Multicast Convolutional Network Codes via Local Encoding Kernels

Morteza Rekab-Eslami, Morteza Esmaeili, and T. Aaron Gulliver

2017

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 15, 2017, accepted March 6, 2017, date of publication April 12, 2017, date of current version May 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2689781

Multicast Convolutional Network Codes via Local Encoding Kernels

MORTEZA REKAB-ESLAMI[1], MORTEZA ESMAEILI[1], AND THOMAS AARON GULLIVER[2]
[1]Isfahan University of Technology, Isfahan 84156-83111, Iran
[2]University of Victoria, Victoria, BC V8W 2Y2, Canada

Corresponding author: Thomas Aaron Gulliver (agullive@ece.uvic.ca)

ABSTRACT A convolutional network (CN) code can be described by either global encoding kernels (GEKs) or local encoding kernels (LEKs). In the literature, the multicast property of a CN code is described using GEKs, so the design algorithms for multicast CN codes employ GEKs to check this property. For cyclic networks, using GEKs makes the design algorithms time-consuming. In this paper, a new approach is proposed for the design of multicast CN codes for networks with cycles. First, a formula is presented to describe the multicast property using LEKs rather than GEKs. Then, this formula is used to develop a design algorithm for multicast CN codes. This algorithm does not use GEKs, which makes it more efficient than GEK-based algorithms, particularly for large cyclic networks.

INDEX TERMS Cyclic network, multicast, edge-disjoint cycles, flow, local encoding kernel.

I. INTRODUCTION

Linear network coding over cyclic networks has attracted significant attention because of the many practical applications [1]–[6]. Over cyclic networks, data propagation around a cycle may be noncausal. To break the deadlock, a time delay is used and this data transmission scheme is called convolutional network (CN) coding [7]–[9]. In a CN code, intermediate nodes perform linear operations on a *rational power series* [2], so through each edge flows a linear combination of the symbols generated by the sources. The coefficients of this linear combination form a vector called the global encoding kernel (GEK) and the coefficients of the linear operation are called the local encoding kernel (LEK).

For a single source multicast network, there exists a CN code over a sufficiently large rational power series that achieves the max-flow, which is the smallest minimum cut between the source node and any sink node [2]. Such a CN code is said to be *multicast*. In the literature, the multicast property of a code is described by GEKs, i.e. a code is multicast when the matrix constructed using the GEKs of the incoming edges of each sink has full-rank. All existing algorithms in the literature for designing multicast CN codes use this condition [1], [2], [10], [11]. Unfortunately, using GEKs makes the design algorithm time consuming for networks with cycles.

In this paper, a new approach is presented to design multicast CN codes. First a formula is proposed to check the multicast property using LEKs rather than GEKs. This formula is then used to develop a design algorithm for multicast CN codes. This algorithm does not use GEKs and so it is more efficient than GEK-based algorithms, particularly for large cyclic networks.

The rest of the paper is organized as follows. In Section II, CN coding on cyclic networks is presented. In Section III, a formula is presented to check the multicast property of a CN code. An algorithm for finding LEKs of a multicast CN code is given in Section IV. Finally, Section V provides a summary of the results.

II. CONVOLUTIONAL NETWORK CODING

In this paper, a single source multicast network is modeled as a finite directed multi-edge graph $\mathcal{N} := (\mathcal{V}, \mathcal{E}_s \cup \mathcal{E}, h)$ where $\mathcal{V}$ is the set of nodes, $\mathcal{E}_s$ is the set of outgoing edges of the source node, $\mathcal{E}$ is the set of other edges, and h is the max-flow of the network. Without loss of generality, we assume that each sink has h incoming edges, and the source has h outgoing edges and no incoming edges. For a node v, the sets of incoming and outgoing edges are denoted by $\mathrm{In}(v)$ and $\mathrm{Out}(v)$, respectively. An ordered pair (d, e) of edges is called an *adjacent pair* when the head of d is the tail of e. Paths and cycles can be represented by sets of adjacent pairs. Thus, the path $e_1, e_2, \ldots, e_{k-1}, e_k$ is represented by the adjacent pair set $\{(e_1, e_2), \ldots, (e_{k-1}, e_k)\}$, and the cycle $e_1, e_2, \ldots, e_k, e_1$ is represented by the adjacent pair set $\{(e_1, e_2), \ldots, (e_k, e_1)\}$.

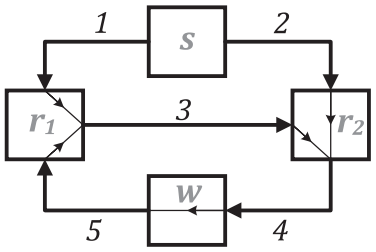

FIGURE 1. A multicast cyclic network with two sinks r_1 and r_2 and max-flow two.

For example, the only cycle in the network shown in Figure 1 is denoted by $\{(3, 4), (4, 5), (5, 3)\}$.

In a CN code, intermediate nodes perform linear operations on a rational power series $\mathbb{F}[(D)]$ which are rational functions of the form $p(D)/(1 + Dq(D))$ where $p(D)$ and $q(D)$ belong to $\mathbb{F}[D]$, i.e. the polynomial ring over the field $\mathbb{F}$ [2].

Definition 1: A CN code $\mathcal{K} = (k_{d,e})$ on a network is defined as the assignment of an element $k_{d,e} \in \mathbb{F}[D]$ to each edge pair (d, e) such that $k_{d,e} = 0$ when (d, e) is not an adjacent pair. The polynomial $k_{d,e}$ is called the *local encoding kernel* (LEK). Further, corresponding to each edge e there is an h-dimensional column vector f_e over $\mathbb{F}[(D)]$, called the *global encoding kernel* (GEK), such that:

1) the set of vectors $\{f_e : e \in \mathcal{E}_s\}$ forms the natural basis of $\mathbb{F}^h$, and
2) $f_e = \sum_{d\in \mathrm{In}(v)} k_{d,e} f_d$ for every edge $e \in \mathrm{Out}(v)$ and node v.

