

Convolutional Network Codes for Cyclic Networks

Elona Erez and Meir Feder

Dept. of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv, 69978, Israel, E-mail: {elona, meir}@eng.tau.ac.il

I. INTRODUCTION

Most data networks contain cycles, but so far most attention in the literature of network coding has been addressed to multicast in acyclic networks. The original paper on network coding [1] did consider cyclic networks but there it was suggested to transform the cyclic network into an acyclic network using the idea of unrolling the network into a layered network. This approach has many drawbacks: it achieves the optimal rate only asymptotically, it leads to time-variant schemes, it has high encoding and decoding complexities and it induces large delay. In [2] it was shown that if each edge in the network has delay, then there exists a time-invariant linear network code that achieves the optimal rate. This approach may, again, introduce large delay since each edge has delay and an efficient construction algorithm is not given. In [3] a heuristic code construction is given for a linear time-invariant code, but the construction is not given explicitly and it is not necessarily efficient.

In this work we give an explicit polynomial time code construction of an optimal multicast linear network code for cyclic networks. As it turns out, it is not necessary for every edge in the network to have delay, as long as we ensure that in each cycle in the network at least one edge has delay. Since delay elements are anyway inserted into the network it seems more natural to focus on convolutional codes for cyclic networks. Nevertheless, our results in this paper are directly applicable for block codes.

II. NOTATIONS AND PRECODING

Consider a cyclic, unit capacity network $G = (V, E)$ where parallel edges are allowed. There is a single source node s and a set of d sinks $T = \{t_1, \dots, t_d\}$. Denote by h the size of the minimal individual min-cut between s and any of the sinks. Let $F(D)$ denote the ring of polynomials over the binary field with variable D . The variable D is a unit time shift. For convenience, as in [6], we add a dummy source s' connected to source s with h edges $\{e_1^0, \dots, e_h^0\}$.

Similarly to [2], we define the directed line graph of $G = (V, E)$ as $L(\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = E \cup s \cup s' \cup T \cup \{e_1^0, \dots, e_h^0\}$ and edge set $\mathcal{E} = \{(e, e') \in E^2 : \text{head}(e) = \text{tail}(e')\} \cup \{(s, e) : e \text{ outgoing from } s\} \cup \{(e, t_i) : e \text{ incoming to } t_i, 1 \leq i \leq d\} \cup \{(s', e_1^0), \dots, (s', e_h^0)\} \cup \{(e_1^0, s), \dots, (e_h^0, s)\}$. We consider in the rest of the paper the line graph $L(\mathcal{V}, \mathcal{E})$. We denote nodes of L as $e \in \mathcal{V}$, and the edges as $(e, e') \in \mathcal{E}$. If there are h edge-disjoint paths between s and t in G , there are corresponding h node-disjoint paths in L .

Node $e \in L$ is associated with an h -dimensional vector global coding vector $\mathbf{v}(e)$, with elements from $F(D)$. The set

of edges incoming into node e is $\Gamma_{in}(e)$ and $m(e_i, e)$ is the coding coefficient of edge (e_i, e) . Denote by $F[D]$ the field of rational functions over the binary field with variable D . The code can be used for multicasting from s to $T = \{t_1, \dots, t_d\}$ if and only if for all $t \in T$, the global coding vectors at the nodes incoming into t span the space $F[D]^h$.

It is assumed that prior to the code construction, all the coding coefficients in the network are set to zero. We have to choose a set of edges E_D in G , such that if we eliminate them from the network G there will be no directed cycles. The nodes corresponding to E_D in $L(\mathcal{V}, \mathcal{E})$ are denoted by \mathcal{E}_D . For edges outgoing from nodes not in \mathcal{E}_D we draw the coding coefficients among polynomials with degree at most M , for some M we will later determine. For edges outgoing from nodes in \mathcal{E}_D we draw the coding coefficients among polynomials with degree at most M , and multiply the result by D . Therefore the set of polynomials we can draw from has the same size for edges outgoing from either nodes in \mathcal{E}_D or nodes not in \mathcal{E}_D , but edges outgoing from nodes in \mathcal{E}_D always introduce at least a single delay. Thus we are guaranteed that each cycle contains at least a single delay.

In order to minimize the delay, it is desired to minimize $|\mathcal{E}_D|$, or equivalently $|E_D|$ in the original network G . Finding the minimal set E_D is essentially the known, long standing problem of finding the minimal arc feedback set. This problem is NP-hard [4]. The best known approximation algorithm with polynomial complexity achieves performance ratio $O(\log |V| \log \log |V|)$ [5]. For our purposes, we can use approximate solutions and insert enough delays in the cycles, before we begin our design algorithm.

III. CODE CONSTRUCTION

The code construction algorithm goes in steps over the terminals. In the l -th step of the algorithm we consider a subgraph L_l of L that consists only of the nodes and edges that participate in the flow from s to the sink t_l . Without loss of generality, we assume that L is given by:

$$L = \cup_{l=1, \dots, d} L_l \quad (1)$$

We can make this assumption, because if there were in the original line graph L nodes that do not participate in any of the flows, we can remove them from the network, and still achieve the same optimal rate. For each node $e \in L_l$ the algorithm will eventually define a coding coefficient $m(e, e') \in F(D)$ for the edge (e, e') that connects this node to the node e' that follows it in the flow. At the beginning of the l -step some of the edges may already have coding coefficients assigned at the previous steps. These coefficients are updated during the l -step, in a manner given in the following.

the stage. If U_l^n is not a basis, we have to change the old coefficient $m'(e_{i,l}, e_{i,l}^n)$ into another coefficient $m(e_{i,l}, e_{i,l}^n)$. Consider the following theorem we prove in Section IV:

Theorem 1: Suppose that with the coefficient $m'(e_{i,l}, e_{i,l}^n)$ the set U_l^n is not a basis. Then by changing it to any other value $m(e_{i,l}, e_{i,l}^n)$ the set U_l^n will be a basis.

