Codes for Graph Erasures

Lev Yohananov
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
Email: levyohananov@campus.technion.ac.il

Eitan Yaakobi
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
Email: yaakobi@cs.technion.ac.il

***Abstract*—Motivated by systems where the information is represented by a graph, such as neural networks, associative memories, and distributed systems, we present in this work a new class of codes, called *codes over graphs*. Under this paradigm, the information is stored on the edges of an undirected graph, and a code over graphs is a set of graphs. A *node failure* is the event where all edges in the neighborhood of the failed node have been erased. We say that a code over graphs can tolerate ρ node failures if it can correct the erased edges of any ρ failed nodes in the graph. While the construction of such codes can be easily accomplished by MDS codes, their field size has to be at least $\mathcal{O}(n^2)$, when n is the number of nodes in the graph. In this work we present several constructions of codes over graphs with smaller field size. In particular, we present optimal codes over graphs correcting two node failures over the binary field, when the number of nodes in the graph is a prime number. We also present a construction of codes over graphs correcting ρ node failures for all ρ over a field of size at least $(n+1)/2-1$, and show how to improve this construction for optimal codes when $\rho = 2, 3$.**

I. Introduction

The traditional setup to represent information is by a vector over some fixed alphabet. Although this commonly used model is the most practical one, especially for storage and communication applications, it does not necessarily fit all information systems. In this work we study a different approach where the information is represented by a *graph*. This model is motivated by several information systems. For example, in *neural networks*, the neural units are connected via *links* which store and transmit information between the neural units [5]. Similarly, in associative memories, the information is stored by associations between different data items [11]. These two examples mimic the brain functionality which stores and processes information by associations between the information units. Furthermore, representing information in a graph can model a distributed storage systems [3] while every two nodes can share a link with the information that is stored between the nodes.

In this paper we present a new class of codes which we call *codes over graphs*. Under this setup we assume that there is an undirected complete graph with n *nodes* (vertices) such that the information is stored on the edges connecting between every two nodes in the graph, including self loops. The information on each edge is a symbol over some fixed alphabet so every graph can be represented by the symbols stored in each of the $\binom{n+1}{2}$ edges, and a code over graphs is simply a set of graphs. A *node failure* is the event where all edges in the node's neighborhood have been erased, and the goal of this work is to construct codes over graphs that can efficiently correct node failures. Namely, we say that a code over graphs can correct ρ node failures if it is possible to correct the erased edges in the neighborhoods of any failed ρ nodes. We study node failures since they correspond to the events of failing neural units in a neural network, data loss in an associative memory, and unavailable and failed nodes in distributed storage systems.

Since every graph can be represented by its adjacency matrix, a natural approach to construct codes over graphs is by their adjacency matrices. Thus, this class of codes is quite similar to the class of *array codes*, such as maximum-rank array codes [7], B codes [10], EVENODD codes [1], RDP code [2], X-codes [10], and regenerating codes [4], [6], [9]. However, there are two main differences between classical array codes and codes over graphs. First, since the graphs are undirected, the matrices are symmetric and square. Second, a failure of the ith node in the graph corresponds to the failure of the ith row and ith column in the adjacency matrix. Most existing constructions of array codes are not designed for symmetric or even square matrices. Furthermore, these constructions do not support the row-column failure model. The closest model to this setup is the one studied by Roth for crisscross errors [7], in which a fixed number of rows and columns have failed. While it is possible to use some of the results from [7], we will show in the paper that they do not provide codes over graphs with good parameters.

Assume a code over graphs with n nodes such that every edge stores a symbol. If ρ nodes have failed then the number of edges that were erased is

$$\binom{n+1}{2} - \binom{n-\rho+1}{2} = \rho n - \binom{\rho}{2}. \tag{1}$$

Therefore, the number of *redundancy edges* for every code which tolerates ρ node failures is at least $\rho n - \binom{\rho}{2}$. A code over graphs which meets this lower bound on the number of redundancy edges will be called an *optimal code over graphs*. Note that it is possible to construct optimal codes over graphs by using an $[\binom{n+1}{2}, \binom{n-\rho+1}{2}, \rho n - \binom{\rho}{2} + 1]$ MDS code. However, this will impose using a field of size at least $\binom{n+1}{2} - 1 = \Theta(n^2)$. Thus, the problem under this setup is the construction of such codes over smaller fields with optimal or close to optimal redundancy.

The rest of this paper is organized as follows. In Section II, we formally define the graph model studied in this paper and some preliminary results. In Section III, we present our main result in the paper of optimal binary codes over graphs correcting two node failures, when the number of nodes is prime. In Section IV, we extend our results for codes over graphs correcting arbitrary number of node failures over a field of size at least $(n+1)/2-1$. While this construction is almost optimal with respect to the bound in (1), we show how to improve it for optimal codes correcting two and three node failures. Lastly, in Section V, we study the existence of codes over graphs correcting $n-2$ node failures. Due to the lack of space some proofs of the results in the paper are omitted.

II. Definitions and Preliminaries

For a positive integer n, the set $\{0, 1, \ldots, n-1\}$ will be denoted by $[n]$. For a prime power q, $\mathbb{F}_q$ is the finite field of size q. A linear code of length n, dimension k, and minimum distance d over $\mathbb{F}_q$ will be denoted by $[n, k, d]_q$.

We will denote an undirected graph by $G = (V, E)$, where $V = \{v_0, v_1, \ldots, v_{n-1}\}$ is a set of n nodes (vertices) and $E \subseteq V \times V$ is its edge set. By a slight abuse of notation, every (undirected) edge in the graph will be denoted by (v_i, v_j)

where the order in this pair does not matter, that is, the pair (v_i, v_j) is identical to the pair (v_j, v_i). When possible we will denote the edges by (v_i, v_j), where $i \geqslant j$. We assume that the graph is simple in the sense that there are no parallel edges, however every node can have a self loop.

