Network Coding Theory Based on Commutative Algebra and Matroids

SUN, Qifu

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Information Engineering

July 2009

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstract

The fundamental result of linear network coding asserts the existence of optimal linear network codes over acyclic networks when the symbol field is sufficiently large. The restriction to just acyclic networks turns out to stem from the customary algebraic structure of data symbols as a finite field.

The first part of this thesis algebraically structures the ensemble of data units as a *principal ideal domain* (PID) instead of a finite field. Fundamental theory of linear algebra deals with vectors over a PID and leads to a theoretic generalization of the basic concepts of linear network coding from acyclic to *cyclic* networks. As a prerequisite for *causal* data propagation around a cycle via a PID-based network code, the code must be *normal* such that the data unit carried on every channel can be unambiguously identified. Moreover, in order to break the deadlock in cyclic transmission, the PID of data units is further restricted to be a *discrete valuation ring* (DVR), which possesses a unique maximal ideal.

The second part connects PID-based network coding with matroid theory, an abstract theory of dependence. Over a network, a *network matroid* is defined on the edge set via the independence structure of edge-disjoint paths. Meanwhile, the linear independence among coding vectors of a normal code naturally induces a matroid on the edge set. Every independent set in the matroid so induced is also independent in the network matroid. Moreover, the two matroids coincide with each other if and only if the normal code is a *generic* one. This shows the optimality of generic codes in terms of linear independence. On the other hand, when the network is acyclic, every representation for the network matroid forms a generic code.

The third part delves into the efficiency issue of code construction. Given a cyclic network, a quadratically large acyclic network is established by de-cycling such that every optimal code on the de-cycled network subject to some straightforward restriction directly induces an optimal code on the cyclic one. In this way, existing construction algorithms for optimal codes on acyclic networks can be adapted to cyclic networks as well.

摘要

綫性網絡編碼最基本的理論保證了當所傳輸的字符域足夠大時于單源組播無環通訊網絡中最優碼的存在。此理論對網絡無環的限制源自對數據字符基於有限域的慣例代數結構模型。

本論文的第一部分用*主理想整環*來代替每個整體數据單元有限域的代數結構。由於綫性代數的基礎理論研究的是基於主理想整環的向量，通過此种模型，綫性網絡編碼的基本概念可以被理論地拓展到*有環*網絡上。作爲數據在網絡環中有因果地通過主理想整環網絡碼傳送的先決條件，網絡碼必須為*標準碼*，即通過其可以明確地決定每一條信道上傳送的數據單元。此外，爲了打破數據繞環傳輸的死環，數據單元的主理想整環結構被進一步限制擁有一個單一的極大理想。在交換代數學中，此种主理想整環被稱爲*離散賦值環*。

本論文的第二部分將主理想整環網絡編碼與擬陣論—一套關於相關性的抽象理論—聯係起來。在一個可能有環存在的網絡中，通過分離鏈路的獨立性結構，一個*網絡擬陣*在邊集合上定義了出來。同時，每一個標準碼的編碼向量中的綫性獨立自然地推導出一個邊集合上的一個擬陣。此擬陣中的每一個獨立集均被證明為網絡擬陣中的一個獨立集。此外，當且僅當此標準碼為*共有碼*時，其推倒導的擬陣成爲一個網絡擬陣。此性質表明了共有碼在綫性獨立方面的最優性。在另一方面，當網絡中沒有環存在時，網絡擬陣的綫性表示可以看成爲共有碼的一种新特征概括。

本論文的第三部分探討高效創建最優碼的問題。給定一個任意網絡，一個二次方大的無環網絡被相對應地建立起來，以使得在無環網上每個受一簡單約束的最優碼均可以直接地引出在原有網絡上的最優碼。通過這種方式，現有的適用于無環網上構造最優碼的算法也可以被應用

於有環網絡上。此種統一的構造方法解決了在有環網絡中構造最優網絡編碼的一個長久難題。此方法以擬陣的對偶性理論為根據，並且能夠構造出不同種類的最優碼。

Acknowledgement

My foremost gratitude is to my supervisor Professor Shuo-Yen Robert Li, not only for his judicious academic instruction since the final year of my undergraduate study, but also for his kindly sharing of living experiences, from which I learnt invaluable knowledge for life. This thesis is the outcome of my collaboration with him. I benefited greatly from his patient guidance on every aspect of this thesis.

Moreover, I would like to thank The Chinese University of Hong Kong for providing me a precious opportunity to pursue both my bachelor and doctoral degrees. I would also appreciate Professor Sidharth Jaggi for his valuable advice on my research. Without the helpful discussion with him, part of this thesis may not be complete. I am grateful for the useful comments by Professor Raymond Wai-Ho Yeung and Mr. Siu Ting Ho as well.

