_h$, where $Z_h \in \mathbb{F}^{l}_{q^m}$. Node h produces the faulty helper data

$$\mathcal{G}^{\tau}_{h,f}(W_h + Z_h) = \mathcal{G}^{\tau}_{h,f}(W_h) + \mathcal{G}^{\tau}_{h,f}(Z_h) = S^{\tau}_{h,f} + \tilde{Z}^{\tau}_{h,f}$$

where $\tilde{Z}_{h,f} \in \mathbb{F}^{\beta_{\tau(h)}}_{q^m}$. Performing the IP transform $I_{h,f}$ on this data, we obtain

$$I^{\tau}_{h,f}(S^{\tau}_{h,f} + \tilde{Z}^{\tau}_{h,f}) = U^{\tau}_{h,f}S^{\tau}_{h,f} + U^{\tau}_{h,f}\tilde{Z}^{\tau}_{h,f}.$$

Note that the rank of the error term $U^{\tau}_{h,f}\tilde{Z}^{\tau}_{h,f}$ is at most $\beta_{\tau(h)}$. Since all the nodes in the network operate

faithfully, oblivious to the data corruption in some of them, at the failed node the helper data has the form

$$\sum_{h\in D} U_{h,f}^{\tau} S_{h,f}^{\tau} + \sum_{h\in T} U_{h,f}^{\tau} \tilde{Z}_{h,f}^{\tau} = W_f + \sum_{h\in T} U_{h,f}^{\tau} \tilde{Z}_{h,f}^{\tau}.$$

Observe that

$$\begin{aligned} \mathrm{rk}_q(\sum_{h\in T} U_{h,f}^{\tau} \tilde{Z}_{h,f}^{\tau}) &\leq \sum_{h\in T} \mathrm{rk}_q(U_{h,f}^{\tau} \tilde{Z}_{h,f}^{\tau}) \\ &\leq \sum_{h\in T} \beta_{\tau(h)} \leq \Omega_t(\mathcal{B}). \end{aligned}$$

By Proposition 4.5.5, W_f is a codeword of an $[N, K]$ Gabidulin code that can correct up to $\frac{N-K}{2} \geq \Omega_t(\mathcal{B})$ rank errors. Hence, the failed node will be able to correct the errors and recover its value. □

Example 4.5.1. *Consider the graph with 10 nodes shown in Fig. 4.2 and assume that the node v_f has failed, and all the other nodes form the helper set D. Suppose that one of the nodes, say node 3, is adversarial, i.e., its contents have been arbitrarily altered. To choose a family of MSR codes, take the codes of [86] with parameters $[n = 10, k = 5, d = 7, l = 15, \beta = 5]$. Using AF repair, we contact $d + 2t = 9$ nodes and download 5 symbols of helper data from each of them, totaling 95 symbol transmissions. Using the code construction described above with an $[n = 10, k = 5, d = 9, l = 25, \beta = 5]$ MSR code and a $[N = 25, K = 15, D = 11]$ MRD code, we will be able to perform IP, correct one error, and achieve the communication complexity of 85 symbols. Note that both codes store a file of size 75.*

Although the code satisfies the k-node data recovery and d-node repair properties, it is not an MSR code because it does not meet the cutset bound with equality. Because of the *two-layer* coding process, the rate of the resulting code is reduced from $\frac{k}{n}$ to $\frac{Kk}{Nn}$ and, as is to be expected, the rate decreases as we go for greater distance in the Gabidulin code, i.e., increase $N - K$.

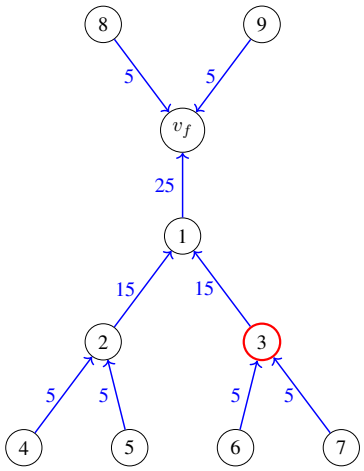


Figure 4.2: Repair with errors. Transmission from Node 1 to v_f is brought down from 35 to 25 by using our code construction.