Unfortunately, changing $m'(e_{i,l}, e_{i,l}^n)$ to an arbitrary value may affect the sinks treated in the previous steps of the algorithm. Thus, if the coding coefficient has to be replaced (i.e., when U_l^n is not a basis) before replacing it to a new value we need to analyze its effect on the other sinks. Specifically, let \mathcal{C}_k be the set of nodes incoming into the sink $t_k, k < l$. We have the following theorem:

Theorem 2: Denote by $V'_k = \{\mathbf{v}'(e_{1,k}), \dots, \mathbf{v}'(e_{h,k})\}$, $e_{j,k} \in \mathcal{C}_k$, the set of global coding vectors of the nodes incoming into t_k before changing $m'(e_{i,l}, e_{i,l}^n)$ to $m(e_{i,l}, e_{i,l}^n)$. If V'_k is a basis, then after the replacement at most a single value of $m(e_{i,l}, e_{i,l}^n)$ will cause the new set of global coding vectors $V_k = \{\mathbf{v}(e_{1,k}), \dots, \mathbf{v}(e_{h,k})\}$ not to be a basis.

Based on these theorems, the procedure for replacing $m'(e_{i,l}, e_{i,l}^n)$ is as follows. If $m'(e_{i,l}, e_{i,l}^n)$ must be replaced i.e., U_l^n is not a basis, we pick a new value $m(e_{i,l}, e_{i,l}^n)$ according to some enumeration. We then check, using the procedure given in the proof of Theorem 2 below, if the independence condition is satisfied for all sinks. If the condition is not satisfied for all sinks, we pick the next coefficient $m(e_{i,l}, e_{i,l}^n)$ in the enumeration. Since for each sink only a single choice of $m(e_{i,l}, e_{i,l}^n)$ is bad, if we have more than d coefficients to choose from, we are guaranteed to have at least a single choice which is good for all previous sinks.

The l -step of the algorithm continues until it reaches the sink t_l , and the algorithm terminates when it goes over all d sinks.

A flow chart of the algorithm is given in Figure 3.

IV. PROOFS OF THEOREMS AND LEMMAS

A. Lemma 1

Lemma 1: Let $\{e_1, \dots, e_h\}$ be a set of nodes and let $W = \{\mathbf{w}(e_1), \dots, \mathbf{w}(e_i), \dots, \mathbf{w}(e_h)\}$, be their coding vectors, which may be partial or global coding vectors. Pick i and consider the coding vectors of the same set of nodes $\tilde{W} = \{\tilde{\mathbf{w}}(e_1), \dots, \tilde{\mathbf{w}}(e_i), \dots, \tilde{\mathbf{w}}(e_h)\}$, where we set $m(e_i, e) = 0$ for $\forall e \in L$. The set W is a basis if and only if the set \tilde{W} is a basis.

We start with the "if" direction. We find the relation between \tilde{W} and W . The only difference is that in the definition of \tilde{W} $m(e_i, e) = 0$ for $\forall e \in L \setminus e_i$. Suppose that $m(e_i, e), \forall e \in L \setminus e_i$ are now set to their true values. Since the code is linear, the effect of the network on the coding vector of e_i is a linear system. We split node e_i into 3 nodes: e_{tail} , e_{mid} and e_{head} , which are connected by edges (e_{tail}, e_{mid}) and (e_{mid}, e_{head}) . We can find a rational function G_{ee} with a variable D which represents the transfer function of the linear system from node e_{head} to node e_{tail} in the network $L \setminus e_{mid}$. The rational function G_{ee} contains the factor D since each cycle must contain the factor D in at least one of its edges, as explained

in Section II. The resulting vector when $m(e_i, e), \forall e \in L \setminus e_i$ are set to their true values is:

$$\mathbf{w}(e_i) = \tilde{\mathbf{w}}(e_i) + G_{ee} \tilde{\mathbf{w}}(e_i) + G_{ee}^2 \tilde{\mathbf{w}}(e_i) \dots = \frac{1}{1 - G_{ee}} \tilde{\mathbf{w}}(e_i) \quad (5)$$

The factor $1 - G_{ee}$ never vanishes since G_{ee} includes the factor D . The other vectors are given by:

$$\mathbf{w}(e_j) = \tilde{\mathbf{w}}(e_j) + F_{ij} \frac{1}{1 - G_{ee}} \tilde{\mathbf{w}}(e_i), j \neq i \quad (6)$$

where F_{ij} is the transfer function from e_i to e_j . The vectors in W can represent rows of some matrix A . Likewise, the vectors in \tilde{W} can represent rows of some matrix \tilde{A} . The matrices A and \tilde{A} have the same rank, since A can be reached from \tilde{A} by multiplication of a row by a non-zero factor and adding a multiplication of this row to the other rows. Therefore if A is full rank, as implied by the conditions of the lemma, so is A .

For the opposite direction, the relation between $\tilde{\mathbf{w}}(e_i)$ and $\mathbf{w}(e_i)$ is

$$\tilde{\mathbf{w}}(e_i) = (1 - G_{ee})\mathbf{w}(e_i) \quad (7)$$

The other vectors are given by:

$$\tilde{\mathbf{w}}(e_j) = \mathbf{w}(e_j) - F_{ij} \mathbf{w}(e_i) = \mathbf{w}(e_j) - F_{ij} \frac{1}{1 - G_{ee}} \tilde{\mathbf{w}}(e_i), j \neq i \quad (8)$$

Similarly to the above, the matrices A and \tilde{A} have the same rank, since \tilde{A} can be reached from A by multiplication of a row by a non-zero factor and subtracting a multiplication of this row from the other rows. Therefore if A is a basis so is the set \tilde{A} .

