Double and Triple Node-Erasure-Correcting Codes Over Complete Graphs

Lev Yohananov, *Student Member, IEEE*, Yuval Efron, and Eitan Yaakobi, *Senior Member, IEEE*

***Abstract*— In this paper we study array-based codes over graphs for correcting multiple node failures. These codes have applications to neural networks, associative memories, and distributed storage systems. We assume that the information is stored on the edges of a complete undirected graph and a *node failure* is the event where all the edges in the neighborhood of a given node have been erased. A code over graphs is called *ρ-node-erasure-correcting* if it allows to reconstruct the erased edges upon the failure of any ρ nodes or less. We present a binary optimal construction for double-node-erasure correction together with an efficient decoding algorithm, when the number of nodes is a prime number. Furthermore, we extend this construction for triple-node-erasure-correcting codes when the number of nodes is a prime number and two is a primitive element in $\mathbb{Z}_n$. These codes are at most a single bit away from optimality.**

***Index Terms*— Array codes, crisscross erasures, codes over graphs, rank metric codes.**

I. Introduction

NETWORKS and distributed storage systems are usually represented as graphs with the information stored in the nodes (vertices) of the graph. In our recent work [23]–[25], we have introduced a new model which assumes that the information is stored on the *edges*. This setup 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 [10]. Similarly, in associative memories, the information is stored by associations between different data items [21]. Furthermore, representing information in a graph can model a distributed storage system [7] while every two nodes can be connected by a link that represents the information that is shared by the nodes.

In [23]–[25], we introduced the notion of *codes over graphs*, which is a class of codes storing the information on the edges of a complete undirected graph (including self-loops). Thus, each codeword is a labeled graph with n *nodes* (vertices) and each of the $\binom{n+1}{2}$ edges stores a symbol over an alphabet Σ. A *node failure* is the event where all the edges incident with a given node have been erased, and a code over graphs is called *ρ-node-erasure-correcting* if it allows to reconstruct the contents of the erased edges upon the failure of any ρ nodes or less.

The information stored in a complete undirected graph can be represented by an $n \times n$ symmetric array and a failure of the ith node corresponds to the erasure of the ith row and ith column in the array. Hence, this problem is translated to the problem of correcting *symmetric crisscross erasures* in square symmetric arrays [15]. By the Singleton bound, the number of *redundancy edges* (i.e., redundancy symbols in the array) of every ρ-node-erasure-correcting code must be at least $n\rho - \binom{\rho}{2}$, and a code meeting this bound will be referred as *optimal*. While the construction of optimal codes is easily accomplished by MDS codes, their alphabet size must be at least the order of n^2, and the task of constructing optimal (or close to optimal) codes over graphs over smaller alphabets remains an intriguing problem.

A natural approach to address this problem is by using the wide existing knowledge on array code constructions such as [2], [4]–[6], [9], [11]–[18], [20]. However, the setup of codes over graphs differs from that of classical array codes in two respects. First, the arrays are symmetric, and, secondly, a failure of the ith node in the graph corresponds to the failure of the ith row and the ith column (for the same i) in the array. Most existing constructions of array codes are not designed for symmetric arrays, and they do not support this special row–column failure model. However, it is still possible to use existing code constructions and modify them to the special structure of the above erasure model in graphs, as was done in [23]–[25]. More specifically, based upon product codes [1], [8], a construction of optimal codes whose alphabet size grows only linearly with n has been proposed. Additionally, using rank-metric codes [15]–[17], binary codes over graphs were designed, however they are relatively close—yet do not attain—the Singleton bound. In [23], [24], a construction of optimal binary codes for two node failures was also presented based upon ideas from EVENODD codes [2].

In this paper we build upon some of the methods that were used in the array code constructions. An example of such an approach is the algebraic representation of EVENODD codes [2]. Another similar construction for optimal $(p-1) \times p$ array codes that can tolerate any ρ column erasures was given in [5] as well as its extensions in [4] and [11]. In these papers, the authors also used an algebraic approach, where each column of the array code is represented as a symbol

Manuscript received June 11, 2019; revised December 9, 2019; accepted January 19, 2020. Date of publication February 6, 2020; date of current version June 18, 2020. This article was presented in part at the 2019 IEEE International Symposium on Information Theory. *(Corresponding author: Eitan Yaakobi.)*

The authors are with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: levyohananov@cs.technion.ac.il; szxrtde@cs.technion.ac.il; yaakobi@cs.technion.ac.il).

Communicated by S. R. Ghorpade, Associate Editor for Coding Theory.

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

Digital Object Identifier 10.1109/TIT.2020.2971997

over a fixed ring, which is then interpreted as a linear MDS code of length p over the ring. Note that these constructions cannot be used directly for the problem studied in the paper for the reasons mentioned above. However, we still show how to take advantage of this algebraic approach in order to construct double- and triple-node-erasure-correcting codes.

Another approach for handling symmetric crisscross erasures (in symmetric arrays) is by using symmetric rank-metric codes. In [16], Schmidt presented a construction of linear $[n \times n, k, d]$ symmetric binary array codes with minimum rank d, where $k = n(n-d+2)/2$ if $n-d$ is even, and $k = (n+1)(n-d+1)/2$ otherwise. Such codes can correct any $d-1$ column or row erasures. Hence, it is possible to use these codes to derive ρ-node-failure-correcting codes while setting $d = 2\rho+1$, as the ρ node failures translate into the erasure of ρ columns and ρ rows. However, the redundancy of these codes is $\binom{\rho}{2}$ symbols away from the Singleton bound for symmetric crisscross erasures (e.g., for $\rho = 2$, their redundancy is $2n$ while the Singleton lower bound is $2n-1$).

In this paper we carry an algebraic approach such as the one presented in [4], [5], and [11] in order to propose new constructions of binary codes over graphs. In Section II, we formally define codes over graphs and review several basic properties from [23], [25] that will be used in the paper. In Section III, we present our optimal binary construction for two-node failures along with its decoding procedure. This construction is not only simpler than the one given in [23], [25], but it also provides a good intuition to understand the triple-node-erasure-correcting codes in the paper. Furthermore, in Section IV, it is shown how to efficiently decode the case of a single node failure for this construction. Then, in Section V, we extend this construction for the three-node failures case. This new construction is only at most a single bit away from the Singleton bound, thereby outperforming the construction obtained from [16]. In Section VI, we show how to efficiently decode the failure of three nodes. Lastly, Section VII concludes the paper.

II. Definitions and Preliminaries

For a positive integer n, the set $\{0, 1, \ldots, n-1\}$ will be denoted by $[n]$ and for a prime power q, $\mathbb{F}_q$ is the finite field of size q. A linear code of length n and dimension k over $\mathbb{F}_q$ will be denoted by $[n, k]_q$ or $[n, k, d]_q$, where d denotes its minimum distance. In the rest of this section, we follow the definitions of our previous work [23] for codes over graphs.

A graph will be denoted by $G = (V_n, E)$, where $V_n = \{v_0, v_1, \ldots, v_{n-1}\}$ is its set of n nodes (vertices) and $E \subseteq V_n \times V_n$ is its edge set. In this paper, we only study complete undirected graphs with self-loops, and in this case, the edge set of an undirected graph G over an alphabet Σ is defined by $E = \{(v_i, v_j) \mid (v_i, v_j) \in V_n \times V_n, i \geqslant j\}$, with a labeling function $L : V_n \times V_n \to \Sigma$. By a slight abuse of notation, every undirected edge in the graph will be denoted by $\langle v_i, v_j \rangle$ where the order in this pair does not matter, that is, the notation $\langle v_i, v_j \rangle$ is identical to the notation $\langle v_j, v_i \rangle$, and thus there are $\binom{n+1}{2}$ edges. We will use the notation $G = (V_n, L)$ for such graphs. For the rest of the paper, whenever we refer to a graph we refer to an undirected graph.

The ***labeling matrix*** of an undirected graph $G = (V_n, L)$ is an $n \times n$ symmetric matrix over Σ denoted by $A_G = [a_{i,j}]_{i=0,j=0}^{n-1,n-1}$, where $a_{i,j} = L\langle v_i, v_j \rangle$. We also use the ***lower-triangle-labeling 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 $a'_{i,j} = 0$. The ***zero graph*** will be denoted by $G_{\mathbf{0}}$ where for all $i, j \in [n]$, $a_{i,j} = 0$.

Let Σ be a ring and G_1 and G_2 be two graphs over Σ with the same node 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 $\alpha \cdot A_{G_1} = A_{G_3}$.

A ***code over graphs*** over Σ of length n and size M is a set of graphs $\mathcal{C} = \{G_i = (V_n, L_i) | i \in [M]\}$ over Σ, and it will be denoted by $(n, M)_\Sigma$. In case that $\Sigma = \{0, 1\}$, we simply use the notation (n, M). The ***dimension*** of a code over graphs $\mathcal{C}$ is $k = \log_{|\Sigma|} M$ and the ***redundancy*** is $r = \binom{n+1}{2} - k$. A code over graphs $\mathcal{C}$ over a ring Σ will be called ***linear*** and will be denoted by $\mathcal{U}$-$[n, k]_\Sigma$ if for every $G_1, G_2 \in \mathcal{C}$ and $\alpha, \beta \in \Sigma$, $\alpha G_1 + \beta G_2 \in \mathcal{C}$.

The ***neighborhood edge set*** of the ith node of an undirected graph $G = (V_n, L)$ is defined by $N_i = \{\langle v_i, v_j \rangle \mid j \in [n]\}$, and it corresponds to the ith column and the ith row in the labeling matrix A_G. The ***node failure*** of the ith node is the event in which all the edges in the neighborhood set of the ith node, i.e. N_i, are erased. We will also denote this edge set by F_i and refer to it by the ***failure set*** of the ith node. A code over graphs is called a ***ρ-node-erasure-correcting code*** if it can correct any failure of at most ρ nodes in each of its graphs.

As discussed in [23]–[25], according to the Singleton bound, the minimum redundancy r of any ρ-node-erasure-correcting code of length n, satisfies

$$r \geqslant \binom{n+1}{2} - \binom{n-\rho+1}{2} = n\rho - \binom{\rho}{2}, \qquad (1)$$

and a code over graphs which satisfies this inequality with equality is called ***optimal***. It was also observed in [23]–[25] that for all n and ρ, an optimal ρ-node-erasure-correcting code exists over a field of size at least $\Theta(n^2)$, and thus the goal is to construct such codes over smaller fields, and ideally over the binary field.

We conclude this section with reviewing the definition of a distance metric over graphs from [25] and its connection to the construction of codes correcting node failures. Let $G = (V_n, L)$ be a graph and let E be the set of all nonzero labeled edges of G, i.e., $E = \{e \in V_n \times V_n \mid L(e) \neq 0\}$. A ***vertex cover*** W of G is a subset of V_n such that for each $\langle v_i, v_j \rangle \in E$ either $v_i \in W$ or $v_j \in W$. The ***graph weight*** of G is defined by

$$w(G) = \min_{W \text{ is a vertex cover of } G} \{|W|\},$$

and the ***graph distance*** between two graphs G_1, G_2 will be denoted by $d(G_1, G_2)$ where it holds that $d(G_1, G_2) = w(G_1 - G_2)$. It was proved in [25] that this graph distance is a metric. The ***minimum distance*** of a code over graphs $\mathcal{C}$, denoted by $d(\mathcal{C})$, is the minimum graph distance between

any two distinct graphs in $\mathcal{C}$, that is

$$d(\mathcal{C}) = \min_{G_1 \neq G_2 \ G_1, G_2 \in \mathcal{C}} \{d(G_1, G_2)\},$$

and in case the code is linear $d(\mathcal{C}) = \min_{G \in \mathcal{C}, G \neq G_0}\{w(G)\}$. Lastly, we state the following theorem from [25] that establishes the connection between the graph distance and the node-erasure-correction capability.

Theorem 1: A linear code over graphs $\mathcal{C}$ is a ρ-node-erasure-correcting code if and only if its minimum distance satisfies $d(\mathcal{C}) \geqslant \rho + 1$.

Let $n \geqslant 2$ be a prime number. Denote by $\mathcal{R}_n$ the ring of polynomials of degree at most $n-1$ over $\mathbb{F}_2$. It is well known that $\mathcal{R}_n$ is isomorphic to the ring of all polynomials in $\mathbb{F}_2[x]$ modulo $x^n - 1$. Denote by $M_n(x) \in \mathcal{R}_n$ the polynomial $M_n(x) = \sum_{\ell=0}^{n-1} x^\ell$ over $\mathbb{F}_2$, where it holds that $M_n(x)(x+1) = x^n - 1$ as a multiplication of polynomials over $\mathbb{F}_2[x]$. To avoid confusion in the sequel, since we are using only polynomials over $\mathbb{F}_2$, the notation $x^\ell + 1$ for all $\ell \in [n]$, will refer to a polynomial in $\mathcal{R}_n$ and for $\ell = n$, we will use the notation $x^n - 1$. It is well known that for all $\ell \in [n]$ it holds that

$$\gcd(x^\ell + 1, x^n - 1) = x^{\gcd(\ell, n)} + 1 = x + 1,$$

and since $M_n(x)(x+1) = x^n - 1$ it can be verified that

$$\gcd(x^\ell + 1, M_n(x)) = 1. \tag{2}$$

Notice also that when 2 is primitive in $\mathbb{Z}_n$, the polynomial $M_n(x)$ is irreducible [3]. The last important and well known property we will use for polynomials over $\mathbb{F}_2$ is that for all $k = 2^j$, $j \in \mathbb{N}$ it holds that $1 + x^{sk} = (1 + x^s)^k$. The notation $\langle a \rangle_n$ will be used to denote the value of $(a \bmod n)$.

III. Optimal Binary Double-Node-Erasure-Correcting Codes

In this section we present a family of optimal binary linear double-node-erasure-correcting codes with n nodes, where n is a prime number.