4.6 Concluding remarks

The results of this chapter suggest that design of codes for a distributed storage system with graphical constraints is governed by the following principles. If we are free to adjust the repair degree in different rounds of repair, then we should use a code that supports *multiple repair degrees* such as the constructions in [86], [43], choosing the number of helpers that optimizes the repair bandwidth under the *uniform download* assumption. If the graph is sufficiently regular so that this degree does not depend on the failed node, we can instead rely on standard code families designed for a *fixed repair degree*. If the repair degree is fixed by the system constraints, we should use the stacking construction of Section 4.2.3, which allows to modify the download amounts based on the distance from the helpers to the failed node in each round of repair. In other words, if a node is included in the set of helpers for a particular failed node and is close to this node, it provides a larger proportion of the repair data. If in the next round this node is again a helper, but is far in the graph from the (new) failed node, it provides a small amount of data or, possibly, no data at all. These solutions depend on the constraints of the storage systems. Whichever of these alternatives the designer follows, it is also possible to incorporate IP repair with any of the mentioned code constructions,

gaining further savings of the repair complexity.

Furthermore, when some parts of the network provide corrupted data during repair, it is still possible to perform node repair on graphs reliably and efficiently without sacrificing the benefits of IP.

The results of this chapter appear in [53, 56].

Chapter 5: Future Directions & Open Problems

Many facets of designing regenerating codes on networks described by general graphs still await their study. While we have hinted at some of these problems throughout the text, we summarize the most interesting ones below.

Improving the protocols for repair of interior point codes: Although the repair protocols proposed for MSR codes in Chapter 2 meet the lower bound on repair complexity, we note that there exists a gap between what is achievable through our proposed IP technique and the lower bounds proved for interior point codes in Chapter 3. We have mentioned that there exists a gap between the tradeoff curve for functional and exact repair and improved bounds on the file size have been obtained for exact repair codes. It is plausible that some of the tools used to derive these bounds could be used to improve the Intermediate Processing bounds proposed in this work. On the other hand, for specific explicit code families, it might be possible to identify better IP protocols by exploiting the underlying algebraic structures of the interior point codes.

Finding better constructions for generalized regenerating codes: We showed that the use of generalized regenerating codes coupled with the idea of nonuniform contributions can reduce the overall repair bandwidth in graphs. We also remarked that the stacking construction, used in Chapter 4 to construct optimal codes at the MSR point, can be used in conjunction with other non-MSR codes to construct families of generalized regenerating codes at the interior points. However, due to the fact that the resultant node size is a sum of that of the component codes, the stacking idea may result in significant system con-

straints. Hence, it is of interest to design alternative algebraic constructions for such codes that support the flexibility of nonuniform contributions during different rounds of repair. This would replace the stacking construction, while possibly reducing the node size (subpacketization) of the coding scheme. While this would be an interesting advance, we believe that this is a difficult problem.

Constructing codes for the adversarial scenario: Another interesting code construction challenge realtes to the adversarial case and involves the construction of coding schemes for the adversarial error model that support optimal overhead for the repair of all nodes. Recall that the linear nature of the IP schemes proposed in this thesis naturally exposes them to error propagation, where a single erroneous symbol supplied by a corrupted helper can spread across all symbols, potentially corrupting all the data of the newly repaired node. At the same time, forgoing IP in the presence of an adversary, while allowing error correction capability, can significantly increase the overall communication bandwidth for node repair in graphs. While we showed that it is possible to combat the effect of errors and maintain (to an extent) the bandwidth saving capabilities of IP, the codes that we constructed are not optimal with respect to the cutset bound of Theorem 4.5.3, and they only allow for the repair of systematic nodes. Hence, lifting this restriction and constructing optimal codes for the repair of all nodes which support both IP and error correction remains an interesting open challenge.

