On Finding the Largest Minimum Distance of Locally Recoverable Codes: A Graph Theory Approach

Majid Khabbazian[a], Muriel Médard[b]

[a] *University of Alberta, Alberta, Canada*
[b] *MIT, Massachusetts, USA*

Abstract

The $[n, k, r]$-Locally recoverable codes (LRC) studied in this work are a well-studied family of $[n, k]$ linear codes for which the value of each symbol can be recovered by a linear combination of at most r other symbols. In this paper, we study the *LMD* problem, which is to find the largest possible minimum distance of $[n, k, r]$-LRCs, denoted by $\mathscr{D}(n, k, r)$. LMD can be approximated within an additive term of one—it is known that $\mathscr{D}(n, k, r)$ is equal to either d^* or $d^* - 1$, where $d^* = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$. Moreover, for a range of parameters, it is known precisely whether the distance $\mathscr{D}(n, k, r)$ is d^* or $d^* - 1$. However, the problem is still open despite a significant effort. In this work, we convert LMD to an equivalent simply-stated problem in graph theory. Using this conversion, we show that an instance of LMD is at least as hard as computing the size of a maximal graph of high girth, a hard problem in extremal graph theory. This is an evidence that LMD—although can be approximated within an additive term of one—is hard to solve in general. As a positive result, thanks to the conversion and the exiting results in extremal graph theory, we solve LMD for a range of code parameters that has not been solved before.

Keywords: locally recoverable codes, minimum distance, distributed storage

1. Introduction

In distributed storage systems, maintaining data availability and redundancy is critical. When a storage node fails, it is essential to quickly reconstruct the lost data block. An effective strategy involves employing erasure codes with low locality, allowing each block to be reconstructed by accessing only a few other nodes.

Locally recoverable codes (LRCs) have received significant attention because of their application in distributed storage systems. The important characteristic of LRCs that distinguishes them from other codes is their small *repair locality*, a term introduced in [1, 2, 3]. An LRC with (all-symbol) repair locality r is a code for which the value of every symbol of the codeword can be recovered from values of a set of r other symbols. As a result, when a storage node fails in distributed storage systems that uses LRC with locality r, only r other storage nodes need to be accessed to repair the failed node. Smaller values of r result in lower I/O complexity and bandwidth overhead to recover a single storage node failure, the dominant failure scenario. Reducing r, however, may come at the cost of a lower minimum distance.

Minimum distance is an important parameter of LRCs—a minimum distance of d guarantees recovery of up to $d-1$ storage node failures, and is one of the main factors in determining the distributed storage reliability. Therefore, given the code parameters n, k, and r, it is desired to find an LRC that has the largest possible minimum distance. Motivated by this, in this work we study the LMD problem defined below.

LMD: For integers $n > k \geq r \geq 1$, let $\mathscr{D}(n,k,r)$ denote the largest possible minimum distance among all linear $[n,k,r]$-LRCs. Then, LMD is defined as the problem of finding the exact value of $\mathscr{D}(n,k,r)$. Note that in this definition, there is no restriction on the order of the finite field used for code construction.

The following relationship between the minimum distance d of an LRC, and its locality r was first derived by Gopalan et al. [1]:

$$d \leq d^* \tag{1}$$

where $d^* = n - k - \left\lceil \frac{k}{r} \right\rceil + 2$. We call an LRC *optimal* if it meets the bound (1) with equality. Many works have studied the design of an optimal LRC [4, 5, 6, 7, 8, 9].

It has been shown that for any code parameters n, k, and r, there exists an $[n,k,r]$-LRC with minimum distance of at least $d^* - 1$ [10]. This result together with the bound (1) imply that LMD can be approximated within an additive term of one. In addition, LMD has been solved in the literature for some ranges of code parameters n, k and r. Before covering these related results, let us define parameters k_1, k_2, n_1, and n_2. These parameters are used throughout the paper.

$$\begin{aligned} k_1 &= \left\lceil \tfrac{k}{r} \right\rceil, \quad & k_2 &= k_1 \cdot r - k, \\ n_1 &= \left\lceil \tfrac{n}{r+1} \right\rceil, \quad & n_2 &= n_1 \cdot (r+1) - n \end{aligned} \tag{2}$$

1.1. Existing Results on Computing $\mathscr{D}(n,k,r)$

For nearly all admissible parameters n, k and r, Kolosov et al. [11] find the largest possible minimum distance of LRCs with disjoint repair groups[1]. Their result, however, does not apply to the general class of LRCs, where repair groups can overlap. Note that LMD does not restrict LRCs to have disjoint repair groups. In addition, LMD does not put any restriction on the size of finite field. Therefore, we exclude bounds that depend on the size of alphabet. For instance, we exclude the bound in [12] as it depends on the size of alphabet, and is stronger than (1) if the size of alphabet is smaller than n. In the following, we enumerate only the existing results/bounds that can be used to compute $\mathscr{D}(n,k,r)$.

1. $\mathscr{D}(n,k,r) = d^*$ if $r = k$. This is because MDS codes achieve (1) with equality.

2. $\mathscr{D}(n,k,r) = d^*$ if $(r+1)|n$ [13, 14].

3. $\mathscr{D}(n,k,r) = d^*$ if $(n \mod r+1) > (k \mod r) > 0$ [13].

4. $\mathscr{D}(n,k,r) = d^* - 1$ if $r < k$, $r|k$ and $(r+1) \nmid n$ [1, 15].

5. $\mathscr{D}(n,k,r) = d^* - 1$ if $n_2 \geq k_2 + 1$ and $k_1 \geq 2k_2 + 2$ [15][2].

6. $\mathscr{D}(n,k,r) \leq n + 1 - (k + l)$, where l is derived from a parameter e_m, which is defined recursively [16].

7. $\mathscr{D}(n,k,r)$ is known when $n_2 < n_1$ [17].

8. $\mathscr{D}(n,k,r)$ is known when $k_2 < k_1 - 1$ [18].

[1] LRCs and repair groups are formally defined in Section 2.

[2] The conditions used in [15] are converted into equivalent conditions on k_1, k_2, n_1, and n_2

1.2. Our Contribution

Our first contribution is Theorem 1. This theorem reduces LMD to an equivalent simply-stated problem in graph theory[3]. We obtain this reduction via a connection between repairable codes and their Tanner graphs. Using Theorem 1, in our next contribution, we prove that a special instance of LMD is at least as hard as a long-standing open problem in extremal graph theory (Theorem 14, Corollary 15). Furthermore, we solve LMD for three new cases (Theorems 10, 11, and 18). In addition, to showcase the power of Theorem 1, in Appendix Appendix A, we demonstrate how to easily derive the existing results covered in Section 1.1, and somewhat extend them.

Theorem 1. *Let n_1, n_2, k_1, and k_2 be the parameters defined in (2). Then, $\mathscr{D}(n,k,r) = d^*$ if and only if there is a multigraph of order[4] n_1 and size n_2 that does not have any subgraph of order k_1 and size greater than k_2. Moreover, any such multigraph can be used for a non-explicit construction of optimal $[n,k,r]$-LRCs over a finite field of size of at least $(d^*-1)\binom{n}{d^*-1}+1$.*

Remainder of this paper Section 2 covers the main definitions and basic tools needed in the rest of the paper. In Section 3, we present our main results including the proof of Theorem 1. We conclude the paper and present possible future work in Section 4.

2. Connecting LMD to Graph Theory

In this section, we will gradually establish the connection between LMD and graph theory. To achieve this, we will introduce novel tools and concepts, including pruned graphs and their minimum distance. The basic results derived here will be utilized in the next section to prove Theorem 1, demonstrate the hardness of LMD, and extend existing results on solving LMD. We start by formally defining LRCs.

Locally Recoverable Code (LRC code) Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a code of length n and cardinality q^k. We say that $\mathcal{C}$ has locality r if for every $i \in$

[3] All the graphs considered in this paper are loopless.

[4] Recall that the order of a graph is the cardinality of its vertex set, and the size of a graph is the cardinality of its edge set.

$\{1, 2, \ldots, n\}$ there is a set $I_i \subset \{1, 2, \ldots, n\} \backslash \{i\}$, $|I_i| \leq r$, such that for every two codewords $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$

$$(\forall j \in I_i : x_j = y_j) \Longrightarrow (x_i = y_i)\,.$$

Informally, this implies that the ith symbol of any codeword is uniquely determined by by its symbols at coordinates associated with I_i. The sets $I_i \cup \{i\}$, $1 \leq i \leq n$, are called *repair groups.* In this work, we restrict ourself to linear LRC codes, and refer to them as $[n, k, r]$-LRC, where r denotes the locality of the code.

Tanner Graphs An $[n, k]$ linear code can be represented by a parity-check matrix $\mathbf{H}$. For example, consider a simple $[7, 4]$ Hamming code with the following parity-check matrix:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3}$$

A vector $V = (v_1, v_2, \ldots, v_7)$ is a codeword if and only if the dot product of V with any row of $\mathbf{H}$ is zero. The matrix $\mathbf{H}$ reveals which subsets of the symbols in the codeword are linearly dependent. For instance, the condition for the first row implies:

$$v_1 + v_3 + v_5 + v_7 = 0,$$

demonstrating the linear dependence among v_1, v_3, v_5, and v_7.

These dependencies can be visualized using a Tanner graph, a bipartite graph consisting of variable nodes and check nodes. An $[n, k]$ Tanner graph includes n variable nodes (v_i, $i \in \{1, \ldots, n\}$), depicted as circles, and $n - k$ check nodes (c_j, $j \in \{1, \ldots, n - k\}$), depicted as squares. Each check node corresponds to a row of $\mathbf{H}$, and it connects to those variable nodes whose indices are involved in the associated linear dependence.