A graph G over an alphabet Σ is defined by a *labeling function* $L : E \to \Sigma$ and will be denoted by $G = (V, E, L)$. In this work we will extend the definition of the labeling function over $V \times V$, i.e. $L : V \times V \to \Sigma$, and have $0 \in \Sigma$ denote the case in which an edge does not exist. Hence, we can fully characterize the graph G by its vertex set V and labeling function L, and will denote it by $G(V, L)$, where we simply consider it as a complete graph with self loops. Under this setup, the *adjacency matrix* of the graph G is an $n \times n$ matrix $A_G = [a_{i,j}]_{i=0,j=0}^{n-1,n-1}$, where $a_{i,j} = L(v_i, v_j)$ for all $i, j \in [n]$. We will also use the *lower-triangle-adjacency matrix* of G to be the $n \times n$ matrix $A'_G = [a'_{i,j}]_{i=0,j=0}^{n-1,n-1}$ such that $a'_{i,j} = a_{i,j}$ if $i \geqslant j$ and otherwise $a'_{i,j} = 0$. For $i \in [n]$, the *neighborhood* of the ith node, denoted by N_i, is the set of edges connected to this node. Since we assumed the graph is complete, the neighborhood is simply the set $N_i = \{(v_i, v_j) | j \in [n]\}$.

Let Σ be a ring and G_1 and G_2 be two graphs over Σ with the same nodes set V. The operator $'' + ''$ between G_1 and G_2 over Σ, is defined by $G_1 + G_2 = G_3$, where G_3 is the unique graph satisfying $A_{G_1} + A_{G_2} = A_{G_3}$. Similarly, the operator $'' \cdot ''$ between G_1 and an element $\alpha \in \Sigma$, is denoted by $\alpha \cdot G_1 = G_3$, where G_3 is the unique graph satisfying $\alpha \cdot A_{G_1} = A_{G_3}$.

Definition 1. *A **code over graphs** of size M, length n over Σ is a set of undirected graphs $\mathcal{C}_\mathcal{G} = \{G_i = (V_n, L_i) | i \in [M]\}$ over Σ where $V_n = \{v_0, \ldots, v_{n-1}\}$. We denote such a code by $\mathcal{G}$-$(n, M)_\Sigma$ and in case $\Sigma = \{0, 1\}$, it will simply be denoted by $\mathcal{G}$-(n, M). The **dimension** of a code over graphs $\mathcal{C}_\mathcal{G}$ is $k_\mathcal{G} = \log_{|\Sigma|} M$, the rate is $R_\mathcal{G} = k_\mathcal{G}/\binom{n+1}{2}$, and the **redundancy** is defined to be $r_\mathcal{G} = \binom{n+1}{2} - k_\mathcal{G}$.*

*A code over graphs $\mathcal{C}_\mathcal{G}$ over a ring Σ will be called **linear** if for every $G_1, G_2 \in \mathcal{C}_\mathcal{G}$ and $\alpha, \beta \in \Sigma$ it holds that $\alpha G_1 + \beta G_2 \in \mathcal{C}_\mathcal{G}$. We denote this family of codes over graphs by $\mathcal{G}$-$[n, k_\mathcal{G}]_\Sigma$.*

*A linear code over graphs whose first k nodes contain the $\binom{k+1}{2}$ unmodified information symbols on their edges, is called a **systematic code over graphs**. All other $\binom{n+1}{2} - \binom{k+1}{2}$ edges in the graph are called **redundancy edges**. In this case we say that there are k **information nodes** and $r = n - k$ **redundancy nodes**. The number of **information edges** is $k_\mathcal{G} = \binom{k+1}{2}$, the redundancy is $r_\mathcal{G} = \binom{n+1}{2} - \binom{k+1}{2}$, and the rate is $R_\mathcal{G} = \binom{k+1}{2}/\binom{n+1}{2}$. We denote such a code by $\mathcal{SG}$-$[n, k]_\Sigma$.*

A *node failure* is the event, where all the edges incident to the failed node are erased in the graph, that is, its neighborhood set. In this case the failed node is known and it is required to complete the values of the edges in the node's neighborhood, which leads us to the following definition.

Definition 2. *A code over graphs is called a **ρ-node-erasure-correcting code** if it can correct the failure of any ρ nodes in each graph in the code.*

The minimum redundancy $r_\mathcal{G}$ of any ρ-node-erasure-correcting code of length n, satisfies

$$r_\mathcal{G} \geqslant \binom{n+1}{2} - \binom{n-\rho+1}{2} = \rho n - \binom{\rho}{2}. \qquad (2)$$

A code over graphs satisfying this inequality with equality will be called *optimal*. Hence for systematic codes over graphs the number of redundancy nodes is at least ρ. Note that for all n and ρ, one can always construct an optimal ρ-node-erasure-correcting code from an $[\binom{n+1}{2}, \binom{n-\rho+1}{2}, \rho n - \binom{\rho}{2} + 1]$ MDS code. However, then the field size of the code will be at least $\binom{n+1}{2} - 1 = \Theta(n^2)$. Our goal in this work is to construct ρ-node-erasure-correcting codes over small fields. When possible, we seek the field size to be binary and in any event at most $\mathcal{O}(n)$.

A closely related construction to our problem was given by Roth in [7]. In this work he presented a construction of maximum-rank array codes that can correct the failure of any combination of some μ rows and columns. Even though his construction results with square matrices, they are not necessarily symmetric. Yet, it is possible to slightly modify his construction in order to achieve $\mathcal{SG}$-$[n, n-2\rho]_q$ ρ-node-erasure-correcting codes over $\mathbb{F}_q$ for $q \geqslant n-1$, and hence the number of redundancy edges is $2\rho n - \binom{2\rho}{2}$. We will show that our construction in Section IV can improve these codes such that the number of redundancy edges to construct ρ-node-erasure-correcting codes will be only ρn. The next example exemplifies the definitions of codes over graphs.