Besides, I have worked in a harmonious laboratory, the Switching Laboratory. The excellent atmosphere here for studying, working, and living is created and maintained by my supervisor Prof. S. Y. R. Li together with my colleagues Dr. Jian Zhu, Dr. Xuesong Joahnathan Tan, Mr. Zhengfeng Qian, Mr. Siu-Ting Ho, Mr. Ziyu Shao, Mr. Zizhou Vincent Wang, Ms. Shuqin Li, Ms. Xiaoming Wu, Ms. Lok-Man Janice Law, Mr. Tong Liang, Mr. Qiwei Li, and Mr. Xiaoming Xiu.

During the past two years, Ms. Dazheng Zhang has supported me a lot in different aspects. Her considerate help and encouragement is a constant inspiration to me.

Last, but not least, I would dedicate this thesis to my dear parents. My eight year study in Hong Kong is ascribed to their constant support and understanding. I can hardly appreciate more for their selfless care and love.

This thesis is dedicated to my parents.

Contents

List of Figures ..... **ix**

List of Tables ..... **x**

Chapter 1. Introduction **1**

1.1 Network Coding Theory 1

1.2 Matroid Theory in Network Coding 5

1.3 Efficient Network Code Design 8

Chapter 2. Network Coding Theory via Commutative Algebra **10**

2.1 Normality of PID-Based Network Coding 11

2.2 Optimal Normal Network Codes 15

2.3 Causal Data Propagation by Network Coding 25

Chapter 3. Linear Network Codes and Matroids **30**

3.1 Basic Definition of Matroids 30

3.2 Network Matroids and Linear Network Codes 32

3.3 Optimality of Generic Linear Network Codes 34

Chapter 4. Efficient Construction of Optimal Network Codes **39**

4.1 Existing Algorithms 39

4.1.1 Acyclic Case 39

4.1.2 Cyclic Case 44

4.2 Algorithm Adaptation from Acyclic to Cyclic Networks 46

4.2.1 A General PID-based Theorem 46

4.2.2 Theorem Proof 48

4.2.3 Adaptation of Existing Algorithms 57

4.2.4 Analysis of Computational Complexity 60

4.3 Construction of Causal Network Codes 62

4.4 Matroid Duality and Theorem Generalization 69

Chapter 5. Summary and Future Work. 73

5.1 Summary 73

5.2 Future Work 76

Appendix A. Preliminaries on Commutative Algebra 77

Appendix B. Preliminaries on Matroids 88

Appendix C. An Example on Network De-cycling 93

Bibliography 95

List of Figures

Figure 1.1. The Butterfly Network 1

Figure 1.2. A finite-state linear time-invariant causal encoder 4

Figure 2.1. The Shuttle Network and a linear multicast on it 18

Figure 2.2. A linear network code that qualifies as a linear dispersion, a linear broadcast, and a linear multicast, but not a generic linear network code. 24

Figure 2.3. A causal linear network code that qualifies as a causal generic linear network code, a causal linear dispersion, a causal linear broadcast, and a causal linear multicast 28

Figure 3.1. A network on which there does not exist a linear network code that can induce the uniform matroid $U_{2,5}$ 32

Figure 3.2. An example to show that the linear network code induced from a representation for the network matroid of a cyclic network is not necessarily normal 38

Figure 4.1. A cyclic network and the associated layered acylic network 48

Figure 4.2. De-cycling process from a cyclic network to the layered acyclic network 51

Figure 4.3. A linear network code that is not normal qualifies as a delay invariant convolutional multicast. 67

Figure 4.4. A t-causal convolutional multicast $(k_{d,e} \cdot D^{t(d,e)})$ such that when the delay function t is changed, the updated convolutional network code $(k_{d,e} \cdot D^{t(d,e)})$ is no longer a convolutional multicast 68

Figure A.1. Relationships among the polynomial ring, the ring of formal power series, the quotient field and the ring of rational power series over a same finite field in indeterminate D 82

Figure B.1. The graphic matroid of a planar graph and its dual 92

List of Tables

Table 4.1. Computational complexities to construct a linear multicast..... 62

Chapter 1.

Introduction

1.1 Network Coding Theory

Network coding is one of the most important breakthroughs in communication engineering in this decade. Before its advent, *store-and-forward* had been the pivotal premise on the study of information flow in the communication network. It is adapted from the paradigm in commodity flow in a transportation network. It has been folklore in data networking that except for data replication, there is no need for data processing at intermediate nodes. However, network coding refutes it via a standard example, called the *Butterfly Network*, as depicted in Figure 1.1.