Figure 1 illustrates the Tanner graph for the described Hamming code. Here, check node c_1 connects to variable nodes v_1, v_3, v_5, and v_7, corresponding to the linear dependencies specified by the first row of $\mathbf{H}$. The check nodes c_2 and c_3 represent the dependencies specified by the second and third rows of $\mathbf{H}$, respectively.

Figure 2 illustrates the structure of a Tanner graph for a general $[n, k]$ linear code. Each variable node v_i is connected to the check node c_j if and

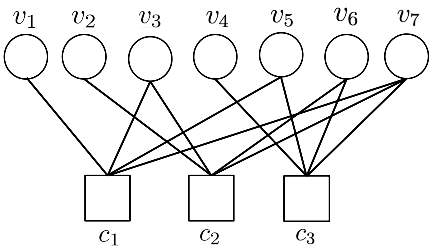

Figure 1: The Tanner graph of the code given by the parity-check matrix $\mathbf{H}$ in (3).

only if the entry $\mathbf{H}[i, j] \neq 0$ in the parity-check matrix $\mathbf{H}_{(n-k)\times n}$. The entry $\mathbf{H}[i, j]$ refers to the element located at the intersection of row i and column j of $\mathbf{H}$. In other words, the set of variable nodes incident to a check node are linearly dependent.

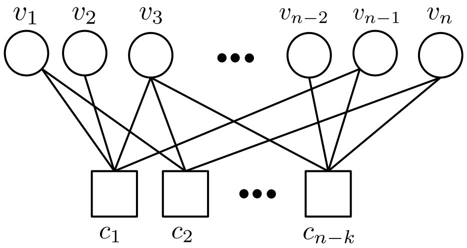

Figure 2: An $[n, k]$-Tanner graph with n variable nodes and $n - k$ check nodes.

Definition 1 (**$[n, k, r]$-Tanner Graph).** *An $[n, k, r]$-Tanner graph is an $[n, k]$-Tanner graph in which every variable node is incident to at least one check node of degree at most $r + 1$.*

Definition 2 (**$[n, k, r]$-Full Tanner Graph).** *An $[n, k, r]$-full Tanner graph is an $[n, k, r]$-Tanner graph in which the degree of each check node is either $r + 1$ or n.*

Definition 3 (Local and Global Check Nodes). *In an $[n, k, r]$-full Tanner graph, a check node is called* local check node *if its degree is $r + 1$; otherwise it is called* global check node.

By the above definitions, each variable node in a $[n, k, r]$-full Tanner graph is adjacent to at least one local check node. Therefore, the number of local check nodes of a $[n, k, r]$-full Tanner is at least $\left\lceil \frac{n}{r+1} \right\rceil = n_1$.

Minimum Distance The minimum distance of a code is the minimum Hamming distance between any two distinct codewords. In the following, we extend the definition of minimum distance to Tanner graphs. We then explain the connection between the minimum distances of an LRC and its corresponding Tanner graph.

Definition 4 (Tanner Graph's Minimum Distance). *The minimum distance of an $[n, k]$-Tanner graph is defined as the largest integer $d \in [2, n-k+1]$ for which we have the following condition: for any integer $\eta \in [n-k-d+2, n-k]$, every set of η check nodes are adjacent to at least $\eta + k$ variable nodes. This is a (reverse engineered) condition to support the Hall's theorem.*

Note that the above definition applies to $[n, k, r]$-Tanner graphs and $[n, k, r]$-full Tanner graphs, because they are all $[n, k]$-Tanner graphs.

Proposition 2. *There exists an $[n, k, r]$-LRC with minimum distance d^* over any finite field of size at least $(d^* - 1)\binom{n}{d^*-1} + 1$ iff there is an $[n, k, r]$-Tanner graph with minimum distance d^*.*

Proof. Appndix Appendix B. □

An $[n, k, r]$-Tanner graph can be easily converted to a $[n, k, r]$-full Tanner graph by adding edges to the check nodes: if the degree of a check nodes is strictly less than $r + 1$, add enough edges to it to make its degree equal to $r + 1$; on the other hand, if the degree of a node is strictly more than $r + 1$, add enough edges to make its degree equal to n. By Definition 4, adding edges does not reduce the minimum distance of a Tanner graph[5]. Therefore, we get the following result using Proposition 2.

Corollary 3. *There is an $[n, k, r]$-LRC with minimum distance d^* iff there is an $[n, k, r]$-full Tanner graph with minimum distance d^*.*

[5] Adding edges may increase the minimum distance of a Tanner graph.

Because of Corollary 3, to solve LMD we can focus only on full Tanner graphs. This somewhat simplifies the problem because full Tanner graphs have only two types of check nodes: local and global.

Recall that the condition in Definition 4 requires every set of η check nodes to be adjacent to at least $\eta + k$ variable nodes. This condition holds if the set of check nodes includes a global check node, because a global check node, by definition, is connected to all the variable nodes. Therefore, to solve LMD we can restrict ourselves to the subgraphs of full Tanner graphs, obtained by removing global check nodes. These subgraphs can be further pruned by removing all the variable nodes of degree one, since the contribution of these nodes in satisfying the condition of Definition 4 can be readily formulated. We call the resulting graphs *pruned graphs*.

Pruned Graphs We start by formally defining a pruned graph. The following definition is based on the fact that a pruned graph is constructed from a full Tanner graph by first removing all its global check nodes, and then all the variable nodes of degree one. We refer to this process as the F2P conversion.

Definition 5 ($[n, k, r]$-Pruned Graph). *An $[n, k, r]$-pruned graph is a subgraph of an $[n, k, r]$-full Tanner graph with the following properties:*

1. *it has m, $0 \leq m \leq n$, variable nodes;*
2. *it has h, $n_1 \leq h \leq n - k$ check nodes;*
3. *the degree of each check node is at most $r + 1$;*
4. *the degree of each variable node is at least two;*
5. *the number of its edges is equal to $h(r + 1) - (n - m)$.*

By the above definition, a pruned graph may not have any variable nodes. In addition, the degree of a check node in a pruned graph can be zero. Therefore, we do not use the term Tanner for these graphs.

Note that the minimum distance defined for Tanner graphs (Definition 4) does not apply to pruned graphs. Before defining the minimum distance of pruned graphs, let us explain how to convert an $[n, k, r]$-pruned graph to an $[n, k, r]$-full Tanner graph.

P2F Conversion: Converting a Pruned Graph to a Full Tanner Graph

1. add $n-m$ variable nodes, $v_1, v_2, \dots v_{n-m}$, of degree zero to the pruned graph; this increases the number of variable nodes to n.

2. perform the following $n-m$ steps: In Step i, $1 \leq i \leq n-m$, connect v_i to any check node whose degree up to that step is less than $r+1$. As will be discussed later, the order of selected check nodes with degree less than $r+1$ does not matter.
   Note that in every step there is at least one check node with degree less than $r+1$, because the number of edges of the pruned graph is $h(r+1)-(n-m)$, and the number of variable nodes added is $n-m$. At the end of the last step, each of the added variable nodes is connected to one check node, and the degree of every check node is exactly $r+1$.

3. add $(n-k)-h$ global check nodes, and connect each added global check node to all the variable nodes (including the new ones).

Definition 6 (Pruned Graph's Minimum Distance). *The minimum distance of a pruned graph is defined to be equal to the minimum distance of a full Tanner graph obtained from it through the above P2F conversion.*

Remark 1. *In each step of the second part of the P2F conversion (item 2), a check node with degree less than $r+1$ is selected. Note that this check node selection is arbitrary, i.e. any check node with degree less than $r+1$ can be selected. Therefore, if a full Tanner graph is converted to a pruned graph and then converted back to a full Tanner graph, the result may be different from the original full Tanner graph. Nevertheless, we will prove (Proposition 5) that all the full Tanner graphs that can be obtained from a fixed pruned graph have the same minimum distance. Hence, the minimum distance of pruned graphs (Definition 6) is well-defined.*

3. Main Results

3.1. Proving Theorem 1

Minimum distance of pruned graphs We start by proving that the minimum distance of pruned graphs is well-defined (Proposition 5).

For a node u in a simple graph G, let $N_G(u)$ denote the set of nodes adjacent to u, and $E_G(u)$ denote the set of edges incident to u. For a set of nodes A, we define

$$N_G(A) = \cup_{u \in A} N_G(u),$$

and

$$E_G(A) = \cup_{u \in A} E_G(u).$$

Lemma 4. *Let $\mathcal{P}$ be a pruned graph, and $\mathcal{T}$ be a corresponding $[n, k, r]$-full Tanner graph i.e., a full Tanner graph constructed from $\mathcal{P}$ using the P2F conversion. Let S be a subset of check nodes of $\mathcal{T}$. Then, we have*

$$|N_{\mathcal{T}}(S)| = \begin{cases} n & \textit{if } S \textit{ includes a global check node;} \\ |N_{\mathcal{P}}(S)| + ((r+1)|S| - |E_{\mathcal{P}}(S)|) & \textit{otherwise,} \end{cases}$$

where $|S|$ denotes the cardinality of S.

Proof. If there is a global check node in S, then $|N_{\mathcal{T}}(S)|$ is equal to n, because in a full Tanner graph each global check node is connected to all the n variable nodes. Therefore, from now assume that all the check nodes in S are local. We have

$$|E_{\mathcal{T}}(S)| = (r+1)|S| \tag{4}$$

because the degree of each check node in S is exactly $r + 1$. Let us call a variable node v *singular* (with respect to S) if

1. v is adjacent to exactly one local check node in $\mathcal{T}$;

2. the local check node that v is incident to is in S.

