Double and Triple Node-Erasure-Correcting Codes over Graphs

Lev Yohananov
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 3200009, Israel
Email: levyohananov@campus.technion.ac.il

Yuval Efron
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 3200009, Israel
Email: szxrtde@cs.technion.ac.il

Eitan Yaakobi
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 3200009, Israel
Email: yaakobi@cs.technion.ac.il

***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.**

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 [17]–[19], 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 [8]. Similarly, in associative memories, the information is stored by associations between different data items [15]. Furthermore, representing information in a graph can model a distributed storage system [6] while every two nodes can be connected by a link that represents the information that is shared by the nodes, e.g., a node may refer to a user and an edge to a file which is shared between two users, and the self-loops are the user's files.

In [17]–[19], 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. In case every node corresponds to a user, a node failure implies that the user's files and those which are shared with the other users are erased.

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 [11]. 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], [10]–[14]. 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 [17], [19]. More specifically, based upon product codes [1], [7], a construction of optimal codes whose alphabet size grows only linearly with n has been proposed. Additionally, using rank-metric codes [11]–[13], binary codes over graphs were designed, however they are relatively close—yet do not attain—the Singleton bound. In [17], [18], a construction of optimal binary codes for two node failures was also presented based upon ideas from EVENODD codes [2].

Another approach for handling symmetric crisscross erasures (in symmetric arrays) is by using symmetric rank-metric codes. In [12], 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 [5] 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 [17], [19] that will be used in the paper. In Section III, we present our optimal binary construction for two-node failures. This construction is simpler than our optimal construction from [17], [19]. Then, in Section IV, 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

978-1-5386-9291-2/19/$31.00 ©2019 IEEE

from [12]. Lastly, Section V concludes the paper. Due to the lack of space some or part of the proofs in the paper are omitted, while they can be found in the long version fo the paper [16].

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 [17] 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.

As discussed in [17]–[19], 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 [17]–[19] 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 [19] and its connection to construct codes correcting node failures. Let $G = (V_n, L)$ be a graph and let E be a 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 [19] 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 $d(\mathcal{C}) = \min_{G \in \mathcal{C}, G \neq G_{\mathbf{0}}} \{w(G)\}$. Lastly, we state the following theorem from [19] 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

$$\gcd(x^\ell + 1, M_n(x)) = 1. \qquad (2)$$

Notice also that when 2 is primitive in $\mathbb{Z}_n$, the polynomial $M_n(x)$ is irreducible [4]. 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 $1 + x^{sk} = (1 + x^s)^k$. Throughout the paper, the notation $\langle a \rangle_n$ will be used to denote the value of $(a \bmod n)$ and the operator " $\equiv$ " is defined to be an equality modulo $x^n - 1$.

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 $n \geqslant 2$ 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

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

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

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

An important observation is that D_m contains only a single self-loop which is the edge $\langle v_{\langle m\cdot 2^{-1}\rangle_n}, v_{\langle m\cdot 2^{-1}\rangle_n}\rangle$.

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 $n \geqslant 2$ 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

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

Therefore the code $\mathcal{C}_2$ has at most $2n-1$ linearly independent constraints which implies that its redundancy is at most $2n-1$. Since we will prove in Theorem 2 that $\mathcal{C}_2$ is a double-node-correcting codes, according to the Singleton bound we get that the redundancy of the code $\mathcal{C}_2$ is exactly $2n-1$, and thus it is an optimal code.

According to Theorem 1, in order to prove that $\mathcal{C}_2$ is a double-node-erasure-correcting code, we need to show that $d(\mathcal{C}_2) \geqslant 3$, that is, for every $G \in \mathcal{C}_2$, $w(G) \geqslant 3$. This will be proved in the next theorem.

Theorem 2. *For all prime number n, the code $\mathcal{C}_2$ is an optimal double-node-erasure-correcting code.*

Proof: Assume in the contrary that $d(\mathcal{C}_2) \leqslant 2$ and let $G \in \mathcal{C}_2, G \neq G_{\mathbf{0}}$ be a nonzero graph such that $w(G) = 2$ (a similar proof will hold in case $w(G) = 1$). Since $w(G) = 2$, the graph G has a vertex cover of size 2, that is, all its nonzero edges are confined to the neighborhoods N_i, N_j of some two nodes v_i, v_j. By symmetry of the graph, it suffices to prove the above property for the case where the two nodes are v_0, v_i for some $i \neq 0$. During the proof, we assume that $a_i(x)$, for $i \in [n]$ are the neighborhood polynomials of the graph G. We first prove the following two claims.