B. Proof of Theorem 1

Let $\tilde{U}_l^n = \{\tilde{\mathbf{u}}^n(e_{1,l}), \dots, \tilde{\mathbf{u}}^n(e_{i,l}^n), \dots, \tilde{\mathbf{u}}^n(e_{h,l})\}$ denote the coding vectors of \mathcal{C}_l^n when all the coefficients in r_l are set to zero. That is, unlike in U_l^n , in the definition of \tilde{U}_l^n the coefficients of the edges outgoing from $e_{i,l}^n$ are set to zero, as well as the coefficients in r_l^n . In the definition of \tilde{U}_l^n , it is assumed that the new value $m(e_{i,l}, e_{i,l}^n)$ is used, while we denote by $\tilde{U}_l^n = \{\tilde{\mathbf{u}}^n(e_{1,l}), \dots, \tilde{\mathbf{u}}^n(e_{i,l}^n), \dots, \tilde{\mathbf{u}}^n(e_{h,l})\}$ the corresponding set with $m'(e_{i,l}, e_{i,l}^n)$. Assume that $U_l = \{\mathbf{u}(e_{1,l}), \dots, \mathbf{u}(e_{i,l}), \dots, \mathbf{u}(e_{h,l})\}$ is a basis, and recall that U_l is also defined when all the coefficients in r_l are zero. From these definitions we have:

$$\tilde{\mathbf{u}}^n(e_{i,l}^n) = \tilde{\mathbf{u}}^n(e_{i,l}^n) + (m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))\mathbf{u}(e_{i,l}) \quad (9)$$

The vector $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ is the coding vector of $e_{i,l}^n$ when the coefficients in r_l are set to zero, before the replacement of $m'(e_{i,l}, e_{i,l}^n)$. The vector $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ may include contributions from all edges incoming into $e_{i,l}^n$. In (9) the vector $\mathbf{u}(e_{i,l})$ is independent of $m'(e_{i,l}, e_{i,l}^n)$ or $m(e_{i,l}, e_{i,l}^n)$ since there is no feedback from $e_{i,l}^n$ to $e_{i,l}$ when the coefficients $m(e_{i,l}^n, e)$ are set to zero $\forall e \in L \setminus e_{i,l}^n$.

We assumed that the previous set of partial coding vectors $U_l = \{\mathbf{u}(e_{1,l}), \dots, \mathbf{u}(e_{i,l}), \dots, \mathbf{u}(e_{h,l})\}$ is a basis. We want to show that the set $\tilde{U}_l^n = \{\tilde{\mathbf{u}}^n(e_{1,l}), \dots, \tilde{\mathbf{u}}^n(e_{i,l}^n), \dots, \tilde{\mathbf{u}}^n(e_{h,l})\}$ is also a basis. We have the relation $U_l \setminus \mathbf{u}(e_{i,l}) = \tilde{U}_l^n \setminus \tilde{\mathbf{u}}^n(e_{i,l}^n)$

since both sets are defined with the coefficients in r_l set to zero. It follows that the vectors in the set $\tilde{U}_l^n \setminus \tilde{\mathbf{u}}^n(e_{i,l}^n) = \{\tilde{\mathbf{u}}^n(e_{1,l}), \dots, \tilde{\mathbf{u}}^n(e_{i-1,l}), \tilde{\mathbf{u}}^n(e_{i+1,l}), \dots, \tilde{\mathbf{u}}^n(e_{h,l})\}$ are independent. It remains to show that $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ is independent of $\tilde{U}_l^n \setminus \tilde{\mathbf{u}}^n(e_{i,l}^n)$. The vector $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ in (9) is dependent on $\tilde{U}_l^n \setminus \tilde{\mathbf{u}}^n(e_{i,l}^n)$ because otherwise according to Lemma 1 the set U_l^n with $m'(e_{i,l}, e_{i,l}^n)$ would be a basis and we would not have to change $m'(e_{i,l}, e_{i,l}^n)$ to another $m(e_{i,l}, e_{i,l}^n)$. Trivially, it follows that the vector $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ is dependent on $U_l \setminus \mathbf{u}(e_{i,l})$. If the new $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ depends on $\tilde{U}_l^n \setminus \tilde{\mathbf{u}}^n(e_{i,l}^n) = U_l \setminus \mathbf{u}(e_{i,l})$ then:

$$\begin{aligned} \tilde{\mathbf{u}}^n(e_{i,l}^n) &= \tilde{\mathbf{u}}^n(e_{i,l}^n) + (m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))\mathbf{u}(e_{i,l}) \\ &= \alpha_1\mathbf{u}(e_{1,l}) + \dots + \alpha_{i-1}\mathbf{u}(e_{i-1,l}) \\ &\quad + \alpha_{i+1}\mathbf{u}(e_{i+1,l}) + \dots + \alpha_h\mathbf{u}(e_{h,l}) \end{aligned} \quad (10)$$

or

$$\begin{aligned} \tilde{\mathbf{u}}^n(e_{i,l}^n) &= \alpha_1\mathbf{u}(e_{1,l}) + \dots + \alpha_{i-1}\mathbf{u}(e_{i-1,l}) \\ &\quad + (m'(e_{i,l}, e_{i,l}^n) - m(e_{i,l}, e_{i,l}^n))\mathbf{u}(e_{i,l}) \\ &\quad + \alpha_{i+1}\mathbf{u}(e_{i+1,l}) + \dots + \alpha_h\mathbf{u}(e_{h,l}) \end{aligned} \quad (11)$$

Since U_l is a basis and since $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ is dependent on $U_l \setminus \mathbf{u}(e_{i,l})$, (11) can be maintained only if $m(e_{i,l}, e_{i,l}^n) = m'(e_{i,l}, e_{i,l}^n)$. Therefore, for any other choice of $m(e_{i,l}, e_{i,l}^n)$ the vector $\tilde{\mathbf{u}}^n(e_{i,l}^n)$ is independent of $\tilde{U}_l^n \setminus \tilde{\mathbf{u}}^n(e_{i,l}^n)$. Thus \tilde{U}_l^n is a basis. It follows from Lemma 1 that since \tilde{U}_l^n is a basis, the set U_l^n is a basis for any $m(e_{i,l}, e_{i,l}^n) \neq m'(e_{i,l}, e_{i,l}^n)$.