Remember that for $i \in [n]$ the ith neighborhood set of the ith node is $N_i = \{\langle v_i, v_j \rangle \mid j \in [n]\}$. Let $n \geqslant 3$ be a prime number and let $G = (V_n, L)$ be a graph with n vertices. For $h \in [n]$ we define the neighborhood of the hth node without itself self-loop by

$$S_h = \big\{\langle v_h, v_\ell \rangle \mid \ell \in [n], h \neq \ell\big\}. \tag{3}$$

We also define for $m \in [n]$, the mth diagonal set by

$$D_m = \{\langle v_k, v_\ell \rangle | k, \ell \in [n], \langle k + \ell \rangle_n = m\}. \tag{4}$$

The sets S_h for $h \in [n]$ will be used to represent parity constraints on the neighborhood of each node and similarly the sets D_m for $m \in [n]$ will be used to represent parity constraints on the diagonals with slope one in the labeling matrix A_G. We state that for all $m \in [n]$, the size of D_m is $\frac{n+1}{2}$. This holds since in each neighborhood $N(v_i)$, there is only a single edge which belongs to D_m, which is the edge $\langle v_i, v_{\langle m-i \rangle_n} \rangle$. Another important observation is that D_m contains only a single self-loop which is the edge $\langle v_{\langle m \cdot 2^{-1} \rangle_n}, v_{\langle m \cdot 2^{-1} \rangle_n} \rangle$.

Example 1: In Fig. 1 we demonstrate the sets S_h and D_m, where $h, m \in [11]$, of a graph $G = (V_{11}, L)$ on its lower-triangle-labeling matrix A'_G.

Motivated by the algebraic approach of the work in [4], [5], and [11], we introduce one more useful notation for graphs. Let $G = (V_n, L)$ be a graph. For $i \in [n]$ we denote the *neighborhood-polynomials* of G to be

$$a'_i(x) = e_{i,0} + e_{i,1}x + e_{i,2}x^2 + \cdots + e_{i,n-1}x^{n-1},$$

where for $i, j \in [n]$, $e_{i,j} = a_{i,j} = L\langle v_i, v_j \rangle$. We also denote the *neighborhood-polynomial without self-loops* of G to be

$$a_i(x) = a'_i(x) - e_{i,i}x^i.$$

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

Construction 1: Let $n \geqslant 3$ be a prime number. The code over graphs $\mathcal{C}_2$ is defined as follows,

$$\mathcal{C}_2 = \left\{ G = (V_n, L) \middle| \begin{array}{ll} (a) & \sum_{\langle v_i, v_j \rangle \in S_h} e_{i,j} = 0, h \in [n] \\ (b) & \sum_{\langle v_i, v_j \rangle \in D_m} e_{i,j} = 0, m \in [n] \end{array} \right\}.$$

Note that for *any* graph G over the binary field, it holds that

$$\sum_{h \in [n]} \sum_{\langle v_i, v_j \rangle \in S_h} e_{i,j} = \sum_{h=0}^{n-1} \sum_{\ell=0, \ell \neq h}^{n-1} e_{h,\ell} = 2 \sum_{h=0}^{n-1} \sum_{\ell=0}^{h-1} e_{h,\ell} = 0. \tag{5}$$

Therefore the code $\mathcal{C}_2$ has at most $2n-1$ linearly independent constraints which implies that its redundancy is at most $2n-1$. Our main result in this section, which is stated in Theorem 2, claims that $\mathcal{C}_2$ is a double-node-correcting code, i.e. its minimum distance is three. Thus, according to the Singleton bound we get that the redundancy of the code $\mathcal{C}_2$ is exactly $2n-1$, which implies that it is an optimal code. In the rest of this section we provide the proof Theorem 2 by showing a complexity optimal decoder for the code $\mathcal{C}_2$ and prove its correctness.

Throughout this section we assume that G is a graph in the code $\mathcal{C}_2$ and $a_\ell(x)$ for $\ell \in [n]$ are its neighborhood polynomials. We also assume that the failed nodes are v_0, v_i. First, we define the following two polynomials $S_1(x), S_2(x) \in \mathcal{R}_n$, which will be called the *syndrome polynomials*

$$S_1(x) = a_0(x) + a_i(x),$$
$$S_2(x) \equiv a_0(x) + a_i(x)x^i (\mathrm{mod} x^n - 1).$$

Next, we prove the following claim.

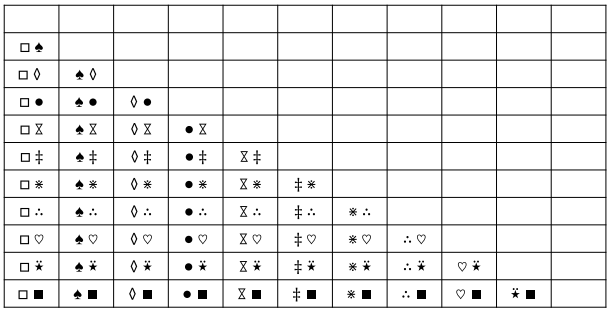
Claim 1: The following properties hold on the graph G:

a) For all $h \in [n] \setminus \{0, i\}$, the value of $e_{h,0} + e_{h,i}$ is known.
b) For all $m \in [n] \setminus \{i\}$, the value of $e_{0,m} + e_{i,\langle m-i \rangle_n}$ is known.
c) The value of $e_{0,i}$ is known.

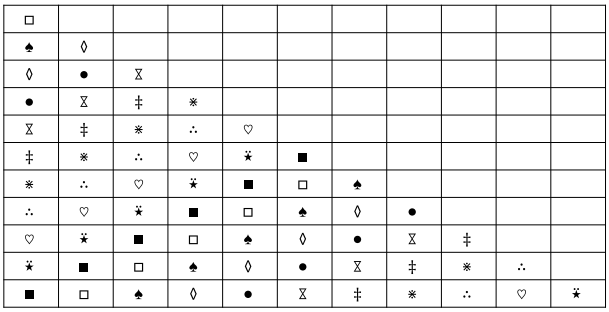
Proof:

a) According to the neighborhood constraint S_h for all $h \in [n] \setminus \{0, i\}$, we have that

$$0 = \sum_{\langle v_h, v_\ell \rangle \in S_h} e_{h,\ell} = \sum_{\ell=0, \ell \neq h}^{n-1} e_{h,\ell} = e_{h,0} + e_{h,i} + \sum_{\ell \in [n] \setminus \{0,i,h\}} e_{h,\ell},$$



(a) Neighborhood-Parity Constraints



(b) Slope-One-Diagonal-Parity Constraints

Fig. 1. The constraints over undirected graphs, represented on the lower-triangle-labeling matrix.

and since $e_{h,\ell}$ are known for all $\ell \in [n]\backslash\{0,i,h\}$, we get that the value of $e_{h,0}+e_{h,i}$ is known.

b) For all $m \in [n]\backslash\{i\}$, the set $D_m\backslash\{\langle v_0,v_m\rangle, \langle v_i,v_{\langle m-i\rangle_n}\rangle\}$ is denoted by D'_m. Therefore, we have that

$$0 = \sum_{\langle v_\ell, v_{\langle m-\ell\rangle_n}\rangle \in D_m} e_{\ell,\langle m-\ell\rangle_n} = \sum_{\langle v_\ell, v_{\langle m-\ell\rangle_n}\rangle \in D'_m} e_{\ell,\langle m-\ell\rangle_n} + e_{0,m} + e_{i,\langle m-i\rangle_n},$$

and since the value of $e_{s,\ell}$ is known for all $\langle v_s, v_\ell\rangle \in D'_m$, we get that the value of $e_{0,m}+e_{i,\langle m-i\rangle_n}$ is known.

c) According to the diagonal constraint D_i we get that

$$0 = \sum_{\langle v_s,v_\ell\rangle \in D_i} e_{s,\ell} = e_{0,i} + \sum_{\langle v_s,v_\ell\rangle \in D_i\backslash\{\langle v_0,v_i\rangle\}} e_{s,\ell},$$

and since the value of $e_{s,\ell}$ is known, for all $\langle v_s, v_\ell\rangle \in D_i \setminus \{\langle v_0, v_i\rangle\}$, the value of $e_{0,i}$ is known.

■

Denote the polynomials $\widetilde{S}_1(x)$ and $\widetilde{S}_2(x)$ by

$$\widetilde{S}_1(x) = \sum_{h=1,h\neq i}^{n-1} \Big(e_{0,h}+e_{i,h}\Big)x^h, \tag{6}$$

$$\widetilde{S}_2(x) = \sum_{m=0,m\neq i}^{n-1} \Big(e_{0,m}+e_{i,\langle m-i\rangle_n}\Big)x^m. \tag{7}$$

Note that it is possible to compute $\widetilde{S}_1(x)$ and $\widetilde{S}_2(x)$ based upon the nodes that did not fail, and as we showed in Claim 1. Also note that calculating $\widetilde{S}_1(x)$ requires $(n-2)(n-4)$ XOR operations since in each of the $n-2$ constraints we read $n-3$ edges. Similarly, $\widetilde{S}_2(x)$ requires $(n-1)(n-5)/2$ XOR operations since in each of the $n-1$ constraints we read $(n-3)/2$ edges. Next we show an important claim.

Claim 2: The following properties hold on the graph G:

a) For all $h \in [n]$, $a_h(1) = 0$.

b) $a_0(x)+a_i(x) = e_{i,0}(1+x^i)+\widetilde{S}_1(x)$.

c) $a_0(x)+a_i(x)x^i \equiv e_{0,0}+e_{i,i}x^{2i}+\widetilde{S}_2(x)(\mathrm{mod}\, x^n-1)$.

Proof:

a) By the definition of the neighborhood constraints, for all $h \in [n]$, $S_h = \{\langle v_h, v_\ell\rangle \mid \ell \in [n], h \neq \ell\}$, and therefore

$$a_h(1) = \sum_{\ell=0,\ell\neq h}^{n-1} e_{h,\ell} = \sum_{\langle v_h,v_\ell\rangle \in S_h} e_{h,\ell} = 0.$$

b)

$$\begin{aligned} a_0(x)+a_i(x) &= \\ &= e_{0,0}+e_{i,i}x^i+\sum_{h=0}^{n-1} e_{0,h}x^h+\sum_{h=0}^{n-1} e_{i,h}x^h \\ &= e_{0,0}+e_{i,i}x^i+\sum_{h=0}^{n-1}\Big(e_{0,h}+e_{i,h}\Big)x^h \\ &= e_{0,0}+e_{i,i}x^i+\Big(e_{0,0}+e_{i,0}\Big)+\Big(e_{0,i}+e_{i,i}\Big)x^i+\widetilde{S}_1(x) \\ &= e_{i,0}(1+x^i)+\widetilde{S}_1(x). \end{aligned}$$

c)

$$\begin{aligned} a_0(x)+a_i(x)x^i &= \\ &= e_{0,0}+e_{i,i}x^{2i}+\sum_{m=0}^{n-1} e_{0,m}x^m+\sum_{m=0}^{n-1} e_{i,m}x^{m+i} \\ &\equiv e_{0,0}+e_{i,i}x^{2i}+\sum_{m=0}^{n-1} e_{0,m}x^m \\ &\quad +\sum_{m=0}^{n-1} e_{i,\langle m-i\rangle_n}x^m(\mathrm{mod}\, x^n-1) \\ &\equiv e_{0,0}+e_{i,i}x^{2i} \\ &\quad +\sum_{m=0}^{n-1}\Big(e_{0,m}+e_{i,\langle m-i\rangle_n}\Big)\cdot x^m(\mathrm{mod}\, x^n-1) \\ &\equiv e_{0,0}+e_{i,i}x^{2i}+\Big(e_{0,i}+e_{i,0}\Big)x^i+\widetilde{S}_2(x)(\mathrm{mod}\, x^n-1) \\ &\equiv e_{0,0}+e_{i,i}x^{2i}+\widetilde{S}_2(x)(\mathrm{mod}\, x^n-1). \end{aligned}$$

■

If no nodes have failed in the graph G, then we can easily compute both of these polynomials since we know the values of all the edges. However in case that v_0, v_i both failed this becomes a far less trivial problem. However, using several properties, that will be proved in this section, we will prove that it is still possible to compute $S_1(x)$ and $S_2(x)$ entirely even though the nodes v_0, v_i failed. First, we show that it is possible to compute $S_1(x)$ and almost compute $S_2(x)$.

Claim 3: Given the two node failures v_0, v_i, it is possible to exactly compute the polynomial $S_1(x)$.

Proof: By using the result of Claim 2(b) we deduce that

$$S_1(x) = a_0(x)+a_i(x) = e_{i,0}(1+x^i)+\widetilde{S}_1(x).$$

According to Claim 1(a) we can compute the polynomial $\widetilde{S}_1(x)$ and due to Claim 1(c) we can compute $e_{i,0}$. Thus, we can compute the polynomial $S_1(x)$. ∎

According to Claim 1(c) it is possible compute the edge $e_{i,0}$ and this calculation requires $(n-3)/2$ XOR operations. Since the calculation of $\widetilde{S}_1(x)$ requires $(n-2)(n-4)$ XOR operations, we deduce that it takes $(n-3)/2 + (n-2)(n-4)$ XOR operations to calculate $S_1(x)$. Next we show how to calculate the polynomial $S_2(x)$.

Claim 4: It is possible to compute all of the coefficients of the polynomial $S_2(x)$ except for the coefficients of x^0 and $x^{\langle 2i \rangle_n}$.

Proof: By using the result of Claim 2(c) we deduce that

$$\begin{aligned} S_2(x) &= a_0(x) + a_i(x)x^i \\ &\equiv e_{0,0} + e_{i,i}x^{2i} + \widetilde{S}_2(x) (\mathrm{mod} x^n - 1). \end{aligned}$$

The polynomial $\widetilde{S}_2(x)$ can be computed due to Claim 1(b). The only coefficients in this polynomial that we can not compute are x^0 and $x^{\langle 2i \rangle_n}$, which are dependent on the edges $e_{0,0}$ and $e_{i,i}$. ∎

After we compute $e_{0,0}$ and $e_{i,i}$, computing $S_2(x)$ requires the same number of XOR operations as we did for $\widetilde{S}_2(x)$ which is $(n-1)(n-5)/2$. We now show how to compute $a_0(x)$ and $a_i(x)$.

Claim 5: Given the values of $e_{0,0}, e_{i,i}$, we can compute the polynomials $a_0(x)$ and $a_i(x)$, i.e., decode the failed nodes v_0, v_i.