In this paper, $[f_e]_{e\in\mathcal{E}'}$ denotes a matrix whose columns are GEKs of the edge set $\mathcal{E}'$. Further, $[k_{d,e}]_{d\in\mathcal{E}',e\in\mathcal{E}''}$ denotes a matrix whose entries are LEKs $k_{d,e}$ where $d \in \mathcal{E}'$ and $e \in \mathcal{E}''$. The second case in Definition 1 can be expressed as $[f_e]_{e\in\mathcal{E}} = [f_e]_{e\in\mathcal{E}}[k_{d,e}]_{d,e\in\mathcal{E}} + [k_{d,e}]_{d\in\mathcal{E}_s,e\in\mathcal{E}}$, which is equivalent to

$$[f_e]_{e\in\mathcal{E}}(I_{|\mathcal{E}|} - [k_{d,e}]_{d,e\in\mathcal{E}}) = [k_{d,e}]_{d\in\mathcal{E}_s,e\in\mathcal{E}}. \tag{1}$$

which is a system of linear equations with discriminant $\det(I_{|\mathcal{E}|} - [k_{d,e}]_{d,e\in\mathcal{E}})$. If the discriminant is zero, then none or multiple solutions exist, otherwise there exists a unique solution on the quotient field of $\mathbb{F}[(D)]$.

Definition 2: The discriminant of a CN code $\mathcal{K}$ over a network is $\delta(\mathcal{K}) := \det(I_{|\mathcal{E}|} - [k_{d,e}]_{d,e\in\mathcal{E}})$. A code is said to be *normal* if it has a nonzero discriminant and determines a unique set of GEKs in $\mathbb{F}^h[(D)]$.

Definition 3: A normal CN code on a network with max-flow h is said to be *multicast* when $\mathrm{rank}([f_e]_{e\in\mathrm{In}(r)}) = h$ for every sink r.

For example, consider a binary CN code for the network in Figure 1. If the LEKs of this code are $k_{1,3} = k_{2,4} = k_{3,4} = k_{4,5} = k_{5,3} = 1$, then the discriminant is zero. However, if the LEKs are $k_{1,3} = k_{2,4} = k_{3,4} = k_{4,5} = k_{5,3} = D$, then the discriminant is nonzero and the unique GEKs are

$$f_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad f_3 = \begin{bmatrix} D/(1-D^3) \\ D^3/(1-D^3) \end{bmatrix},$$
$$f_4 = \begin{bmatrix} D^2/(1-D^3) \\ D/(1-D^3) \end{bmatrix}, \quad f_5 = \begin{bmatrix} D^3/(1-D^3) \\ D^2/(1-D^3) \end{bmatrix}.$$

This CN code is normal because all GEKs are in $\mathbb{F}^2[(D)]$. Further, it is multicast because

$$\mathrm{rank}([f_e]_{e\in\mathrm{In}(r_1)}) = \mathrm{rank}\left(\begin{bmatrix} 1 & D^3/(1-D^3) \\ 0 & D^2/(1-D^3) \end{bmatrix}\right) = 2,$$
$$\mathrm{rank}([f_e]_{e\in\mathrm{In}(r_2)}) = \mathrm{rank}\left(\begin{bmatrix} 0 & D/(1-D^3) \\ 1 & D^3/(1-D^3) \end{bmatrix}\right) = 2.$$

III. DESCRIPTION OF THE MULTICAST PROPERTY USING LEKs

In this section, we propose a formula to check the multicast property of a CN code. This formula uses the concepts of a *multiple-cycle* and the *partial discriminant* defined as follows.

Definition 4: A multiple-cycle in a network is defined as a union of some edge-disjoint cycles, where a cycle is a special multiple-cycle that is formed by one cycle. The sign of a multiple-cycle C is defined as $\mathrm{sgn}(C) := (-1)^{\sigma_C}$ where σ_C is the number of cycles that form C. The set of all multiple-cycles is denoted by $\mathcal{C}$. For example, the cyclic multicast network shown in Figure 2 has only one multiple-cycle which is $\{(15, 16), (16, 17), (17, 18), (18, 15)\}$.

It was shown in [3] that the discriminant of a CN code $\mathcal{K} = (k_{d,e})$ on a network can be obtained via LEKs as

$$\delta(\mathcal{K}) = 1 + \sum_{C\in\mathcal{C}} \mathrm{sgn}(C) \prod_{(d,e)\in C} k_{d,e}. \tag{2}$$

Inspired by this formula, we define the partial discriminant as follows.

Definition 5: In a network, a flow F for sink r is defined as a union of h edge-disjoint paths from $\mathcal{E}_s$ to sink r. Let $\mathcal{C}_F \subseteq \mathcal{C}$ be the set of all multiple-cycles which have no common edge with F. The partial discriminant of a CN code $\mathcal{K} = (k_{d,e})$ with respect to flow F is defined as

$$\delta_F(\mathcal{K}) := 1 + \sum_{C\in\mathcal{C}_F} \mathrm{sgn}(C) \prod_{(d,e)\in C} k_{d,e}.$$

Denote the set of all flows for a sink r by $\mathcal{F}_r$. The following theorem uses the concept of a partial discriminant to give a formula to check the multicast property of a CN code. The proof is given in the Appendix.

Theorem 1: A normal CN code $\mathcal{K} = (k_{d,e})$ on a network is multicast if and only if for each sink r

$$\sum_{F\in\mathcal{F}_r} \delta_F(\mathcal{K}) \prod_{(d,e)\in F} k_{d,e} \neq 0.$$

IV. FINDING THE LEKs OF A MULTICAST CN CODE

In this section, Theorem 1 is used to present an algorithm for finding the LEKs of a multicast CN code. This algorithm is based on LEKs and does not use GEKs. In the following, first the algorithm is described, and then its complexity is determined and compared with GEK-based algorithms. Finally, an example is given for this algorithm.