C. Proof of Theorem 2

Suppose that $V'_k = \{\mathbf{v}'(e_{1,k}), \dots, \mathbf{v}'(e_{h,k})\}$, defined at the Theorem, is a basis. We want to analyze under which conditions the set of global coding vectors $V_k = \{\mathbf{v}(e_{1,k}), \dots, \mathbf{v}(e_{h,k})\}$, obtained after replacing $m'(e_{i,l}, e_{i,l}^n)$ to $m(e_{i,l}, e_{i,l}^n)$, is also basis.

We divide the sinks into two types: (a) sinks that are at the head of $e_{i,l}^n$ and (b) other sinks. We have divided the sinks into two types since sinks of type (a) are simpler to analyze. In this case $e_{i,l}^n \in \mathcal{C}_k$ since $e_{i,l}^n$ is incoming into t_k and without loss of generality, assume $e_{i,l}^n = e_{i,k}$. According to Lemma 1 it suffices to show that the new set of global coding vectors of \mathcal{C}_k are a basis when for all edges outgoing from $e_{i,l}^n = e_{i,k}$, the coefficients $m(e_{i,l}^n, e)$, $\forall e \in L \setminus e_{i,l}^n$ are set to zero. Denote as $\tilde{V}'_k = \{\tilde{\mathbf{v}}'(e_{1,k}), \dots, \tilde{\mathbf{v}}'(e_{h,k})\}$ the coding vectors of \mathcal{C}_k before the replacement of $m'(e_{i,l}, e_{i,l}^n)$ when the coefficients $m(e_{i,l}^n, e) = 0 \forall e \in L \setminus e_{i,l}^n$ (and all the rest of the coefficients in the network have their true current value). Denote as $\tilde{V}_k = \{\tilde{\mathbf{v}}(e_{1,k}), \dots, \tilde{\mathbf{v}}(e_{h,k})\}$ the coding vectors of \mathcal{C}_k after the replacement to $m(e_{i,l}, e_{i,l}^n)$ when the coefficients $m(e_{i,l}^n, e) = 0, \forall e \in L \setminus e_{i,l}^n$. For $j \neq i$ we have $\tilde{\mathbf{v}}'(e_{j,k}) = \tilde{\mathbf{v}}(e_{j,k})$ since when the coefficients $m(e_{i,l}^n, e) = 0, \forall e \in L \setminus e_{i,l}^n$ the coding vectors of the other edges in the network do not depend on $m(e_{i,l}, e_{i,l}^n)$ or $m'(e_{i,l}, e_{i,l}^n)$. Similarly, the coding vector of $e_{i,l}$ when the coefficients $m(e_{i,l}^n, e) = 0 \forall e \in L \setminus e_{i,l}^n$ does not depend on $m(e_{i,l}, e_{i,l}^n)$ or $m'(e_{i,l}, e_{i,l}^n)$ and is denoted

as $\tilde{\mathbf{v}}(e_{i,l})$. When $m'(e_{i,l}, e_{i,l}^n)$ is replaced by $m(e_{i,l}, e_{i,l}^n)$, the new vector $\tilde{\mathbf{v}}(e_{i,l}^n)$ is related to the old vector $\tilde{\mathbf{v}}'(e_{i,l}^n)$ as:

$$\tilde{\mathbf{v}}(e_{i,l}^n) = \tilde{\mathbf{v}}'(e_{i,l}^n) + (m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))\tilde{\mathbf{v}}(e_{i,l}) \quad (12)$$

or equivalently,

$$\tilde{\mathbf{v}}(e_{i,k}) = \tilde{\mathbf{v}}'(e_{i,k}) + (m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))\tilde{\mathbf{v}}(e_{i,l}) \quad (13)$$

We know that before changing $m'(e_{i,l}, e_{i,l}^n)$ to $m(e_{i,l}, e_{i,l}^n)$ the set of vectors $V'_k = \{\mathbf{v}'(e_{1,k}), \dots, \mathbf{v}'(e_{h,k})\}$ was a basis. Therefore according to Lemma 1 (the "only if" direction), the set $\tilde{V}'_k = \{\tilde{\mathbf{v}}'(e_{1,k}), \dots, \tilde{\mathbf{v}}'(e_{i,k}), \dots, \tilde{\mathbf{v}}'(e_{h,k})\}$ is also a basis. It follows trivially that the set of vectors $\{\tilde{\mathbf{v}}(e_{1,k}), \dots, \tilde{\mathbf{v}}(e_{i-1,k}), \tilde{\mathbf{v}}'(e_{i,k}), \tilde{\mathbf{v}}(e_{i+1,k}), \dots, \tilde{\mathbf{v}}(e_{h,k})\}$ is a basis and it remains to show for which $m(e_{i,l}, e_{i,l}^n) \neq m'(e_{i,l}, e_{i,l}^n)$, we can replace $\tilde{\mathbf{v}}'(e_{i,k})$ by $\tilde{\mathbf{v}}(e_{i,k})$ and still have a basis. Suppose that $\tilde{\mathbf{v}}(e_{i,k})$ is dependent on $\{\tilde{\mathbf{v}}(e_{1,k}), \dots, \tilde{\mathbf{v}}(e_{i-1,k}), \tilde{\mathbf{v}}(e_{i+1,k}), \dots, \tilde{\mathbf{v}}(e_{h,k})\}$, then:

$$\begin{aligned} \tilde{\mathbf{v}}(e_{i,k}) &= \tilde{\mathbf{v}}'(e_{i,k}) + (m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))\tilde{\mathbf{v}}(e_{i,l}) \\ &= \alpha_1\tilde{\mathbf{v}}(e_{1,k}) + \dots + \alpha_{i-1}\tilde{\mathbf{v}}(e_{i-1,k}) + \alpha_{i+1}\tilde{\mathbf{v}}(e_{i+1,k}) \\ &\quad + \dots + \alpha_h\tilde{\mathbf{v}}(e_{h,k}) \end{aligned} \quad (14)$$