Proof: Assume that the values of $e_{0,0}, e_{i,i}$ are known. This implies that we can compute exactly the polynomials $S_1(x)$ as well as $S_2(x)$ and let us denote

$$S_1(x) + S_2(x) \equiv \sum_{k=0}^{n-1} s_k x^k (\mathrm{mod} x^n - 1),$$

that is, the coefficients s_k for $k \in [n]$ are known. By the definition of $S_1(x)$ and $S_2(x)$ we have that

$$\begin{aligned} S_1(x) &= a_0(x) + a_i(x), \\ S_2(x) &\equiv a_0(x) + a_i(x)x^i (\mathrm{mod} x^n - 1). \end{aligned}$$

Adding up these two equations results with

$$S_1(x) + S_2(x) \equiv a_i(x) + a_i(x)x^i (\mathrm{mod} x^n - 1).$$

Thus, we get the following n equations with the n variables $e_{i,k}$ for $k \in [n]$. For all $k \in [n] \setminus \{i, \langle 2i \rangle_n\}$ we get the equation

$$e_{i,k} + e_{i,\langle k-i \rangle_n} = s_k, \tag{8}$$

for $k = i$ we get the equation

$$e_{i,0} = s_i,$$

and lastly for $k = \langle 2i \rangle_n$ we get the equation

$$e_{i,\langle 2i \rangle_n} = s_{\langle 2i \rangle_n}.$$

Since $e_{i,0}$ and s_k for $k \in [n]$ are known and since n is a prime number, by a simple induction using (8) the edges $e_{i,\langle -i \rangle_n}, e_{i,\langle -2i \rangle_n}, \ldots, e_{i,\langle 4i \rangle_n}, e_{i,\langle 3i \rangle_n}$ can be calculated. Hence, at this point all of the coefficients of $a_i(x)$ are known besides $e_{i,i}$, which assures the claim's statement for $a_i(x)$.

Algorithm 1 Decoding of $e_{0,\langle 2i \rangle_n}$

```
Decode e_{0,i} using the D_i constraint
ℓ = 3
sum = e_{0,i}
while ℓ < n − 1 do
    Compute d_ℓ = e_{0,⟨ℓ·i⟩_n} + e_{i,⟨ℓ·i⟩_n}
    Compute f_ℓ = e_{i,⟨ℓ·i⟩_n} + e_{0,⟨(ℓ+1)·i⟩_n}
    sum = sum + d_ℓ + f_ℓ
    ℓ = ℓ + 2
e_{0,⟨2i⟩_n} = sum
```

Lastly, the polynomial $a_0(x)$ can be decoded by the value of $e_{0,0}$ and the equality $a_0(x) = S_1(x) + a_i(x)$. ∎

An important observation is that calculating $S_1(x) + S_2(x)$ requires n XOR operations. After that, calculating $a_i(x)$ requires $n-3$ XORs due to (8). The polynomial $a_0(x)$ requires n more XORs by $a_0(x) = S_1(x) + a_i(x)$. We will show that $e_{0,0}$ and $e_{i,i}$ requires $5(n-3)$ XORs and so, computing $a_0(x)$ and $a_i(x)$ requires

$$\begin{aligned} &(n-2)(n-4) + (n-3)/2 + (n-1)(n-5)/2 + 5(n-3) \\ &+ 2n + n - 3 = \frac{3}{2}n^2 - \frac{1}{2}n - 9 \end{aligned}$$

XOR operations.

Now all that is left to show is the decoding of $e_{0,0}, e_{i,i}$. This will be done in two steps; first we will decode the values of $e_{i,n-i}, e_{0,\langle 2i \rangle_n}$ and then we will derive the values of $e_{0,0}, e_{i,i}$. The former edges will be decoded as described in Algorithm 1.

Using a similar algorithm we decode the value $e_{i,n-i}$ as well. To prove the correctness of Algorithm 1, it suffices that we prove the following claim.

Claim 6: All steps in Algorithm 1 are possible to compute and furthermore, $\mathrm{sum} = e_{0,\langle 2i \rangle_n}$.

Proof: First we compute the edge $e_{0,i}$ due to Claim 1(c). Next, the values ℓ receives in the while loop of the algorithm are $3, 5, \ldots, n-2$ and for every value of ℓ it is possible to compute d_ℓ by the neighborhood constraint of $S_{\langle \ell \cdot i \rangle_n}$. Similarly, the value of f_ℓ is computed by the diagonal constraint $D_{\langle (\ell+1) \cdot i \rangle_n}$.

From the while loop of Algorithm 1, we have that

$$\begin{aligned} \mathrm{sum} &= e_{0,i} + \sum_{k=1}^{\frac{n-3}{2}} (d_{2k+1} + f_{2k+1}) \\ &= e_{0,i} + \sum_{k=1}^{\frac{n-3}{2}} (e_{0,\langle (2k+1) \cdot i \rangle_n} + e_{0,\langle (2k+2) \cdot i \rangle_n}) \\ &= \sum_{\ell=1, \ell \neq 2}^{n-1} e_{0,\langle \ell \cdot i \rangle_n} \overset{(a)}{=} \sum_{\ell=1, \ell \neq \langle 2i \rangle_n}^{n-1} e_{0,\ell} \overset{(b)}{=} e_{0,\langle 2i \rangle_n}. \end{aligned}$$

Step (a) holds since i is a generator of the group $\mathbb{Z}_n$, and thus $\{\langle 3i \rangle_n, \langle 4i \rangle_n, \ldots, \langle (n-1) \cdot i \rangle_n\}$ are all distinct elements in $\mathbb{Z}_n$, and since we also added the term $e_{0,i}$ to this summation. Lastly, Step (b) holds by the neighborhood constraint of S_0 and we get that $\mathrm{sum} = e_{0,\langle 2i \rangle_n}$. ∎

Algorithm 2

1: Compute $S_1(x), S_2(x)$
2: Compute $S_1(x) + S_2(x)$
3: Solve the linear system of equations induced from the equality

$$S_1(x) + S_2(x) \equiv a_i(x) + a_i(x)x^i (\mathrm{mod}(x^n - 1))$$

in order to decode $a_i(x)$
4: Use the equality $a_0(x) = S_1(x) + a_i(x)$ in order to decode $a_0(x)$

Given the value of $e_{i,0}$, computing the edge $e_{0,\langle 2i \rangle_n}$ in Algorithm 1 requires $2(n-3)$ XORs. Next, the edge $e_{i,i}$ is calculated using the diagonal constraint $D_{\langle 2i \rangle_n}$, which requires $(n-3)/2$ XORs. The edge $e_{0,0}$ us calculated in the same manner and requires $5(n-3)/2$ XOR operations.

To summarize, given the values of $e_{i,i}, e_{0,0}$, an efficient decoding procedure with time complexity $\Theta(n^2)$ is described in Algorithm 2.

Finally, using the properties above we conclude that it is possible to decode the polynomials $a_0(x)$ and $a_i(x)$ using $\frac{3}{2}n^2 - \frac{1}{2}n - 9$ XOR operations and the following theorem is established. Note that the decoding complexity is optimal since the input size is $\Theta(n^2)$.

Theorem 2: The decoding Algorithm 2 to the code $\mathcal{C}_2$, efficiently corrects any two node failures. Its complexity is $\Theta(n^2)$, where n is the number of nodes.

IV. Single Node Regeneration

In this section, we follow the recent works on regenerating codes [7], [14], [18] in order to analyze a sufficient number of edges we have to read in order to correct a single node erasure while using the code $\mathcal{C}_2$. Our goal is to show that in order to correct a single node erasure, we are not required to read the rest of the graph in its entirety. Namely, while the number of edges in the graph is $\frac{n(n+1)}{2}$, we show that it is enough to read only $\frac{5}{12}n^2 + \mathcal{O}(n)$ edges in order to decode a single node failure. The main result of this section is summarized in the following theorem.

Theorem 3: For any graph $G \in \mathcal{C}_2$ with a single node failure, it suffices to read $\frac{5}{12}n^2 + \mathcal{O}(n)$ edges in order to correct the node failure.

From the symmetry of the code, the algorithm can assume that v_0 is the failed node and it is decoded as follows. The x edges $e_{0,n-1}, \ldots, e_{0,n-x}$ will be corrected using the neighborhood constraints $S_i, i \in [n]$. The rest of the edges will be corrected using the diagonal constraints $D_i, i \in [n]$. Throughout the section, to simplify the calculations, we assume that when we read the sets S_i for $i \in [n]$, we also read the self loop edge $e_{i,i}$, that is, we read the neighborhood edge set $N_i = S_i \cup \{e_{i,i}\}$. We also assume that the edges of the node v_0 are read at the decoding algorithm as well. Note that these two assumptions can only weaken the result on the number of edges that are read in order to decode the node v_0 and they do not affect the statement in Theorem 3. Let $R(x)$ be the set of edges that are read in order to correct the x edges

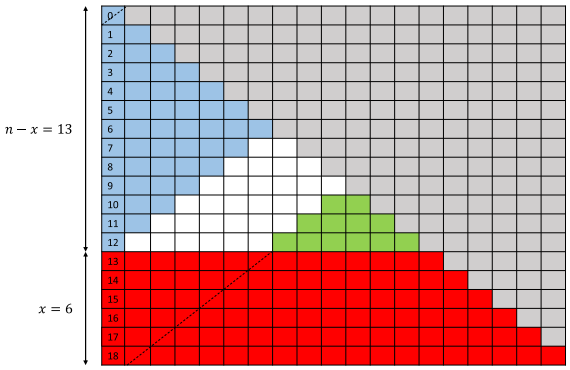


Fig. 2. The red part is the edges which are read in order to decode the edges $e_{0,13}, e_{0,14}, \ldots, e_{0,18}$ by the neighborhood constraints $S_i, i \in [19] \setminus [13]$. The green and blue parts are additional edges which are read in order to decode the edges $e_{0,0}, e_{0,1}, \ldots, e_{0,12}$ by the diagonal constraints $D_i, i \in [13]$.

$e_{0,n-1}, \ldots, e_{0,n-x}$, that is, $R(x) = \bigcup_{1 \leqslant i \leqslant x} N_{n-i}$. We begin with the following claim.

Claim 7: Let $k \in [n]$. For all $A \subseteq [n]$, such that $|A| = k$, it holds that $|\bigcup_{t \in A} N_t| = nk - \binom{k}{2}$.

Proof: Clearly, $\sum_{t \in A} |N_t| = nk$, and for all distinct $t_1, t_2 \in [n]$ it holds that $|N_{t_1} \cap N_{t_2}| = 1$. Note that if $k \geqslant 3$, for all $B \subseteq [n]$, such that $3 \leqslant |B| \leqslant k$, it holds that

$$|\bigcap_{t \in B} N_t| = 0.$$

Thus, we deduce that

$$|\bigcup_{t \in A} N_t| = \sum_{t \in A} |N_t| - \sum_{t_1, t_2 \in A, t_1 \neq t_2} |N_{t_1} \cap N_{t_2}| = nk - \binom{k}{2}.$$

■

By Claim 7 we immediately deduce that $|R(x)| = nx - \binom{x}{2}$.

Example 2: Fig. 2 demonstrates the edges that are read for the case of $n = 19, x = 6$ when the failed node is v_0.

Next, we prove the following claim where we assume that $x \leqslant \lfloor n/2 \rfloor$. For convenience, for all $i \in [n-x]$, we denote the set $A_i(x) = \{e_{n-z,i+z} | 1 \leqslant z \leqslant x\}$. Remember that $D_i = \{e_{k,\ell} | k, \ell \in [n], \langle k + \ell \rangle_n = i\}$.

Claim 8: The following properties hold:

a) For all $i \in [n-x]$, $D_i \cap R(x) = A_i(x)$.
b) For all $i \in [n-2x]$ and $1 \leqslant z \leqslant x$, $n - z > i + z$.
c) For all $i \in [n-2x]$, $|A_i(x)| = x$.
d) For all $i \in [n-x] \setminus [n-2x]$ and $1 \leqslant z < \lfloor \frac{n-i}{2} \rfloor$, $n - z > i + z$.
e) For all $i \in [n-x] \setminus [n-2x]$, $|A_i(x)| = \lfloor \frac{n-i}{2} \rfloor$.

Proof:

a) For all $1 \leqslant z \leqslant x$ it holds that $e_{n-z,i+z} \in N_{n-z}$ and thus $A_i(x) \subseteq R(x)$. Clearly, $e_{n-z,i+z} \in D_i$. Hence, $A_i(x) \subseteq D_i \cap R(x)$ and it is possible to verify the other direction, so $A_i(x) = D_i \cap R(x)$.
b) It holds that

$$n - z \overset{(a)}{\geqslant} n - x \overset{(b)}{>} i + x \overset{(c)}{\geqslant} i + z,$$

where (a) and (c) hold since $1 \leqslant z \leqslant x$ and (b) holds since $i \in [n-2x]$.

c) By the definition of $A_i(x)$, for all $i \in [n-2x]$, $|A_i(x)| \leqslant x$. By (b), for all $i \in [n-2x]$ and $1 \leqslant z \leqslant x$, $n-z > i+z$. Therefore, every value of z between 1 and x generates a unique edge $e_{n-z,i+z}$. Thus, by the definition of $A_i(x)$, we deduce that $|A_i(x)| = x$.

d) It holds that

$$n - z \overset{(a)}{>} n - \left\lfloor \frac{n-i}{2} \right\rfloor \overset{(b)}{\geqslant} i + \left\lfloor \frac{n-i}{2} \right\rfloor \overset{(c)}{>} i + z,$$

where (a) and (c) hold since $1 \leqslant z < \lfloor \frac{n-i}{2} \rfloor$ and (b) holds since $n - 2\lfloor \frac{n-i}{2} \rfloor \geqslant i$.

e) By (d), for all $i \in [n-x] \setminus [n-2x]$ and $1 \leqslant z < \lfloor \frac{n-i}{2} \rfloor$, $n-z > i+z$. Therefore, every value of z between 1 and $\lfloor \frac{n-i}{2} \rfloor - 1$ generates a unique edge $e_{n-z,i+z}$. Moreover, for $z = \lfloor \frac{n-i}{2} \rfloor$, $n - z \geqslant i + z$. Thus, $|A_i(x)| \geqslant \lfloor \frac{n-i}{2} \rfloor$. Next, for all $\lfloor \frac{n-i}{2} \rfloor + 1 \leqslant z' \leqslant x$, the positive integer $z = n - i - z'$ satisfies $1 \leqslant z \leqslant \lfloor \frac{n-i}{2} \rfloor$. Hence, we deduce that the edge $e_{i+z',n-z'} = e_{i+z,n-z}$ already appears in $A_i(x)$, and since we already counted all the edges $e_{i+z,n-z} \in A_i(x)$, we conclude that $|A_i(x)| = \lfloor \frac{n-i}{2} \rfloor$.