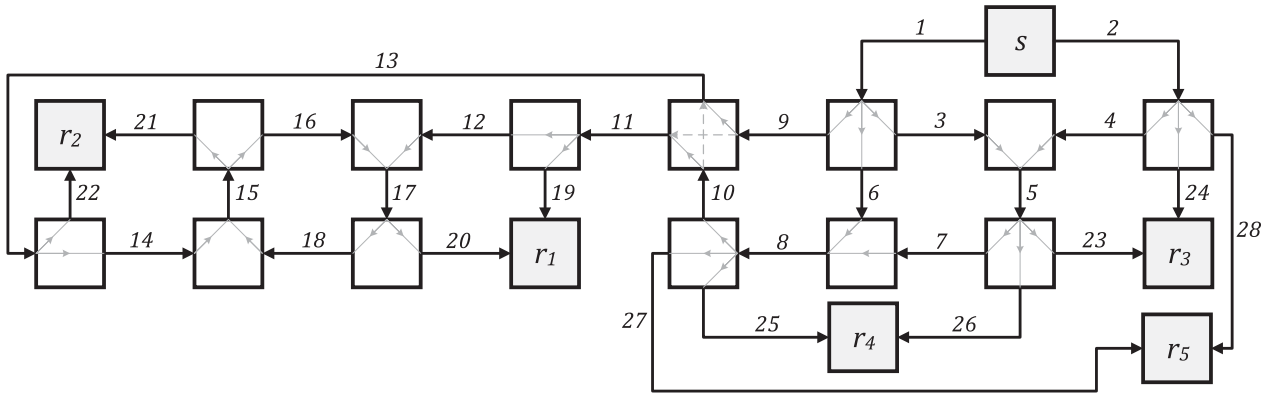

FIGURE 2. A multicast cyclic network with five sinks $r_1, \ldots, r_5$ and max-flow two.

Algorithm 1 Algorithm for Finding the LEKs of a Multicast CN Code

1) Find a flow $F_{r,0}$ for each sink r.
2) For each sink r, form $\hat{\mathcal{F}}(r)$ and for each $F \in \hat{\mathcal{F}}(r)$, set $\Pi_F = \text{sgn}(F)$.
3) Assign zero to the adjacent pairs not in $\hat{\mathcal{N}}$, consider an order on the edges, and for each edge e, do
   3-1) set $\sigma_r = 0$ for sink $r \in \hat{\mathcal{R}}$,
   3-2) for each adjacent pair $(d, e) \in \mathcal{A}^e$,
      3-2-1) for each sink $r \in \hat{\mathcal{R}}_{d,e}$, the impermissible value of $k_{d,e}$ is $-\sigma_r/(\sum_{F\in\hat{\mathcal{F}}_{r,d,e}} \Pi_F)$,
      3-2-2) assign a value to $k_{d,e}$ other than an impermissible value, and for each sink $r \in \hat{\mathcal{R}}_{d,e}$, set $\sigma_r = \sigma_r + k_{d,e} \sum_{F\in\hat{\mathcal{F}}_{r,d,e}} \Pi_F$,
      3-2-3) for each $r \in \hat{\mathcal{R}}$ and $F \in \hat{\mathcal{F}}_{r,d,e}$, set $\Pi_F = k_{d,e}\Pi_F$.

A. ALGORITHM FOR FINDING LEKs

The algorithm consists of three steps. In the first step, a flow $F_{r,0}$ is found to each sink r. The union of these flows is called a *flow path graph* and denoted by $\hat{\mathcal{N}}$. In fact, $\hat{\mathcal{N}}$ is constructed by eliminating the nodes, edges and adjacent pairs from $\mathcal{N}$ that do not participate in any flow. We assign zero to the adjacent pairs not in $\hat{\mathcal{N}}$.

There may exist more than one flow in $\hat{\mathcal{N}}$ to each sink. In the second step, these other flows are found and the following set is formed for each sink r

$$\hat{\mathcal{F}}_r := \{F : F \in \mathcal{F}_r,\ F \subset \hat{\mathcal{N}}\} \cup \{F \cup C : F \in \mathcal{F}_r,\ C \in \mathcal{C}_F,\ F \cup C \subset \hat{\mathcal{N}}\}.$$

In fact, each $F \in \hat{\mathcal{F}}_r$ is a flow for sink r or the union of a flow F' for sink r and a multiple-cycle C such that C and F' are edge-disjoint. Note that the flows and multiple-cycles must be in $\hat{\mathcal{N}}$. The sign of F is defined as $\text{sgn}(F) := \text{sgn}(C)$. As a special case, if F is a flow, then the sign of F is defined as $\text{sgn}(F) := +1$. Using this notation, the condition in Theorem 1 can be written as

$$\sum_{F\in\hat{\mathcal{F}}_r} \text{sgn}(F) \prod_{(d,e)\in F} k_{d,e} \neq 0. \tag{3}$$

In the third step, LEKs for a code are found such that (3) is satisfied for every sink. For this, we first consider an order on the edges such that the last edges are the sink inputs and then an iterative process is employed according to the ordering of the edges. At the iteration associated with edge e, we find suitable LEKs for the adjacent pairs in $\mathcal{A}^e := \{(d, e) : d \in \mathcal{E} \cup \mathcal{E}_s\}$.

In the following, we describe the method for finding these LEKs. At the iteration associated with edge e, the LEKs of adjacent pairs in $\mathcal{A}^{e'}$ for every $e' < e$ have been determined in previous iterations. Let $\hat{\mathcal{R}}$ be the set of all sinks with at least two flows in $\hat{\mathcal{N}}$. For each $F \in \hat{\mathcal{F}}_r$, let $\Pi_F = \text{sgn}(F) \prod_{(d,e')\in F, e'<e} k_{d,e'}$, and for each sink $r \in \hat{\mathcal{R}}$ and adjacent pair $(d, e) \in \mathcal{A}^e$, let $\hat{\mathcal{F}}_{r,d,e}$ be the set of elements of $\hat{\mathcal{F}}_r$ that contain adjacent pair (d, e) and their indeterminant LEKs are the same as for flow $F_{r,0}$. To obtain the LEKs of a multicast CN code, it is sufficient that the LEKs of adjacent pairs in $\mathcal{A}^e$ are chosen such that for each sink $r \in \hat{\mathcal{R}}$

$$\sum_{(d,e)\in\mathcal{A}^e} \left(k_{d,e} \sum_{F\in\hat{\mathcal{F}}_{r,d,e}} \Pi_F \right) \neq 0. \tag{4}$$