Claim 3. *The following properties hold on the graph G:*

(a) For all $h \in [n] \setminus \{0, i\}$, $e_{h,0} + e_{h,i} = 0$.
(b) For all $h \in [n] \setminus \{i\}$, $e_{0,h} + e_{i,\langle h-i\rangle_n} = 0$.
(c) $e_{0,i} = 0$.

Proof:

(b) For $h \in [n] \setminus \{i\}$, the set $D_h \setminus \left\{\langle v_0, v_h\rangle, \langle v_i, v_{\langle h-i\rangle_n}\rangle\right\}$ will be denoted by D'_h. Therefore, we have that

$$\begin{aligned} 0 &= \sum_{\langle v_\ell, v_{\langle h-\ell\rangle_n}\rangle \in D_h} e_{\ell,\langle h-\ell\rangle_n} \\ &= \sum_{\langle v_\ell, v_{\langle h-\ell\rangle_n}\rangle \in D'_h} e_{\ell,\langle h-\ell\rangle_n} + e_{0,h} + e_{i,\langle h-i\rangle_n}, \end{aligned}$$

and since $e_{s,\ell} = 0$ for all $\langle v_s, v_\ell\rangle \in D'_h$, we get that $e_{0,h} + e_{i,\langle h-i\rangle_n} = 0$.

■

Claim 4. *The following properties hold on the graph G:*

(a) For all $h \in [n]$, $a_h(1) = 0$.
(b) $a_0(x) + a_i(x) = 0$.
(c) $a_0(x) + a_i(x)x^i \equiv e_{0,0} + e_{i,i}x^{2i}$.

Proof:

$$\begin{aligned} \text{(c) } a_0(x) + a_i(x)x^i &= e_{0,0} + e_{i,i}x^{2i} + \sum_{\ell=0}^{n-1} e_{0,\ell}x^\ell + \sum_{\ell=0}^{n-1} e_{i,\ell}x^{\ell+i} \\ &\equiv e_{0,0} + e_{i,i}x^{2i} + \sum_{\ell=0}^{n-1} e_{0,\ell}x^\ell + \sum_{\ell=0}^{n-1} e_{i,\langle \ell-i\rangle_n}x^\ell \\ &\equiv e_{0,0} + e_{i,i}x^{2i} + \sum_{\ell=0}^{n-1}\left(e_{0,\ell} + e_{i,\langle\ell-i\rangle_n}\right)x^\ell \\ &\overset{(a)}{\equiv} e_{0,0} + e_{i,i}x^{2i} + \left(e_{0,i} + e_{i,0}\right)x^i \equiv e_{0,0} + e_{i,i}x^{2i}, \end{aligned}$$

where Step (a) holds since by Claim 3(b) for all $\ell \in [n] \setminus \{i\}$, $e_{0,\ell} + e_{i,\langle \ell-i\rangle_n} = 0$. ■

The summation of the equations from Claims 4(b) and 4(c) results with $a_i(x)(1+x^i) \equiv e_{0,0} + e_{i,i}x^{2i}$. 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 $a_i(x)(1+x^i) \equiv 1 + x^{2i}$. Since $1 + x^{2i} = (1+x^i)^2$, it holds that $(1+x^i)(1+x^i+a_i(x)) \equiv 0$. 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 $a_i(x)(1+x^i) \equiv 0$. Notice that $a_i(x) \in \mathcal{R}_n$ and by Claim 4(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, from Claim 4(b) we get also that $a_0(x) = 0$ and together we get that $G = G_{\mathbf{0}}$, which is a contradiction. This completes the proof. ■

Note that whenever two nodes fail, the number of unknown variables is $2n-1$, and so a naive decoding solution for the code $\mathcal{C}_2$ is to solve the linear equation system of $2n-1$ constraints with the $2n-1$ variables. However, the complexity of such a solution will be $O(n^\omega)$, where it is only known that $2 \leqslant \omega \leqslant 2.37286$ as it requires the inversion of a $(2n-1) \times (2n-1)$ matrix [9]. In [16] we present a decoding algorithm for $\mathcal{C}_2$ of time complexity $\Theta(n^2)$. That is, we prove the following theorem.

Theorem 5. *There exists an efficient decoding procedure to the code $\mathcal{C}_2$ given any two node failures. Its complexity is $\Theta(n^2)$, where n is the number of nodes.*

Clearly, this time complexity is optimal since the complexity of the input size of the graph is $\Theta(n^2)$.