Example 1. The following codes over graphs, given in Fig. 1, is a binary systematic single-node-erasure-correcting code of length 3. The information bits are stored in the edges of the subgraph containing nodes v_0 and v_1. The remaining three redundancy edges are completed in a way that the edges in the neighborhood of each node belong to a simple parity code of length three.

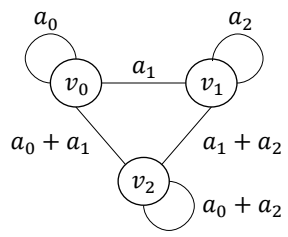


Fig. 1. An $\mathcal{SG}$-$[3, 2]$ single-node-erasure-correcting code.

The code construction from Example 1 is easily extended for arbitrary number of nodes by simple parity constraints for the neighborhoods of each node. Next we study the more interesting case of double-node-erasure-correcting codes.

III. Double-Node-Erasure-Correcting Codes

In this section we present a construction of binary double-node-erasure-correcting codes. We use the notation $\langle a \rangle_n$ to denote the value of $(a \bmod n)$.

Throughout this section we assume that $n \geqslant 5$ is a prime number. Let $G = (V_n, L)$ be a code over graphs with n vertices. Let us define for $m \in [n-1]$

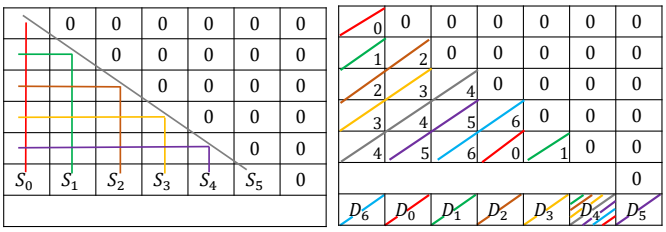
$$S_m = \begin{cases} \{(v_m, v_\ell) \mid \ell \in [n-1]\} & , m \in [n-2], \\ \{(v_\ell, v_\ell) \mid \ell \in [n-1]\} & , m = n-2. \end{cases}$$

and for $m \in [n]$

$$D_m = \{(v_k, v_\ell) | k, \ell \in [n] \backslash \{n-2\}, \langle k+\ell \rangle_n = m\} \cup \{(v_{n-1}, v_{n-2})\}.$$

The sets S_m where $m \in [n-2]$, will be used to represent parity constraints on the neighborhood of each node, which correspond to rows in the adjacency matrix. Similarly, the sets D_m will represent parity constraints on the diagonals of the adjacency matrix. Note that for all $m \in [n-1]$, $|S_m| = n - 1$ and for all $m \in [n]$, $|D_m| = \frac{n+1}{2}$.

Example 2. The sets S_m, D_m for $n = 7$ are marked in Fig. 2. Note that entries on lines with the same color belong to the same parity constraints.



(a) Neighborhood Parity Paths (b) Diagonal Parity Paths

Fig. 2. The neighborhoods and diagonals sets.

Recall that for $m \in [n]$ the failure set F_m of the mth node is its neighborhood set which we denote by $F_m = \{(v_m, v_\ell) \mid \ell \in [n]\}$.

Claim 1 *The sets S_m, D_m, F_m satisfy the following properties.*

(a) For all pairwise distinct $i, j, h \in [n-2]$, $S_h \cap (F_i \cup F_j) = \{(v_h, v_i), (v_h, v_j)\}$.
(b) For all distinct $i, j \in [n-2]$, $S_{n-2} \cap (F_i \cup F_j) = \{(v_i, v_i), (v_j, v_j)\}$.
(c) For all $i \in [n-2]$, $s \in [n] \setminus \{\langle i-2\rangle_n\}$, $D_s \cap F_i = \{(v_{\langle s-i\rangle_n}, v_i)\}$.
(d) For all distinct $i, j \in [n-2]$, $D_{\langle j-2\rangle_n} \cap (F_i \cup F_j) = \{(v_{\langle j-i-2\rangle_n}, v_i)\}$.
(e) For all distinct $i, j \in [n-2]$, $D_{\langle i+j\rangle_n} \cap (F_i \cup F_j) = \{(v_i, v_j)\}$.

We are now ready to present the construction of binary $\mathcal{SG}$-$[n, n-2]$ double-node-erasure-correcting codes.

Construction 1 For all $n \geqslant 5$ prime number let $\mathcal{C}_{\mathcal{G}_1}$ be the following code:

$$\mathcal{C}_{\mathcal{G}_1} = \left\{ G = (V_n, L) \,\middle|\, \begin{array}{l} (a) \sum_{(v_i,v_j)\in S_m} L(v_i, v_j) = 0, m \in [n-1] \\ (b) \sum_{(v_i,v_j)\in D_m} L(v_i, v_j) = 0, m \in [n] \end{array} \right\}.$$

Note that in this construction we had two sets of constraints. In the first set we had $n-1$ constraints and we call them constraint S_m for $m \in [n-1]$. Similarly, we call constraint D_m for $m \in [n]$. Furthermore, the edge (v_{n-1}, v_{n-2}) appears in each of the diagonal sets in order to have successul decoding when the failed nodes are $i \in [n-2]$ and $j = n-2$ (due to the lack of space we do not consider this case in the proof). Lastly, the correctness of this construction could be proved by defining a minimum distance for codes over graphs and showing that the minimum distance of this code is 3. However, this will not provide a decoding algorithm as we present in the following proof.