By the definition of pruned graph, each variable node that is in $N_{\mathcal{T}}(S)$ but not in $N_{\mathcal{P}}(S)$ must be a singular variable node. It is because among edges incident to a local check node in S, exactly those that are incident to a singular variable node are removed in the F2P conversion. Therefore, $|N_{\mathcal{T}}(S)| - |N_{\mathcal{P}}(S)|$ is equal to the number of singular variable nodes in $\mathcal{T}$. The number of singular variable nodes, on the other hand, is equal to $|E_{\mathcal{T}}| - |E_{\mathcal{P}}|$, because each singular variable node is incident to exactly one edge (which is in $E_{\mathcal{T}}(S)$ but not in $E_{\mathcal{P}}(S)$). Thus,

$$|N_{\mathcal{T}}(S)| - |N_{\mathcal{P}}(S) = |E_{\mathcal{T}}(S)| - |E_{\mathcal{P}}(S)|,$$

hence

$$\begin{aligned}|N_{\mathcal{T}}(S)| &= |N_{\mathcal{P}}(S)| + |E_{\mathcal{T}}(S)| - |E_{\mathcal{P}}(S)| \\ &= |N_{\mathcal{P}}(S)| + ((r+1)|S| - |E_{\mathcal{P}}(S)|),\end{aligned}$$

where the second equality is by (4).

□

Proposition 5. *All the full Tanner graphs that can be constructed from a fixed pruned graph using the P2F conversion have identical minimum distances.*

Proof. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two full Tanner graphs constructed from an $[n,k,r]$-pruned graph $\mathcal{P}$. By Lemma 4, we have $|N_{\mathcal{T}_1}(S)| = |N_{\mathcal{T}_2}(S)|$ for any subset of check nodes S. Thus, by Definition 4, the minimum distances of $\mathcal{T}_1$ and $\mathcal{T}_2$ are identical. □

Refining Pruned Graphs Our objective here is to reduce the number of check nodes of an $[n,k,r]$-pruned graph to exactly n_1, and the degree of all variable nodes to exactly two. The challenge is to preserve the minimum distance of the pruned graph throughout the conversion. We start with reducing the number of check nodes.

Lemma 6. *Any $[n,k,r]$-pruned graph $\mathcal{P}_1$ with minimum distance d, and $h_1 > n_1$ check nodes can be converted into a $[n,k,r]$-pruned graph $\mathcal{P}_2$ with minimum distance at least d and $h_2 = h_1 - 1$ check nodes.*

Proof. Let m_1 be the number of variable nodes in $\mathcal{P}_1$. We convert $\mathcal{P}_1$ into $\mathcal{P}_2$ through the following process.

Check node reduction process:

Step 1: An arbitrary check node is selected and is removed from $\mathcal{P}_1$. Let l be the degree of the removed check node.

Step 2: An arbitrary variable node with degree at least two is selected and one of its edges is removed. This operation is done $r+1-l$ times[6].

[6] A variable node may be selected multiple times. Also, note that $r+1-l \geq 0$ because $l \leq r+1$.

This is possible because the total number of edges of $\mathcal{P}_1$ after Step 1 is

$$\begin{aligned}&h_1(r+1)-(n-m_1)-l\\&\geq (n_1+1)(r+1)-(n-m_1)-l\\&= n_2+(r+1-l)+m_1\\&\geq (r+1-l)+m_1.\end{aligned}$$

Step 3: All variable nodes of degree one are removed.

Suppose the number of remaining variable nodes is m_2. The total number of edges removed is then

$$l+(r+1-l)+(m_1-m_2)=r+1+(m_1-m_2).$$

Thus the total number of remaining edges is

$$\begin{aligned}&(h_1(r+1)-(n-m_1))-((r+1)+(m_1-m_2))\\&=(h_1-1)(r+1)-(n-m_2)\\&=h_2(r+1)-(n-m_2),\end{aligned}$$

which is equal to the number of edges of an $[n,k,r]$-pruned graph with $h_2 = h_1-1$ check nodes, and m_2 variable nodes. Note that the degree of each variable node in the constructed pruned graph $\mathcal{P}_2$ is at least two, and the degree of each check node is at most $r+1$. Therefore, the constructed graph $\mathcal{P}_2$ is indeed an $[n,k,r]$-pruned graph.

Now, let us compare the minimum distances of the two pruned graphs $\mathcal{P}_1$ and $\mathcal{P}_2$. Let $\mathcal{T}_1$, $\mathcal{T}_2$ be two $[n,k,r]$-full Tanner graphs corresponding to $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Next, we show that

$$|N_{\mathcal{T}_2}(S)|\geq |N_{\mathcal{T}_1}(S)|,$$

for every set S of check nodes in the full Tanner graph. By Definition 4, this implies that the minimum distance of $\mathcal{T}_2$ is not smaller than that of $\mathcal{T}_1$.

Let S be an arbitrary set of check nodes of $\mathcal{T}_2$. If S includes any global check node of $\mathcal{T}_2$, then $|N_{\mathcal{T}_2}(S)|=n$ which yields the above inequality, because $|N_{\mathcal{T}_1}(S)|$ is at most equal to n. Thus, assume that S is a subset of local check nodes of $\mathcal{T}_2$ (i.e., S is a subset of check nodes of $\mathcal{P}_2$). We have

$$|E_{\mathcal{P}_1}(S)|-|E_{\mathcal{P}_2}(S)|\geq |N_{\mathcal{P}_1}(S)|-|N_{\mathcal{P}_2}(S)|,$$

because the reduction in size of $N_{\mathcal{P}_1}(S)$ as the result of edge removal in the *check node reduction process* is at most equal to the number of edges removed from $E_{\mathcal{P}_1}(S)$. Equivalently,

$$|N_{\mathcal{P}_2}(S)| - |E_{\mathcal{P}_2}(S)| \geq |N_{\mathcal{P}_1}(S)| - |E_{\mathcal{P}_1}(S)|.$$

Hence, by Lemma 4, we get

$$\begin{aligned} |N_{\mathcal{T}_2}(S)| &= |N_{\mathcal{P}_2}(S)| + (|S|(r+1) - |E_{\mathcal{P}_2}(S)|) \\ &= (|N_{\mathcal{P}_2}(S)| - |E_{\mathcal{P}_2}(S)|) + |S|(r+1) \\ &\geq (|N_{\mathcal{P}_1}(S)| - |E_{\mathcal{P}_1}(S)|) + |S|(r+1) \\ &= |N_{\mathcal{P}_1}(S)| + (|S|(r+1) - |E_{\mathcal{P}_1}(S)|) \\ &= |N_{\mathcal{T}_1}(S)|. \end{aligned}$$

□

Next, we reduce the degree of all variable nodes to two, while keeping the number of check nodes at n_1.

Proposition 7. *Any $[n, k, r]$-pruned graph with minimum distance d can be converted to a $[n, k, r]$-pruned graph with minimum distance at least d in which the degree of every variable node is exactly two, the number of check nodes is exactly n_1, and the number of variable nodes is n_2.*

Proof. By repeatedly applying Lemma 6, we first convert the given $[n, k, r]$-pruned graph into one with n_1 check nodes. Let us represent the new pruned graph by $\mathcal{P}_1$. By the definition of pruned graphs, the number of edges of $\mathcal{P}_1$ is

$$n_1(r+1) - (n - m_1) = n_2 + m_1,$$

where m_1 is the number of its variable nodes. Since the degree of each variable node is at least two, we get that the number of edges of $\mathcal{P}_1$ is at least $2m_1$, thus

$$2m_1 \leq n_2 + m_1,$$

hence $m_1 \leq n_2$. Therefore, $m_1 \leq r$ and $m_1 < n$, because $n_2 \leq r$, and $n_2 < n$, respectively. Since the total number of variable nodes, m_1, is at most equal to r, we get that the degree of each check node in $\mathcal{P}_1$ is *strictly* less than $r + 1$.

Let v be a variable node which has the maximum degree among all variable nodes in $\mathcal{P}_1$. If the degree of v is two, we are done, because this implies that the degree of all variable nodes in $\mathcal{P}_1$ is two. Therefore, assume that the degree of v is more than two. Let c_1 and c_2 be two check nodes adjacent to v. From $\mathcal{P}_1$, we construct $\mathcal{P}_2$ as follows: First, we add a variable node v' (of degree zero) to $\mathcal{P}_1$. Note that, after this addition, the number of variable nodes does not exceed n because $m_1 < n$. We connect the variable node v' to both check nodes c_1, and c_2, and remove the edge between v and c_2. This edge removal reduces the degree of the variable node v by one. The degree of v, however, remains at least two, as v's degree, before removal, was more than two. Also, the degrees of c_1 and c_2 will not exceed $r+1$, because the degree of each node was strictly less than $r+1$. By the definition of pruned graphs, the constructed graph $\mathcal{P}_2$ is an $[n,k,r]$-pruned graph with $m_2 = m_1 + 1$ variable nodes, n_1 check nodes, and

$$\begin{aligned} n_1(r+1) - (n - m_1) + 2 - 1 &= n_1(r+1) - (n - (m_1 + 1)) \\ &= n_1(r+1) - (n - m_2) \end{aligned}$$

edges. Next, we show that the minimum distance of $\mathcal{P}_2$ is not less than that of $\mathcal{P}_1$. This will conclude the proof, as by using the above process, the maximum degree can be always decremented if it is more than two; repeating this will yield an $[n,k,r]$-pruned graph in which all variable nodes have degree two.