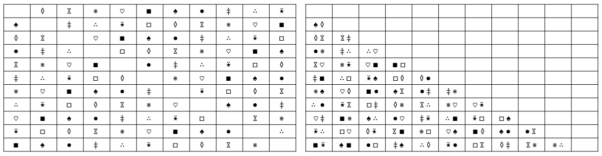
IV. Binary Triple-Node-Erasure-Correcting Codes

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

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

Example 1. In Fig. 1 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.



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

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

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,

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

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 6 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.

Theorem 6. *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 $H_{i,j}=\{i,j,\langle 2i\rangle_n,\langle 2j\rangle_n,\langle 2i+j\rangle_n,\langle 2j+i\rangle_n\}$. For all $\ell\in[n]$ denote by $h_{i,j}(\ell)$ the sum

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

The next claim presents several useful properties.

Claim 7. *The following properties hold on the graph G:*

(a) For all $\ell\in[n]\setminus\{0,i,j\}$, $e_{0,\ell}+e_{i,\ell}+e_{j,\ell}=0$.

(b) For all $\ell\in[n]\setminus\{i,j,\langle i+j\rangle_n\}$, $e_{0,\ell}+e_{i,\langle\ell-i\rangle_n}+e_{j,\langle\ell-j\rangle_n}=0$.

(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 $\ell\in[n]\setminus H_{i,j}$, it holds that $h_{i,j}(\ell)=0$.

(e) It holds that

$$\sum_{\ell\in H_{i,j}} h_{i,j}(\ell)x^\ell \equiv e_{i,0}(x^i+x^{2i})+e_{j,0}(x^j+x^{2j})+e_{j,i}(x^{2i+j}+x^{i+2j}).$$

Proof: Remember that for $s,\ell\in[n]\setminus\{0,i,j\}$, $e_{s,\ell}=0$.

(d) For all $\ell\in[n]$, let B_ℓ be the following edge set

$$\begin{aligned} B_\ell=\{&\langle v_0,v_\ell\rangle,\langle v_i,v_{\langle\ell-2i\rangle_n}\rangle,\langle v_j,v_{\langle\ell-2j\rangle_n}\rangle, \\ &\langle v_0,v_{\langle 2^{-1}\ell\rangle_n}\rangle,\langle v_i,v_{\langle 2^{-1}(\ell-i)\rangle_n}\rangle,\langle v_j,v_{\langle 2^{-1}(\ell-j)\rangle_n}\rangle\}. \end{aligned} \quad (6)$$

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

$$0=\sum_{\langle v_k,v_m\rangle\in T_\ell} e_{k,m}=\sum_{\langle v_k,v_m\rangle\in B_\ell} e_{k,m}=h_{i,j}(\ell).$$

Moreover, for $\ell=0$, $\langle v_0,v_\ell\rangle=\langle v_0,v_{\langle 2^{-1}\ell\rangle_n}\rangle=\langle v_0,v_0\rangle$, 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 $s\in\{0,i,j\}, \ell\in\{0,\langle 3i\rangle_n,\langle 3j\rangle_n\}$, if $\langle v_s,v_{\langle\ell-2s\rangle_n}\rangle\in B_\ell$ then it holds that $\langle v_s,v_{\langle\ell-2s\rangle_n}\rangle\in T_\ell\cup\{\langle v_s,v_s\rangle\}$, i.e., $B_\ell\subseteq T_\ell\cup\{\langle v_s,v_s\rangle\}$. Therefore again, by the definition of the diagonal constraint (c) in Construction 2 we deduce that for all $\ell\in\{0,\langle 3i\rangle_n,\langle 3j\rangle_n\}$,

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

(e) For all $\ell\in H_{i,j}$ let B_ℓ be the edge set from (6). Similarly to the proof of (d) it can be verified that for all $\ell\in H_{i,j}$, $|B_\ell|=5$, and the edge set B_ℓ consists of all the edges incident to at least one of the nodes v_0,v_i and v_j in T_ℓ, i.e., $B_\ell\subseteq T_\ell$. Therefore we deduce that for $\ell\in\{i,j\}$,

$$e_{\ell,0}=\sum_{\langle v_k,v_m\rangle\in T_\ell} e_{k,m}+e_{\ell,0}=\sum_{\langle v_k,v_m\rangle\in B_\ell} e_{k,m}+e_{\ell,0}=h_{i,j}(\ell),$$

and the coefficient of the monomial x^i, x^j in the polynomial $\sum_{\ell\in H_{i,j}} h_{i,j}(\ell)x^\ell$ 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. ■

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 8. *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) $a_0(x)+a_i(x)x^i+a_j(x)x^j$
$\equiv e_{0,0}+e_{i,i}x^{2i}+e_{j,j}x^{2j}+e_{i,0}x^i+e_{j,0}x^j+e_{j,i}x^{i+j}$.