We can divide by $m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n)$ since $m(e_{i,l}, e_{i,l}^n) \neq m'(e_{i,l}, e_{i,l}^n)$,

$$\begin{aligned} \tilde{\mathbf{v}}(e_{i,l}) &= \frac{1}{m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n)}(\alpha_1\tilde{\mathbf{v}}(e_{1,k}) + \dots \\ &\quad + \alpha_{i-1}\tilde{\mathbf{v}}(e_{i-1,k}) - \tilde{\mathbf{v}}'(e_{i,k}) + \alpha_{i+1}\tilde{\mathbf{v}}(e_{i+1,k}) \\ &\quad + \dots + \alpha_h\tilde{\mathbf{v}}(e_{h,k})) \end{aligned} \quad (15)$$

Suppose that the representation of $\tilde{\mathbf{v}}(e_{i,l})$ in the basis $\{\tilde{\mathbf{v}}(e_{1,k}), \dots, \tilde{\mathbf{v}}(e_{i-1,k}), \tilde{\mathbf{v}}'(e_{i,k}), \tilde{\mathbf{v}}(e_{i+1,k}), \dots, \tilde{\mathbf{v}}(e_{h,k})\}$ is:

$$\begin{aligned} \tilde{\mathbf{v}}(e_{i,l}) &= \beta_1\tilde{\mathbf{v}}(e_{1,k}) + \dots + \beta_{i-1}\tilde{\mathbf{v}}(e_{i-1,k}) + \beta_i\tilde{\mathbf{v}}'(e_{i,k}) \\ &\quad + \beta_{i+1}\tilde{\mathbf{v}}(e_{i+1,k}) + \dots + \beta_h\tilde{\mathbf{v}}(e_{h,k}) \end{aligned} \quad (16)$$

Then in order for (15) to be maintain it is required that:

$$-\frac{1}{m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n)} = \beta_i \quad (17)$$

If $\beta_i = 0$, no choice of $m(e_{i,l}, e_{i,l}^n)$ will satisfy this relation. If $\beta_i \neq 0$, there is a single solution of (17)

$$m(e_{i,l}, e_{i,l}^n) = m'(e_{i,l}, e_{i,l}^n) - \frac{1}{\beta_i} \quad (18)$$

Therefore at most a single choice of $m(e_{i,l}, e_{i,l}^n)$ might cause the set \tilde{V}_k not to be a basis. Note that such a choice, $m'(e_{i,l}, e_{i,l}^n) - \frac{1}{\beta_i}$ must also be a polynomial in the set we are using for the code. According to Lemma 1 if \tilde{V}_k is a basis then V_k is also a basis.

We turn now our attention to sinks of type (b). Assume that the edges outgoing from $e_{i,l}$, except $(e_{i,l}, e_{i,l}^n)$, are $\Gamma_O = \{(e_{i,l}, e_1), \dots, (e_{i,l}, e_q)\}$. The system G_{ee} defined in Section IV-A can be expressed as $G_{ee} = G_1 + m(e_{i,l}, e_{i,l}^n)G_2$, where G_1 is a transfer function, defined as G_{ee} but in the network $L \setminus (e_{i,l}, e_{i,l}^n)$ and $m(e_{i,l}, e_{i,l}^n)G_2$ is the transfer function defined in $G \setminus \Gamma_O$. The vector $\tilde{\mathbf{v}}(e_{i,l})$ is the coding vector of $e_{i,l}$ when

the coefficients of the edges outgoing from $e_{i,l}$ are all zero. Then for an arbitrary coefficient $m(e_{i,l}, e_{i,l}^n)$ the global coding vector of $e_{i,l}$ is $\mathbf{v}(e_{i,l})$ and is given by:

$$\begin{aligned} \mathbf{v}(e_{i,l}) &= \tilde{\mathbf{v}}(e_{i,l}) + G_{ee}\tilde{\mathbf{v}}(e_{i,l}) + \dots = \frac{1}{1 - G_{ee}}\tilde{\mathbf{v}}(e_{i,l}) \\ &= \frac{1}{1 - G_1 - m(e_{i,l}, e_{i,l}^n)G_2}\tilde{\mathbf{v}}(e_{i,l}) \\ &= \frac{1}{(1 - G_1)\left(1 - \frac{m(e_{i,l}, e_{i,l}^n)G_2}{1 - G_1}\right)}\tilde{\mathbf{v}}(e_{i,l}) \end{aligned} \quad (19)$$

where we can divide by $1 - G_1$ since G_1 always contains the factor D . Define as $\mathbf{y}(e_{i,l})$ the coding vector of $e_{i,l}$ when $m(e_{i,l}, e_{i,l}^n) = 0$. From (19) we observe that when $m(e_{i,l}, e_{i,l}^n) = 0$, $\mathbf{v}(e_{i,l}) = \mathbf{y}(e_{i,l}) = \tilde{\mathbf{v}}(e_{i,l})/(1 - G_1)$. Thus,

$$\mathbf{v}(e_{i,l}) = \frac{1}{1 - m(e_{i,l}, e_{i,l}^n)Q}\mathbf{y}(e_{i,l}) \quad (20)$$

where $Q = G_2/(1 - G_1)$. Note that since G_1 and G_2 both contain the term D , so does Q . Similarly, for coefficient $m'(e_{i,l}, e_{i,l}^n)$ we have,

$$\mathbf{v}'(e_{i,l}) = \frac{1}{1 - m'(e_{i,l}, e_{i,l}^n)Q}\mathbf{y}(e_{i,l}) \quad (21)$$