■

Claim 9: It holds that

$$\sum_{i=n-2x}^{n-x-1} |A_i(x)| = x^2 - \sum_{i=0}^{x-1} \left\lfloor \frac{i}{2} \right\rfloor.$$

Proof: Note that,

$$\begin{aligned}\sum_{i=n-2x}^{n-x-1} |A_i(x)| &= \sum_{i=n-2x}^{n-x-1} \left\lfloor \frac{n-i}{2} \right\rfloor \\ &= \sum_{i=0}^{x-1} \left\lfloor \frac{n-i-n+2x}{2} \right\rfloor = \sum_{i=0}^{x-1} \left(x - \left\lfloor \frac{i}{2} \right\rfloor \right) \\ &= x^2 - \sum_{i=0}^{x-1} \left\lfloor \frac{i}{2} \right\rfloor.\end{aligned}$$

■

Denote by $F(x)$ the number of edges that we have to read in order to reconstruct all the edges in N_0. Now we are ready to present the proof of Theorem 3.

Proof of Theorem 3: Note that the value $|R(x)|$ corresponds to the number of edges that are read to decode the x edges $e_{0,n-x}, \ldots, e_{0,n-1}$, and $\sum_{i=0}^{n-x-1} \Big(|D_i| - |A_i(x)| \Big)$ is the remaining number of edges in order to decode the first $n-x$ edges $e_{0,0}, \ldots, e_{0,n-x-1}$. Thus,

$$\begin{aligned}F(x) &= |R(x)| + \sum_{i=0}^{n-x-1} \Big(|D_i| - |A_i(x)| \Big) \\ &= |R(x)| + \sum_{i=0}^{n-x-1} |D_i| - \sum_{i=0}^{n-2x-1} |A_i(x)| - \sum_{i=n-2x}^{n-x-1} |A_i(x)| \\ &\overset{(a)}{=} nx - \binom{x}{2} + (n-x)\frac{n+1}{2} - (n-2x)x \\ &\quad - \Big(x^2 - \sum_{i=0}^{x-1} \left\lfloor \frac{i}{2} \right\rfloor \Big).\end{aligned}$$

In the last equation, Step (a) holds since by Claim 7, $|R(x)| = nx - \binom{x}{2}$, using the fact that for all $i \in [n]$, $|D_i| = \frac{n+1}{2}$, also by Claim 8(c), for all $i \in [n-2x]$, $|A_i(x)| = x$, and also using Claim 9 in which

$$\sum_{i=n-2x}^{n-x-1} |A_i(x)| = x^2 - \sum_{i=0}^{x-1} \left\lfloor \frac{i}{2} \right\rfloor.$$

Lastly, it is possible to check that applying $x = \lceil \frac{n}{3} \rceil$ provides that $F(\lceil \frac{n}{3} \rceil) \leqslant \frac{5}{12}n^2 + \frac{n}{2} = \frac{5}{12}n^2 + \mathcal{O}(n)$ as required. Thus it is sufficient to read only $\frac{5}{12}n^2 + \mathcal{O}(n)$ edges in this decoding algorithm for v_0. This concludes the proof of Theorem 3.

■

V. Binary Triple-Node-Erasure-Correcting Codes

In this section we present a construction of binary triple-node-erasure-correcting codes for undirected graphs. Let $n \geqslant 5$ be a prime number such that 2 is a primitive number in $\mathbb{Z}_n$. Let $G = (V_n, L)$ be a graph with n vertices. We will use in this construction the edge sets S_h, D_m for $h \in [n], m \in [n]$ which were defined in (3),(4), respectively. In addition, for $s \in [n]$ we define the edge set

$$T_s = \{ \langle v_k, v_\ell \rangle | k, \ell \in [n], \langle k + 2\ell \rangle_n = s, k \neq \ell \}.$$

In this construction we impose the same constraints from Construction 1, that is, the sets S_h will be used to represent parity constraints on the neighborhood of each node, the sets D_m will represent parity constraints on the diagonals with slope one of A_G, and furthermore the sets T_s will represent parity constraints on the diagonals with slope two of A_G.

Example 3: In Fig. 3 we present the sets T_s, $s \in [11]$ of a graph $G = (V_{11}, L)$ on its labeling matrix A_G, and its lower-triangle-labeling matrix A'_G.

We are now ready to show the following construction.

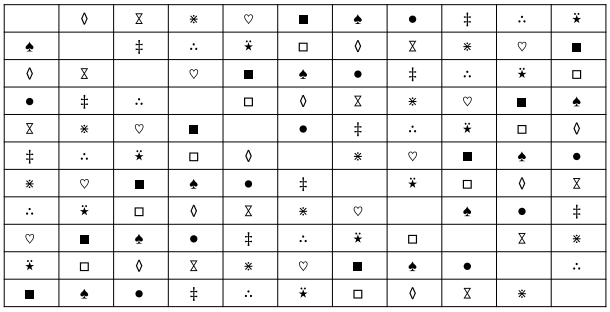
Construction 2: For all prime number $n \geqslant 5$ where 2 is primitive in $\mathbb{Z}_n$, let $\mathcal{C}_3$ be the following code:

$$\mathcal{C}_3 = \left\{ G = (V_n, L) \left| \begin{array}{ll} (a) & \sum_{\langle v_i, v_j \rangle \in S_h} e_{i,j} = 0, h \in [n] \\ (b) & \sum_{\langle v_i, v_j \rangle \in D_m} e_{i,j} = 0, m \in [n] \\ (c) & \sum_{\langle v_i, v_j \rangle \in T_s} e_{i,j} = 0, s \in [n] \end{array} \right. \right\}.$$

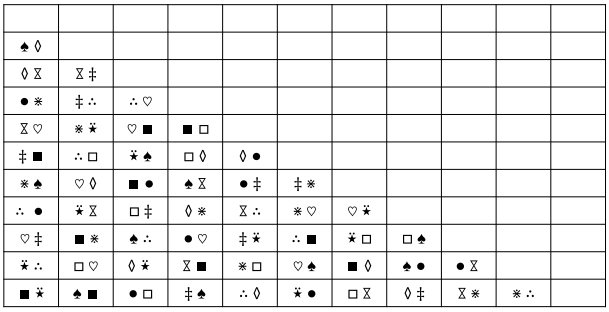
Note that the code $\mathcal{C}_3$ is a subcode of the code $\mathcal{C}_2$ and for *any* graph G over the binary field, by (5) there are only $n-1$ independent constraints (a) in Construction 2, and by the same principle,

$$\sum_{s \in [n]} \sum_{\langle v_i, v_j \rangle \in T_s} e_{i,j} = \sum_{s=0}^{n-1} \sum_{\substack{\ell=0 \\ \ell \neq \langle 3^{-1} s \rangle_n}}^{n-1} e_{\langle s-2\ell \rangle_n, \ell} = 2 \sum_{h=0}^{n-1} \sum_{\ell=0}^{h-1} e_{h,\ell} = 0. \tag{9}$$

Therefore the code $\mathcal{C}_3$ has at most $3n-2$ linearly independent constraints which implies that its redundancy is not greater than $3n-2$. Since we will prove in Theorem 4 that $\mathcal{C}_3$ is a triple-node-correcting codes, according to the Singleton bound we get that the code redundancy is at most a single bit away from optimality. Our main result in this section is showing the following theorem.



(a) Slope-Two-Diagonal-Parity Constraints on A_G



(b) Slope-Two-Diagonal-Parity Constraints on A'_G

Fig. 3. The slope-two-diagonal constraints over undirected graphs, represented on the labeling matrix and the lower-triangle-labeling matrix.

Theorem 4: For all prime number $n \geqslant 5$ such that 2 is primitive in $\mathbb{Z}_n$, the code $\mathcal{C}_3$ is a triple-node-erasure-correcting code. It is at most a single bit away from optimality.

Proof: Assume on the contrary that there is a graph $G = (V_n, L) \in \mathcal{C}_3$ where $w(G) \leqslant 3$. We prove here only the case that $w(G) = 3$ since the case of $w(G) \leqslant 2$ holds according to Theorem 2. By the symmetry of Construction 2, it is sufficient to assume that a vertex cover W of G is $W = \{v_0, v_i, v_j\}$ for distinct $i, j \in [n] \setminus \{0\}$, while all other cases hold by relabeling the indices $0, i, j$. We will show that $G = G_{\mathbf{0}}$.

Denote by $H_{i,j}$ the set

$$H_{i,j} = \{i, j, \langle 2i\rangle_n, \langle 2j\rangle_n, \langle 2i+j\rangle_n, \langle 2j+i\rangle_n\}, \tag{10}$$

and for all $s \in [n]$, denote by $h_{i,j}(s)$ the sum

$$\begin{aligned} h_{i,j}(s) = e_{0,s} + e_{i,\langle s-2i\rangle_n} + e_{j,\langle s-2j\rangle_n} \\ + e_{0,\langle 2^{-1}s\rangle_n} + e_{i,\langle 2^{-1}(s-i)\rangle_n} + e_{j,\langle 2^{-1}(s-j)\rangle_n}. \end{aligned} \tag{11}$$

The next claim presents several useful properties.

Claim 10: The following properties hold on the graph G:

a) For all $h \in [n] \setminus \{0, i, j\}$, $e_{0,h} + e_{i,h} + e_{j,h} = 0$.
b) For all $m \in [n] \setminus \{i, j, \langle i+j\rangle_n\}$, $e_{0,m} + e_{i,\langle m-i\rangle_n} + e_{j,\langle m-j\rangle_n} = 0$.
c) $e_{0,i} + e_{j,\langle i-j\rangle_n} = e_{0,j} + e_{i,\langle j-i\rangle_n} = e_{j,i} + e_{0,\langle i+j\rangle_n} = 0$.
d) For all $s \in [n] \setminus H_{i,j}$, it holds that $h_{i,j}(s) = 0$.
e) It holds that

$$\begin{aligned} &\sum_{s\in H_{i,j}} h_{i,j}(s)x^s \\ &\equiv e_{i,0}(x^i + x^{2i}) + e_{j,0}(x^j + x^{2j}) + e_{j,i}(x^{2i+j} + x^{i+2j}) \\ &\qquad (\mathrm{mod} x^n - 1). \end{aligned}$$

Proof: We remind that for all $k, \ell \in [n] \setminus \{0, i, j\}$, $e_{k,\ell} = 0$.

a) We know that for all $h \in [n] \setminus \{0, i, j\}, s \in [n] \setminus \{h\}$, $\langle v_s, v_h\rangle \in S_h$, and therefore by the definition of the constraint (a) in Construction 2 we get that

$$0 = \sum_{\langle v_s, v_h\rangle \in S_h} e_{s,h} = \sum_{s=0, s\neq h}^{n-1} e_{s,h} = e_{0,h} + e_{i,h} + e_{j,h}.$$

b) For all $m \in [n] \setminus \{i, j, i+j\}$, denote by D'_m the set

$$D'_m = D_m \setminus \left\{\langle v_0, v_m\rangle, \langle v_i, v_{\langle m-i\rangle_n}\rangle, \langle v_j, v_{\langle m-j\rangle_n}\rangle\right\}.$$

Therefore, we have that

$$\begin{aligned} 0 = \sum_{\langle v_j, v_{\langle m-j\rangle_n}\rangle \in D_m} e_{j,\langle m-j\rangle_n} = \\ \sum_{\langle v_j, v_{\langle m-j\rangle_n}\rangle \in D'_m} e_{j,\langle m-j\rangle_n} + e_{0,m} + e_{i,\langle m-i\rangle_n} + e_{j,\langle m-j\rangle_n}, \end{aligned}$$

and since $e_{s,k} = 0$ for all $\langle v_s, v_k\rangle \in D'_m$, we get that $e_{0,m} + e_{i,\langle m-i\rangle_n} + e_{i,\langle m-i\rangle_n} = 0$.

c) Similarly to (b), for $m = i$ we get that $\langle v_0, v_m\rangle = \langle v_i, v_{\langle m-i\rangle_n}\rangle$ and therefore by the definition of the constraint (b) in Construction 2 we get that $e_{0,i} + e_{j,\langle i-j\rangle_n} = 0$. It can be similarly verified that for $m = j$ we get that $e_{0,j} + e_{i,\langle j-i\rangle_n} = 0$ and for $m = \langle i+j\rangle_n$ we get that $e_{j,i} + e_{0,\langle i+j\rangle_n} = 0$.

d) For all $s \in [n]$, let B_s be the following edge set

$$\begin{aligned} B_s = \{\langle v_0, v_s\rangle, \langle v_i, v_{\langle s-2i\rangle_n}\rangle, \langle v_j, v_{\langle s-2j\rangle_n}\rangle, \\ \langle v_0, v_{\langle 2^{-1}s\rangle_n}\rangle, \langle v_i, v_{\langle 2^{-1}(s-i)\rangle_n}\rangle, \langle v_j, v_{\langle 2^{-1}(s-j)\rangle_n}\rangle\}. \end{aligned} \tag{12}$$

It can be readily verified that for $s \notin \{0, \langle 3i\rangle_n, \langle 3j\rangle_n\} \cup H_{i,j}$, $|B_s| = 6$. For all $k \in \{0, i, j\}$ and for all $s \in [n] \setminus \{0, \langle 3i\rangle_n, \langle 3j\rangle_n\}$ it holds that $k \neq \langle s-2k\rangle_n$ and therefore, if $\langle v_k, v_{\langle s-2k\rangle_n}\rangle \in B_s$ then $\langle v_k, v_{\langle s-2k\rangle_n}\rangle \in T_s$, i.e., $B_s \subseteq T_s$. Therefore, by the definition of the diagonal constraint (c) in Construction 2 we deduce that for all $s \notin \{0, \langle 3i\rangle_n, \langle 3j\rangle_n\} \cup H_{i,j}$,

$$0 = \sum_{\langle v_k, v_m\rangle \in T_s} e_{k,m} = \sum_{\langle v_k, v_m\rangle \in B_s} e_{k,m} = h_{i,j}(s).$$