Extensions to general distributed function computation: Finally, we remark that the idea of node repair using regenerating codes is a special case of distributed function computation on encoded data. The general description of the later problem is as follows. Suppose that we encode a data vector $x \in \mathbb{F}^k$ to an n-length codeword $c = \mathcal{C}(x)$ using an error-correcting code $\mathcal{C}$ and store each coordinate of the codeword in a different storage node. A function evaluator wishes to compute some function $F(x)$ of the data where $F : \mathbb{F}^k \rightarrow \mathbb{F}$ belongs in some class of functions $\mathcal{F}$. Using an MDS code for encoding, this can be done by connecting to any k nodes and downloading their contents, resulting in download complexity

of k symbols. The question then arises if this computation can be performed with lower complexity by possibly contacting more than k nodes and downloading smaller amounts from them. Such a low-bandwidth function evaluation requirement arises in several practical scenarios, including gradient coding [77], homomorphic secret sharing [9] and general coded computation [17, 39, 92].

It is clear that this question reduces to the problem of regenerating codes when $\mathcal{F} := \{f_i(x) = \mathcal{C}(x)_i : i = 1, \cdots, n\}$. Recently, it was shown [72] that low-bandwidth function evaluation schemes for linear functions exist for Reed-Solomon codes. Based on the motivation and results of this thesis, it is therefore an interesting open direction to explore the problem of low-bandwidth distributed function computation when communication avenues between the nodes of the network are constrained by a graph, potentially extending our results.

Appendix A: Omitted Proofs

A.1 Proof of Lemma 2.2.4

1) By the assumption (2.10), given the contents of all the nodes in $D\backslash E$, the information contained in R_E^F is sufficient to repair the nodes $\{v_i : i \in F\}$, i.e.,

$$H(W_F|R_E^F, W_{D\backslash E}) = 0. \tag{A.1}$$

We have $|D\backslash E| \leq k - h$. Consider a set $A \subset E$ with $|A| = k - h - |D\backslash E|$. Now,

$$H(R_E^F, W_{D\backslash E}, W_A) = H(R_E^F, W_{D\backslash E}, W_F, W_A) \geq kl \tag{A.2}$$

where the equality in (A.2) follows from (A.1) and the chain rule, and the inequality follows from the MDS property of MSR codes because $|D\backslash E| + |A| + |F| = k$. Next observe that

$$\begin{aligned} H(R_E^F, W_{D\backslash E}, W_A) &\leq H(R_E^F) + H(W_{D\backslash E}, W_A) \\ &= H(R_E^f) + (k-h)l \end{aligned} \tag{A.3}$$

where the equality again uses the independence of any $k - h$ coordinates in an MDS code. Combining (A.2) and (A.3), we obtain the claimed inequality.

For Part (2), let $A \subseteq D\backslash E$ such that $|A| = k - h$ and let $I = D\backslash\{E \cup A\}$. By the assumption

(2.10), we have

$$H(W_F|R_E^F, W_A, S_I^F) = 0. \tag{A.4}$$

Now,

$$H(R_E^F, W_A, S_I^F) = H(R_E^F, W_F, W_A, S_I^F) \geq kl \tag{A.5}$$

where the equality in (A.5) follows from (A.4) and the chain rule, and the inequality follows from the MDS property and the fact that $|A| = k - h$. Next observe that

$$\begin{aligned} & H(R_E^F, W_A, S_I^F) \\ & \leq H(R_E^F) + H(W_A) + H(S_I^F) \\ & \leq H(R_E^f) + H(W_A) + \sum_{i \in D \setminus \{E \cup A\}} H(S_i^F) \\ & = H(R_E^F) + (k-h)l + \frac{h(d-(k-h)-|E|)l}{d-k+h} \end{aligned} \tag{A.6}$$

where we again use the independence of any $k - h$ coordinates in an MDS code. Combining (A.5) and (A.6), we obtain the claimed inequality.