The difference between the two vectors is:

$$\begin{aligned} \mathbf{v}(e_{i,l}) - \mathbf{v}'(e_{i,l}) &= \left(\frac{1}{1 - m(e_{i,l}, e_{i,l}^n)Q} - \frac{1}{1 - m'(e_{i,l}, e_{i,l}^n)Q} \right) \mathbf{y}(e_{i,l}) \\ &= \left(\frac{m(e_{i,l}, e_{i,l}^n)Q}{1 - m(e_{i,l}, e_{i,l}^n)Q} - \frac{m'(e_{i,l}, e_{i,l}^n)Q}{1 - m'(e_{i,l}, e_{i,l}^n)Q} \right) \mathbf{y}(e_{i,l}) \\ &= \frac{(m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))Q}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} \mathbf{y}(e_{i,l}) \\ &\triangleq f(m(e_{i,l}, e_{i,l}^n)) \cdot Q \cdot \mathbf{y}(e_{i,l}) \end{aligned} \quad (22)$$

We note that $f(m(e_{i,l}, e_{i,l}^n))$ does not diverge to infinity since $m(e_{i,l}, e_{i,l}^n)Q$ (and $m'(e_{i,l}, e_{i,l}^n)Q$) contains D as a factor and thus $(1 - m(e_{i,l}, e_{i,l}^n)Q)$ never vanishes. Since the linear network code is equivalent to a linear system operating on the coding vectors, the new set of vectors at the input of another sink t_k satisfies:

$$\begin{aligned} \mathbf{v}(e_{j,k}) - \mathbf{v}'(e_{j,k}) &= (m(e_{i,l}, e_{i,l}^n)F_{1,j} + F_{2,j})\mathbf{v}(e_{i,l}) \\ &\quad - (m'(e_{i,l}, e_{i,l}^n)F_{1,j} + F_{2,j})\mathbf{v}'(e_{i,l}) \\ &= \left(\frac{m(e_{i,l}, e_{i,l}^n)F_{1,j} + F_{2,j}}{1 - m(e_{i,l}, e_{i,l}^n)Q} \right. \\ &\quad \left. - \frac{m'(e_{i,l}, e_{i,l}^n)F_{1,j} + F_{2,j}}{1 - m'(e_{i,l}, e_{i,l}^n)Q} \right) \mathbf{y}(e_{i,l}), \\ &\quad 1 \leq j \leq h \end{aligned} \quad (23)$$

The rational function $F_{1,j}$ is the transfer function from $e_{i,l}$ to $e_{j,k}$, when $m(e_{i,l}, e) = 0, \forall e \in \Gamma_O \setminus (e_{i,l}, e_{i,l}^n)$. The rational function $F_{2,j}$ is the transfer function from $e_{i,l}$ to $e_{j,k}$, when

only the coefficient $m(e_{i,l}, e_{i,l}^n) = 0$. Due to superposition in linear systems, the total transfer function from $e_{i,l}$ to $e_{j,k}$ before replacing $m'(e_{i,l}, e_{i,l}^n)$ is $F_j' = m'(e_{i,l}, e_{i,l}^n)F_{1,j} + F_{2,j}$ and after the replacement $F_j = m(e_{i,l}, e_{i,l}^n)F_{1,j} + F_{2,j}$. Rearranging terms in (23):

$$\begin{aligned} \mathbf{v}(e_{j,k}) - \mathbf{v}'(e_{j,k}) &= \left(\frac{m(e_{i,l}, e_{i,l}^n)F_{1,j}}{1 - m(e_{i,l}, e_{i,l}^n)Q} - \frac{m'(e_{i,l}, e_{i,l}^n)F_{1,j}}{1 - m'(e_{i,l}, e_{i,l}^n)Q} \right. \\ &\quad \left. + \frac{F_{2,j}}{1 - m(e_{i,l}, e_{i,l}^n)Q} - \frac{F_{2,j}}{1 - m'(e_{i,l}, e_{i,l}^n)Q} \right) \mathbf{y}(e_{i,l}) \\ &= \left(\frac{(m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))F_{1,j}}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} \right. \\ &\quad \left. + \frac{(m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))QF_{2,j}}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} \right) \mathbf{y}(e_{i,l}) \\ &= \left(\frac{(m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n))}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} \right) \\ &\quad (F_{1,j} + QF_{2,j})\mathbf{y}(e_{i,l}) \\ &= f(m(e_{i,l}, e_{i,l}^n)) \cdot H_j \cdot \mathbf{y}(e_{i,l}), 1 \leq j \leq h \end{aligned} \quad (24)$$

where $H_j \equiv F_{1,j} + QF_{2,j}$. We know that before changing $m'(e_{i,l}, e_{i,l}^n)$ to $m(e_{i,l}, e_{i,l}^n)$ the set of vectors $V_k' = \{\mathbf{v}'(e_{1,k}), \dots, \mathbf{v}'(e_{h,k})\}$ was a basis. We analyze under which conditions on $m(e_{i,l}, e_{i,l}^n)$ the new set of coding vectors $V_k = \{\mathbf{v}(e_{1,k}), \dots, \mathbf{v}(e_{h,k})\}$ is also a basis. Suppose the representation of $\mathbf{y}(e_{i,l})$ in basis V_k' is:

$$\mathbf{y}(e_{i,l}) = \beta_1\mathbf{v}'(e_{1,k}) + \beta_2\mathbf{v}'(e_{2,k}) + \dots + \beta_h\mathbf{v}'(e_{h,k}) \quad (25)$$

We examine independence of the vectors in V_k according to definition:

$$\alpha_1\mathbf{v}(e_{1,k}) + \dots + \alpha_h\mathbf{v}(e_{h,k}) = 0 \quad (26)$$

If this equation has a solution other than $\alpha_1 = \dots = \alpha_h = 0$, the vectors in V_k are dependent. Using (24):

$$\begin{aligned} \alpha_1 \{ \mathbf{v}'(e_{1,k}) + f(m(e_{i,l}, e_{i,l}^n)) \cdot H_1 \mathbf{y}(e_{i,l}) \} + \dots \\ + \alpha_h \{ \mathbf{v}'(e_{h,k}) + f(m(e_{i,l}, e_{i,l}^n)) \cdot H_h \mathbf{y}(e_{i,l}) \} = 0 \end{aligned} \quad (27)$$