These LEKs are chosen according to an ordering on the adjacent pairs in $\mathcal{A}^j$ defined as $(d, e) < (d', e)$ if $d < d'$. Let $\hat{\mathcal{R}}_{d,e}$ be the set of all sinks $r \in \hat{\mathcal{R}}$ for which $\hat{\mathcal{F}}_{r,d,e} \neq \emptyset$ and $\hat{\mathcal{F}}_{r,d',e} = \emptyset$ for every $d' > d$. To find a suitable value for $k_{d,e}$, first the values of $k_{d,e}$ are found that do not satisfy condition (4) for each sink $r \in \hat{\mathcal{R}}_{d,e}$. These are called the impermissible values of $k_{d,e}$. Then, an arbitrary value other

than an impermissible value is assigned to $k_{d,e}$. Note that because of causality, the LEK of at least one adjacent pair on each cycle in $\hat{\mathcal{N}}$ must be divisible by D. This algorithm is summarized in Algorithm 1.

The following theorem gives an upper bound on the required alphabet size to ensure the existence of a multicast CN code on a *unit-delay* network. In a unit-delay network, a symbol is transmitted on every channel with a transmission delay of exactly one time unit.

Theorem 2: For a unit-delay network, there is a multicast CN code on $\mathbb{F}[(D)]$ if

$$|\mathbb{F}| > \min\{|\hat{\mathcal{R}}| + 1, |\mathcal{R}|\}.$$

Proof: For each sink in $\hat{\mathcal{R}}$, there is at most one value that violates condition (4), and for each sink not in $\hat{\mathcal{R}}$, the only value that can violate this condition is zero. Thus when $\hat{\mathcal{R}} \subset \mathcal{R}$, we can find LEKs from $\mathbb{F}[(D)]$ if $|\mathbb{F}| > |\hat{\mathcal{R}}| + 1$, and when $\hat{\mathcal{R}} = \mathcal{R}$ we can find LEKs from $\mathbb{F}[(D)]$ if $|\mathbb{F}| > |\hat{\mathcal{R}}| = |\mathcal{R}|$. Therefore, LEKs can be found from $\mathbb{F}[(D)]$ if $|\mathbb{F}| > \min\{|\hat{\mathcal{R}}| + 1, |\mathcal{R}|\}$. □

B. ALGORITHM VERIFICATION

Consider a causal CN code for which the LEKs of adjacent pairs not in $\hat{\mathcal{N}}$ are zero. From Theorem 1, this code is multicast if and only if $\sum_{F\in\mathcal{F}_r} \delta_F(\mathcal{K}) \prod_{(d,e)\in F} k_{d,e} \neq 0$ for each sink r. When there is exactly one flow $F_{r,0}$ to sink r in $\hat{\mathcal{N}}$, this condition is reduced to $\delta_{F_{r,0}}(\mathcal{K}) \prod_{(d,e)\in F_{r,0}} k_{d,e} \neq 0$, so it is not necessary to check the multicast property for this sink if the LEKs of the adjacent pairs in $F_{r,0}$ are not zero. Hence, we focus only on the sinks with at least two flows, i.e. sinks in $\hat{\mathcal{R}}$.

In the following, we show that at the end of the algorithm, condition (3) holds for all sinks. Let $x_{d,e}$ be the indeterminant LEK associated with adjacent pair (d, e). Before the iteration associated with edge e, condition (3) for each sink r is a multivariable polynomial equation

$$\sum_{F\in\hat{\mathcal{F}}_r} \left(\prod_{(d,e')\in F,\ e'<e} k_{d,e'}\right)\left(\prod_{(d,e')\in F,\ e'\geq e} x_{d,e'}\right) = \sum_{F\in\hat{\mathcal{F}}_r} \Pi_F \left(\prod_{(d,e')\in F,\ e'\geq e} x_{d,e'}\right) \neq 0. \quad (5)$$

To obtain the LEKs of a multicast CN code, it is sufficient to assign $k_{d,e}$ to each $(d, e) \in \mathcal{A}^e$ such that (5) is nonzero for each sink $r \in \hat{\mathcal{R}}$. The sum of the terms in (5) divisible by $\prod_{(d,e')\in F_{r,0},\ e'>e} x_{d,e}$ forms the polynomial

$$\left(\prod_{(d,e')\in F,\ e'>e} x_{d,e'}\right) \sum_{(d,e)\in\mathcal{A}^e} \left(x_{d,e} \sum_{F\in\hat{\mathcal{F}}_{r,d,e}} \Pi_F\right). \quad (6)$$

Clearly if (6) is nonzero, then (5) is also nonzero, so we assign an LEK $k_{d,e}$ to each $(d, e) \in \mathcal{A}^e$ such that $\sum_{(d,e)\in\mathcal{A}^e} \left(k_{d,e} \sum_{F\in\hat{\mathcal{F}}_{r,d,e}} \Pi_F\right) \neq 0$ for each sink $r \in \hat{\mathcal{R}}$. This condition is the same as condition (4) in the algorithm. Hence, at the end of the algorithm, condition (3) holds for all sinks, and so the code obtained is multicast.

C. COMPLEXITY ANALYSIS

Let $\hat{\mathcal{F}} := \bigcup_{r\in\mathcal{R}} \hat{\mathcal{F}}_r$. The following theorem provides the time complexity of the proposed algorithm.

Theorem 3: For a given network, the time complexity of finding the LEKs for a multicast CN code is $O\left(|\hat{\mathcal{F}}||\mathcal{E}| + h|\mathcal{R}||\mathcal{E}|\right)$.