A.2 Proof of Lemma 3.5.1

(see [22]; [58, p.631-2]) Let $A_{\sim i} = A \setminus \{i\}, A_y = A \cup \{y\}, A_{\sim i,y} = (A_y)_{\sim i}$. Below we denote the elements of the matrices R, D and Φ by lowercase letters. Recall also that $w_{S,i}$ denote elements of the set $\mathcal{W}$.

$$\begin{aligned} \sum_{i \in A} (-1)^{\tau_A(i)} R_{A\setminus\{i\},:} D_{:,i} &= \sum_{i \in A} (-1)^{\tau_A(i)} \sum_{L \subset [d], |L|=m} r_{A \sim i, L} d_{L,i} \\ &= \sum_{i \in A} (-1)^{\tau_A(i)} r_{A \sim i, A} d_{A,i} + \sum_{i \in A} (-1)^{\tau_A(i)} \sum_{y \in [d] \setminus A} r_{A_{\sim i}, A_{\sim i,y}} d_{A_{\sim i,y}, i} \end{aligned}$$

$$= \sum_{i\in A} \phi_{i,1} d_{A,i} + \sum_{i\in A} (-1)^{\tau_A(i)} \sum_{y\in[d]\setminus A} (-1)^{\tau_{A_{\sim i,y}}(y)} \phi_{y,1} d_{A_{\sim i,y},i}$$

$$= \sum_{i\in A} \phi_{i,1} d_{A,i} + \sum_{y\in[d]\setminus A} \phi_{y,1} \sum_{i\in A} (-1)^{\tau_A(i)+\tau_{A_{\sim i,y}}(y)} w_{A_y,i}.$$

Now for $i \neq y$,

$$\begin{aligned}
\tau_A(i) + \tau_{A_{\sim i,y}}(y) &= |\{j \in A : j \leq i\}| + |\{x \in A_{\sim i,y} : x \leq y\}| \\
&= |\{j \in A_y : j \leq i\} - \mathbb{1}(y < i) + |\{x \in A_y : x \leq y\}| - \mathbb{1}(i < y) \\
&= |\{j \in A_y : j \leq i\} + |\{x \in A_y : x \leq y\}| - 1 \\
&= \tau_{A_y}(i) + \tau_{A_y}(y) - 1,
\end{aligned}$$

and we obtain

$$\begin{aligned}
\sum_{i\in A} (-1)^{\tau_A(i)} R_{A\setminus\{i\},:} D_{:,i} &= \sum_{i\in A} \phi_{i,1} d_{A,i} + \sum_{y\in[d]\setminus A} \phi_{y,1} \sum_{i\in A} (-1)^{\tau_{A_y}(i)+\tau_{A_y}(y)-1} w_{A_y,i} \\
&= \sum_{i\in A} \phi_{i,1} d_{A,i} + \sum_{y\in[d]\setminus A} (-1)^{\tau_{A_y}(y)} \phi_{y,1} \sum_{i\in A} [-(-1)^{\tau_{A_y}(i)} w_{A_y,i}] \\
&= \sum_{i\in A} \phi_{i,1} d_{A,i} + \sum_{y\in[d]\setminus A} (-1)^{\tau_{A_y}(y)} \phi_{y,1} (-1)^{\tau_{A_y}(y)} w_{A_y,y} \\
&= \sum_{i\in A} \phi_{i,1} d_{A,i} + \sum_{y\in[d]\setminus A} \phi_{y,1} d_{A,y} = \sum_{i\in[d]} d_{A,i} \phi_{i,1} = c_{A,1}.
\end{aligned}$$

A.3 Proof of Theorem 4.4.1:

Let $\mathcal{L}$ be the solution space of $\{\beta_i\}$s such that $\beta_1 \leq \beta_2 \leq \cdots \leq \beta_{n-k} = \beta_{n-k+1} = \cdots = \beta_{n-1}$. We know that the optimal solution of the LP lies in $\mathcal{L}$. We call an assignment of $\{\beta_i\}$s satisfying the constraints of the problem and having $\beta_i = 0$ for $i < n - d$ a d-repair scheme. Let $\mathcal{L}_d$ be the set of feasible d-repair

schemes, i.e.,

$$\begin{aligned}\mathcal{L}_d = \{\{\beta_i\} : 0 = \beta_1 = \cdots = \beta_{n-d-1} < \beta_{n-d} \\ \leq \beta_{n-d+1} \leq \cdots \leq \beta_{n-k} = \beta_{n-k+1} = \cdots = \beta_{n-1}\}.\end{aligned} \tag{A.7}$$