Moreover, for $s = 0$, $\langle v_0, v_s\rangle = \langle v_0, v_{\langle 2^{-1}s\rangle_n}\rangle = \langle v_0, v_0\rangle$ and therefore $|B_0| = 5$. It can be similarly verified that $|B_{\langle 3i\rangle_n}| = |B_{\langle 3j\rangle_n}| = 5$. Notice that for all $k \in \{0, i, j\}, s \in \{0, \langle 3i\rangle_n, \langle 3j\rangle_n\}$, if $\langle v_k, v_{\langle s-2k\rangle_n}\rangle \in B_s$ then it holds that $\langle v_k, v_{\langle s-2k\rangle_n}\rangle \in T_s \cup \{\langle v_k, v_k\rangle\}$, i.e., $B_s \subseteq T_s \cup \{\langle v_k, v_k\rangle\}$. Therefore again, by the definition of the diagonal constraint (c) in Construction 2 we deduce that for all $s \in \{0, \langle 3i\rangle_n, \langle 3j\rangle_n\}$,

$$\begin{aligned} 0 = \sum_{\langle v_k, v_m\rangle \in T_s \cup \{\langle v_{\langle 3^{-1}s\rangle_n}, v_{\langle 3^{-1}s\rangle_n}\rangle\}} e_{k,m} + e_{\langle 3^{-1}s\rangle_n, \langle 3^{-1}s\rangle_n} \\ = \sum_{\langle v_k, v_m\rangle \in B_s} e_{k,m} + e_{\langle 3^{-1}s\rangle_n, \langle 3^{-1}s\rangle_n} = h_{i,j}(s). \end{aligned}$$

e) For all $s \in H_{i,j}$ let B_s be the edge set from (12). Notice that for $s = i$ we get that $\langle v_0, v_s\rangle = \langle v_i, v_{\langle 2^{-1}(s-i)\rangle_n}\rangle$,

for $s=\langle 2i\rangle_n$ we get that $\langle v_i, v_{\langle s-2i\rangle_n}\rangle = \langle v_0, v_{\langle 2^{-1}s\rangle_n}\rangle$, and for $s=\langle 2i+j\rangle_n$ we get that $\langle v_i, v_{\langle s-2i\rangle_n}\rangle = \langle v_j, v_{\langle 2^{-1}(s-j)\rangle_n}\rangle$, and therefore for all $s\in H_{i,j}$, $|B_s|=5$. Similarly to the proof of (d), the edge set B_s consists of all the edges incident to at least one of the nodes v_0, v_i and v_j in T_s, i.e., $B_s\subseteq T_s$. Therefore we deduce that for $s\in\{i,j\}$,

$$e_{s,0}=\sum_{\langle v_k,v_m\rangle\in T_s} e_{k,m}+e_{s,0}=\sum_{\langle v_k,v_m\rangle\in B_s} e_{k,m}+e_{s,0}=h_{i,j}(s),$$

and the coefficient of the monomial x^i, x^j in the polynomial $\sum_{s\in H_{i,j}} h_{i,j}(s)x^s$ is $e_{i,0}, e_{j,0}$, respectively. The proof that the coefficient of $x^{2i}, x^{2j}, x^{2i+j}, x^{2j+i}$ in this polynomial is $e_{i,0}, e_{j,0}, e_{j,i}, e_{j,i}$ is similar, respectively. ∎

From this claim we deduce the following equations.

$$\sum_{h=1,h\notin\{i,j\}}^{n-1}\Big(e_{0,h}+e_{i,h}+e_{j,h}\Big)x^h=0, \tag{13}$$

$$\sum_{m=0,m\notin\{i,j,\langle i+j\rangle_n\}}^{n-1}\Big(e_{0,m}+e_{i,\langle m-i\rangle_n}+e_{j,\langle m-j\rangle_n}\Big)x^m=0, \tag{14}$$

$$\sum_{s=0,s\notin H_{i,j}}^{n-1} h_{i,j}(s)x^s=0. \tag{15}$$

Next, let $a_0(x), a_i(x)$ and $a_j(x)$ be the neighborhood polynomials without self-loops of G. The following lemma presents a few equalities that will be used to decode the values of $a_0(x), a_i(x)$ and $a_j(x)$.

Lemma 5: The following properties hold:

a) $a_0(x)+a_i(x)+a_j(x)$
$=e_{i,0}(1+x^i)+e_{j,0}(1+x^j)+e_{j,i}(x^i+x^j)$.

b) $a_0(x)+a_i(x)x^i+a_j(x)x^j$
$\equiv e_{0,0}+e_{i,i}x^{2i}+e_{j,j}x^{2j}+e_{i,0}x^i+e_{j,0}x^j+e_{j,i}x^{i+j}$
$(\bmod x^n-1)$.

c) $a_0(x)+a_i(x)x^{2i}+a_j(x)x^{2j}+a_0^2(x)+a_i^2(x)x^i+a_j^2(x)x^j$
$\equiv e_{i,0}(x^i+x^{2i})+e_{j,0}(x^j+x^{2j})+e_{j,i}(x^{2i+j}+x^{i+2j})$
$(\bmod x^n-1)$.

Proof:

a) According to the neighborhood-polynomials definition we can write

$$\begin{aligned}
&a_0(x)+a_i(x)+a_j(x)=\\
&=\sum_{h\in\{0,i,j\}} e_{h,h}x^h+\sum_{h=0}^{n-1}e_{0,h}x^h+\sum_{h=0}^{n-1}e_{i,h}x^h+\sum_{h=0}^{n-1}e_{j,h}x^h\\
&=\sum_{h\in\{0,i,j\}} e_{h,h}x^h+\sum_{h=0}^{n-1}\Big(e_{0,h}+e_{i,h}+e_{j,h}\Big)x^h\\
&=e_{0,0}+e_{i,i}x^i+e_{j,j}x^j+\Big(e_{0,0}+e_{i,0}+e_{j,0}\Big)\\
&+\Big(e_{0,i}+e_{i,i}+e_{j,i}\Big)x^i+\Big(e_{0,j}+e_{i,j}+e_{j,j}\Big)x^j\\
&+\sum_{h=1,h\notin\{i,j\}}^{n-1}\Big(e_{0,h}+e_{i,h}+e_{j,h}\Big)x^h\\
&\overset{(a)}{=}e_{i,0}(1+x^i)+e_{j,0}(1+x^j)+e_{j,i}(x^i+x^j),
\end{aligned}$$

where Step (a) holds due to (13).

b)

$$\begin{aligned}
&a_0(x)+a_i(x)x^i+a_j(x)x^j=\\
&=\sum_{m\in\{0,i,j\}} e_{m,m}x^{2m}+\sum_{m=0}^{n-1}e_{0,m}x^m+\sum_{m=0}^{n-1}e_{i,m}x^{m+i}\\
&+\sum_{m=0}^{n-1}e_{j,m}x^{m+j}\equiv\sum_{m\in\{0,i,j\}} e_{m,m}x^{2m}+\sum_{m=0}^{n-1}e_{0,m}x^m\\
&+\sum_{m=0}^{n-1}e_{i,\langle m-i\rangle_n}x^m+\sum_{m=0}^{n-1}e_{j,\langle m-j\rangle_n}x^m(\bmod x^n-1)\\
&=\sum_{m\in\{0,i,j\}} e_{m,m}x^{2m}+\sum_{m=0}^{n-1}\Big(e_{0,m}+e_{i,\langle m-i\rangle_n}+e_{j,\langle m-j\rangle_n}\Big)x^m\\
&=e_{0,0}+e_{i,i}x^{2i}+e_{j,j}x^{2j}\\
&+\Big(e_{0,i}+e_{i,0}+e_{j,\langle i-j\rangle_n}\Big)x^i+\Big(e_{0,j}+e_{i,\langle j-i\rangle_n}+e_{j,0}\Big)x^j\\
&+\Big(e_{0,\langle i+j\rangle_n}+e_{i,j}+e_{j,i}\Big)x^{i+j}\\
&+\sum_{m=0,m\notin\{i,j,\langle i+j\rangle_n\}}^{n-1}\Big(e_{0,m}+e_{i,\langle m-i\rangle_n}+e_{j,\langle m-j\rangle_n}\Big)x^m\\
&\overset{(a)}{=}e_{0,0}+e_{i,i}x^{2i}+e_{j,j}x^{2j}+e_{i,0}x^i+e_{j,0}x^j+e_{j,i}x^{i+j},
\end{aligned}$$

where Step (a) holds due to (14).

c)

$$\begin{aligned}
&a_0(x)+a_i(x)x^{2i}+a_j(x)x^{2j}+a_0^2(x)+a_i^2(x)x^i+a_j^2(x)x^j\\
&=\sum_{s\in\{0,i,j\}} e_{s,s}x^{3s}+\sum_{s=0}^{n-1}e_{0,s}x^s+\sum_{s=0}^{n-1}e_{i,s}x^{s+2i}+\sum_{s=0}^{n-1}e_{j,s}x^{s+2j}\\
&+\sum_{s\in\{0,i,j\}} e_{s,s}x^{3s}+\sum_{s=0}^{n-1}e_{0,s}x^{2s}+\sum_{s=0}^{n-1}e_{i,s}x^{2s+i}+\sum_{s=0}^{n-1}e_{j,s}x^{2s+j}\\
&\equiv\sum_{s=0}^{n-1}e_{0,s}x^s+\sum_{s=0}^{n-1}e_{i,\langle s-2i\rangle_n}x^s+\sum_{s=0}^{n-1}e_{j,\langle s-2j\rangle_n}x^s\\
&\qquad\qquad(\bmod x^n-1)\\
&+\sum_{s=0}^{n-1}e_{0,\langle 2^{-1}s\rangle_n}x^s+\sum_{s=0}^{n-1}e_{i,\langle 2^{-1}(s-i)\rangle_n}x^s+\sum_{s=0}^{n-1}e_{j,\langle 2^{-1}(s-j)\rangle_n}x^s\\
&=\sum_{s=0}^{n-1}h_{i,j}(s)x^s=\sum_{s\in H_{i,j}}h_{i,j}(s)x^s+\sum_{s=0,s\notin H_{i,j}}^{n-1}h_{i,j}(s)x^s\\
&\overset{(a)}{=}e_{i,0}(x^i+x^{2i})+e_{j,0}(x^j+x^{2j})+e_{j,i}(x^{2i+j}+x^{i+2j}),
\end{aligned}$$

where Step (a) holds due to (15).

∎

Notice that by setting $x=1$ in the equation of Lemma 5(b) we get that

$$e_{0,0}+e_{i,i}+e_{j,j}+e_{i,0}+e_{j,0}+e_{j,i}=0. \tag{16}$$

Using the result of Lemma 5 we get the next three equalities. The proof of this lemma is given in Appendix A

Lemma 6: The following equations hold

a) $a_j(x)(1+x^i)+a_j^2(x)\equiv e_{j,j}(1+x^j)(x^i+x^j)$
$(\bmod x^n-1)$.

b) $a_i(x)(1+x^j)+a_i^2(x)\equiv e_{i,i}(1+x^i)(x^i+x^j)$
$(\bmod x^n-1)$.

c) $a_0(x)(x^i+x^j)+a_0^2(x)\equiv e_{0,0}(1+x^i)(1+x^j)$
$(\bmod x^n-1)$.

Our next step is showing that the value of at least one of the self-loops $e_{j,j}, e_{i,i}$ or $e_{0,0}$ is zero. For this goal, we show another important claim where its proof is given in Appendix B.

Lemma 7: It holds that $e_{0,0} + e_{i,i} + e_{j,j} = e_{j,0} + e_{j,0} + e_{j,i} = 0$.

By Lemma 7, we know that at least one of the self-loops $e_{j,j}, e_{i,i}$ or $e_{0,0}$ is zero, and our next step is showing that one of the polynomials $a_0(x), a_i(x)$ or $a_j(x)$ is zero. We assume that $e_{j,j}$ is zero, while the proof of the other two cases will be similar based upon Lemma 6(b) and 6(c). By Lemma 6(a), we get that

$$a_j(x)[1 + x^i + a_j(x)] \equiv 0 (\mathrm{mod} x^n - 1).$$

Denote by $p(x)$ the polynomial $p(x) = 1 + x^i + a_j(x)$ which is clearly in $\mathcal{R}_n$. Since $M_n(x)$ is irreducible, either $M_n(x)|a_j(x)$ or $M_n(x)|p(x)$. Since $1 + x|a_j(x)$ and $1 + x|p(x)$ it is possible to derive that either $a_j(x) = 0$ or $p(x) = 0$. We will show that $p(x) \neq 0$ which will lead to the fact that $a_j(x) = 0$. Assume on a contrary that $p(x) = 0$. Therefore we deduce that $a_j(x) = 1 + x^i$ and thus $e_{j,i} = e_{j,0} = 1$. Notice that in this case, by Lemma 7 we have that $e_{i,0} = 0$. By Lemma 5(a) we deduce that

$$\begin{aligned} &a_0(x) + a_i(x) + 1 + x^i \\ &= a_0(x) + a_i(x) + a_j(x) \\ &= e_{i,0}(1 + x^i) + e_{j,0}(1 + x^j) + e_{j,i}(x^i + x^j) \\ &= (1 + x^j) + (x^i + x^j) = 1 + x^i, \end{aligned}$$

and therefore $a_0(x) + a_i(x) = 0$. Again, by Lemma 7 we know that $e_{0,0} + e_{i,i} + e_{j,j} = 0$ and therefore, since $e_{j,j} = 0$, we get that $e_{i,i} = e_{0,0}$. By Lemma 5(b) we deduce that

$$\begin{aligned} &a_0(x) + a_i(x)x^i + (1 + x^i)x^j \\ &= a_0(x) + a_i(x)x^i + a_j(x)x^j \\ &\equiv e_{0,0} + e_{i,i}x^{2i} + e_{j,j}x^{2j} + e_{i,0}x^i + e_{j,0}x^j \\ &\quad + e_{j,i}x^{i+j} (\mathrm{mod} x^n - 1) \\ &\equiv e_{0,0} + e_{i,i}x^{2i} + x^j + x^{i+j} (\mathrm{mod} x^n - 1) \\ &\equiv e_{0,0} + e_{i,i}x^{2i} + (1 + x^i)x^j (\mathrm{mod} x^n - 1), \end{aligned}$$

and therefore $a_0(x) + a_i(x)x^i \equiv e_{0,0} + e_{i,i}x^{2i} (\mathrm{mod} x^n - 1)$. Next, we show an important claim.

Claim 11: If

$$\begin{aligned} &a_0(x) + a_i(x) = 0, \\ &a_0(x) + a_i(x)x^i \equiv e_{0,0} + e_{i,i}x^{2i} (\mathrm{mod} x^n - 1), \end{aligned}$$

then $a_0(x) = a_i(x) = 0$.