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

Proof:

(c) According to the neighborhood-polynomials definition we can write

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

where Step (a) holds since by Claim 7(d) for all $\ell \in [n]\setminus H_{i,j}$ the coefficient of x^{ℓ} is zero, and Step (b) is a direct result of Claim 7(e). ■

Notice that by setting $x=1$ in the equation of Lemma 8(b) we get that

$$e_{0,0}+e_{i,i}+e_{j,j}+e_{i,0}+e_{j,0}+e_{j,i}=0. \tag{7}$$

Using the result of Lemma 8 we get the next equalities.

Lemma 9. *The following equations hold*

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

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 omitted.

Lemma 10. *It holds* $e_{0,0}+e_{i,i}+e_{j,j}=e_{j,0}+e_{j,0}+e_{j,i}=0$.

By Lemma 10, 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 9(b) and 9(c). By Lemma 9(a), we get that $a_j(x)[1+x^i+a_j(x)]\equiv 0$. 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 10 we have that $e_{i,0}=0$. By Lemma 8(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 10 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 8(b) we deduce that

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

and therefore $a_0(x)+a_i(x)x^i\equiv e_{0,0}+e_{i,i}x^{2i}$. As we showed in the proof of Theorem 2, since the conditions of Claim 4 hold, we deduce also here that $a_0(x)=a_i(x)=0$, and therefore we get a contradiction since $e_{j,i}=e_{j,0}=1$. Therefore, it holds $a_j(x)=0$ and since $\mathcal{C}_3$ is a sub code of $\mathcal{C}_2$, we again get that $a_0(x)=a_i(x)=0$, which concludes the proof. ■

V. Conclusion

In this paper we continued our research on codes over graphs from [17], [18]. We presented an optimal binary construction for codes correcting a failure of two nodes together with a decoding procedure that is complexity 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.

Acknowledgments

The authors would like to thank Gil Kupfer for his contribution to the result of Lemma 10.

References

[1] N. Abramson, "Cascade decoding of cyclic product codes," *IEEE Trans. Communication Technology*, vol. 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. Computers*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[3] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 529–542, Mar. 1996.

[4] 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, Mar. 2013.

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

[6] 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.

[7] P. Elias, "Error free coding," *IRE Trans. of the IRE Professional Group on Inf. Theory*, vol. 4, no. 4, pp. 29–37, Sep. 1954.

[8] J. Hopfield, *Neurocomputing: foundations of research*, MIT Press Cambridge, MA, USA, pp. 457–464, 1988.

[9] F. Le Gall, "Powers of tensors and fast matrix multiplication," *Proc. 39th Int. Symp. on Symbolic and Algebraic Computation*, pp. 296–303, 2014.

[10] P. MCorbett, R. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," *Proc. 3rd USENIX Symp. on File and Storage Technologies*, pp. 1–14, San Francisco, CA, USA, Apr. 2004.

[11] R.M. 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.

[12] K.-U. Schmidt, "Symmetric bilinear forms over finite fields of even characteristic," *J. of Combinatorial Theory, Series A*, vol. 117, no. 8, pp. 1011–1026, May 2010.

[13] K.-U. Schmidt, "Symmetric bilinear forms over finite fields with applications to coding theory," *J. of Algebraic Combinatorics*, vol. 42, no. 2, pp. 635–679, Sep. 2015.

[14] 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.

[15] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," *Proc. IEEE Int. Symp. Inf. Theory*, vol. 45, no. 6, pp. 106–110, Cambridge, MA, USA, Jul. 2012.

[16] L. Yohananov, Y. Efron, and E. Yaakobi, "Double and triple node-erasure-correcting codes over graphs," arXiv:1812.00485v2, Dec. 2018.

[17] L. Yohananov and E. Yaakobi, "Codes for graph erasures," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 844–848, Aachen, Germany, Jul. 2017.

[18] L. Yohananov and E. Yaakobi, "Codes for erasures over directed graphs," *Proc. IEEE Int. Inf. Theory Workshop*, pp. 116–120, Kaohsiung, Taiwan, Nov. 2017.

[19] L. Yohananov and E. Yaakobi, "Codes for graph erasures," submitted to *IEEE Trans. Inf. Theory*, 2018.