Theorem 3. *The code $\mathcal{C}_{\mathcal{G}_1}$ is an optimal binary double-node-erasure-correcting code.*

Proof: Assume that nodes $i, j \in [n]$, where $i < j$ are the failed nodes. We will show the correctness of this construction by explicitly showing its decoding algorithm. We will only consider the more difficult case of $i, j \in [n-2]$.

In this case we show an explicit algorithm which decodes all the erased edges. First, we denote the *single parity syndromes* for $m \in [n-1] \setminus \{i, j\}$ by

$$\widehat{S}_m = \sum_{(v_k,v_\ell)\in S_m \setminus (F_i \cup F_j)} L(v_k, v_\ell),$$

and the *diagonal parity syndromes* for $m \in [n]$ by

$$\widehat{D}_m = \sum_{(v_k,v_\ell)\in D_m \setminus (F_i \cup F_j)} L(v_k, v_\ell).$$

Let $d = \langle j-i\rangle_n$, $x = \langle -1-d^{-1}\rangle_n$ and $y = \langle -1+d^{-1}\rangle_n$. The decoding procedure is described in Algorithm 1.

Algorithm 1

$b_{prev} \leftarrow 0$	$b_{prev} \leftarrow 0$
for $t = 0, 1, \ldots, x$ **do**	**for** $t = 0, 1, \ldots, y$ **do**
$s_1 \leftarrow \langle -d(t+1) - 2\rangle_n$	$s_1 \leftarrow \langle d(t+1) - 2\rangle_n$
$s_2 \leftarrow \langle s_1 + j\rangle_n$	$s_2 \leftarrow \langle s_1 + i\rangle_n$
if $(s_1 \notin \{i, j, n-1\})$ **then**	**if** $(s_1 \notin \{i, j, n-1\})$ **then**
$L(v_{s_1}, v_j) \leftarrow \widehat{D}_{s_2} + b_{prev}$	$L(v_{s_1}, v_i) \leftarrow \widehat{D}_{s_2} + b_{prev}$
$L(v_{s_1}, v_i) \leftarrow \widehat{S}_{s_1} + L(v_{s_1}, v_j)$	$L(v_{s_1}, v_j) \leftarrow \widehat{S}_{s_1} + L(v_{s_1}, v_i)$
$b_{prev} \leftarrow L(v_{s_1}, v_i)$	$b_{prev} \leftarrow L(v_{s_1}, v_j)$
if $(s_1 = j)$ **then**	**if** $(s_1 = i)$ **then**
$L(v_{s_1}, v_j) \leftarrow \widehat{D}_{s_2} + b_{prev}$	$L(v_{s_1}, v_i) \leftarrow \widehat{D}_{s_2} + b_{prev}$
$L(v_i, v_i) \leftarrow \widehat{S}_{n-2} + L(v_{s_1}, v_j)$	$L(v_j, v_j) \leftarrow \widehat{S}_{n-2} + L(v_{s_1}, v_i)$
$b_{prev} \leftarrow L(v_i, v_i)$	$b_{prev} \leftarrow L(v_j, v_j)$
if $s_1 = n-1$ **then**	**if** $s_1 = n-1$ **then**
$L(v_{s_1}, v_j) \leftarrow \widehat{D}_{s_2} + b_{prev}$	$L(v_{s_1}, v_i) \leftarrow \widehat{D}_{s_2} + b_{prev}$

Denote $F^{(t)}$ as the set of uncorrected edges in the first loop and $\widetilde{F}^{(t)}$ in the second loop, so $F^{(0)} = F_i \cup F_j$ and $\widetilde{F}^{(0)} = F^{(x)}$. Denote $s_1^{(t)}$ and $s_2^{(t)}$ the values of s_1 and s_2 respectively on iteration t in the first loop. The values of $\widetilde{s}_1^{(t)}$ and $\widetilde{s}_2^{(t)}$ will be defined similarly for the second loop. The values of $s_1^{(t)}$ and $s_2^{(t)}$ are given by:

$$s_1^{(t)} = \langle -d(t+1) - 2\rangle_n, s_2^{(t)} = \langle s_1^{(t)} + j\rangle_n = \langle -dt + i - 2\rangle_n.$$

Similar expressions can be derived for $\widetilde{s}_1^{(t)}$ and $\widetilde{s}_2^{(t)}$. Next, we denote the following sets.

$$A = \{s_1^{(t_1)} | 0 \leqslant t_1 \leqslant x\}, B = \{\widetilde{s}_1^{(t_2)} | 0 \leqslant t_2 \leqslant y\}.$$

Claim 2 *The following properties holds:*

(a) $x \neq y$ and $x + y = n - 2$.
(b) $s_1^{(x)} = \widetilde{s}_1^{(y)} = n - 1$.
(c) $i, j \in A$ or $i, j \in B$ but not in both.
(d) $n - 2 \notin A \cup B$.
(e) $|A| = x + 1$, $|B| = y + 1$ and $A \cap B = \{n-1\}$

According to Claim 2(c), the variable s_1 in Algorithm 1 gets the values of i and j either in the first or the second loop. Let us assume for the rest of the proof that this happens in the first loop, i.e. $i, j \in A$, while the second case is proved similarly. We are now ready to show the correctness of the first loop by induction while the proof for the second loop is very similar.