Proof: The summation of these equations results with

$$a_i(x)(1 + x^i) \equiv e_{0,0} + e_{i,i}x^{2i} (\mathrm{mod} x^n - 1).$$

It holds that $e_{0,0} = e_{i,i}$ by applying $x = 1$ in the last equation. Assume that $e_{0,0} = e_{i,i} = 1$, so we get that

$$a_i(x)(1 + x^i) \equiv 1 + x^{2i} (\mathrm{mod} x^n - 1).$$

Since $1 + x^{2i} = (1 + x^i)^2$, it holds that

$$(1 + x^i)(1 + x^i + a_i(x)) \equiv 0 (\mathrm{mod} x^n - 1).$$

Denote by $p(x)$ the polynomial $p(x) = 1 + x^i + a_i(x)$, and since $p(1) = 0$, it holds that $1 + x|p(x)$. As stated in (2), it holds that $\gcd(x^i + 1, M_n(x)) = 1$, and since

$$(1 + x^i)p(x) = (x^n - 1)s(x) = M_n(x)(x + 1)s(x)$$

for some polynomial $s(x)$ over $\mathbb{F}_2$, we deduce that $M_n(x)|p(x)$. Therefore we get that $x^n - 1|p(x)$, however $p(x) \in \mathcal{R}_n$, and so we deduce that $p(x) = 0$, that is, $a_i(x) = 1 + x^i$. This results with a contradiction since the coefficient of x^i in $a_i(x)$ is 0. Thus $e_{0,0} = e_{i,i} = 0$ and

$$a_i(x)(1 + x^i) \equiv 0 (\mathrm{mod} x^n - 1).$$

Notice that $a_i(x) \in \mathcal{R}_n$ and by Claim 2(a) it also holds $a_i(1) = 0$. Since $\gcd(x^i + 1, M_n(x)) = 1$, we derive that $x^n - 1|a_i(x)$ and since $a_i(x) \in \mathcal{R}_n$, we immediately get that $a_i(x) = 0$. Finally, since $a_0(x) + a_i(x) = 0$, we also get that $a_0(x) = 0$, and that completes the proof. ∎

Using Claim 11 we get a contradiction since $e_{j,i} = e_{j,0} = 1$. Therefore, it holds that $a_j(x) = 0$ and since $\mathcal{C}_3$ is a sub code of $\mathcal{C}_2$, we again get that $a_0(x) = a_i(x) = 0$, and that concludes the proof. ∎

Next, it is proved that the redundancy of the code $\mathcal{C}_3$ is exactly $3n - 2$. Every linear code over undirected graphs $\mathcal{U}$-$[n, k]_{\mathbb{F}_q}$ can be represented by a parity-check matrix of dimension $r \times \binom{n+1}{2}$ over $\mathbb{F}_q$, when $r \geqslant \binom{n+1}{2} - k$. Let $\mathbf{H}$ be the parity check matrix of the code $\mathcal{C}_3$ of dimension $3n \times \binom{n+1}{2}$, which is constructed as follows. The first n rows of $\mathbf{H}$, denoted by $\boldsymbol{s}_h, h \in [n]$, are formed by the neighborhood constraints $S_h, h \in [n]$. The next set of n rows, denoted by $\boldsymbol{d}_m, m \in [n]$, are formed by the diagonal constraints $D_m, m \in [n]$. Lastly, the last set of n rows, denoted by $\boldsymbol{t}_s, s \in [n]$, are formed by the constraints $T_s, s \in [n]$. For a vector $\boldsymbol{u} \in \mathbb{F}_2^n$ denote by $\mathbf{w}(\boldsymbol{u})$ its Hamming weight, i.e., the number of non-zero entries in $\boldsymbol{u}$.

Theorem 8: The redundancy of the code $\mathcal{C}_3$ is $3n - 2$.

Proof: By the Singleton bound in (1), the redundancy of $\mathcal{C}_3$ is at least $3n - 3$ and therefore $\dim(\ker \mathbf{H}) \leqslant 3$. Our goal is to show that $\dim(\ker \mathbf{H}) = 2$. Let $\mathbf{0}, \mathbf{1}$ be a length-n vector of zeros, ones, respectively. Denote by $\boldsymbol{u}_0, \boldsymbol{u}_1$ the following vector of length $3n$.

$$\boldsymbol{u}_0 = (\mathbf{1}, \mathbf{0}, \mathbf{0}), \boldsymbol{u}_\mathbf{1} = (\mathbf{0}, \mathbf{0}, \mathbf{1}),$$

respectively. By (5), (9), the sum of the first, last n rows of $\mathbf{H}$ is zero, and thus it holds that the vector $\boldsymbol{u}_0, \boldsymbol{u}_1$ is in $\ker \mathbf{H}$, respectively. Clearly $\boldsymbol{u}_0$ and $\boldsymbol{u}_1$ are linearly independent and therefore $\dim(\ker \mathbf{H}) \geqslant 2$. Furthermore, every self-loop $e_{m,m}, m \in [n]$ appears only in the constraint $D_m, m \in [n]$, and thus only vectors of the form $(\boldsymbol{x}, \mathbf{0}, \boldsymbol{y})$ can be in $\ker \mathbf{H}$, where $\boldsymbol{x}, \boldsymbol{y}$ is a binary vector of length n, respectively.

Now, assume in the contrary that $\dim(\ker \mathbf{H}) = 3$. Therefore, we can deduce that there is another vector $\boldsymbol{u}_2 = (\boldsymbol{x}, \mathbf{0}, \boldsymbol{y}) \in \ker \mathbf{H}$ which is not in $\mathrm{span}\{\boldsymbol{u}_0, \boldsymbol{u}_1\}$. Since the redundancy of the double-node-erasure-correcting code $\mathcal{C}_2$ is $2n - 1$, it holds that every $n - 1$ rows of the first n rows of $\mathbf{H}$ are linearly independent. Hence, $\boldsymbol{y} \notin \{\mathbf{0}, \mathbf{1}\}$ and it can be similarly proved that $\boldsymbol{x} \notin \{\mathbf{0}, \mathbf{1}\}$. Another observation is that we can choose a vector $\boldsymbol{x}$ such that $\mathbf{w}(\boldsymbol{x}) \leqslant (n - 1)/2$,

since otherwise we can choose a vector $\boldsymbol{x}'$ such that $(\boldsymbol{x}', \mathbf{0}, \boldsymbol{y}) = (\boldsymbol{x}, \mathbf{0}, \boldsymbol{y}) + \boldsymbol{u_0}$ which is also in $\ker \mathbf{H}$ and $\mathbf{w}(\boldsymbol{x}') \leqslant (n-1)/2$. The same property holds for the vector $\boldsymbol{y}$ and thus it is possible to choose vectors $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ such that $\mathbf{w}(\boldsymbol{x}_0), \mathbf{w}(\boldsymbol{y}_0) \leqslant (n-1)/2$ and $\boldsymbol{u}_2 = (\boldsymbol{x}_0, \mathbf{0}, \boldsymbol{y}_0) \in \ker \mathbf{H}$. Let $\boldsymbol{x}_1, \boldsymbol{y}_1$ be a vector which results from a single right cyclic shift of the vector $\boldsymbol{x}_0, \boldsymbol{y}_0$. By symmetry of the construction we can relabel the nodes of every graph in the code, and deduce that the vector $\boldsymbol{u}_3 = (\boldsymbol{x}_1, \mathbf{0}, \boldsymbol{y}_1)$ is also in $\ker \mathbf{H}$. Since $\mathbf{w}(\boldsymbol{x}_0) = \mathbf{w}(\boldsymbol{x}_1) \leqslant (n-1)/2$, there are indices i, j such that $(\boldsymbol{x}_0)_i = 1, (\boldsymbol{x}_1)_i = 0$ and $(\boldsymbol{x}_0)_j = 0, (\boldsymbol{x}_1)_j = 1$ and therefore, the vector $\boldsymbol{u}_3$ is not in $\mathrm{span}\{\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2\}$. Thus $\dim(\ker \mathbf{H}) \geqslant 4$ and we get a contradiction. ■

VI. Decoding of the Triple-Node-Erasure-Correcting Codes

In Section V, we proved that the code $\mathcal{C}_3$ can correct the failure of any three nodes in the graph. Note that whenever three nodes fail, the number of unknown variables is $3n-3$, and so a naive decoding solution for the code $\mathcal{C}_3$ is to solve the linear equation system of $3n-2$ constraints with the $3n-3$ variables. In this section we show how to efficiently solve this linear equation system for $\mathcal{C}_3$ with time complexity $\Theta(n^2)$. Clearly, this time complexity is optimal since the complexity of the input size of the graph is $\Theta(n^2)$.

Assume that G is a graph in the code $\mathcal{C}_3$ and that the failed nodes are v_0, v_i, and v_j. Let $\boldsymbol{v}$ be a binary vector of length $3n-2$ denoted by

$$\begin{aligned}\boldsymbol{v} =&(e_{0,0}, e_{0,1}, \ldots, e_{0,n-1}, e_{i,1}, \ldots, e_{i,n-1},\\ &e_{j,1}, \ldots, e_{j,i-1}, e_{j,i+1}, \ldots, e_{j,n-1}).\end{aligned}$$

Using the $3n-2$ constraints in the code $\mathcal{C}_3$, it is possible to form the linear equation system as

$$\mathbf{H} \cdot \boldsymbol{v} = \boldsymbol{s}. \tag{17}$$

Here, $\mathbf{H}$ is a binary $(3n-2)\times(3n-3)$ matrix that its columns are indexed by the edges in $\boldsymbol{v}$ and its rows indexed by the constraints $S_0, S_1, \ldots S_{n-2}, D_0, D_1, \ldots, D_{n-1}, T_0, T_1, \ldots, T_{n-2}$, and $\boldsymbol{s}$ is a syndrome vector of length $3n-2$, indexed by the same order of the $3n-2$ constraints, that is calculated from the surviving edges. In this case, the binary matrix has $\mathbf{H}$ has $\mathcal{O}(n)$ non zero entries (three rows with $n-1$ one's and all other rows with at most 6 one's) and furthermore, it has a unique solution since we proved the minimum distance of the code is four. Hence, according to Wiedemann [19], the vector $\boldsymbol{v}$ can be found in time complexity $\mathcal{O}(n^2)$.

VII. Conclusion

In this paper we continued our research on codes over graphs from [23], [24]. We presented an optimal binary construction for codes correcting a failure of two nodes together with a decoding procedure that its complexity is optimal. We then extended this construction for triple-node-erasure-correcting codes which are at most a single bit away from optimality with respect to the Singleton bound.

Appendix A

Remember that i and j are indices such that $i, j \in [n] \setminus \{0\}$.

Lemma 6: The following equations hold

a) $a_j(x)(1+x^i) + a_j^2(x) \equiv e_{j,j}(1+x^j)(x^i+x^j) \pmod{x^n-1}$.

b) $a_i(x)(1+x^j) + a_i^2(x) \equiv e_{i,i}(1+x^i)(x^i+x^j) \pmod{x^n-1}$.

c) $a_0(x)(x^i+x^j) + a_0^2(x) \equiv e_{0,0}(1+x^i)(1+x^j) \pmod{x^n-1}$.

Proof: We only prove equation a while the other two hold by relabelling of the construction. First we prove two useful properties on polynomials.

Claim 12: The following equation holds

$$\begin{aligned}&(x^{2i}+x^{2j})(1+x^i) + (1+x^{2j})(1+x^i)x^{2i}\\ &+ (x^{2i}+x^{2j})^2 + (1+x^{2j})^2 x^i\\ &= (1+x^i)(1+x^{2j})(x^{2i}+x^{2j}) + (1+x^i)^3 x^i.\end{aligned}$$

Proof: This equation can be rewritten by

$$\begin{aligned}&(x^{2i}+x^{2j})(1+x^i) + (1+x^{2j})(1+x^i)x^{2i}\\ &+ (1+x^i)(1+x^{2j})(x^{2i}+x^{2j}) + (1+x^i)^3 x^i\\ &= (x^{2i}+x^{2j})^2 + (1+x^{2j})^2 x^i,\end{aligned}$$

or,

$$\begin{aligned}&(1+x^i)[(x^{2i}+x^{2j}) + (1+x^{2j})x^{2i}\\ &+ (1+x^{2j})(x^{2i}+x^{2j}) + (1+x^i)^2 x^i]\\ &= (x^{2i}+x^{2j})^2 + (1+x^{2j})^2 x^i.\end{aligned}$$

Moreover, it can be rewritten by

$$\begin{aligned}&(1+x^i)[x^{2i}+x^{2j}+x^{2i}+x^{2j+2i}+\\ &+ (1+x^{2j})(x^{2i}+x^{2j}) + x^i + x^{3i}]\\ &= (x^{2i}+x^{2j})^2 + (1+x^{2j})^2 x^i,\end{aligned}$$

or

$$\begin{aligned}&(1+x^i)[x^{2j} + x^{2j+2i}\\ &+ x^{2i} + x^{2j} + x^{2i+2j} + x^{4j} + x^i + x^{3i}]\\ &= x^{4i} + x^{4j} + x^i + x^{4j+i}.\end{aligned}$$

We finally rewrite it by

$$\begin{aligned}&(1+x^i)(x^{2i} + x^{4j} + x^i + x^{3i})\\ &= x^{4i} + x^{4j} + x^i + x^{4j+i},\end{aligned}$$

which holds since $(1+x^i)(x^{2i}+x^{4j}+x^i+x^{3i}) = x^{2i}+x^{4j}+x^i+x^{3i}+x^{3i}+x^{4j+i}+x^{2i}+x^{4i} = x^{4i}+x^{4j}+x^i+x^{4j+i}$. ■

Let $e_{0,0}, e_{i,i}, e_{j,j}$ be the label on the self-loop of node v_0, v_i, v_j, respectively, and let $e_{i,0}, e_{j,0}, e_{j,i}$ be the label on the edge $\langle v_i, v_0\rangle, \langle v_j, v_0\rangle, \langle v_j, v_i\rangle$, respectively. We define the

following three polynomials

$$p(x) = e_{i,i}(1+x^{2i}) + e_{j,j}(1+x^{2j}) + e_{j,i}(1+x^i)(1+x^j), \tag{18}$$

$$q(x) = e_{0,0}(1+x^{2i}) + e_{j,j}(x^{2i}+x^{2j}) + e_{j,0}(x^i+x^j)(1+x^i), \tag{19}$$

$$s(x) = e_{i,0}(x^i+x^{2i}) + e_{j,0}(x^j+x^{2j}) + e_{j,i}(x^{2i+j}+x^{i+2j}). \tag{20}$$