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two $[n,k,r]$-full Tanner graphs corresponding to $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. To prove the above claim, by Definition 4, it is sufficient to show that for every set S of check nodes we have

$$|N_{\mathcal{T}_2}(S)| \geq |N_{\mathcal{T}_1}(S)|. \tag{5}$$

If S includes any global check node of $\mathcal{T}_2$, then $|N_{\mathcal{T}_2}(S)| = n$, hence the inequality. Therefore, assume that S is a subset of local check nodes of $\mathcal{T}_2$. If S does not contain any of the check nodes c_1 and c_2, we will have $|N_{\mathcal{T}_2}(S)| = |N_{\mathcal{T}_1}(S)|$. This is by Lemma 4 and the fact that, except check nodes c_1 and c_2, every other check node of $\mathcal{P}_2$ is identical to its original one in $\mathcal{P}_1$. Using Lemma 4, the inequality (5) can be verified for the remaining cases where S includes one or both check nodes c_1 and c_2: If S contains c_1 but not c_2 or if it contains both c_1 and c_2, then we have $|E_{\mathcal{P}_2}(S)| = |E_{\mathcal{P}_1}(S)| + 1$ and $|N_{\mathcal{P}_2}(S)| = |N_{\mathcal{P}_1}(S)| + 1$ hence by Lemma 4, we get $|N_{\mathcal{T}_2}(S)| = |N_{\mathcal{T}_1}(S)|$. If S includes c_2 but not c_1, then we have two cases based on whether or not v is in $N_{\mathcal{P}_1}(S\backslash\{c_2\})$. If $v \notin N_{\mathcal{P}_1}(S\backslash\{c_2\})$, then $|N_{\mathcal{P}_2}(S)| = |N_{\mathcal{P}_1}(S)|$, and $|E_{\mathcal{P}_2}(S)| = |E_{\mathcal{P}_1}(S)|$, hence $|N_{\mathcal{T}_2}(S)| = |N_{\mathcal{T}_1}(S)|$. If $v \in N_{\mathcal{P}_1}(S\backslash\{c_2\})$, however, we will have $|N_{\mathcal{P}_2}| = |N_{\mathcal{P}_1}| + 1$ and $|E_{\mathcal{P}_2}(S)| = |E_{\mathcal{P}_1}(S)|$, thus $|N_{\mathcal{T}_2}(S)| = |N_{\mathcal{T}_1}(S)| + 1$, hence the inequality (5).

Let $\mathcal{P}$ be the constructed pruned graph. The number of edges of $\mathcal{P}$ is equal to $2m$, where m denotes the number of variable nodes of $\mathcal{P}$. This is because the degree of each variable node is exactly two. Alternatively, by the definition of pruned graphs, the number of edges of $\mathcal{P}$ is

$$n_1(r+1) - (n-m).$$

Thus, we must have

$$n_1(r+1) - (n-m) = 2m,$$

hence

$$m = n_1(r+1) - n = n_2.$$

□

We are ready now to prove Theorem 1.
Proof. [Theorem 1] So far, we have shown the following:

1. There is $[n,k,r]$-LRC with minimum distance d^*, iff there is an $[n,k,r]$-full Tanner graph with minimum distance d^* (Proposition 5).

2. There is an $[n,k,r]$-full Tanner graph with minimum distance d^* iff there is an $[n,k,r]$-pruned graph with minimum distance d^* (Definition 6 and Proposition 5).

3. There is an $[n,k,r]$-pruned graph with minimum distance d^* iff there is an $[n,k,r]$-pruned graph $\mathcal{P}$ with minimum distance d^* in which the degree of every variable node is exactly two, the number of check nodes is n_1, and the number of variable nodes is n_2 (Proposition 7).

Suppose $\mathscr{D}(n,k,r) = d^*$. Therefore, there exists an $[n,k,r]$-pruned graph $\mathcal{P}$ with minimum distance d^* in which the degree of every variable node is two, the number of check nodes is n_1, and the number of variable nodes in n_2. Let $G = (V,E)$ be a multigraph, where the vertex set V is the set of check nodes of $\mathcal{P}$, and $(u,v) \in E$ iff there is variable node in $\mathcal{P}$ that is connected to both check nodes u and v. Since the degree of each variable node in $\mathcal{P}$ is exactly two, the size of G will be equal to the number of variable nodes in $\mathcal{P}$, i.e. $|E| = n_2$. Also, $|V| = n_1$, because V is the set of check nodes of $\mathcal{P}$. For every subset S of check nodes of $\mathcal{P}$, we have

$$|N_{\mathcal{P}}(S)| = |E_{\mathcal{P}}(S)| - |G[S]|,$$

where $|G[S]|$ denotes the size of the subgraph induced in G by S. Therefore, by Lemma 4, we get

$$\begin{aligned} |N_{\mathcal{T}}(S)| &= |N_{\mathcal{P}}(S)| + ((r+1)|S| - |E_{\mathcal{P}}(S)|) \\ &= (r+1)|S| - |G[S]|, \end{aligned} \tag{6}$$

where $\mathcal{T}$ is an $[n, k, r]$-full Tanner graph obtained from $\mathcal{P}$ using the P2F conversion method. Since the minimum distance of $\mathcal{T}$ is d^*, by Definition 4, for every set S of $n - k - d^* + 2 = \lceil \frac{k}{r} \rceil = k_1$ local check nodes of $\mathcal{T}$, we must have

$$|N_{\mathcal{T}}(S)| \geq k + |S| = k + k_1. \tag{7}$$

By (6), the above inequality is equivalent to

$$\begin{aligned} |G[S]| &\leq (r+1)|S| - k_1 - k \\ &= (r+1)k_1 - k_1 - k \\ &= rk_1 - k \\ &= k_2. \end{aligned}$$

Note that by (6), $|N_{\mathcal{T}}(S)|$ increases with the size of the set S. It is because the degree of each node in G (hence in $G[S]$) is strictly less than $r+1$, since the size of G (which is equal to n_2) is strictly less than $r+1$. Therefore, if (7) holds for every set S of size k_1, we will have

$$|N_{\mathcal{T}}(S)| \geq k + |S|$$

for every set S of size at least k_1. Therefore, a necessary and sufficient condition for $\mathcal{T}$ to have a minimum distance of d^* is that $|G[S]| \leq k_2$, for every set S, $|S| = k_1$.

Conversely, if such a multigraph G exists, then we can construct a pruned graph, and consequently a full Tanner graph with minimum distance d^*. The full Tanner graph, determines the zero elements of the optimal code's parity check matrix $\mathbf{H}$. If the non-zero elements of $\mathbf{H}$ are selected uniformly at random from a finite field of order n^{d^*}, we get that the minimum distance of the corresponding code is d^* with high probability (i.e, with probability at least $1-\frac{1}{n}$).[7] Similar to the proof of Proposition 2, this can be easily derived from the Schwartz-Zippel theorem and the union bound. □

3.2. LMD and Extremal Graph Theory

For a family of so called prohibited graphs $\mathcal{F}$, let $ex(n,\mathcal{F})$ denote the maximum number of edges that an n-vertex graph can have without containing a subgraph from $\mathcal{F}$. We use the notation $eX(n,\mathcal{F})$ when multiple/parallel edges are permitted.

Let $\mathscr{F}_{k_1,k_2}$ denote the family of all multigraphs of order k_1 and size strictly greater than k_2. The following corollary is a direct result of Theorem 1.

Corollary 8. *$\mathscr{D}(n,k,r)=d^*$ iff $n_2 \leq eX(n_1,\mathscr{F}_{k_1,k_2})$.*

We have

$$eX(n_1,\mathscr{F}_{k_1,k_2}) \geq ex(n_1,\mathscr{F}_{k_1,k_2}),$$

because simple graphs are subset of multigraphs. Thus, we also get the following corollary from Theorem 1.

Corollary 9. *$\mathscr{D}(n,k,r)=d^*$ if $n_2 \leq ex(n_1,\mathscr{F}_{k_1,k_2})$.*

Corollaries 8 and 9 allow us to approach LMD using existing results in extremal graph theory. For example, when $k_1=3$ and $k_2=2$, we get that $\mathscr{D}(n,k,r)=d^*$ iff $n_2 \leq \left\lfloor \frac{n_1^2}{4} \right\rfloor$. This is derived using the Mantel's theorem on triangle-free maximal graphs [19].

Theorem 10. *Suppose $k_1=3$ and $k_2=2$. In this case, the dimension of the code satisfies $k=3r-2$.*
Then, $\mathscr{D}(n,k,r)=d^*$ iff $n_2 \leq \left\lfloor \frac{n_1^2}{4} \right\rfloor$.

[7] In general, this probability can be set to at least $(1-\epsilon)$ by setting the order of the finite field to be at least $\frac{n^{d^*-1}}{\epsilon}$.

Proof. A simple graph is $\mathscr{F}_{3,2}$-free iff it is triangle-free. By Mantel's theorem, the maximum size of a triangle-free simple graph on n_1 vertices is $\left\lfloor \frac{n_1^2}{4} \right\rfloor$. In other words, $ex(n_1, \mathscr{F}_{3,2}) = \left\lfloor \frac{n_1^2}{4} \right\rfloor$. Therefore, by Corollary 9, we get that $\mathscr{D}(n,k,r) = d^*$ if $n_2 \leq \left\lfloor \frac{n_1^2}{4} \right\rfloor$. By Mantel's theorem, we know that n_1-vertex simple graphs of size greater than $\left\lfloor \frac{n_1^2}{4} \right\rfloor$ are not triangle-free, hence are not $\mathscr{F}_{3,2}$-free. This is also the case for multigraphs: n_1-vertex multigraphs of size greater than $\left\lfloor \frac{n_1^2}{4} \right\rfloor$ are not $\mathscr{F}_{3,2}$-free.