We call an assignment of download amounts *uniform* if $\beta_i = \frac{l}{d-k+1}$ for all $i \geq n-d$ any other assignment with at least one $\beta_i \neq \beta := \frac{l}{d-k+1}$ a *nonuniform* one. Clearly,

$$\mathcal{L} = \cup_{d=k}^{n-1} \mathcal{L}_d.$$

A priori, the uniform assignment is not necessarily a minimizer of the objective function in (4.15). Depending on the costs, there may exist nonuniform assignments that give a lower value of the objective function than the uniform assignment in some $\mathcal{L}_d$. The proof idea is to show that for any such nonuniform assignment in $\mathcal{L}_d$, there exists a $d' < d$ for which the uniform assignment in $\mathcal{L}_{d'}$ gives at most the same value of the objective function as given by the nonuniform assignment in $\mathcal{L}_d$. Let $\{\beta_i\}$ be such a nonuniform assignment in $\mathcal{L}_d$ such that

$$\sum_{i=n-d}^{n-1} b_i \beta_i < \sum_{i=n-d}^{n-1} b_i \beta. \tag{A.8}$$

Since

$$\sum_{i=1}^{n-k} \beta_i = \sum_{i=n-d}^{n-k} \beta_i \geq l, \tag{A.9}$$

and since the β_i's are nondecreasing, for any nonuniform assignment,

$$\beta_{n-k} = \beta + c \tag{A.10}$$

for some integer $c, 0 < c \leq l - \beta$.

At the same time, there will be at least one $i \in \{n-d, \ldots, n-k-1\}$ for which $\beta_i < \beta$. To see that inequality (A.8) can hold, rewrite it as

$$\sum_{i=n-d}^{n-1} b_i(\beta_i - \beta) < 0$$

and observe that, since $\beta_i = \beta + c$ for all $i \geq n-k$, (A.8) implies that

$$\sum_{i=n-k}^{n-1} b_i c < \sum_{i=n-d}^{n-k-1} b_i(\beta - \beta_i) \leq b_{n-d} c.$$

Here the last inequality holds because the b_i's are nonincreasing. So if $b_{n-d} > \sum_{i=n-k}^{n-1} b_i$, then a nonuniform assignment in $\mathcal{L}_d$ is going to give a lower value of the objective function than the uniform assignment in $\mathcal{L}_d$. Now we show that, even if this is the case, we can lower the value of d to some d' such that, with uniform assignment in $\mathcal{L}_{d'}$, the objective function does not increase.

Below we will assume that $(\beta + c)|l$ (it is always possible to choose l large enough so that it is divisible by the l.c.m.$(k+1, \ldots, n-1)$). With this assumption, let us take $d' = \frac{l}{\beta+c} + k - 1$ and consider the uniform assignment in $\mathcal{L}_{d'}$. Letting $\beta' = \beta + c$, we have $l = (d' - k + 1)\beta'$ and

$$\sum_{i=n-d}^{n-k-1} \beta_i \geq l - \beta' = \sum_{i=n-d'}^{n-k-1} \beta',$$

Since the b_i's are nonincreasing on i, the sum $\sum_{i=n-d}^{n-k-1} b_i \beta_i$ takes the smallest value when $\beta_i = 0$ for $n - d \leq i \leq n - d' - 1$ and $\beta_i = \beta'$ for $n - d' \leq i \leq n - 1$. Hence, the uniform assignment in $\mathcal{L}_{d'}$ results in communication cost at most equal to the cost of the initial nonuniform assignment in $\mathcal{L}_d$ that we started with. Since this argument applies for any d, there always exists a minimizing solution to Problem 4.15 in the form of Eq. (4.16) for some d.

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] S. Akhlaghi, A. Kiani, and M. R. Ghanavati, "Cost-bandwidth tradeoff in distributed storage systems," *Computer Communications*, vol. 33, no. 17, pp. 2105–2115, 2010.