Rearranging terms:

$$\begin{aligned} \mathbf{v}'(e_{1,k}) \{ \alpha_1 + \alpha_1 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_1 \beta_1 \\ + \alpha_2 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_2 \beta_1 + \dots \\ + \alpha_h f(m(e_{i,l}, e_{i,l}^n)) \cdot H_h \beta_1 \} \\ + \dots + \\ \mathbf{v}'(e_{h,k}) \{ \alpha_h + \alpha_1 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_1 \beta_h \\ + \alpha_2 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_2 \beta_h + \dots \\ + \alpha_h f(m(e_{i,l}, e_{i,l}^n)) \cdot H_h \beta_h \} = 0 \end{aligned} \quad (28)$$

Since V'_k is a basis it is required that:

$$\begin{aligned} \alpha_1 + \alpha_1 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_1 \beta_1 + \alpha_2 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_2 \beta_1 + \dots \\ + \alpha_h f(m(e_{i,l}, e_{i,l}^n)) \cdot H_h \beta_1 = 0 \\ \vdots \\ \alpha_h + \alpha_1 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_1 \beta_h + \alpha_2 f(m(e_{i,l}, e_{i,l}^n)) \cdot H_2 \beta_h + \dots \\ + \alpha_h f(m(e_{i,l}, e_{i,l}^n)) \cdot H_h \beta_h = 0 \end{aligned} \quad (29)$$

or in matrix notation:

$$-\frac{1}{f(m(e_{i,l}, e_{i,l}^n))} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_h \end{pmatrix} = A \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_h \end{pmatrix} \quad (30)$$

where

$$A = \begin{pmatrix} H_1 \beta_1 & H_2 \beta_1 & \cdots & H_h \beta_1 \\ H_1 \beta_2 & H_2 \beta_2 & \cdots & H_h \beta_2 \\ \vdots & \cdots & \ddots & \vdots \\ H_1 \beta_h & H_2 \beta_h & \cdots & H_h \beta_h \end{pmatrix} \quad (31)$$

We could have divided by $f(m(e_{i,l}, e_{i,l}^n))$ since $f(m(e_{i,l}, e_{i,l}^n)) = 0$ only for $m(e_{i,l}, e_{i,l}^n) = m'(e_{i,l}, e_{i,l}^n)$, but we examine the case where $m(e_{i,l}, e_{i,l}^n) \neq m'(e_{i,l}, e_{i,l}^n)$. We see that (30) can be maintained only if $\alpha_1 = \cdots = \alpha_h = 0$ or if A has eigenvalue λ such that:

$$\lambda = -\frac{1}{f(m(e_{i,l}, e_{i,l}^n))} \quad (32)$$

In the following we show that a matrix of the form of A has eigenvalue 0 with multitude $h - 1$ and eigenvalue:

$$\lambda = \text{trace}(A) = H_1 \beta_1 + H_2 \beta_2 + \cdots + H_h \beta_h \quad (33)$$

with multitude 1. The rank of the $h \times h$ matrix A is 1. Therefore, the solution of the equation $Av = 0$ has dimension $h - 1$. Thus the geometric multiplicity of the eigenvalue $\lambda = 0$ is $h - 1$. The algebraic multiplicity of an eigenvalue is not smaller than the geometric multiplicity [8, proposition(6.4.3)]. Therefore A has an eigenvalue 0 of algebraic multiplicity at least $h - 1$. Since the sum of the eigenvalues of a matrix equals its trace, A also has an eigenvalue $\text{trace}(A)$ of algebraic multiplicity 1.

According to (32) $\lambda \neq 0$ since $f(m(e_{i,l}, e_{i,l}^n))$ does not rise to infinity. Therefore, we have only to consider $\lambda = \text{trace}(A)$. Suppose,

$$\lambda = -\frac{1}{f(m(e_{i,l}, e_{i,l}^n))} = \text{trace}(A) \quad (34)$$

Then according to the definition of $f(m(e_{i,l}, e_{i,l}^n))$,

$$\begin{aligned} f(m(e_{i,l}, e_{i,l}^n)) &= \frac{m(e_{i,l}, e_{i,l}^n) - m'(e_{i,l}, e_{i,l}^n)}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} \\ &= -\frac{1}{\text{trace}(A)} \end{aligned} \quad (35)$$

where we could have divided by $\text{trace}(A)$, since we already showed that if $\lambda = \text{trace}(A) = 0$, then (32) cannot

be maintained. We could multiply by $f(m(e_{i,l}, e_{i,l}^n))$, since $f(m(e_{i,l}, e_{i,l}^n)) \neq 0$ if $m(e_{i,l}, e_{i,l}^n) \neq m'(e_{i,l}, e_{i,l}^n)$. Thus,

$$\begin{aligned} \frac{m(e_{i,l}, e_{i,l}^n)}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} \\ = \frac{m'(e_{i,l}, e_{i,l}^n)}{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)} - \frac{1}{\text{trace}(A)} \end{aligned} \quad (36)$$

Or,

$$\begin{aligned} m(e_{i,l}, e_{i,l}^n) &= \quad (37) \\ m'(e_{i,l}, e_{i,l}^n) &- \frac{(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)}{\text{trace}(A)} \end{aligned}$$

where we could multiply by $(1 - m(e_{i,l}, e_{i,l}^n)Q)(1 - m'(e_{i,l}, e_{i,l}^n)Q)$ since, as we discussed, it does not vanish to zero. Rearranging terms we set,

$$\begin{aligned} m(e_{i,l}, e_{i,l}^n) \left(1 - Q \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)} \right) \\ = m'(e_{i,l}, e_{i,l}^n) - \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)} \end{aligned} \quad (38)$$