Lemma 4. *For all $0 \leqslant t \leqslant x$, the following properties hold:*

1) If $s_1^{(t)} \notin \{i, j, n-1\}$ then $D_{s_2^{(t)}} \cap F^{(t)} = \{(v_{s_1^{(t)}}, v_j)\}$, $S_{s_1^{(t)}} \cap F^{(t)} = \{(v_{s_1^{(t)}}, v_i), (v_{s_1^{(t)}}, v_j)\}$, and the edges $(v_{s_1^{(t)}}, v_j), (v_{s_1^{(t)}}, v_i)$ are corrected on the tth iteration.
2) If $s_1^{(t)} = j$ then $D_{s_2^{(t)}} \cap F^{(t)} = \{(v_j, v_j)\}$, $S_{n-2} \cap F^{(t)} = \{(v_i, v_i), (v_j, v_j)\}$ and the edges $(v_i, v_i), (v_j, v_j)$ is corrected on the tth iteration.
3) If $s_1^{(t)} = n - 1$ then $D_{s_2^{(t)}} \cap F^{(t)} = \{(v_{n-1}, v_j)\}$ and the edge (v_{n-1}, v_j) is corrected on the tth iteration.

Proof: We prove this claim by induction on t.

Base: For $t = 0$ we have $s_1^{(0)} = \langle -d-2\rangle_n$, $s_2^{(0)} = \langle i - 2\rangle_n$, and $F^{(0)} = F_i \cup F_j$. Note that in this case, $s_1^{(0)} \neq i$, we will also assume that $s_1^{(0)} \notin \{j, n-1\}$ since these cases are proved similarly. Hence, we need to show that,

1) $D_{\langle i-2\rangle_n} \cap (F_i \cup F_j) = \{(v_{\langle -d-2\rangle_n}, v_j)\}$,
2) $S_{\langle -d-2\rangle_n} \cap (F_i \cup F_j) = \{(v_{\langle -d-2\rangle_n}, v_i), (v_{\langle -d-2\rangle_n}, v_j)\}$.
3) The edges $(v_{\langle -d-2\rangle_n}, v_j)$ and $(v_{\langle -d-2\rangle_n}, v_i)$ are corrected on this iteration.

The proof consists of the following observations:

- According to Claim 1(d) we deduce that $D_{\langle i-2\rangle_n} \cap (F_i \cup F_j) = \{(v_{\langle i-j-2\rangle_n}, v_j)\} = \{(v_{\langle -d-2\rangle_n}, v_j)\}$ and therefore the edge $(v_{\langle -d-2\rangle_n}, v_j)$ is corrected in Step 6 according to the constraint $D_{\langle i-2\rangle_n}$, therefore, $L(v_{\langle -d-2\rangle_n}, v_j) = \widehat{D}_{\langle i-2\rangle_n}$.
- According to Claim 1(a) we get $S_{\langle -d-2\rangle_n} \cap (F_i \cup F_j) = \{(v_{\langle -d-2\rangle_n}, v_i), (v_{\langle -d-2\rangle_n}, v_j)\}$, and therefore the edge $(v_{\langle -d-2\rangle_n}, v_i)$ is corrected in Step 7 according to the constraint $S_{\langle -d-2\rangle_n}$, by
$$L(v_{\langle -d-2\rangle_n}, v_i) = \widehat{S}_{\langle -d-2\rangle_n} + L(v_{\langle -d-2\rangle_n}, v_j).$$

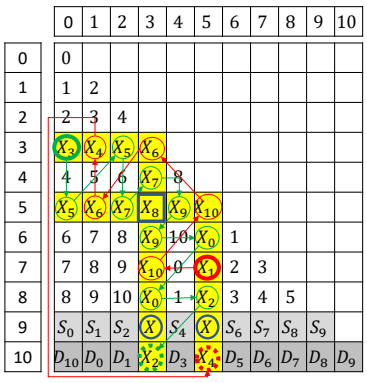
Step: Assume that the property holds for $t-1$, where $t \leqslant x$ and we prove its correctness for t. In this case, by Claim 2(b) and Claim 2(e) we have that $s_1^{(t-1)} \neq n-1$, so we will need to distinguish between the following cases: 1. $s_1^{(t-1)} = j$, 2. $s_1^{(t-1)} = i$, and 3. $s_1^{(t-1)} \neq i, j$. We will prove the claim for the third case while the first two cases are handled similarly. Hence, we assume that the edges $(v_{s_1^{(t-1)}}, v_j)$ and $(v_{s_1^{(t-1)}}, v_i)$, were corrected on iteration $t-1$. Since we assumed that $s_1^{(t-1)} \neq j$, we can deduce that $s_1^{(t)} \neq i$, and we consider three cases:

1) $s_1^{(t)} \neq j, n-1$: It is possible to show that $s_2^{(t)} \notin \{\langle i-2\rangle_n, \langle j-2\rangle_n\}$ and by Claim 1(c), we deduce that
$$D_{s_2^{(t)}} \cap (F_i \cup F_j) = \{(v_{\langle s_2^{(t)}-i\rangle_n}, v_i), (v_{\langle s_2^{(t)}-j\rangle_n}, v_j)\} = \{(v_{s_1^{(t-1)}}, v_i), (v_{s_1^{(t)}}, v_j)\}.$$
By the induction assumption $(v_{s_1^{(t-1)}}, v_i)$ was corrected, so,
$$D_{s_2^{(t)}} \cap (F_i^{(t)} \cup F_j^{(t)}) = \{(v_{s_1^{(t)}}, v_j)\},$$
and the edge $(v_{s_1^{(t)}}, v_j)$ is successfully corrected in Step 6 by constraint $D_{s_2^{(t)}}$. Furthermore, by Claim 2(d) $s_1^{(t)} \neq n-2$ and since $s_1^{(t)} \neq n-1$, by Claim 1(a),
$$S_{s_1^{(t)}} \cap (F_i \cup F_j) = \{(v_{s_1^{(t)}}, v_i), (v_{s_1^{(t)}}, v_j)\},$$
so it holds that $S_{s_1^{(t)}} \cap (F_i^{(t)} \cup F_j^{(t)}) = \{(v_{s_1^{(t)}}, v_i), (v_{s_1^{(t)}}, v_j)\}$ and therefore the edge $(v_{s_1^{(t)}}, v_i)$ can be successfully corrected in Step 7 by constraint $S_{s_1^{(t)}}$ and the value of $L(v_{s_1^{(t)}}, v_j)$.
2) $s_1^{(t)} = j$: As in the previous case, we first correct the edge $(v_{s_1^{(t)}}, v_j) = (v_j, v_j)$ by constraint $D_{s_2^{(t)}}$. Then, by Claim 1(b), $S_{n-2} \cap (F_i \cup F_j) = \{(v_i, v_i), (v_j, v_j)\}$ so it holds that $S_{n-2} \cap (F_i^{(t)} \cup F_j^{(t)}) = \{(v_i, v_i), (v_j, v_j)\}$ and therefore the edge (v_i, v_i) is corrected in Step 11 by constraint S_{n-2} and the value of $L(v_j, v_j)$.
3) $s_1^{(t)} = n-1$: This case is proved similarly as in the previous case. ■