[3] ——, "Cost-bandwidth tradeoff in distributed storage systems," *Computer Communications*, vol. 33, no. 17, pp. 2105–2115, 2010.

[4] A. Beemer, A. Kılıç, and A. Ravagnani, "Network decoding," *IEEE Transactions on Information Theory*, vol. 69, no. 6, pp. 3708–3730, 2023.

[5] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Broadcasting with side information: Bounding and approximating the broadcast rate," *IEEE Trans. Inform. Theory*, vol. 59, no. 9, pp. 5811–5823, 2013.

[6] È. L. Blokh and V. V. Zyablov, "Coding of generalized concatenated codes," *Problemy Peredachi Informatsii*, vol. 10, no. 3, pp. 45–50, 1974.

[7] B. Bollobás, "A probabilistic proof of an asymptotic formula for the number of labelled regular graphs," *European Journal of Combinatorics*, vol. 1, no. 4, pp. 311–316, 1980.

[8] ——, "The diameter of random graphs," *Trans. AMS*, vol. 267, no. 1, pp. 41–52, 1981.

[9] E. Boyle, N. Gilboa, Y. Ishai, H. Lin, and S. Tessaro, "Foundations of homomorphic secret sharing," in *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 94. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, pp. 21:1–21:21.

[10] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length mds codes with optimal repair in distributed storage," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 1850–1854.

[11] J. Castura and Y. Mao, "Rateless coding and relay networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 27–35, 2007.

[12] Z. Chen and A. Barg, "Explicit constructions of MSR codes for clustered distributed storage: The rack-aware storage model," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, pp. 866–879, 2020.

[13] Z. Chen, "Rack-aware MSR codes with optimal access," in *2022 IEEE Information Theory Workshop (ITW)*. IEEE, 2022, pp. 19–24.

[14] F. Chung and L. Lu, "The diameter of sparse random graphs," *Advances in Applied Mathematics*, vol. 26, no. 4, pp. 257 – 279, 2001.

[15] T. M. Cover and A. El Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

[16] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.

[17] S. Dutta, V. Cadambe, and P. Grover, "Short-Dot: Computing large linear transforms distributedly using coded short dot products," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 2100–2108. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/0a1bf96b6c5cce1c7806a9d9b9f2d5e2-Paper.pdf

[18] I. M. Duursma, "Outer bounds for exact repair codes," *arXiv preprint arXiv:1406.4852*, 2014.

[19] I. M. Duursma, X. Li, and H.-P. Wang, "Multilinear algebra for distributed storage," *SIAM J. Appl. Algebra Geom.*, vol. 5, pp. 552–587, 2021.

[20] I. M. Duursma and H.-P. Wang, "Multilinear algebra for minimum storage regenerating codes: a generalization of the product-matrix construction," *Applicable Algebra in Engineering, Communication and Computing*, pp. 1–27, 2023. [Online]. Available: https://doi.org/10.1007/s00200-021-00526-3

[21] O. Elishco and A. Barg, "Recoverable systems," *IEEE Transactions on Information Theory*, vol. 68, no. 6, pp. 3681–3699, 2022.

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

[23] M. Elyasi and S. Mohajer, "Determinant coding: A novel framework for exact-repair regenerating codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6683–6697, 2016.

[24] ——, "Determinant codes with helper-independent repair for single and multiple failures," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5469–5483, 2019.

[25] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[26] A. Frieze and M. Karońsky, *Introduction to Random Graphs*. Cambridge University Press, 2016.

[27] M. Gadouleau, "Finite dynamical systems, hat games, and coding theory," *SIAM J. Discrete Math.*, vol. 32, no. 3, pp. 1922–1945, 2018.

[28] M. Gerami and M. Xiao, "Exact optimized-cost repair in multi-hop distributed storage networks," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 4120–4124.

[29] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6318–6328, 2017.

[30] S. Gupta, B. R. Devi, and V. Lalitha, "On rack-aware cooperative regenerating codes and epsilon-MSCR codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 2, pp. 362–378, 2022.