Proof: In step 1, a flow path graph for a network is found in $O(h|\mathcal{R}||\mathcal{E}|)$ time. In step 2, all flows in the flow path graph are found in $O(|\hat{\mathcal{F}}||\mathcal{E}|)$ time. In step 3, for each edge e, in the process of assigning LEKs to adjacent pairs in $\mathcal{A}^e$, condition (4) is checked at most once for each sink $r \in \hat{\mathcal{R}}$, and each check operation takes at most $O(|\hat{\mathcal{F}}_r|)$ time. Thus, for each edge e, steps 3-2-1 and 3-2-2 take at most $O(|\hat{\mathcal{F}}|)$ time. Further, for each edge e and sink r, the process of updating the parameter Π_F takes at most $O(|\hat{\mathcal{F}}_r|)$ time, so for each edge e, step 3-2-3 takes at most $O(|\hat{\mathcal{F}}|)$ time. The total time complexity of the proposed algorithm is then

$$O\left(h|\mathcal{R}||\mathcal{E}| + |\hat{\mathcal{F}}||\mathcal{E}| + |\hat{\mathcal{F}}||\mathcal{E}|\right) = O\left(|\hat{\mathcal{F}}||\mathcal{E}| + h|\mathcal{R}||\mathcal{E}|\right).$$

□

In the following, the proposed algorithm is compared with GEK-based algorithms for designing multicast CN codes in networks with cycles.

1) COMPARISON WITH THE LIFE ALGORITHM

For acyclic networks, the best existing algorithm for designing multicast linear codes is the LIF algorithm for which the time complexity is $O(|\mathcal{E}||\mathcal{R}|h^2)$ [10]. For cyclic networks without knots, the LIF algorithm was generalized to the LIFE algorithm which has the same time complexity as the LIF algorithm [11]. A knot is a special collection of cycles defined in [11]. In comparison, our algorithm can be applied for networks with knots. Further, considering $\mu = |\hat{\mathcal{F}}|/|\mathcal{R}|$, our algorithm is more efficient than the LIFE algorithm when μ is low and h is high.

2) COMPARISON WITH THE ALGORITHM IN [1]

The first polynomial time algorithm for designing multicast CN codes over cyclic networks was presented in [1]. This algorithm uses GEKs to check the multicast property. It updates the GEKs after finding the LEKs associated with each edge. This update process is time consuming and results in a high complexity algorithm. The time complexity of the algorithm is $O(|\mathcal{R}|^3|\mathcal{E}|^{\omega+2})$ where $2 \leq \omega < 2.73$. Because of the power of $|\mathcal{E}|$ in the time complexity, our algorithm is more efficient than the algorithm presented in [1] when the number of edges is high, i.e. the network is large. The key reason for this advantage is that our algorithm uses LEKs, but the algorithm in [1] uses GEKs to check the multicast property, and updating GEKs is a time consuming process.

3) COMPARISON WITH THE DECYCLING METHOD

The decycling method [2] is another polynomial time algorithm for designing multicast CN codes over cyclic networks. This method first associates every cyclic network with a four layer acyclic network with max-flow $|\mathcal{E}|$ such that every multicast linear network code on the acyclic network induces a multicast linear network code on the cyclic network. Then existing algorithms for acyclic networks are used to design a multicast linear code for the acyclic network. Thus with this method, $|\mathcal{E}|$-dimensional GEKs are used to check the multicast property. This check process is time consuming for large networks because the dimension of the GEKs is high. The time complexity of this algorithm is $O(|\mathcal{R}||\mathcal{E}|^3)$. Because of the power of $|\mathcal{E}|$ in the time complexity, our algorithm is more efficient than the decycling method when the number of edges is high, i.e. the network is large. The main reason for this advantage is that our algorithm uses LEKs, while the decycling method uses GEKs with high dimension to check the multicast property.

D. FINDING LEKs FOR AN EXAMPLE NETWORK

In this subsection, Algorithm 1 is used to find the LEKs of a multicast CN code for the network shown in Figure 2. In this figure, each square is a node, each directed edge connecting two squares is a channel, and each directed edge in a square is an adjacent pair. For this network, each flow is the union of two edge-disjoint paths. We consider the following flows to construct a flow path graph

$$\begin{aligned}
F_{r_1,0} &= \{(1,9),(9,13),(13,14),(14,15),(15,16),\\
&\qquad (16,17),(17,20)\}\\
&\cup \{(2,4),(4,5),(5,7),(7,8),(8,10),(10,11),\\
&\qquad (11,19)\},\\
F_{r_2,0} &= \{(1,9),(9,13),(13,22)\}\\
&\cup \{(2,4),(4,5),(5,7),(7,8),(8,10),(10,11),\\
&\qquad (11,12),(12,17),(17,18),(18,15),(15,21)\},\\
F_{r_3,0} &= \{(1,3),(3,5),(5,23)\}\cup\{(2,24)\},\\
F_{r_4,0} &= \{(1,6),(6,8),(8,25)\}\\
&\cup \{(2,4),(4,5),(5,26)\},\\
F_{r_5,0} &= \{(1,6),(6,8),(8,27)\}\cup\{(2,28)\}.
\end{aligned}$$

The flow path graph contains all adjacent pairs except those denoted by the dashed lines in the figure. In this flow path graph, there is exactly one flow to every sink except r_5, so we have $\hat{\mathcal{R}}=\{r_5\}$ and the multicast property is checked only for sink r_5. Sink r_5 has a flow different than flow $F_{r_5,0}$ which is $F_{r_5,1}=\{(1,3),(3,5),(5,7),(7,8),(8,27)\}\cup\{(2,28)\}$. There is only one multiple-cycle in the flow path graph which is $C=\{(15,16),(16,17),(17,18),(18,15)\}$, so we have

$$\hat{\mathcal{F}}_{r_5}=\{F_{r_5,0},F_{r_5,1},F_{r_5,2}=C\cup F_{r_5,0},F_{r_5,3}=C\cup F_{r_5,1}\}.$$

We assume that the network is unit-delay. In the following, the LEKs are obtained for a multicast CN code on $\mathbb{F}_3[(D)]$, where $\mathbb{F}_3=\{0,1,2\}$.