Let G be a maximal $\mathscr{F}_{3,2}$-free multigraph on n_1 vertices. By induction on n_1, we prove that the size of G is at most $\left\lfloor \frac{n_1^2}{4} \right\rfloor$. The assertion clearly holds for $n_1 = 3$ and $n_1 = 4$. Suppose G has multiple edges between two distinct vertices u and v. The maximum number of parallel edges between u and v is two, as otherwise G will not be $\mathscr{F}$-free. Also, any vertex $w \notin \{u, v\}$ is not connected to either u or v, as otherwise the graph induced by $\{u, v, w\}$ will have a size of at least 3. Therefore, by induction hypothesis, the maximum size of G will be

$$2 + \left\lfloor \frac{(n_1 - 2)^2}{4} \right\rfloor < \left\lfloor \frac{n_1^2}{4} \right\rfloor,$$

for $n_1 \geq 5$. □

The following theorem can be similarly derived from Corollary 9, and Turán's theorem in extremal graph theory [19]. A Turán graph $T(n, k)$ is a complete multipartite graph on n vertices, and k partitions with the size of partitions being as equal as possible. By Turán's theorem, the Turán graph has the maximum possible number of edges among all $(k + 1)$-clique-free graphs with n vertices [19].

Theorem 11. *Suppose $k_2 = \binom{k_1}{2} - 1$. Then, $\mathscr{D}(n,k,r) = d^*$ if $n_2 \leq t_{k_1}(n_1)$, where $t_{k_1}(n_1)$ denotes the size of Turán's graph on n_1 vertices, and $k_1 - 1$ partitions.*

3.3. Hardness of LMD

LMD is approximable within an additive term of one—the largest minimum distance is either d^* or $d^* - 1$. However, as will be discussed shortly, it appears that LMD is hard to solve in general[8]. In the remaining of this section, we prove that solving LMD is at least as hard as computing the size of a maximal graph of high girth, a challenging problem in extremal graph theory. We start by proving a few lemmas.

Lemma 12. *We have*

$$eX(n, \mathscr{F}_{k,k-1}) = ex(n, \mathscr{F}_{k,k-1}),$$

where $n \geq k \geq 1$.

Proof.
We have $eX(n, \mathscr{F}_{k,k-1}) \geq ex(n, \mathscr{F}_{k,k-1})$, because simple graphs are subset of multigraphs. Therefore, we only need to show that $eX(n, \mathscr{F}_{k,k-1}) \leq ex(n, \mathscr{F}_{k,k-1})$. To this end, we show that any $\mathscr{F}_{k,k-1}$-free multigraph of order n and size m can be converted to a simple $\mathscr{F}_{k,k-1}$-free graph of order n and size at least m.

Let G be a $\mathscr{F}_{k,k-1}$-free multigraph of order n and size m. Suppose G is connected, and assume that G has two vertices u and v connected with multiple edges. Then, any connected subgraph of G of order k will have at least k edges if the subgraph includes u and v. Therefore, a connected $\mathscr{F}_{k,k-1}$-free graph cannot have parallel edges.

Now suppose that G has $c > 1$ connected components denoted $G_i = (V_i, E_i)$, $i \in [c]$. Any connected component of order at least k must be a simple graph; otherwise, by the above argument, it will not be $\mathscr{F}_{k,k-1}$-free (hence G will not be $\mathscr{F}_{k,k-1}$-free). Therefore, if G does not have any connected component of order less than k, we are done.

Without loss of generality, suppose $G_i = (V_i, E_i)$, $i \in [c_1]$, where $c_1 \in [c]$, is the number of connected components of G that have less than k vertices. We show that

$$\sum_{i=1}^{c_1} |E_i| \leq \sum_{i=1}^{c_1} |V_i|. \tag{8}$$

The above inequality clearly holds if

$$\forall i \in [c_1] \quad |E_i| < |V_i|.$$

[8] This may remind the reader of the few NP-hard problems (e.g., edge coloring [20], and 3-colorability of planar graphs [21]) that are approximable within an additive term of one, but are hard to solve.

If not, we must have $|E_i| \geq |V_i|$ for some connected components G_j, $j \in [c_1]$. Without loss of generality, suppose $|E_i| \geq |V_i|$ for $i \in [c_2]$, where $c_2 \in [c_1]$. Also, assume that $|E_i| - |V_i| \geq |E_j| - |V_j|$ for every $i < j$, where $i, j \in [c_2]$. Note that for the remaining connected components G_i, $c_2 < i \leq c_1$, we must have $|E_i| = |V_i| - 1$.

Let us extract a k-vertex subgraph of G in k steps as follows. In the first step, we select an arbitrary vertex from G_1. In every consecutive step, we find a vertex that is connected to at least one of the vertices that we have selected so far, and add that vertex to the set of selected vertices. If none exist, we move on to the next connected component G_2 and then G_3 and so on. We continue the above process until we select k vertices.

Let H denote the subgraph induced by the selected k vertices. Suppose that G_t, $t \in [c_1+1]$ is the last connected graph from which a vertex has been selected. The size of H will be at least

$$\begin{aligned} &\sum_{i=1}^{t-1} |E_i| + \left(k - \sum_{i=1}^{t-1} |V_i|\right) - 1 \\ &= k + \left(\sum_{i=1}^{t-1} |E_i| - \sum_{i=1}^{t-1} |V_i|\right) - 1 \end{aligned} \tag{9}$$

If (8) does not hold, then the term $\left(\sum_{i=1}^{t-1} |E_i| - \sum_{i=1}^{t-1} |V_i|\right)$ in (9) will be at least equal to one. This means that the size of H will be at least k, which is not possible since G is $\mathscr{F}_{k,k-1}$-free. Thus (8) must hold. In the special case, where $k \leq \sum_{i=1}^{c_1} |V_i|$ (i.e., $t \leq c_1$), we must have

$$\sum_{i=1}^{c_1} |E_i| < \sum_{i=1}^{c_1} |V_i|, \tag{10}$$

as otherwise the size of H will be at least k.

Let us now construct a n-vertex $\mathscr{F}_{k,k-1}$-free simple graph of size at least m from G. To do so, we replace the connected components G_i, $i \in [c_1]$ with a path graph of order $\sum_{i=1}^{c_1} |V_i|$. We then connect the path graph (by an edge) to one of the remaining connected components of G if there are any. The new graph G' is a n-vertex $\mathscr{F}_{k,k-1}$-free simple graph. Also, by (8) and (10), the order of G' is not less than that of G.

□

Let C_k denote the cycle of length k, and define

$$\mathscr{C}_k = \{C_3, C_4, ..., C_k\}.$$

Lemma 13. *We have*

$$ex(n, \mathscr{F}_{k,k-1}) = ex(n, \mathscr{C}_k),$$

where $n \geq k \geq 3$.

Proof. If a simple graph is $\mathscr{C}_k$-free, it is $\mathscr{F}_{k,k-1}$-free, too[9]. If not, it has a k-vertex subgraph of size at least k. Such a subgraph must have a cycle of length at most k, which contradicts the fact that the graph is $\mathscr{C}_k$-free. Therefore, we have

$$ex(n, \mathscr{F}_{k,k-1}) \geq ex(n, \mathscr{C}_k). \tag{11}$$

Let $G = (V, E)$ be an arbitrary n-vertex $\mathscr{F}_{k,k-1}$-free simple graph. Any connected component of G of order at least k must be $\mathscr{C}_k$-free. It is because, otherwise, any connected k-vertex subgraph of that component which includes the cycle will be of size at least k. If G does not have any connected component of order less than k, we are done, because by the above argument, each connected component of G is $\mathscr{C}_k$-free, hence G is $\mathscr{C}_k$-free.

Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_c = (V_c, E_c)$ be the $c \geq 1$ connected components of G that have order less than k. Similar to the proof of Lemma 12 (Inequality 8), we get

$$\sum_{i=1}^{c} |E_i| \leq \sum_{i=1}^{c} |V_i|.$$

Therefore

$$|E| \leq ex(n - n', \mathscr{C}_k) + n', \tag{12}$$

where $n' = \sum_{i=1}^{c} |V_i|$. For any integer n', $0 \leq n' \leq n$, we have

$$ex(n - n', \mathscr{C}_k) + n' \leq ex(n, \mathscr{C}_k). \tag{13}$$

[9] The converse is not true; there are $\mathscr{F}_{k,k-1}$-free simple graphs that are not $\mathscr{C}_k$-free. For instance, a triangle and $k - 3 \geq 1$ isolated vertices is $\mathscr{F}_{k,k-1}$-free but not $\mathscr{C}_k$-free

It is because we can make a n-vertex $\mathscr{C}_k$-free graph by connecting (using an edge) a n'-vertex path graph to a $(n-n')$-vertex $\mathscr{C}_k$-free graph. From (12) and (13), we get

$$|E| \leq ex(n, \mathscr{C}_k). \tag{14}$$

Since $G=(V,E)$ is an arbitrary n-vertex $\mathscr{F}_{k,k-1}$-free simple graph, by (14), we get

$$ex(n, \mathscr{F}_{k,k-1}) \leq \mathrm{ex}(n, \mathscr{C}_k).$$

Combining this with the inequality (11) we arrive at the desired result:

$$ex(n, \mathscr{F}_{k,k-1}) = \mathrm{ex}(n, \mathscr{C}_k).$$

□

Theorem 14. *Let $k_2 = k_1 - 1$, and $k_1 \geq 3$. Then, $\mathscr{D}(n,k,r) = d^*$ iff $n_2 \leq ex(n_1, \mathscr{C}_{k_1})$.*

Proof. By Corollary 8, $\mathscr{D}(n,k,r) = d^*$ iff $n_2 \leq eX(n_1, \mathscr{F}_{k_1,k_1-1})$. By Lemma 12, we have $eX(n_1, \mathscr{F}_{k_1,k_1-1}) = ex(n_1, \mathscr{F}_{k_1,k_1-1})$. Also, Lemma 13 states that $ex(n_1, \mathscr{F}_{k_1,k_1-1}) = ex(n_1, \mathscr{C}_{k_1})$, when $k_1 \geq 3$. Therefore, when $k_1 \geq 3$, $\mathscr{D}(n,k,r) = d^*$ iff $n_2 \leq ex(n_1, \mathscr{C}_{k_1})$. □

The following corollary is an immediate result of Theorem 14.