[31] J. L. Hafner, "Weaver codes: highly fault tolerant erasure codes for storage systems," in *Proceedings of the 4th Conference on USENIX Conference on File and Storage Technologies - Volume 4*, ser. FAST'05. USA: USENIX Association, 2005, p. 16.

[32] A. Host-Madsen and J. Zhang, "Capacity bounds and power allocation for wireless relay channels," *IEEE Trans. Inf. Theor.*, vol. 51, no. 6, p. 2020–2040, Jun. 2005. [Online]. Available: https://doi.org/10.1109/TIT.2005.847703

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

[34] A. Khina, O. Ordentlich, U. Erez, Y. Kochman, and G. W. Wornell, "Decode-and-forward for the gaussian relay channel via standard awgn coding and decoding," in *2012 IEEE Information Theory Workshop (ITW)*. IEEE, 2012, pp. 457–461.

[35] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[36] O. Kosut, L. Tong, and D. N. C. Tse, "Polytope codes against adversaries in networks," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3308–3344, 2014.

[37] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[38] H. Lee and J. Lee, "An outer bound on the storage-bandwidth tradeoff of exact-repair regenerating codes and its asymptotic optimality in high rates," in *2016 IEEE Information Theory Workshop (ITW)*, 2016, pp. 255–259.

[39] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 1143–1147.

[40] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[41] Z. Li, W. H. Mow, L. Deng, and T.-Y. Wu, "Optimal-repair-cost MDS array codes for a class of heterogeneous distributed storage systems," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 2379–2384.

[42] ——, "Optimal-repair-cost MDS array codes for a class of heterogeneous distributed storage systems," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 2379–2384.

[43] Y. Liu, J. Li, and X. Tang, "A generic transformation to enable optimal repair/access mds array codes with multiple repair degrees," *IEEE Transactions on Information Theory*, vol. 69, no. 7, pp. 4407–4428, 2023.

[44] J. Lu, X. Guang, and F.-W. Fu, "Distributed storage over unidirectional ring networks," in *2014 International Symposium on Information Theory and its Applications*, 2014, pp. 368–372.

[45] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[46] G. A. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71–78, 1982.

[47] A. Mazumdar, "Storage capacity of repairable networks," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 5810–5821, 2015.

[48] A. Mazumdar, A. McGregor, and S. Vorotnikova, "Storage capacity as an information-theoretic vertex cover and the index coding rate," *IEEE Transactions on Information Theory*, vol. 65, no. 9, pp. 5580–5591, 2019.

[49] S. Mohajer and R. Tandon, "New bounds on the (n, k, d) storage systems with exact repair," in *2015 IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 2056–2060.

[50] A. Patra and A. Barg, "Regenerating codes on graphs," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2197–2202.

[51] ——, "Interior-point regenerating codes on graphs," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 1560–1565.

[52] ——, "Node repair on connected graphs," *IEEE Transactions on Information Theory*, vol. 68, no. 5, pp. 3081–3095, 2022.

[53] ——, "Node repair on connected graphs, Part II," 2022, eprint arXiv:2211.00797.

[54] ——, "Node repair for adversarial graphical networks," in *2023 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2023, pp. 2284–2289.

[55] ——, "More results for regenerating codes on graphs," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 813–818.

[56] ——, "Generalized regenerating codes and node repair on graphs," *IEEE Transactions on Information Theory*, vol. 71, no. 3, pp. 1613–1630, 2025.

[57] N. Prakash, V. Abdrashitov, and M. Médard, "The storage versus repair-bandwidth trade-off for clustered storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5783–5805, 2018.

[58] V. Ramkumar, S. Balaji, B. Sasidharan, M. Vajha, M. N. Krishnan, and P. V. Kumar, "Codes for distributed storage," *Foundations and Trends in Communications and Information Theory*, vol. 19, pp. 547–813, 2022.

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

[60] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Regenerating codes for errors and erasures in distributed storage," in *Proc. IEEE International Symposium on Information Theory, Cambridge, MA, USA*, 2012, pp. 1202–1206.