We start from edge 3 because $\mathcal{A}^1=\mathcal{A}^2=\emptyset$. As $|\mathcal{A}^e|=1$ for $e=3,4$, we can assign any nonzero value to the adjacent pairs in these sets, so set $k_{1,3}=k_{2,4}=D$. For $\mathcal{A}^5$, because $\hat{\mathcal{F}}_{r_5,3,5}=\hat{\mathcal{F}}_{r_5,4,5}=\emptyset$, we can assign any nonzero value to the adjacent pairs in this set, so set $k_{3,5}=k_{4,5}=D$. Because $|\mathcal{A}^e|=1$ for $e=6,7$, we can assign any nonzero value to the adjacent pairs in these sets, so set $k_{1,6}=k_{5,7}=D$.

For $\mathcal{A}^8$, we have $\hat{\mathcal{F}}_{r_5,6,8}=\{F_{r_5,0}\}$ and $\hat{\mathcal{F}}_{r_5,7,8}=\{F_{r_5,1}\}$. As $\hat{\mathcal{R}}_{6,8}=\emptyset$, we can assign any nonzero value to $k_{6,8}$, so set $k_{6,8}=D$. Because $\hat{\mathcal{R}}_{7,8}=\{r_5\}$, we obtain the impermissible value of $k_{7,8}$ associated with sink r_5. This value is $-\Pi_{F_{r_5,0}}/\Pi_{F_{r_5,1}}=-D^2/D^3=-1/D$, so we can set $k_{6,8}=D$. For $e=9,\ldots,14$, because the adjacent pairs in $\mathcal{A}^e$ do not belong to any elements of $\hat{\mathcal{F}}_{r_5}$, we can assign any nonzero LEK to these adjacent pairs, so assign D to these adjacent pairs.

For $\mathcal{A}^{15}$, we have $\hat{\mathcal{F}}_{r_5,14,15}=\emptyset$ and $\hat{\mathcal{F}}_{r_5,18,15}=\{F_{r_5,2},F_{r_5,3}\}$. As $\hat{\mathcal{R}}_{14,15}=\emptyset$, we can assign any nonzero value to $k_{14,15}$, so set $k_{14,15}=D$. Because $\hat{\mathcal{R}}_{18,15}=\{r_5\}$, we obtain the impermissible value of $k_{18,15}$ associated with sink r_5. This value is

$$\begin{aligned}
&-(\Pi_{F_{r_5,0}}+\Pi_{F_{r_5,1}})/(\Pi_{F_{r_5,2}}+\Pi_{F_{r_5,3}})\\
&\quad=-(D^2+D^4)/(-D^2-D^4)=1,
\end{aligned}$$

so we can set $k_{18,15}=D$. As $|\mathcal{A}^{16}|=1$, we can assign any nonzero value to the adjacent pairs, so set $k_{15,16}=D$.

For $\mathcal{A}^{17}$, we have $\hat{\mathcal{F}}_{r_5,12,17}=\emptyset$ and $\hat{\mathcal{F}}_{r_5,16,17}=\{F_{r_5,2},F_{r_5,3}\}$. As $\hat{\mathcal{R}}_{12,17}=\emptyset$, we can assign any nonzero value to $k_{12,17}$, so set $k_{12,17}=D$. Because $\hat{\mathcal{R}}_{16,17}=\{r_5\}$, we require the impermissible value of $k_{16,17}$ associated with sink r_5. This value is

$$\begin{aligned}
&-(\Pi_{F_{r_5,0}}+\Pi_{F_{r_5,1}})/(\Pi_{F_{r_5,2}}+\Pi_{F_{r_5,3}})\\
&\quad=-(D^2+D^4)/(-D^4-D^6)=1/D^2,
\end{aligned}$$

so we can set $k_{16,17}=D$. For $e=18,\ldots,26$, because the adjacent pairs in $\mathcal{A}^e$ do not belong to any elements of $\hat{\mathcal{F}}_{r_5}$, we can assign any nonzero LEK to these adjacent pairs, so we assign D to these adjacent pairs. Because $|\mathcal{A}^e|=1$ for $e=27,28$, we can assign any nonzero value to the adjacent pairs in these sets, so set $k_{8,27}=k_{2,28}=D$. In summary, D is assigned to all LEKs, so this code can also be designed on the binary field.

V. CONCLUSION

In this paper, the concept of multiple-cycles was introduced to develop a formula to check the multicast property of CN codes. This formula is based on LEKs and does not use GEKs. It was used to develop an algorithm for obtaining the LEKs of a causal multicast CN code with time complexity $O\left(|\hat{\mathcal{F}}||\mathcal{E}|+h|\mathcal{R}||\mathcal{E}|\right)$, where h is the max-flow, $\mathcal{E}$ is the set of edges, $\mathcal{R}$ is the set of sinks and each element of $\hat{\mathcal{F}}$ is a flow to a sink or the union of a multiple-cycle and a flow to a sink such that the flow and the multiple-cycle are edge-disjoint. This algorithm is based on LEKs and does not use

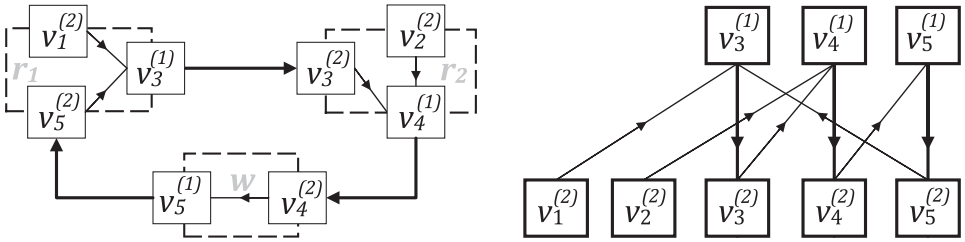

FIGURE 3. A multicast network and its associated bipartite graph.

GEKs. Further, it was shown that this algorithm is more efficient than GEK-based algorithms, particularly for large cyclic networks.

APPENDIX
PROOF OF THEOREM 1

In this appendix, we first convert a network into a bipartite graph and then introduce a relation between the multiple-cycles and flows of the network and complete matchings of the bipartite graph. Finally, this relation is used to prove Theorem 1.

A network $\mathcal{N}$ can be converted into a bipartite graph $\mathcal{N}_B := (V^{(1)} \cup V^{(2)}, E)$ with two parts $V^{(1)}$ and $V^{(2)}$ and edge set E. This bipartite graph is constructed as follows.