Corollary 15. *Solving LMD is at least as hard as computing $ex(n, \mathscr{C}_k)$.*

Proof. Using binary search, one can compute $ex(n, \mathscr{C}_k)$ by calling the LMD solution $\mathcal{O}(\log n)$ times. □

Computing $ex(n, \mathscr{C}_k)$ is a challenging open problem in extremal graph theory. In fact, we do not even know the asymptotic behaviour of $ex(n, \mathscr{C}_k)$ for almost any value of k. In particular, the following conjecture of Erdös and Simonovits is still one of the main open problems in extremal graph theory.

Conjecture 1. (Erdös and Simonovits [22]*) For all $k \geq 2$, $ex(n, \mathscr{C}_{2k}) = \theta(n^{1+\frac{1}{k}})$.*

3.4. LMD and Graph Theory

In the previous sections, we discussed the connection between LMD and extremal graph theory. This connection, as showed, can be used to solve LMD for more special cases, or recognize cases that are hard to solve. Theorem 1 also allows us to use tools from the general field of graph theory to tackle LMD. As an example, let us solve another special instance of LMD, where $n_1 - k_1 = 1$.[10] To this end, we use some basic results from graph realization[11].

A sequence $d = \langle d_1, ..., d_n \rangle$ of non-negative integers is called *graphic* if it is the degree sequence of some multigraph G. Such a multigraph G is called a *realization* of sequence d. Degree sequences of simple graphs are well-understood—they can be efficiently recognized [23] and realized [24]. A general realizability test for multigraphs follows.

Lemma 16. (Harary [25]) *The sequence $d = \langle d_1, ..., d_n \rangle$, where $d_1 = \max(d)$, is graphic iff $\sum_{i=1}^{n} d_i$ is even and $d_1 \leq \sum_{i=2}^{n} d_i$.*

We call a multigraph *almost regular* if the degrees of its vertices differ by at most one. The following corollary is a direct result of Lemma 16.

Corollary 17. *For any integers $n \geq 2$ and $m \geq 0$ there exists an almost-regular multigraph of order n and size m.*

Proof. Let $t = (2m \mod n)$. The following degree sequence satisfies the conditions of Lemma 16, hence is realizable.

$$\langle d_1 = \left\lceil \frac{2m}{n} \right\rceil, \ldots, d_t = \left\lceil \frac{2m}{n} \right\rceil, d_{t+1} = \left\lfloor \frac{2m}{n} \right\rfloor \ldots d_n = \left\lfloor \frac{2m}{n} \right\rfloor \rangle$$

Note that $\sum_{i=1}^{n} d_i = 2m$. Therefore, a realization of the above degree sequence is an almost-regular multigraph of order n and size m.

□

Theorem 18. *Suppose $n_1 - k_1 = 1$. Then, $\mathscr{D}(n, k, r) = d^*$ iff*

$$n_2 - \left\lfloor \frac{2n_2}{n_1} \right\rfloor \leq k_2.$$

[10] The case $n_1 - k_1 = 1$ holds for typical range of practical LRCs, as well as LRCs with almost optimal rate; for $[n, k, r]$-LRCs we have $\frac{k}{n} \leq \frac{r}{r+1}$ [1].

[11] Similar approach/tools can be used to extend this result to $n_1 - k_1 \leq 3$.

Proof. Suppose $\mathscr{D}(n,k,r) = d^*$. Then, by Corollary 8, there is a $\mathscr{F}_{k_1,k_2}$-free multigraph G of order n_1 and size n_2. Since G has n_2 edges, it must have a vertex v of degree at most $\left\lfloor \frac{2n_2}{n_1} \right\rfloor$. Removing v from G we get a k_1-vertex subgraph of G of size at least $n_2 - \left\lfloor \frac{2n_2}{n_1} \right\rfloor$. Since G is $\mathscr{F}_{k_1,k_2}$-free, we must have

$$n_2 - \left\lfloor \frac{2n_2}{n_1} \right\rfloor \leq k_2. \tag{15}$$

Now, suppose (15) holds. Let G be an almost-regular multigraph of order n_1 and size n_2. By Corollary 17, such multigraph G exists. Let H be a k_1-vertex subgraph of G obtained by removing a vertex v from G. Since G is an almost-regular graph, the degree of v is at least equal to $\left\lfloor \frac{2n_2}{n_1} \right\rfloor$, thus the size of H is at most $n_2 - \left\lfloor \frac{2n_2}{n_1} \right\rfloor$ which by (15) is bounded by k_2. This implies that G is $\mathscr{F}_{k_1,k_2}$-free. □

There is an infinite range of code parameters for which the existing results in the literature cannot solve LMD but Theorem 18 does. This range includes the case where

$$n_2 > k_2 \geq k_1 \geq 3, \quad k_2 \geq n_2 - \left\lfloor \frac{2n_2}{n_1} \right\rfloor, \text{ and } \quad n_1 = k_1 + 1.$$

Theorem 18 addresses several gaps in the landscape of known code parameters. For instance, while the literature addresses LMD for $[16,8,4]$-LRC and $[16,11,4]$-LRC, Theorem 18 bridges the gap by also solving LMD for $[16,9,4]$-LRC and $[16,10,4]$-LRC. Similarly, although solutions exist for $[19,10,5]$-LRC and $[19,14,5]$-LRC, Theorem 18 further expands coverage to include $[19,11,5]$-LRC, $[19,12,5]$-LRC, and $[19,13,5]$-LRC.

4. Conclusion and Future Research

We studied the problem of finding the largest possible minimum distance of LRCs, a problem referred to as LMD. We converted LMD to an equivalent simply stated graph theory problem. Using this result, we showed how to easily derive and extend the existing results in the literature. In addition, we established a connection between an instance of LMD and a well-known open problem in extremal graph theory; an indication that LMD is hard to be solved, in general.

As a future direction, this work can be extended to LRCs with multiple recovering sets such as those considered in [26, 27, 28, 29]. Also, there are a number of interesting questions that remain unanswered. For example, all the solved instances of LMD in the literature and in this paper have a corresponding almost-regular multigraph solution. An interesting question is whether this is the case for every instance of LMD. If so, future research may focus on such graphs. Another interesting question is whether or not $eX(n_1, \mathscr{F}_{k_1,k_2}) = ex(n_1, \mathscr{F}_{k_1,k_2})$ when $k_2 \leq \binom{k_1}{2}$. In this work, we proved this for some special cases, e.g. when $k_2 \leq k_1 - 1$.

Appendix A.

Using Theorem 1, we can easily derive and somewhat extend the existing results in the literature.

1. **Case** $r = k$: In this case, we have $k_1 = 1$ and $k_2 = 0$, because $k_1 = \lceil \frac{k}{r} \rceil$, and $k_2 = k_1 \cdot r - k$. Note that the size of every $(k_1 = 1)$-vertex subgraph of a multigraph is zero. Therefore, replacing k_1 and k_2 with one and zero in Theorem 1, we get $\mathscr{D}(n, k, r) = d^*$. Using Theorem 1, we can easily extend this result to $k_1 = 2$ as follows:

   Corollary 19. *Suppose $k_1 = 2$. Then, $\mathscr{D}(n, k, r) = d^*$ iff*

   $$n_2 \leq \binom{n_1}{2} \cdot k_2,$$

   Proof. The size of a n_1-vertex multigraph that has at most k_2 parallel edges between any pair of vertices is clearly bounded by $\binom{n_1}{2} \cdot k_2$. □

2. **Case** $(r + 1)|n$: This case is equivalent to $n_2 = 0$. The size of any subgraph of a multigraph of size $n_2 = 0$ is zero, hence upper bounded by k_2. Therefore, $\mathscr{D}(n, k, r) = d^*$ by Theorem 1.

3. **Case** $(n \mod r + 1) > (k \mod r) > 0$: This case is equivalent to $k_2 > n_2 > 0$. Clearly, the size of any subgraph of a multigraph of size n_2 is at most n_2. Since $n_2 < k_2$ in this case, by Theorem 1, we get $\mathscr{D}(n, k, r) = d^*$. In fact, by Theorem 1, this result still holds if $k_2 = n_2$. Therefore, with this little extension, we get $\mathscr{D}(n, k, r) = d^*$ if $(n \mod r + 1) \geq (k \mod r) > 0$.