To show that we can divide (38) by the term

$$1 - Q \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)} \quad (39)$$

we have to prove that it does not vanish to zero. Suppose that it does vanish to zero:

$$1 - Q \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)} = 0 \quad (40)$$

Then according to (38) it follows that,

$$m'(e_{i,l}, e_{i,l}^n) - \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)} = 0 \quad (41)$$

Therefore (40) becomes:

$$1 - Q \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)} = 1 - Q m'(e_{i,l}, e_{i,l}^n) = 0 \quad (42)$$

But if $1 - Q m'(e_{i,l}, e_{i,l}^n) = 0$, clearly (40) cannot be maintained. We conclude that the term in (39) does not vanish to zero and we can therefore divide (38) by this expression, which yields:

$$m(e_{i,l}, e_{i,l}^n) = \frac{m'(e_{i,l}, e_{i,l}^n) - \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)}}{1 - Q \frac{1 - m'(e_{i,l}, e_{i,l}^n)Q}{\text{trace}(A)}} \quad (43)$$

We conclude that for at most a single choice of $m(e_{i,l}, e_{i,l}^n)$, the one given in (43), the set V_k will not be a basis.

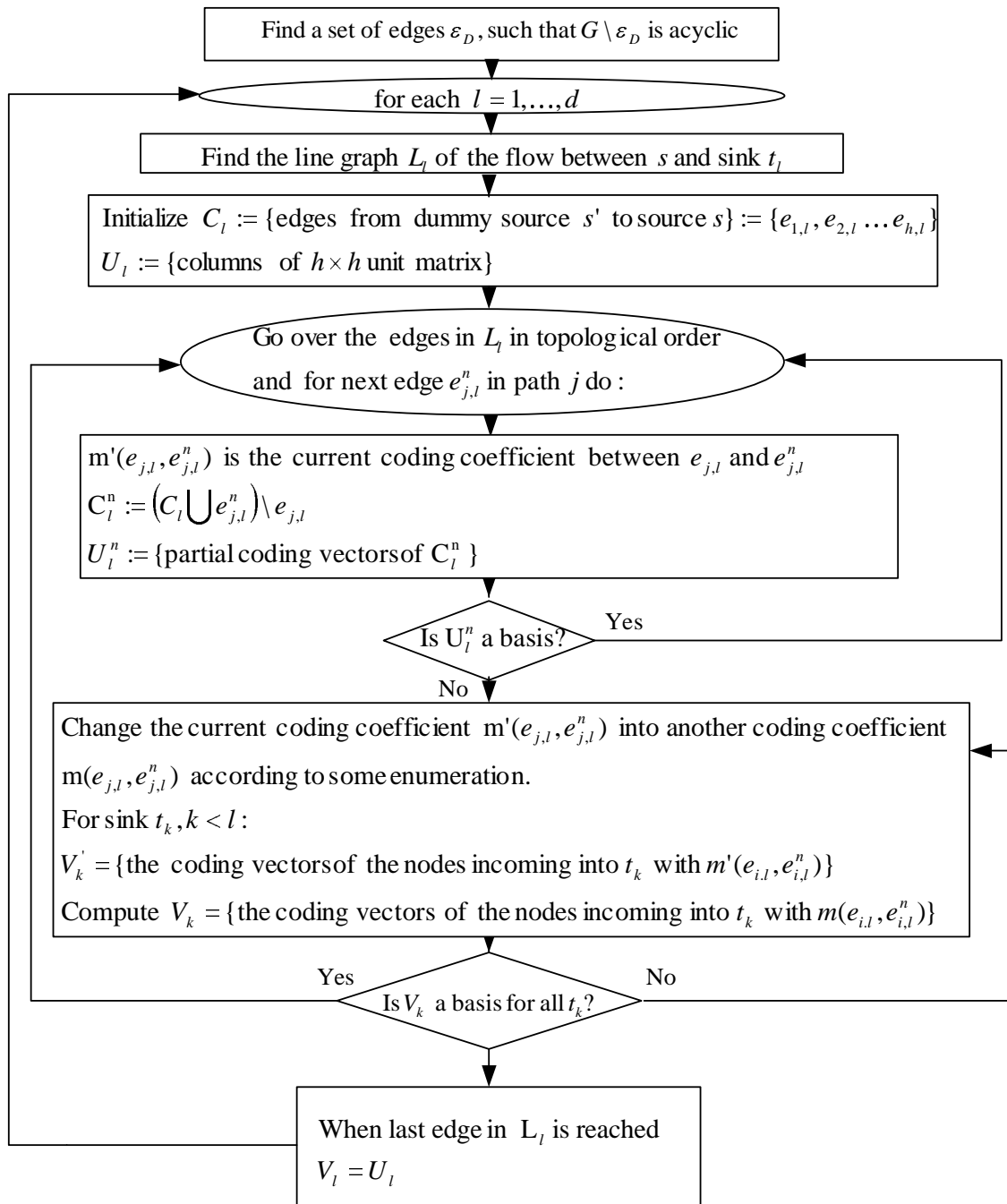


Fig. 3. Flow chart of algorithm

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R.W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.
- [2] R. Koetter and M. Médard, "An algebraic approach to network coding," *Proceedings of INFOCOM*, 2002.
- [3] S.Y.R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Inform. Theory*, vol. 49, pp. 371–381, 2003.
- [4] R.M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations*, pp. 85–104, 1972.
- [5] G. Even, J. Naor, B. Schieber, and M. Sudan, "Approximating minimum feedback sets and multicuts in directed graphs," *Algorithmica*, vol. 20, pp. 151–174, 1998.
- [6] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," 2003.
- [7] E. Erez and M. Feder, "Convolutional network codes," *IEEE International Symposium on Information Theory*, 2004.
- [8] Hans Schneider and George Philip Barker, *Matrices and linear algebra*, New York : Holt, Rinehart and Winston, 2nd edition, 1973.