[61] ——, "Information-theoretically secure erasure codes for distributed storage," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1621–1646, 2018.

[62] N. Raviv, N. Silberstein, and T. Etzion, "Constructions of high-rate minimum storage regenerating codes over small fields," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2015–2038, 2017.

[63] N. Raviv, I. Tamo, and E. Yaakobi, "Private information retrieval in graph-based replication systems," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3590–3602, 2020.

[64] J. Rotman, *An Introduction to Homological Algebra*, 2nd ed. New York, N.Y.: Springer, 2009.

[65] B. Sasidharan, K. Senthoor, and P. V. Kumar, "An improved outer bound on the storage-repair-bandwidth tradeoff of exact-repair regenerating codes," in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 2430–2434.

[66] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "Xoring elephants: novel erasure codes for big data," *Proc. VLDB Endow.*, vol. 6, no. 5, p. 325–336, Mar. 2013. [Online]. Available: https://doi.org/10.14778/2535573.2488339

[67] K. Senthoor, B. Sasidharan, and P. V. Kumar, "Improved layered regenerating codes characterizing the exact-repair storage-repair bandwidth tradeoff for certain parameter sets," in *2015 IEEE Information Theory Workshop (ITW)*, 2015, pp. 1–5.

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

[69] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, 2010, pp. 1–5.

[70] ——, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2134–2158, 2012.

[71] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Trans. Inform. Theory*, vol. 59, no. 11, pp. 7229–7258, 2013.

[72] N. Shutty and M. Wootters, "Low-bandwidth recovery of linear functions of reed-solomon-encoded data." Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.117

[73] N. Silberstein, A. S. Rawat, and S. Vishwanath, "Error-correcting regenerating and locally repairable codes via rank-metric codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 11, pp. 5765–5778, 2015.

[74] J. Y. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of clustered distributed storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 81–107, 2019.

[75] I. Tamo, Z. Wang, and J. Bruck, "Access versus bandwidth in codes for storage," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2028–2037, 2014.

[76] ——, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.

[77] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3368–3376. [Online]. Available: https://proceedings.mlr.press/v70/tandon17a.html

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

[79] E. C. Van Der Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, p. 120–154, 1971.

[80] J. Wang and Z. Chen, "Low-access repair of Reed-Solomon codes in rack-aware storage," in *2023 IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 1142–1147.

[81] J. Wang, D. Zheng, S. Li, and X. Tang, "Rack-aware MSR codes with error correction capability for multiple erasure tolerance," *IEEE Transactions on Information Theory*, vol. 69, no. 10, pp. 6428–6442, 2023.

[82] J. Wang, Y. Luo, and K. W. Shum, "Storage and repair bandwidth tradeoff for heterogeneous cluster distributed storage systems," *Science China Information Sciences*, vol. 63, pp. 1–15, 2020.

[83] Y. Wang, D. Wei, X. Yin, and X. Wang, "Heterogeneity-aware data regeneration in distributed storage systems," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1878–1886.

[84] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 272–276, 1999.

[85] S. Yang, A. Hareedy, R. Calderbank, and L. Dolecek, "Hierarchical coding for cloud storage: Topology-adaptivity, scalability, and flexibility," *IEEE Transactions on Information Theory*, vol. 68, no. 6, pp. 3657–3680, 2022.

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

[87] M. Ye, "New constructions of cooperative MSR codes: Reducing node size to $\exp(o(n))$," *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7457–7464, 2020.

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

[89] ——, "Cooperative repair: Constructions of optimal mds codes for all admissible parameters," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1639–1656, 2019.

[90] R. W. Yeung and N. Cai, "Network error correction, Part I: Basic concepts and upper bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 19 – 35, 2006.

[91] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory Part I: Single source," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 4, pp. 241–329, 2006.

[92] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4406–4416.

[93] Q. Yu, K. W. Shum, and C. W. Sung, "Tradeoff between storage cost and repair cost in heterogeneous distributed storage systems," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 10, pp. 1201–1211, 2015.