4. **Case** $r < k$, $r|k$ and $(r+1) \nmid n$: This case is equivalent to $k_1 \geq 2$, $k_2 = 0$ and $n_2 \geq 1$, respectively. Since $r < k$, and $k < n$, we get $r+1 < n$, thus $n_1 \geq 2$. Clearly, any $(n_1 \geq 2)$-vertex multigraph of size $n_2 \geq 1$ always has a $(k_1 \geq 2)$-vertex subgraph of size greater than $k_2 = 0$. Thus, by Theorem 1 we get that $\mathscr{D}(n,k,r) \neq d^*$, which implies $\mathscr{D}(n,k,r) = d^* - 1$.

5. **Case** $n_2 \geq k_2 + 1$ and $k_1 \geq 2k_2 + 2$: Let G be any multigraph of size n_2. Pick $k_2 + 1$ edges of G. The result is a subgraph of order at most $2k_2 + 2 \leq k_1$, and size grater than k_2. Therefore, any multigraph of size $n_2 \geq k_2 + 1$ has a k_1-vertex subgraph of size greater than k_2. Thus, by Theorem 1, we get $\mathscr{D}(n,k,r) = d^* - 1$.

6. **Case** $\mathscr{D}(n,k,r) \leq n + 1 - (k + l)$: Since $\mathscr{D}(n,k,r) \geq d^* - 1$, the only advantage of this upper bound—or any other upper bound on $\mathscr{D}(n,k,r)$—over (1) is when the right side of the inequality becomes equal to $d^* - 1$; that is exactly when the inequality implies $\mathscr{D}(n,k,r) = d^* - 1$. In the above case, this happens iff

   $$t_{k_1} > k_2, \tag{A.1}$$

   where

   $$t_{m-1} = t_m - \left\lceil \frac{2t_m}{m} \right\rceil, \quad 2 \leq m \leq n_1, \quad t_{n_1} = n_2, \tag{A.2}$$

   is a recursive equation obtained for the one defined in [16] by substituting their parameter e_m with $t_m = m(r+1) - e_m$. Let G be any n_1-vertex multigraph of size n_2. Let $T_{n_1} = G$ and T_{m-1}, $2 \leq m \leq n_1 - 1$, be the $(m-1)$-vertex graph obtained from T_m by removing its vertex with the smallest degree. Since the smallest degree of T_m is at most equal to $\lceil \frac{2t_m}{m} \rceil$, by (A.2) we get that the size of T_{m-1} is at least t_{m-1}. Therefore, t_m is a lower bound on the size of graph T_m. Thus, the condition (A.1) means that the size of T_{k_1} (which is a k_1-vertex subgraph of G) is greater than k_2. By Theorem 1, we then get $\mathscr{D}(n,k,r) \neq d^*$, which implies $\mathscr{D}(n,k,r) = d^* - 1$.

   By the above argument, an improvement over the upper bound of [16] is obtained by replacing $\lceil \frac{2t_m}{m} \rceil$ with $\lfloor \frac{2t_m}{m} \rfloor$ in (A.2)—note that $\lfloor \frac{2t_m}{m} \rfloor$ is a better upper bound on the smallest degree of T_m.

7. **Case** $n_2 < n_1$: Using Theorem 1, we can also solve LMD for this case. The intuition is as follows: Let us define k*-density* of a multigraph as the maximum size of any of its k-vertex subgraphs. To solve LMD, we need a multigraph with minimum k_1-density among all the n_1-vertex multigraphs of size n_2. Let us call such a multigraph k_1*-dense.* It is not hard to show that a forest with almost equally sized trees (i.e. with trees whose order differ by at most one) is always k_1-dense. To extend the result of [17] a bit further, one can show that a cycle graph is k_1-dense when $n_2 = n_1$. This observation extends the result of [17] from the case $n_2 < n_1$ to $n_2 \leq n_1$.

   Instead of providing the technical details for the above intuition, we show how to solve LMD for a similar case, i.e. for the case where $k_2 < k_1 - 1$. The reasons for doing so are I) the case $k_2 < k_1 - 1$ is solved using a similar technique and graphs (forests with almost equally sized trees); II) unlike the case $n_2 < n_1$, which we showed that can be extended to $n_2 \leq n_1$, the case $k_2 < k_1 - 1$ is hard to be extended to $k_2 \leq k_1 - 1$ as proven earlier.

8. **Case** $k_2 < k_1 - 1$:

   Theorem 20. *Suppose* $k_2 < k_1 - 1$*. Then,* $\mathscr{D}(n, k, r) = d^*$ *iff*

$$n_2 \leq n_1 - \left(\left\lceil \frac{n_1 - k_1 + 1}{\left\lfloor \frac{k_1}{k_1 - k_2 - 1} \right\rfloor} \right\rceil + k_1 - k_2 - 1 \right).$$

   Proof.

   Let $\mathscr{F}_{k_1,k_2}$ be the set of all k_1-vertex multigraphs of size greater than k_2. We say a graph G is $\mathscr{F}_{k_1,k_2}$-free if G does not have a subgraph of order k_1 and size greaters than k_2. We first prove a necessary and sufficient condition for a n_1-vertex forest to be $\mathscr{F}_{k_1,k_2}$-free, when $k_2 < k_1 - 1$. Then, we show that this condition applies to all multigraphs on n_1 vertices.

   Let G be a forest on n_1 vertices. Let $t \geq 1$ be the minimum number of connected components of G that are needed to collect k_1 vertices. The maximum size of a k_1-vertex subgraph of G is then exactly $k_1 - t$. Therefore, G is $\mathscr{F}_{k_1,k_2}$-free, iff $k_1 - t \leq k_2$, or equivalently $t \geq k_1 - k_2$.

Note that, by the above argument, only the order of the connected components of G determines whether or not G is $\mathscr{F}_{k_1,k_2}$-free. Thus, we can safely assume that each connected component of G (which is a tree) is a path graph.

If the order of two connected components of G differ by at least two, we can remove one vertex from one end of the larger connected component (which is a path) and add one vertex and connect it with an edge to one end of the smaller connected component. If G is $\mathscr{F}_{k_1,k_2}$-free, so is the new forest—the value of t for the new forest is not smaller than that for G. Therefore, in pursuing a necessary condition for a forest to be $\mathscr{F}_{k_1,k_2}$-free, we can safely assume that G is a forest with almost equally sized trees, where each tree is a path graph.

Suppose G has c connected components (thus, $n_2 = n_1 - c$). Since the connected components of G are almost equally sized, and the total number of vertices in any $k_1 - k_2 - 1$ connected components of G is at most $k_1 - 1$, we can have at most $A = (k_1 - 1) \mod (k_1 - k_2 - 1)$ connected components of order $\left\lceil \frac{k_1}{k_1-k_2-1} \right\rceil$, and $B = c - A$ connected components of order $\left\lfloor \frac{k_1}{k_1-k_2-1} \right\rfloor$. Thus,

$$n_1 \leq A \cdot \left\lceil \frac{k_1}{k_1 - k_2 - 1} \right\rceil + B \cdot \left\lfloor \frac{k_1}{k_1 - k_2 - 1} \right\rfloor,$$

which is simplified to

$$n_1 \leq (k_1 - 1) + (c - (k_1 - k_2 - 1)) \left\lfloor \frac{k_1}{k_1 - k_2 - 1} \right\rfloor.$$

This yields

$$c \geq \left\lceil \frac{n_1 - k_1 + 1}{\left\lfloor \frac{k_1}{k_1-k_2-1} \right\rfloor} \right\rceil + k_1 - k_2 - 1,$$

from which we get

$$n_2 \leq n_1 - \left(\left\lceil \frac{n_1 - k_1 + 1}{\left\lfloor \frac{k_1}{k_1-k_2-1} \right\rfloor} \right\rceil + k_1 - k_2 - 1 \right) \tag{A.3}$$

because $n_2 = n_1 - c$. Note that if (A.3) holds, we can divide n_1 vertices into

$$c = \left\lceil \frac{n_1 - k_1 + 1}{\left\lfloor \frac{k_1}{k_1 - k_2 - 1} \right\rfloor} \right\rceil + k_1 - k_2 - 1$$

groups such that the total sum of vertices in every $k_1 - k_2 - 1$ groups is at most $k_1 - 1$. Therefore, (A.3) is both necessary and sufficient to have a $\mathscr{F}_{k_1,k_2}$-free forest of order n_1 and size n_2.

Now, let us cover the case where G is $\mathscr{F}_{k_1,k_2}$-free but not a forest. We first convert G into a forest G' of the same order and size as G. Then, we prove that G' is $\mathscr{F}_{k_1,k_2}$-free. This will imply the bound (A.3), and conclude the proof.

Let $G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_c = (V_c, E_c)$ be the connected components of G. Suppose that the fist $c_1 \geq 1$ connected components of G are not tree, that is $|E_i| \geq |V_i|$ for every $i \in [c_1]$. Since the remaining components are tree, we have $|E_i| = |V_i| - 1$ for $c_1 < i \leq c$. Let t be the smallest integer for which we have

$$\sum_{i=1}^{t} |E_i| = \sum_{i=1}^{t} |V_i| - 1.$$

Since G is $\mathscr{F}_{k_1,k_2}$-free, such t must exist. Note that for any integer h, $1 \leq h \leq \sum_{i=1}^{t} |V_i| - 1$, the first t connected components of G (i.e. $G_1, G_2, \ldots, G_t$) have a h-vertex subgraph of size at least $h - 1$.