1) Corresponding to each edge $e \in \mathcal{E}_s$ in $\mathcal{N}$, there is a vertex $v_e^{(2)} \in V^{(2)}$.
2) Corresponding to each edge $e \in \mathcal{E}$ in $\mathcal{N}$, there are two vertices $v_e^{(1)} \in V^{(1)}$ and $v_e^{(2)} \in V^{(2)}$ and an edge $\overline{ee} \in E$ from $v_e^{(1)}$ to $v_e^{(2)}$.
3) Corresponding to each adjacent pair (d, e), there is an edge $\overline{de} \in E$ from $v_d^{(2)}$ to $v_e^{(1)}$.

Thus for a network $\mathcal{N}$, a bipartite graph $\mathcal{N}_B$ is constructed by adding edge $\overline{de}$ to the network corresponding to each adjacent pair (d, e). For example, the graph on the left in Figure 3 is the bipartite graph of the network in Figure 1. The graph on the right illustrates the two parts of the bipartite graph.

A. RELATIONSHIP BETWEEN COMPLETE MATCHINGS AND MULTIPLE-CYCLES AND FLOWS

A *complete matching* is a matching that covers all vertices in one part of a given bipartite graph. A subset of $V^{(2)}$ covered by a complete matching is called a *transversal*. The set of matchings that cover transversal T is denoted by $\mathcal{M}[T]$. Due to the structure of bipartite graph $\mathcal{N}_B$, the set $M_0 := \{\overline{ee} | e \in \mathcal{E}\}$ is a complete matching for $\mathcal{N}_B$ that covers transversal $\{v_e^{(2)} | e \in \mathcal{E}\}$.

For each sink r, let $\bar{\mathcal{F}}_r := \mathcal{F}_r \cup \{F \cup C : F \in \mathcal{F}_r, C \in \mathcal{C}$ such that C and F are edge disjoint$\}$, where $\mathcal{F}_r$ is the set of all flows for sink r. In fact, each element of $\bar{\mathcal{F}}_r$ is a flow for r or the union of a flow F for sink r and a multiple-cycle C such that C and F are edge-disjoint. The following lemma gives the relationship between $\bar{\mathcal{F}}_r$ and the set of complete matchings that cover transversal $T_r := \{v_e^{(2)} : e \notin \text{In}(r)\}$.

Lemma 1: For each sink r, the function $\mu : \mathcal{M}[T_r] \to \bar{\mathcal{F}}_r$, $\mu(M) = \{(d, e) : \overline{de} \in M \backslash M_0\}$ is bijective.

Proof: If $M \in \mathcal{M}[T_r]$, then from [13], $M \Delta M_0 := M \cup M_0 - M \cap M_0$ is a set of edge-disjoint M_0-alternating cycles and paths and vice versa where the paths are from $\{v_e^{(2)} : e \in \mathcal{E}_s\}$ to $\{v_e^{(2)} : e \in \text{In}(r)\}$. An M_0-alternating cycle (or path) is a cycle (or path) whose edges belong alternatively to matching M_0 and not to M_0. Due to the structure of bipartite graph $\mathcal{N}_B$, $M \Delta M_0$ is a set of edge-disjoint M_0-alternating cycles and paths from $\{v_e^{(2)} : e \in \mathcal{E}_s\}$ to $\{v_e^{(2)} : e \in \text{In}(r)\}$ in $\mathcal{N}_B$ if and only if the set $\{(d, e) : \overline{de} \in M \backslash M_0\}$ is a flow for sink r or the union of a flow for sink r and a multiple-cycle. □

For each nontrivial matching M, the sign function of M is defined as $\text{sgn}(M) := (-1)^{|M - M_0| - \sigma_M}$ where σ_M is the number of edge-disjoint paths and cycles of $M \Delta M_0$. Let $\text{sgn}(M) = C$, then $\text{sgn}(M) = (-1)^{|M - M_0| - \sigma_M} = (-1)^{|C| - \sigma_M} = (-1)^{|C|}\text{sgn}(C)$.

B. PROOF OF THEOREM 1 USING $\mathcal{N}_B$

For a given CN code $\mathcal{K} = (k_{d,e})$, define the $|\mathcal{E}| \times (|\mathcal{E}| + h)$ matrix $K' = [k'_{d,e}]$ as

$$K' := \begin{bmatrix} [k_{d,e}]_{d \in \mathcal{E}_s, e \in \mathcal{E}} \\ I_{|\mathcal{E}|} - [k_{d,e}]_{d \in \mathcal{E}, e \in \mathcal{E}} \end{bmatrix}^T = \left[[k_{d,e}]^T_{d \in \mathcal{E}_s, e \in \mathcal{E}} \mid (I_{|\mathcal{E}|} - [k_{d,e}]_{d,e \in \mathcal{E}})^T \right]. \quad (7)$$

This is the incidence matrix of $\mathcal{N}_B$ if the LEKs are replaced by one. Using this fact and the Leibniz formula [14], we have

$$\det(K'[T]) = \sum_{M \in \mathcal{M}[T]} \text{sgn}(M) \prod_{\overline{de} \in M} k'_{e,d}, \quad (8)$$

where $K'[T]$ is the $|\mathcal{E}| \times |\mathcal{E}|$ matrix formed from the columns of K' corresponding to transversal T. We now use Lemma 1 to prove Theorem 1.