The correctness of the second loop is proved similarly. Finally, it is possible to show that at the end of the algorithm the set of uncorrected edges is
$$\widetilde{F}^{(y)} = \{(v_i, v_j), (v_{n-2}, v_i), (v_{n-2}, v_j)\},$$
which can be corrected by the same arguments we used before. This completes the theorem's proof. ■

The decoding algorithm presented in the proof of Theorem 3 is demonstrated in the next example.

Example 3. We consider the case where $n = 11$ and the failed nodes are v_3 and v_5, that is, $i = 3, j = 5$. Therefore $d = 2$ and $x = 4, y = 5$. We use here the lower-triangle-adjacency matrix. The first loop starts with the edge (v_7, v_5), and ends with the edge (v_{10}, v_5). Similarly, the second loop starts with the edge (v_3, v_0) and ends with the edges (v_{10}, v_3). At the end of this algorithm, $(v_5, v_3), (v_9, v_3), (v_9, v_5)$ are the uncorrected edges and are marked in blue.



(a) Simulation of the algorithm

First Loop (Red)

s_2	s_1	(v_{s_1}, v_j)	(v_{s_1}, v_i)
1	7	(v_7, v_5)	(v_7, v_3)
10	5	(v_5, v_5)	(v_3, v_3)
8	3	–	–
6	1	(v_1, v_5)	(v_1, v_3)
4	10	(v_{10}, v_5)	

Second Loop (Green)

s_2	s_1	(v_{s_1}, v_i)	(v_{s_1}, v_i)
3	0	(v_0, v_3)	(v_0, v_5)
5	2	(v_2, v_3)	(v_2, v_5)
7	4	(v_4, v_3)	(v_4, v_5)
9	6	(v_6, v_3)	(v_6, v_5)
0	8	(v_8, v_3)	(v_8, v_5)
2	10	(v_{10}, v_3)	

(b) Corrected edge order

Fig. 3. The first loop is the red loop, and the second loop is the green loop.

IV. Multiple-Node-Erasure-Correcting Codes

In this section we present constructions of ρ-node-erasure-correcting codes for arbitrary ρ. An $[n \times n, k, d]$ symmetric linear array code $\mathcal{C}$ over a field $\mathbb{F}$ is a k-dimensional linear space of $n \times n$ symmetric matrices over $\mathbb{F}$, where the minimum rank of all nonzero matrices in $\mathcal{C}$ is d. According to [7], these codes can correct the erasure of any $d-1$ rows or columns in the array. A construction of binary $[n \times n, k, d]$ symmetric linear array codes where
$$k = \begin{cases} n(n-d+2)/2 & , n-d \text{ is even,} \\ (n+1)(n-d+1)/2 & , n-d \text{ is odd,} \end{cases}$$
was shown in [8]. Based on these codes, we present the following construction of binary ρ-node-erasure-correcting codes.

Construction 2 Let $\mathcal{C}$ be an $[n \times n, k, d = 2\rho+1]$ symmetric binary array code from [8], where
$$k = \begin{cases} n(n-2\rho+1)/2 & , n \text{ is odd,} \\ (n+1)(n-2\rho)/2 & , n \text{ is even,} \end{cases}$$
and $\rho < n/2$. The code over graphs $\mathcal{C}_{\mathcal{G}_2}$ is defined as follows,
$$\mathcal{C}_{\mathcal{G}_2} = \{G = (V_n, L) \mid A_G \in \mathcal{C}\}.$$

Theorem 5. *For all $\rho < n/2$ and*
$$k_{\mathcal{G}} = \begin{cases} n(n-2\rho+1)/2 & , n \text{ is odd,} \\ (n+1)(n-2\rho)/2 & , n \text{ is even,} \end{cases}$$
the code $\mathcal{C}_{\mathcal{G}_2}$ is a $\mathcal{G}$-$[n, k_{\mathcal{G}}]$ ρ-node-erasure-correcting code.

Note that this construction does not provide optimal ρ-node-erasure-correcting codes since
$$r_{\mathcal{G}} = \begin{cases} n\rho & , n \text{ is odd,} \\ (n+1)\rho & , n \text{ is even,} \end{cases}$$
which does not meet the bound in (2). For example, for $\rho = 2$ the difference between the code redundancy and the bound is one redundancy bit for n odd and three bits for n even. We note that for n even we have another construction with redundancy $r_{\mathcal{G}} = n\rho$, however it requires a field of size $q \geqslant n/2$. Lastly, we prove that it is possible to construct optimal systematic codes for $\rho = 2, 3$ for $q \geqslant n+1$; due to the lack of space we only present the construction for $\rho = 3$.