Let us now change G to a forest G' by replacing the first t connected components of G with a path graph of order $\sum_{i=1}^{t} |V_i|$ and size $\sum_{i=1}^{t} |E_i|$. Towards showing a contradiction, assume that G' has a subgraph H' of order k_1 and size greater than k_2. Suppose h vertices of H' are from the path graph added. We replace these h vertices with h vertices from the first t connected component of G that induce a subgraph of size at least $h - 1$. These new set of h vertices together with the $k_1 - h$ remaining vertices of H' induce a k_1-subgraph of size greater than k_2 in G. This is a contradiction because G is $\mathscr{F}_{k_1,k_2}$-free.

□

Appendix B. Proof of Proposition 2

Let $\mathcal{T}$ be an $[n,k,r]$-Tanner graph with minimum distance d^*. A Tanner graph determines the zero elements of code's parity-check matrix. Let $\mathbf{H}_{(n-k)\times n}$ be a parity-check matrix whose zero elements are set by $\mathcal{T}$, and the non-zero elements are chosen uniformly at random from $GF(q)$. Let V be any set of d^*-1 variable nodes. By Definition 4 and Hall's theorem [30], we get that there is a perfect matching between V and a set of d^*-1 check nodes, denoted C. Let $\mathbf{h}$ be the submatrix of $\mathbf{H}$ whose rows and columns correspond to the sets C and V, respectively. Using the Schwartz-Zippel theorem we get that the determinant of matrix $\mathbf{h}$ is non-zero with probability at least $1-\frac{d^*-1}{q}$. In other words, the d^*-1 failures corresponding to variable nodes V are recoverable with probability at least $1-\frac{d^*-1}{q}$. There are in total $\binom{n}{d^*-1}$ possible d^*-1 node failure combinations. By the union bound, the probability that every d^*-1 failures are recoverable is at least

$$1-\frac{d^*-1}{q}\binom{n}{d^*-1},$$

which is positive if $q > (d^*-1)\binom{n}{d^*-1}$. Therefore, there exists an $[n,k,r]$-LRC with minimum distance d^* over any finite field of size at least $(d^*-1)\binom{n}{d^*-1}+1$.

Now let us prove the converse. Suppose there is a $[n,k,r]$-LRC with minimum distance d^*. Let $\mathbf{H}_{(n-k)\times n}$ be a parity-check matrix of the LRC that has the maximum number of rows of Hamming weight at most $r+1$. Let $\mathcal{T}$ be the $[n,k]$-Tanner graph corresponding to $\mathbf{H}$. Note that every variable node in $\mathcal{T}$ must be adjacent to at least one check node of degree at most $r+1$; otherwise, by the construction of $\mathbf{H}$, we get that the code's locality is greater than r. Therefore, $\mathcal{T}$ is indeed an $[n,k,r]$-Tanner graph.

Let η be any integer in the interval $[n-k-d^*+2, n-k]$. We show that every set of η check nodes are adjacent to at least $\eta+k$ variable nodes. Towards showing a contradiction, suppose that there is a set C of η check nodes that are connected to at most $\eta+k-1$ variable nodes. There are in total n variable nodes, thus there is a set V of $n-(\eta+k-1)$ variable nodes that are not connected to any of the check nodes in C. In other words, there is a set V of $n-(\eta+k-1)$

variable nodes that are connected to at most $n-k-|C| = n-k-\eta$ check nodes. Note that $|V| = n-k-\eta+1$ and $\eta \in [n-k-d^*+2, n-k]$, thus $1 \leq |V| \leq d^* - 1$. If all the nodes in V fail, there will be not enough number of equations to recover them, because the number of check nodes connected to the variable nodes in V is less than the number of variable nodes in V. This is a contradiction, because any $d^* - 1$ failures are recoverable as the code's minimum distance is d^*. Therefore, for any integer $\eta \in [n-k-d^*+2, n-k]$, every set of η check nodes are adjacent to at least $\eta + k$ variable nodes. Consequently, by Definition 4, the minimum distance of $\mathcal{T}$ is at least d^*. This implies that the minimum distance of $\mathcal{T}$ is exactly d^*; otherwise, by the first part of this proof, there exists an $[n, k, r]$-LRC with minimum distance greater than d^*, which is not possible.

Acknowledgement

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada discovery grant.

References

[1] P. Gopalan, C. Huang, H. Simitci, S. Yekhanin, On the locality of codeword symbols, IEEE Trans. Inf. Theory 58 (11) (2012) 6925–6934.

[2] F. Oggier, A. Datta, Self-repairing homomorphic codes for distributed storage systems, in: INFOCOM, 2011, pp. 1215–1223.

[3] D. Papailiopoulos, J. Luo, A. Dimakis, C. Huang, J. Li, Simple regenerating codes: Network coding for cloud storage, in: INFOCOM, 2012, pp. 2801–2805.

[4] X. Li, L. Ma, C. Xing, Optimal locally repairable codes via elliptic curves, IEEE Trans. Inf. Theory 65 (1) (2019) 108–117.

[5] B. Chen, W. Fang, S. Xia, F. Fu, Constructions of optimal (r, δ) locally repairable codes via constacyclic codes, IEEE Trans. Commun. 67 (8) (2019) 5253–5263.

[6] L. Jin, Explicit construction of optimal locally recoverable codes of distance 5 and 6 via binary constant weight codes, IEEE Trans. Inf. Theory 65 (8) (2019) 4658–4663.

[7] J. Hao, S. Xia, K. W. Shum, B. Chen, F. Fu, Y. Yang, Bounds and constructions of locally repairable codes: Parity-check matrix approach, IEEE Trans. Inf. Theory 66 (12) (2020) 7465–7474.

[8] L. Jin, L. Ma, C. Xing, Construction of optimal locally repairable codes via automorphism groups of rational function fields, IEEE Trans. Inf. Theory 66 (1) (2020) 210–221.

[9] B. Chen, W. Fang, S. Xia, J. Hao, F. Fu, Improved bounds and singleton-optimal constructions of locally repairable codes with minimum distance 5 and 6, IEEE Trans. Inf. Theory 67 (1) (2021) 217–231.

[10] I. Tamo, A. Barg, A family of optimal locally recoverable codes, IEEE Trans. Inf. Theory 60 (8) (2014) 4661–4676.

[11] O. Kolosov, A. Barg, I. Tamo, G. Yadgar, Optimal LRC codes for all lengths n≤q, CoRR abs/1802.00157 (2018).
URL `http://arxiv.org/abs/1802.00157`

[12] V. Cadambe, A. Mazumdar, Bounds on the size of locally recoverable codes, IEEE Trans. Inf. Theory 61 (11) (2015) 5787–5794.

[13] N. Silberstein, A. Rawat, S. Vishwanath, Error-correcting regenerating and locally repairable codes via rank-metric codes, IEEE Trans. Inf. Theory 61 (11) (2015) 5765–5778.

[14] I. Tamo, D. Papailiopoulos, A. Dimakis, Optimal locally repairable codes and connections to matroid theory, IEEE Trans. Inf. Theory 62 (12) (2016) 6661–6671.

[15] W. Song, S. Dau, C. Yuen, T. Li, Optimal locally repairable linear codes, IEEE J. Sel. Areas Commun. 32 (5) (2014) 1019–1036.

[16] N. Prakash, V. Lalitha, P. V. Kumar, Codes with locality for two erasures, in: IEEE Int. Symp. Inf. Theory (ISIT), 2014, pp. 1962–1966.

[17] A. Wang, Z. Zhang, An integer programming-based bound for locally repairable codes, IEEE Trans. Inf. Theory 61 (10) (2015) 5280–5294.

[18] T. Westerbäck, R. Freij-Hollanti, T. Ernvall, C. Hollanti, On the combinatorics of locally repairable codes via matroid theory, IEEE Trans. Inf. Theory 62 (10) (2016) 5296–5315.

[19] B. Bollobás, Extremal Graph Theory, Dover, 2004.

[20] I. Holyer, The NP-completeness of edge-coloring, SIAM J. Comput. 10 (4) (1981) 718–720.

[21] M. Garey, D. Johnson, L. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comput. Sci. 1 (3) (1976) 237–267.

[22] P. Erdös, M. Simonovits, Compactness results in extremal graph theory, Combinatorica 2 (3) (1982) 275–288.

[23] P. Erdös, T. Gallai, Graphs with prescribed degrees of vertices (in hungarian), Matematikai Lapok 11 (1960) 264–274.

[24] S. L. Hakimi, On realizability of a set of integers as degrees of the vertices of a linear graph, SIAM J. Discrete Math. 10 (3) (1962) 496–506.

[25] F. Harary, Graph theory, Addison-Wesley, 1991.

[26] N. Prakash, G. Kamath, V. Lalitha, P. V. Kumar, Optimal linear codes with a local-error-correction property, in: IEEE Int. Symp. Inf. Theory (ISIT), 2012, pp. 2776–2780.

[27] A. Wang, Z. Zhang, Repair locality with multiple erasure tolerance, IEEE Trans. Inf. Theory 60 (11) (2014) 6979–6987.

[28] I. Tamo, A. Barg, A. Frolov, Bounds on the parameters of locally recoverable codes, IEEE Trans. Inf. Theory 62 (6) (2016) 3070–3083.

[29] A. Rawat, D. Papailiopoulos, A. Dimakis, S. Vishwanath, Locality and availability in distributed storage, IEEE Trans. Inf. Theory 62 (8) (2016) 4481–4493.

[30] J. Bondy, U. Murthy, Graph Theory with Applications, Elsevier, 1976.