Proof: Let $A := I_{|\mathcal{E}|} - [k_{d,e}]_{d,e \in \mathcal{E}}$ and $B := [k_{d,e}]_{d \in \mathcal{E}_s, e \in \mathcal{E}}$. From (1), we have $[f_e]_{e \in \mathcal{E}_s \cup \mathcal{E}} = [I_{|\mathcal{E}|} | BA^{-1}]$. Since A is invertible, we have $\left[(BA^{-1})^T \mid I_{|\mathcal{E}|}\right] = \left[(A^{-1})^T B^T \mid I_{|\mathcal{E}|}\right] = (A^{-1})^T [B^T \mid A^T] = (A^{-1})^T K'$, so from the *duality theorem for vector matroids* [15], the columns of $[f]_{\text{In}(r)}$ are linearly independent if and only if the determinant of $K'[T_r]$ is nonzero. The determinant of $K'[T_r]$

is given by

$$\begin{aligned}
\det(K'[T_r]) &\overset{(a)}{=} \sum_{M\in\mathcal{M}[T_r]} \operatorname{sgn}(M) \prod_{\overline{de}\in M} k'_{e,d} \\
&\overset{(b)}{=} \sum_{M\in\mathcal{M}[T_r]} \operatorname{sgn}(M) \prod_{\overline{de}\in M\setminus M_0} k'_{e,d} \\
&\overset{(c)}{=} \sum_{F\in\bar{\mathcal{F}}_r} (-1)^{|F|}\operatorname{sgn}(F) \prod_{(d,e)\in F} k'_{e,d} \\
&\overset{(d)}{=} \sum_{F\in\bar{\mathcal{F}}_r} \operatorname{sgn}(F) \prod_{\substack{(d,e)\in F\\ d\in\mathcal{E}_s}} -k_{d,e} \prod_{\substack{(d,e)\in F\\ d\in\mathcal{E}}} \\
&= (-1)^h \sum_{F\in\bar{\mathcal{F}}_r} \operatorname{sgn}(F) \prod_{(d,e)\in F} k_{d,e} \\
&\overset{(e)}{=} (-1)^{h+1} \sum_{F\in\mathcal{F}_r} ((1+\sum_{C\in\mathcal{C}_F} \operatorname{sgn}(C) \\
&\quad \times \prod_{(d,e)\in C} k_{d,e}) \prod_{(d,e)\in F} k_{d,e}) \\
&= (-1)^{h+1} \sum_{F\in\mathcal{F}_r} \delta_F \prod_{(d,e)\in F} k_{d,e}.
\end{aligned}$$

Equality (a) is a consequence of (8). From (7), $k'_{e,e} = 1$ for every $e \in \mathcal{E}$, and hence equality (b) holds. Equality (c) is a consequence of Lemma 1 and equality (d) holds from (7). Finally, equality (e) holds by factoring the LEKs of adjacent pairs in flows. □

REFERENCES

[1] E. Erez and M. Feder, "Efficient network code design for cyclic networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3862–3878, Aug. 2010.
[2] S.-Y. R. Li and Q. T. Sun, "Network coding theory via commutative algebra," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 403–415, Jan. 2011.
[3] X. Zhao and W. Guo, "Equivalent conditions to determine the GEKs by the LEKs in a convolutional network code over a cyclic network," *IEICE Trans. Fundam.*, vol. E95–A, no. 9, pp. 1570–1576, Sep. 2012.
[4] X. Zhao, "An efficient basic convolutional network code construction algorithm on cyclic networks," *AEU-Int. J. Electron. Commun.*, vol. 67, no. 12, pp. 1072–1078, Dec. 2013.
[5] Q. T. Sun and S.-Y. R. Li, "On decoding of DVR-based linear network codes," *Appl. Algebra Eng. Commun. Comput.*, vol. 26, no. 6, pp. 527–542, Dec. 2015.
[6] V. Samadi-Khaftari, M. Esmaeili, and T. A. Gulliver, "Some connections between classical coding and network coding over erroneous cyclic networks," *IEEE Access*, vol. 4, pp. 5889–5895, Sep. 2016.
[7] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
[8] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
[9] C. Fragouli and E. Soljanin, "A connection between network coding and convolutional codes," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, Jun. 2004, pp. 661–666.
[10] S. Jaggi *et al.*, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1983, Jun. 2005.
[11] A. I. Barbero and O. Ytrehus, "Cycle-logical treatment for 'cyclopathic' networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2795–2804, Jun. 2006.
[12] S.-Y. R. Li and R. W. Yeung, "On convolutional network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, USA, Jul. 2006, pp. 1743–1747.
[13] D. B. West, *Introduction to Graph Theory*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[14] K. Janich, *Linear Algebra*. New York, NY, USA: Springer-Verlag, 1994.
[15] J. G. Oxley, *Matroid Theory*. New York, NY, USA: Oxford Univ. Press, 1992.

MORTEZA REKAB-ESLAMI received the B.E. degree in applied mathematics from Shahid Chamran University of Ahvaz, Ahvaz, Iran, in 2008, and the M.Sc. degree in applied mathematics from the Amirkabir University of Technology, Tehran, Iran, in 2010. He is currently pursuing the Ph.D. degree at Isfahan University of Technology, Isfahan, Iran. His research interests include network coding, channel coding, and matroid theory.

MORTEZA ESMAEILI received the M.S. degree in mathematics from the Teacher Training University of Tehran, Iran, in 1988, and the Ph.D. degree in mathematics (coding theory) from Carleton University, Ottawa, Canada, in 1996. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada, for two years. Since 1998, he has been with the Department of Mathematical Sciences, Isfahan University of Technology, Isfahan, Iran, where he is currently a Professor. He joined the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada, as an Adjunct Professor in 2009. His current research interests include coding and information theory, cryptography, and combinatorics and its application to communication theory.

THOMAS AARON GULLIVER received the Ph.D. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 1989. From 1989 to 1991, he was a Defence Scientist with Defence Research Establishment Ottawa, Ottawa, ON, Canada. He has held academic appointments at Carleton University, Ottawa, and the University of Canterbury, Christchurch, New Zealand. He joined the University of Victoria in 1999, where he is currently a Professor with the Department of Electrical and Computer Engineering. His research interests include information theory and communication theory, algebraic coding theory, multicarrier systems, smart grid, and security. In 2002, he became a fellow of the Engineering Institute of Canada. In 2012, he became a fellow of the Canadian Academy of Engineering. From 2000 to 2003, he was the Secretary and a member of the Board of Governors of the IEEE Information Theory Society. He is currently an Area Editor of the IEEE Transactions on Wireless Communications. His research interests include information theory and communication theory, algebraic coding theory, multicarrier systems, smart grid, and security.