Let $G = (V_n, L)$ be a graph over a field $\mathbb{F}_q$. For a set of edges $U \subseteq V_n \times V_n$, we define $\boldsymbol{c}_U \in \mathbb{F}_q^{|U|}$ to be a

vector over $\mathbb{F}_q$ of length $|U|$, where its entries are the labels of the edges in the set U, in their lexicographic order, while we treat every edge as (v_i, v_j), with $i \geqslant j$. For example, if $U = \{(v_1, v_0), (v_6, v_3), (v_6, v_2), (v_5, v_3)\}$, then $\boldsymbol{c}_U = \big(L(v_1, v_0), L(v_5, v_3), L(v_6, v_2), L(v_6, v_3)\big)$.

For $\rho \geqslant 2$, let P_ρ be the set of all edges connecting between the first $n-\rho$ nodes but without the self loops, i.e.,

$$P_\rho = \{(v_k, v_\ell) \mid k, \ell \in [n-\rho], k > \ell\},$$

and let $\widetilde{P}_3 = P_3 \cup \{(v_{n-2}, v_{n-2}), (v_{n-1}, v_{n-2}), (v_{n-1}, v_{n-1})\}$. Next we define a new family of codes. Let $\mathbb{F}_q$ be a field of size at least $n+1$ and let $\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \in \mathbb{F}_q$ be n nonzero different elements in the field. Let $\mathcal{H}_{\widetilde{P}_3}$ be the following $3 \times (\binom{n-3}{2} + 3)$ matrix. The columns of the matrix are indexed by the entries corresponding to the edge set $\widetilde{P}_3$, that is, the set

$$\{(i,j) \in [n-3]^2 | i > j\} \cup \{(n-2, n-2), (n-1, n-2), (n-1, n-1)\},$$

in their lexicographic order,

$$\mathcal{H}_{\widetilde{P}_3} = \begin{array}{c} \begin{array}{ccccccc} (1,0) & (2,0) & \cdots & (n-3,n-4) & (n-2,n-2) & (n-1,n-2) & (n-1,n-1) \end{array} \\ \begin{bmatrix} \alpha^0_{\langle 1+0\rangle_n} & \alpha^0_{\langle 2+0\rangle_n} & \cdots & \alpha^0_{\langle n-3+n-4\rangle_n} & 1 & 0 & 0 \\ \alpha^1_{\langle 1+0\rangle_n} & \alpha^1_{\langle 2+0\rangle_n} & \cdots & \alpha^1_{\langle n-3+n-4\rangle_n} & 0 & 1 & 0 \\ \alpha^2_{\langle 1+0\rangle_n} & \alpha^2_{\langle 2+0\rangle_n} & \cdots & \alpha^2_{\langle n-3+n-4\rangle_n} & 0 & 0 & 1 \end{bmatrix} \end{array}.$$

Hence, each column in the matrix with index (i, j) represents the edge (v_i, v_j) in the graph. The $[\binom{n-3}{2} + 3, \binom{n-3}{2}]_q$ code over a field of size $q \geqslant n+1$, whose parity check matrix is $\mathcal{H}_{\widetilde{P}_3}$ will be denoted by $\mathcal{C}_{\widetilde{P}_3}$.

Claim 3 *For all $i, j, k \in [n-3]$, where $i < j < k$ the columns in the matrix $\mathcal{H}_{\widetilde{P}_3}$ with indices $(i, j), (i, k), (j, k)$ are linearly independent.*

Proof: For all $i, j, k \in [n-3]$, where $i < j < k$ we have that the elements $\alpha_{\langle i+j\rangle_n}, \alpha_{\langle i+k\rangle_n}, \alpha_{\langle j+k\rangle_n}$ are all different from each other. Therefore, the columns in the matrix $\mathcal{H}_{\widetilde{P}_3}$ with indices $(i, j), (i, k), (j, k)$ form a 3×3 Vandermonde matrix and in particular are linearly independent. ■

Claim 4 *For all pairwise distinct $i, j, k, m \in [n]$,*

$$|N_m \cap (F_i \cup F_j \cup F_k)| = 3.$$

We are now ready to present the construction of triple-node-erasure-correcting codes.

Construction 3 Let $n \geqslant 3$ be a positive integer and $q \geqslant n+1$ be a prime power. Let $\mathcal{C}_N$ be an $[n, n-3, 4]_q$ MDS code. Let $\mathcal{C}_{\widetilde{P}_3}$ be an $[\binom{n-3}{2} + 3, \binom{n-3}{2}]_q$ code. The code $\mathcal{C}_{\mathcal{G}_3}$ is defined as follows,

$$\mathcal{C}_{\mathcal{G}_3} = \Big\{ G = (V_n, L) \Big| \forall m \in [n-2], \boldsymbol{c}_{N_m} \in \mathcal{C}_N, \boldsymbol{c}_{\widetilde{P}_3} \in \mathcal{C}_{\widetilde{P}_3} \Big\}.$$

Theorem 6. *The code $\mathcal{C}_{\mathcal{G}_3}$ is a $\mathcal{G}$-$[n, k_{\mathcal{G}} = \binom{n-2}{2}]_q$ triple-node-erasure-correcting code, for $q \geqslant n+1$.*

Proof: Let v_i and v_j and v_k be the failed nodes. We will prove the more interesting case for which $i < j < k < n-2$. First, by Claim 4, for $m \in [n-2] \setminus \{i, j, k\}$, $|N_m \cap (F_i \cup F_j \cup F_k)| = 3$, therefore each of the $n-5$ codewords $\boldsymbol{c}_{N_m}$, for $m \in [n-2] \setminus \{i, j, k\}$, is reconstructed with the decoder of the code $\mathcal{C}_N$. Next, the edges $(v_i, v_j), (v_j, v_k), (v_i, v_k)$ will be corrected with the code $\mathcal{C}_{\widetilde{P}_3}$. The word $\boldsymbol{c}_{\widetilde{P}_3}$ has three erasures in the indices $(i, j), (j, k), (i, k)$ and by Claim 3, the appropriate columns of $\mathcal{H}_{\widetilde{P}_3}$ are linearly independent. Therefore they are corrected with the decoder of the code $\mathcal{C}_{\widetilde{P}_3}$. Finally, the codewords $\boldsymbol{c}_{N_i}, \boldsymbol{c}_{N_j}$ and $\boldsymbol{c}_{N_k}$ each has three missing edges $(v_\ell, v_\ell), (v_\ell, v_{n-2}), (v_\ell, v_{n-1})$ where $\ell \in \{i, j, k\}$, and they are corrected with the decoder of $\mathcal{C}_N$. ■

Lastly, we note that in this case we can also have a systematic construction of the codes in Construction 3.