Claim 13: The following equation holds

$$q(x)(1+x^i) + p(x)(1+x^i)x^{2i} + q^2(x) + p^2(x)x^i + s(x)(1+x^i)^2 = e_{j,j}(1+x^i)(1+x^{2j})(x^{2i}+x^{2j}).$$

Proof: By the definition,

$$\begin{aligned}
&q(x)(1+x^i) + p(x)(1+x^i)x^{2i} + q^2(x) + p^2(x)x^i \\
&+ s(x)(1+x^i)^2 = \\
&[e_{0,0}(1+x^{2i}) + e_{j,j}(x^{2i}+x^{2j}) + e_{j,0}(x^i+x^j)(1+x^i)](1+x^i) \\
&+ [e_{i,i}(1+x^{2i}) + e_{j,j}(1+x^{2j}) + e_{j,i}(1+x^i)(1+x^j)] \\
&\qquad \cdot (1+x^i)x^{2i} \\
&+ [e_{0,0}(1+x^{2i}) + e_{j,j}(x^{2i}+x^{2j}) + e_{j,0}(x^i+x^j)(1+x^i)]^2 \\
&+ [e_{i,i}(1+x^{2i}) + e_{j,j}(1+x^{2j}) + e_{j,i}(1+x^i)(1+x^j)]^2 x^i \\
&+ [e_{i,0}(x^i+x^{2i}) + e_{j,0}(x^j+x^{2j}) + e_{j,i}(x^{2i+j}+x^{i+2j})] \\
&\qquad \cdot (1+x^i)^2 \\
&= e_{0,0}(1+x^i)^3x^i + e_{i,i}[(1+x^i)^3x^{2i} + (1+x)^{4i}x^i] \\
&+ e_{i,0}(x^i+x^{2i})(1+x^i)^2 \\
&+ e_{j,0}(1+x^i)^2[(x^i+x^j) + (x^i+x^j)^2 + (x^j+x^{2j})] \\
&+ e_{j,i}(1+x^i)^2[(1+x^j)x^{2i} + (1+x^j)^2x^i + (x^{2i+j}+x^{i+2j})] \\
&+ e_{j,j}[(x^{2i}+x^{2j})(1+x^i) + (1+x^{2j})(1+x^i)x^{2i} \\
&\qquad + (x^{2i}+x^{2j})^2 + (1+x^{2j})^2x^i] \\
&\overset{(a)}{=} e_{0,0}(1+x^i)^3x^i + e_{i,i}(1+x^i)^3x^i + e_{i,0}(1+x^i)^3x^i \\
&+ e_{j,0}(1+x^i)^2(x^i+x^{2i}) \\
&+ e_{j,i}(1+x^i)^2[x^{2i}+x^{2i+j}+x^i+x^{i+2j}+x^{2i+j}+x^{i+2j}] \\
&+ e_{j,j}[(1+x^i)(1+x^{2j})(x^{2i}+x^{2j}) + (1+x^i)^3x^i] \\
&= e_{0,0}(1+x^i)^3x^i + e_{i,i}(1+x^i)^3x^i + e_{i,0}(1+x^i)^3x^i \\
&+ e_{j,0}(1+x^i)^3x^i + e_{j,i}(1+x^i)^3x^i \\
&+ e_{j,j}[(1+x^i)(1+x^{2j})(x^{2i}+x^{2j}) + (1+x^i)^3x^i] \\
&\overset{(b)}{=} e_{j,j}(1+x^i)(1+x^{2j})(x^{2i}+x^{2j}),
\end{aligned}$$

where Step (a) holds since by Claim 12,

$$\begin{aligned}
&e_{j,j}[(x^{2i}+x^{2j})(1+x^i) + (1+x^{2j})(1+x^i)x^{2i} \\
&+ (x^{2i}+x^{2j})^2 + (1+x^{2j})^2x^i] \\
&= e_{j,j}[(1+x^i)(1+x^{2j})(x^{2i}+x^{2j}) + (1+x^i)^3x^i],
\end{aligned}$$

and Step (b) holds since by equation (16) $e_{0,0}+e_{i,i}+e_{j,j}+e_{i,0}+e_{j,0}+e_{j,i}=0$. ∎

Summing the equation of Lemma 5(a) with the equation of Lemma 5(b) we get

$$\begin{aligned}
&a_i(x)(1+x^i) + a_j(x)(1+x^j) \equiv e_{0,0} + e_{i,i}x^{2i} + e_{j,j}x^{2j} \\
&+ e_{i,0} + e_{j,0} + e_{j,i}(x^i+x^j+x^{i+j}) (\mathrm{mod} x^n-1),
\end{aligned}$$

and since $e_{0,0} = e_{i,i}+e_{j,j}+e_{i,0}+e_{j,0}+e_{j,i}$ we rewrite it as

$$\begin{aligned}
&a_i(x)(1+x^i) + a_j(x)(1+x^j) \\
&\equiv e_{i,i}(1+x^{2i}) + e_{j,j}(1+x^{2j}) + e_{j,i}(1+x^i)(1+x^j) \\
&= p(x) (\mathrm{mod} x^n-1).
\end{aligned} \tag{21}$$

Multiplying the equation of Lemma 5(a) by x^i and adding it to the equation of Lemma 5(b) we get

$$\begin{aligned}
&a_0(x)(1+x^i) + a_j(x)(x^i+x^j) \equiv e_{0,0} + e_{i,i}x^{2i} + e_{j,j}x^{2j} \\
&+ e_{i,0}x^{2i} + e_{j,0}(x^i+x^j+x^{i+j}) + e_{j,i}x^{2i} (\mathrm{mod} x^n-1),
\end{aligned}$$

and since $e_{i,i} = e_{0,0}+e_{j,j}+e_{i,0}+e_{j,0}+e_{j,i}$ we rewrite it as

$$\begin{aligned}
&a_0(x)(1+x^i) + a_j(x)(x^i+x^j) \equiv e_{0,0}(1+x^{2i}) \\
&+ e_{j,j}(x^{2i}+x^{2j}) + e_{j,0}(x^i+x^j)(1+x^i) \\
&= q(x) (\mathrm{mod} x^n-1).
\end{aligned} \tag{22}$$

Next, we multiply the equation of Lemma 5(c) by $(x^i+1)^2$. In the left-hand side of this equation we set the value of $a_0(x)(1+x^i)$ from equation (22) and the value of $a_i(x)(1+x^i)$ from equation (21) to get that

$$\begin{aligned}
&a_0(x)(1+x^i)^2 + a_i(x)(1+x^i)^2x^{2i} + a_j(x)(1+x^i)^2x^{2j} \\
&+ a_0^2(x)(1+x^i)^2 + a_i^2(x)(1+x^i)^2x^i + a_j^2(x)(1+x^i)^2x^j \\
&\equiv [a_j(x)(x^i+x^j) + q(x)](1+x^i) \\
&+ [a_j(x)(1+x^j) + p(x)](1+x^i)x^{2i} \\
&+ [a_j(x)(x^i+x^j) + q(x)]^2 \\
&+ [a_j(x)(1+x^j) + p(x)]^2\ x^i \\
&+ a_j(x)(1+x^i)^2x^{2j} + a_j^2(x)(1+x^i)^2x^j (\mathrm{mod} x^n-1).
\end{aligned}$$

The right-hand side of this equation is

$$s(x)(1+x^i)^2 (\mathrm{mod} x^n-1).$$

Now, we proceed with the calculations, while having on the left-hand side only the values that depend on $a_j(x)$, so we receive that,

$$\begin{aligned}
&a_j(x)[(x^i+x^j)(1+x^i) + (1+x^j)(1+x^i)x^{2i} + (1+x^i)^2x^{2j}] \\
&+ a_j^2(x)[(x^i+x^j)^2 + (1+x^j)^2x^i + (1+x^i)^2x^j] \\
&\equiv a_j(x)(1+x^i)(x^i+x^j+x^{2i}+x^{2j}+x^{2i+j}+x^{i+2j}) \\
&+ a_j^2(x)(x^i+x^j+x^{2i}+x^{2j}+x^{2i+j}+x^{i+2j}) (\mathrm{mod} x^n-1) \\
&\equiv a_j(x)(1+x^i)^2(1+x^j)(x^i+x^j) \\
&+ a_j^2(x)(1+x^i)(1+x^j)(x^i+x^j) (\mathrm{mod} x^n-1).
\end{aligned}$$

The right-hand side of the last equation is rewritten to be

$$\begin{aligned}
&q(x)(1+x^i) + p(x)(1+x^i)x^{2i} + q^2(x) + p^2(x)x^i \\
&+ s(x)(1+x^i)^2 (\mathrm{mod} x^n-1),
\end{aligned} \tag{23}$$

which is equal to $e_{j,j}(1+x^i)(1+x^{2j})(x^{2i}+x^{2j})$ by Claim 13. Combining both sides together we deduce that

$$\begin{aligned}
&a_j(x)(1+x^i)^2(1+x^j)(x^i+x^j) \\
&+ a_j^2(x)(1+x^i)(1+x^j)(x^i+x^j) \\
&\equiv e_{j,j}(1+x^i)(1+x^{2j})(x^{2i}+x^{2j}) (\mathrm{mod} x^n-1),
\end{aligned}$$

which can be rewritten by,

$$(1+x^i)(1+x^j)(x^i+x^j) \cdot [a_j(x)(1+x^i)+a_j^2(x)+e_{j,j}(1+x^j)(x^i+x^j)] \equiv 0 \pmod{x^n-1}.$$

Lastly, denote by $m(x)$ the polynomial

$$m(x) = a_j(x)(1+x^i)+a_j^2(x)+e_{j,j}(1+x^j)(x^i+x^j),$$

where it holds that $1+x|m(x)$ since $m(1) \equiv 0(\bmod x^n-1)$. Notice that the polynomials $1+x^i$ and $1+x^j$ are in $\mathcal{R}_n$ and by (2) they are co-prime to $M_n(x)$. Similarly, the polynomial x^i+x^j is also in $\mathcal{R}_n$ and thus is co-prime to $M_n(x)$ as well. Therefore, we deduce that $M_n(x)|m(x)$ and $m(x) \equiv 0(\bmod x^n-1)$, which leads to,

$$a_j(x)(1+x^i)+a_j^2(x) \equiv e_{j,j}(1+x^j)(x^i+x^j)(\bmod x^n-1).$$

■

APPENDIX B

Lemma 7: It holds that $e_{0,0}+e_{i,i}+e_{j,j} = e_{j,0}+e_{j,0}+e_{j,i} = 0$.

Proof: We start with proving several important claims.

Claim 14: If

$$a_j(x)(1+x^i)+a_j^2(x) \equiv e_{j,j}(x^j+x^{i+j}+x^{2j})(\bmod x^n-1), \tag{24}$$

then for all $s \in [n]$

$$e_{j,s} = e_{j,\langle 2s\rangle_n} + e_{j,\langle 2s-i\rangle_n}, \tag{25}$$

and for all $1 \leqslant t \leqslant n-1$ it holds that

$$e_{j,s} = \sum_{\ell=0}^{2^t-1} e_{j,\langle 2^t s-\ell i\rangle_n}. \tag{26}$$

Proof: First notice that by calculating the coefficient of $x^{\langle 2s\rangle_n}$ of equation (24) for all $s \in [n]$ such that $\langle 2s\rangle_n \notin \{j, \langle i+j\rangle_n, \langle 2j\rangle_n\}$ it holds that

$$e_{j,\langle 2s\rangle_n} + e_{j,\langle 2s-i\rangle_n} + e_{j,s} = 0.$$

For $\langle 2s\rangle_n = j, \langle 2s\rangle_n = \langle i+j\rangle_n, \langle 2s\rangle_n = \langle 2j\rangle_n$, since the coefficient of $x^j, x^{\langle i+j\rangle_n}, x^{\langle 2j\rangle_n}$ in $a_j(x), a_j(x)x^i, a_j^2(x)$ is zero, respectively, we deduce that also in this case we can write

$$e_{j,\langle 2s\rangle_n} + e_{j,\langle 2s-i\rangle_n} + e_{j,s} = 0,$$

which proves the correctness of equation (25).

Next, we prove the rest of this claim by induction on t where $1 \leqslant t \leqslant n-1$.

Base: for $t=1$, as we showed above, by calculating the coefficient of $x^{\langle 2s\rangle_n}$ of equation (24) we deduce that for all $s \in [n]$ it holds

$$e_{j,s} = e_{j,\langle 2s\rangle_n} + e_{j,\langle 2s-i\rangle_n}.$$

Step: assume that the claim holds for all τ where $1 \leqslant \tau \leqslant t-1 \leqslant n-2$, that is,

$$e_{j,s} = \sum_{\ell=0}^{2^\tau-1} e_{j,\langle 2^\tau s-\ell i\rangle_n}.$$

By the correctness of equation (25) and replacing s with $\langle 2^\tau s-\ell i\rangle_n$ we deduce that

$$e_{j,\langle 2^\tau s-\ell i\rangle_n} = e_{j,\langle 2(2^\tau s-\ell i)\rangle_n} + e_{j,\langle 2(2^\tau s-\ell i)-i\rangle_n},$$

and for $\tau = t-1$ we get that

$$e_{j,s} = \sum_{\ell=0}^{2^{t-1}-1} e_{j,\langle 2^{t-1}s-\ell i\rangle_n} = \sum_{\ell=0}^{2^{t-1}-1} \left(e_{j,\langle 2^t s-2\ell i\rangle_n} + e_{j,\langle 2^t s-2\ell i-i\rangle_n}\right) = \sum_{\ell=0}^{2^t-1} e_{j,\langle 2^t s-\ell i\rangle_n}.$$

■

For $a,t,s \in [n]$ denote by $I_{a,t,s}$ the number

$$I_{a,t,s} = |\{(\tau,\ell) \mid \langle 2^\tau s-\ell a\rangle_n = \langle a2^{-1}\rangle_n, \tau \in [t], \ell \in [2^\tau]\}|.$$

Corollary 9: For all $1 \leqslant t \leqslant n-1$ and $s \in [n]$ it holds that

$$e_{j,s} = \sum_{\ell=0}^{2^t-1} e_{j,\langle 2^t s-\ell i\rangle_n} + e_{j,j}I_{i,t,s}.$$

Proof: By adding the monomial $e_{j,j}x^i$ to equation (24) we get the same expression as in Claim 6(a) on the right-hand side. According to equation (25), by calculating the coefficient of $x^{\langle 2s\rangle_n}$, for all $s \in [n] \setminus \{\langle 2^{-1}i\rangle_n\}$ we will get $e_{j,s} = e_{j,\langle 2s\rangle_n} + e_{j,\langle 2s-i\rangle_n}$ and for $s = \langle 2^{-1}i\rangle_n$ we will get $e_{j,s} = e_{j,\langle 2s\rangle_n} + e_{j,\langle 2s-i\rangle_n} + e_{j,j}$. According to the modification of equation (25) for $s = \langle 2^{-1}i\rangle_n$, we need to similarly adjust equation (26) by counting the number of times the self-loop $e_{j,j}$ should be added to the equation. Hence, by the same arguments of the proof of Claim 14, we deduce that

$$e_{j,s} = \sum_{\ell=0}^{2^t-1} e_{j,\langle 2^t s-\ell i\rangle_n} + e_{j,j}I_{i,t,s},$$

where by definition, $I_{i,t,s}$ is the number of pairs (τ,ℓ) where $\tau \in [t]$ and $\ell \in [2^\tau]$ such that $\langle 2^\tau s-\ell i\rangle_n = \langle i2^{-1}\rangle_n$. ■

Next we show another important claim.

Claim 15: For all $s \in [n]$ it holds

$$e_{j,s} = e_{j,\langle i-s\rangle_n} + e_{j,j}(1+I_{i,\frac{n-1}{2},s}).$$

Proof: By Corollary 9 we know that for all $t,s \in [n]$,

$$e_{j,s} = \sum_{\ell=0}^{2^t-1} e_{j,\langle 2^t s-\ell i\rangle_n} + e_{j,j}I_{i,t,s}.$$

Since 2 is primitive in $\mathbb{Z}_n$, there exists a t for which $\langle 2^t\rangle_n = n-1$, or equivalently, there is an odd positive number h such that $2^t = hn-1$. It can be verified that in this

case $t=(n-1)/2$. Therefore,

$$\begin{aligned}
e_{j,s} &= \sum_{\ell=0}^{nh-2} e_{j,\langle -s-\ell i\rangle_n} + e_{j,j} I_{i,\frac{n-1}{2},s} \\
&= \sum_{\ell=0}^{n(h-1)-1} e_{j,\langle -s-\ell i\rangle_n} + \sum_{\ell=nh-n}^{nh-2} e_{j,\langle -s-\ell i\rangle_n} + e_{j,j} I_{i,\frac{n-1}{2},s} \\
&= \sum_{\ell=0}^{n(h-1)-1} e_{j,\langle -s-\ell i\rangle_n} + \sum_{\ell=0}^{n-2} e_{j,\langle -s-\ell i\rangle_n} + e_{j,j} I_{i,\frac{n-1}{2},s} \\
&\overset{(a)}{=} \sum_{\ell=0}^{n-2} e_{j,\langle -s-\ell i\rangle_n} + e_{j,j} I_{i,\frac{n-1}{2},s} \\
&= e_{j,\langle i-s\rangle_n} + \sum_{\ell=0}^{n-1} e_{j,\langle -s-\ell i\rangle_n} + e_{j,j} I_{i,\frac{n-1}{2},s} \\
&\overset{(b)}{=} e_{j,\langle i-s\rangle_n} + e_{j,j} + e_{j,j} I_{i,\frac{n-1}{2},s} \\
&= e_{j,\langle i-s\rangle_n} + e_{j,j}(1 + I_{i,\frac{n-1}{2},s}).
\end{aligned}$$

Note that the summation $\sum_{\ell=0}^{n(h-1)-1} e_{j,\langle -s-\ell i\rangle_n}$ expresses the neighborhood of the jth node (including its self-loop) $h-1$ times (i.e., an even number of times). Hence, in Step (a) we noticed that $\sum_{\ell=0}^{n(h-1)-1} e_{j,\langle -s-\ell i\rangle_n} = 0$. Step (b) holds since $\sum_{\ell=0}^{n-1} e_{j,\ell} = e_{j,j}$. ∎

Our next goal is to show that the value of $I_{i,\frac{n-1}{2},i}$ is even. For that we show two more claims. First, we define for $t \in [\frac{n+1}{2}]$ the indicator bit x_t as follows:

$$x_t = \begin{cases} 0 & \text{if } \langle 2^{t-1}\rangle_n < \langle 2^{-1}\rangle_n, \\ 1 & \text{if } \langle 2^{t-1}\rangle_n \geqslant \langle 2^{-1}\rangle_n. \end{cases}$$

Claim 16: For all $2 \leqslant t \leqslant \frac{n-1}{2}$,

$$I_{i,t,i} - I_{i,t-1,i} = 2(I_{i,t-1,i} - I_{i,t-2,i}) - x_{t-1} + x_t.$$

Proof: By definition, for $t \in [\frac{n-1}{2}]$, $I_{i,t,i}$ is given by

$$\begin{aligned}
I_{i,t,i} &= |\{(\tau,\ell) | \langle 2^\tau i - \ell i\rangle_n = \langle i 2^{-1}\rangle_n, \tau \in [t], \ell \in [2^\tau]\}| \\
&= |\{(\tau,\ell) | \langle 2^\tau - \ell\rangle_n = \langle 2^{-1}\rangle_n, \tau \in [t], \ell \in [2^\tau]\}| \\
&= |\{(\tau,m) | \langle m\rangle_n = \langle 2^{-1}\rangle_n, \tau \in [t], 1 \leqslant m \leqslant 2^\tau\}|.
\end{aligned}$$

Therefore, it holds that for all $2 \leqslant t \leqslant \frac{n-1}{2}$

$$\begin{aligned}
I_{i,t,i} - I_{i,t-1,i} &= |\{m \mid \langle m\rangle_n = \langle 2^{-1}\rangle_n, 1 \leqslant m \leqslant 2^{t-1}\}| \\
&= \left\lfloor \frac{2^{t-1}}{n} \right\rfloor + x_t = \left\lfloor 2 \cdot \left(\frac{2^{t-2}}{n} \right) \right\rfloor + x_t \\
&= 2 \left\lfloor \frac{2^{t-2}}{n} \right\rfloor + x_{t-1} + x_t \\
&\overset{(a)}{=} 2(I_{i,t-1,i} - I_{i,t-2,i}) - x_{t-1} + x_t,
\end{aligned}$$

where in Step (a) we used the property that $I_{i,t,i} - I_{i,t-1,i} = \left\lfloor \frac{2^{t-2}}{n} \right\rfloor + x_{t-1}$. ∎

Claim 17: For all $t \in [\frac{n+1}{2}]$ it holds that $\langle I_{i,t,i} + x_t\rangle_2 = 0$.

Proof: We will prove this claim by induction on $t \in [\frac{n-1}{2}]$.

Base: For $t=0$, $\langle 2^0\rangle_n = 1$ which is smaller than $\langle 2^{-1}\rangle_n$ and indeed $I_{i,0,i} + x_0 = 0$. Similarly, for $t=1$, $\langle 2^1\rangle_n = 2$ which is smaller than $\langle 2^{-1}\rangle_n$ for all prime $n \geqslant 5$ and therefore again $I_{i,1,i} + x_1 = 0$ is even.

Step: Assume that the claim holds for $t-1$ where $2 \leqslant t < \frac{n-1}{2}$. By the induction assumption, we have that $\langle I_{i,t-1,i} + x_{t-1}\rangle_2 = 0$, and by Claim 16 we know that

$$I_{i,t,i} - I_{i,t-1,i} = 2(I_{i,t-1,i} - I_{i,t-2,i}) - x_{t-1} + x_t,$$

or similarly

$$I_{i,t,i} = 3I_{i,t-1,i} - 2I_{i,t-2,i} - x_{t-1} + x_t,$$

and therefore

$$\begin{aligned}
\langle I_{i,t,i} + x_t\rangle_2 &= \langle 3I_{i,t-1,i} + 2I_{i,t-2,i} + x_{t-1}\rangle_2 \\
&= \langle I_{i,t-1,i} + x_{t-1}\rangle_2 = 0.
\end{aligned}$$

∎

Corollary 10: The value of $I_{i,\frac{n-1}{2},i}$ is even.

Proof: By Claim 17, it holds that $x_{\frac{n-1}{2}} = 0$ since $\langle 2^{\frac{n-1}{2}-1}\rangle_n = \frac{n-1}{2} < \frac{n+1}{2} = \langle 2^{-1}\rangle_n$ and we immediately deduce that $I_{i,\frac{n-1}{2},i}$ is even. ∎

By Claim 15 we know that

$$e_{j,s} = e_{j,\langle i-s\rangle_n} + e_{j,j}(1 + I_{i,\frac{n-1}{2},s}).$$

Since $I_{i,\frac{n-1}{2},i}$ is even, we get

$$e_{j,i} = e_{j,0} + e_{j,j}.$$

By symmetry of the construction, we also get

$$\begin{aligned}
e_{j,i} + e_{i,0} + e_{i,i} &= 0, \\
e_{j,0} + e_{i,0} + e_{0,0} &= 0.
\end{aligned}$$

The summation of the last three equalities results with

$$e_{j,j} + e_{i,i} + e_{0,0} = 0,$$

and since

$$e_{j,j} + e_{i,i} + e_{0,0} + e_{j,i} + e_{i,0} + e_{j,0} = 0,$$

we deduce that

$$e_{j,i} + e_{i,0} + e_{j,0} = 0.$$

∎

ACKNOWLEDGMENT

The authors would like to thank G. Kupfer for his contribution to the result of Corollary 10 in Appendix B. They also thank the two anonymous reviewers and the Associate Editor S. Ghorpade for their valuable comments and suggestions, which have contributed to the clarity of the article and its presentation.

REFERENCES

[1] N. Abramson, "Cascade decoding of cyclic product codes," *IEEE Trans. Commun.*, vol. COM-16, no. 3, pp. 398–402, Jun. 1968.

[2] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[3] M. Blaum, J. L. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4510–4519, Jul. 2013.

[4] M. Blaum and S. R. Hetzler, "Array codes with local properties," 2019, *arXiv:1906.11731*. [Online]. Available: https://arxiv.org/abs/1906.11731

[5] M. Blaum and R. Roth, "New array codes for multiple phased burst correction," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 66–77, Jan. 1993.

[6] P. Corbett *et al.*, "Row-diagonal parity for double disk failure correction," in *Proc. 3rd USENIX Symp. File Storage Technol.*, San Francisco, CA, USA, Apr. 2004, pp. 1–14.
[7] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.
[8] P. Elias, "Error free coding," *IRE Trans. IRE Prof. Group Inf. Theory*, vol. 4, no. 4, pp. 29–37, Sep. 1954.
[9] E. En Gad, R. Mateescu, F. Blagojevic, C. Guyot, and Z. Bandic, "Repair-optimal MDS array codes over GF(2)," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 887–891.
[10] J. Hopfield, *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988, pp. 457–464.
[11] H. Hou, Y. S. Han, K. W. Shum, and H. Li, "A unified form of EVENODD and RDP codes and their efficient decoding," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5053–5066, Nov. 2018.
[12] C. Huang and L. Xu, "STAR: An efficient coding scheme for correcting triple storage node failures," *IEEE Trans. Comput.*, vol. 57, no. 7, pp. 889–901, Jul. 2008.
[13] S. K. Mohammed, E. Viterbo, Y. Hong, and A. Chockalingam, "X-codes: A low complexity full-rate high-diversity achieving precoder for TDD MIMO systems," in *Proc. IEEE Int. Conf. Commun.*, Cape Town, South Africa, May 2010, pp. 1–5.
[14] N. Raviv, N. Silberstein, and T. Etzion, "Constructions of high-rate minimum storage regenerating codes over small fields," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2015–2038, Apr. 2017.
[15] R. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 328–336, Mar. 1991.
[16] K.-U. Schmidt, "Symmetric bilinear forms over finite fields of even characteristic," *J. Combinat. Theory A*, vol. 117, no. 8, pp. 1011–1026, Nov. 2010.
[17] K.-U. Schmidt, "Symmetric bilinear forms over finite fields with applications to coding theory," *J. Algebraic Combinatorics*, vol. 42, no. 2, pp. 635–679, Sep. 2015.
[18] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.
[19] D. Wiedemann, "Solving sparse linear equations over finite fields," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 54–62, Jan. 1986.
[20] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner, "Low-density MDS codes and factors of complete graphs," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1817–1836, Mar. 1999.
[21] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, vol. 45, no. 6, pp. 106–110.
[22] L. Yohananov, Y. Efron, and E. Yaakobi, "Double and triple node-erasure-correcting codes over graphs," in *Proc. IEEE Int. Symp. Inf. Theory*, Paris, France, Jul. 2019, pp. 1582–1586.
[23] L. Yohananov and E. Yaakobi, "Codes for graph erasures," in *Proc. IEEE Int. Symp. Inf. Theory*, Aachen, Germany, Jul. 2017, pp. 844–848.
[24] L. Yohananov and E. Yaakobi, "Codes for erasures over directed graphs," in *Proc. IEEE Int. Inf. Theory Workshop*, Kaohsiung, Taiwan, Nov. 2017, pp. 116–120.
[25] L. Yohananov and E. Yaakobi, "Codes for graph erasures," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5433–5453, Sep. 2019.

Lev Yohananov (Student Member, IEEE) received the B.Sc. and the M.Sc. degrees in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Computer Science Department. His research interests include algebraic error-correction codes, codes over graphs, and combinatorics.

Yuval Efron received the B.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2018, where he is currently pursuing the M.Sc. degree with the Computer Science Department, Technion—Israel Institute of Technology. His research interests include coding theory and distributed graph algorithms.

Eitan Yaakobi (Senior Member, IEEE) received the B.A. degree in computer science and mathematics, and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California at San Diego in 2011. From 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology. He is currently an Associate Professor with the Computer Science Department, Technion—Israel Institute of Technology. His research interests include information and coding theory with applications to nonvolatile memories, associative memories, data storage and retrieval, and voting theory. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship from 2010 to 2011.