V. $(n-2)$-Node-Erasure-Correcting Codes

In this section we study $(n-2)$-node-erasure-correcting codes over $\mathbb{F}_q$. In particular, we will find necessary and sufficient conditions for the existence of optimal codes, and in case they exist we will find the number of such codes.

Every code over graphs $\mathcal{G}$-$[n, k_{\mathcal{G}}]_q$ can be represented by a generator matrix $\mathbf{G}$ of dimensions $k_{\mathcal{G}} \times \binom{n+1}{2}$ over $\mathbb{F}_q$. As done before we denote the columns of the generator matrix $\mathbf{G}$ by the indices of the set $\{(i, j) \in [n]^2 \mid i \geqslant j\}$, in their lexicographic order, and so the column indexed by (i, j) is represented by a vector $\boldsymbol{g}_{i,j} \in \mathbb{F}_q^{k_{\mathcal{G}}}$.

Lemma 7. *Let $\mathbf{G}$ be a generator matrix of a $\mathcal{G}$-$[n, 3]_q$ code over graphs $\mathcal{C}_{\mathcal{G}}$. Then, $\mathcal{C}_{\mathcal{G}}$ is an optimal $(n-2)$-node-erasure-correcting code if and only if for all $i, j \in [n]$, the vectors $\boldsymbol{g}_{i,i}, \boldsymbol{g}_{i,j}$ and $\boldsymbol{g}_{j,j}$ are linearly independent.*

Proof: Denote by u_0, u_1, u_2 the information symbols in $\mathbb{F}_q$ that are encoded with $\mathcal{C}_{\mathcal{G}}$. After a failure of $n-2$ nodes we will have two nodes v_i and v_j where $i, j \in [n]$, with three symbols on the edges that were not erased $c_{i,i}, c_{i,j}, c_{j,j}$. Finding the correct values of the information symbols u_0, u_1, u_2 can be achieved if and only if the equation system

$$[u_0, u_1, u_2] \cdot [\boldsymbol{g}_{i,i}, \boldsymbol{g}_{i,j}, \boldsymbol{g}_{j,j}] = [c_{i,i}, c_{i,j}, c_{j,j}],$$

has a unique solution, that is, if and only if $\boldsymbol{g}_{i,i}, \boldsymbol{g}_{i,j}$ and $\boldsymbol{g}_{j,j}$ are linearly independent. ■

Lastly, we conclude with the following theorem.

Theorem 8. *For all positive integer $n \geqslant 3$ and prime power q, there exists an optimal $(n-2)$-node-erasure-correcting code over $\mathbb{F}_q$ if and only if $\frac{q^3-1}{q-1} > n-1$, and in this case, the number of such codes over graphs is*

$$q^{2\binom{n}{2}} (q-1)^{\binom{n+1}{2}} \frac{(q^2+q+1)!}{(q^2+q+1-n)!}.$$

Acknowledgments

The authors would like to thank Jehoshua Bruck for valuable discussions and Ron M. Roth for his contribution to the results in Section V and sharing reference [8] with them.

References

[1] M. Blaum, J. Brady, J. Bruck, and J. Menon. EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures. *IEEE Transactions on Computers*, 44(2):192–202, 1995.

[2] P. F. Corbett, R. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar. Row-diagonal parity for double disk failure correction (awarded best paper!). In C. Thekkath, editor, *FAST*, pages 1–14. USENIX, 2004.

[3] 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*, 56(9):4539–4551, Sept 2010.

[4] E. E. Gad, R. Mateescu, F. Blagojevic, C. Guyot, and Z. Bandic. Repair-optimal mds array codes over GF(2). In *ISIT*, pages 887–891. IEEE, 2013.

[5] J. J. Hopfield. Neurocomputing: Foundations of research. chapter Neural Networks and Physical Systems with Emergent Collective Computational Abilities, pages 457–464. MIT Press, Cambridge, MA, USA, 1988.

[6] N. Raviv, N. Silberstein, and T. Etzion. Constructions of high-rate minimum storage regenerating codes over small fields. In *ISIT*, pages 61–65. IEEE, 2016.

[7] R. M. Roth. Maximum-rank array codes and their application to criss-cross error correction. *IEEE Transactions on Information Theory*, 37(2):328–336, 1991.

[8] K.-U. Schmidt. Symmetric bilinear forms over finite fields of even characteristic. *J. Comb. Theory, Ser. A*, 117(8):1011–1026, 2010.

[9] I. Tamo, Z. Wang, and J. Bruck. Zigzag codes: MDS array codes with optimal rebuilding. *IEEE Transactions on Information Theory*, 59(3):1597–1616, 2013.

[10] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner. Low-density MDS codes and factors of complete graphs. *IEEE Transactions on Information Theory*, 45(6):1817–1836, 1999.

[11] E. Yaakobi and J. Bruck. On the uncertainty of information retrieval in associative memories. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 106–110, July 2012.