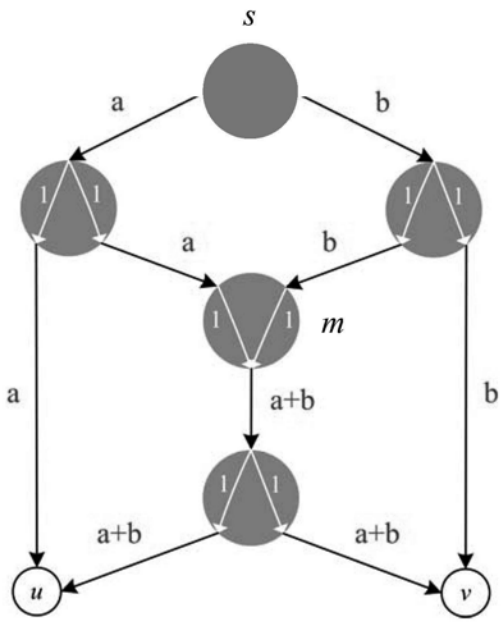


Figure 1.1 The *Butterfly Network* shows that the data processing at intermediate nodes helps to increase data transmission throughput. In the network, every directed edge represents a noiseless channel transmitting one bit per unit time, and the source node s is to send two bits a and b to both node u and v. In the store-and-forward fashion, the outgoing link of node m can only transmit either a or b at a time. However, if it transmits the summation $a+b$ instead, then upon receiving $a+b$, both u and v can respectively decode b and a whereas one timeslot transmission is saved.

The concept of network coding originated from both the nonlinear theory [1][2] and the linear theory [29][31]. The latter restricts the data processing at intermediate nodes to be a linear operation in a communication network. It, especially, carries wide applications to many established fields including coding theory, computer networks, computer science, distributed data storage, information security, information theory, optimization theory, peer-to-peer (P2P) content delivery, switching theory, and wireless/satellite communications. It models a communication network as a finite acyclic directed multi-graph with a single source node. An edge represents a noiseless communication channel of unit capacity. The symbol alphabet is structured as a finite field. The source generates a message per unit time, which is formulated as a vector over the symbol field. Upstream-to-downstream ordering of nodes enables concerted synchronization among all nodes so that the encoding and transmission of a message is independent of sequential messages. This finesses the issue of data communication delay and thereby allows the theory to deal with each message individually. When the symbol field is sufficiently large, the main theorem guarantees the maximum data rate toward every eligible receiver, which has the maximum information flow from the source no less than the message dimension.

Despite the restricted formulation over just acyclic networks, practical applications are not so restricted. What makes the difference is the time-multiplexed deployment of the transmission medium. When every channel in a cyclic network transmits a time series of data symbols, it may be viewed as the transmission of a data symbol on every channel in the trellis network that unfolds the time multiplexing. As time is unidirectional, this

trellis network is acyclic. The coding scheme at a node is time variant, but the propagation of sequential messages may convolve together. On the surface, it seems that the theoretic restriction to acyclic networks has been because of the need of upstream-to-downstream node ordering. Actually, this restriction is a much deeper mystery to be explored in the first part of the present thesis.

Chapter 2 below algebraically structures the ensemble of data units as a *principal ideal domain* (PID) instead of a finite field. Fundamental theory of linear algebra deals with vectors over a PID and leads to a theoretical generalization of the basic concepts of linear network coding from acyclic to cyclic networks. As a prerequisite for practical data propagation via a linear network code, the code must be *normal* such that the data unit carried on every channel can be unambiguously identified. Still, data propagation around a cycle via a normal PID-based linear network code may be non causal, because the PID-structure of data units lacks a unidirectional element to break the transmission deadlock. To resolve this problem, the PID of data units is further restricted to possess a unique maximal ideal. In commutative algebra, such a PID is called a *discrete valuation ring* (DVR). Let $\mathbb{M}$ denote the maximal ideal. Then, all ideals in the DVR form the infinite strictly descending chain

$$\mathbb{M} \supset \mathbb{M}^2 \supset \ldots \supset \mathbb{M}^t \supset \ldots$$

that converges to 0. Monotonicity of this chain is a "unidirectional attribute" of the DVR, which generalizes the unidirectional nature of time. It serves to break the deadlock in cyclic transmission. Meanwhile, every field is a PID with the unique maximal ideal 0 and hence can be regarded as a degenerated DVR. In this sense, the conventional formulation of linear network coding becomes a degenerated case of DVR-based network coding.

Convolutional network coding [31][22][30] is a special case of DVR-based network coding. The data unit is a time series of data symbols. Every time-multiplexed edge carries such a data unit, which is represented by a formal power series over the symbol field $\mathbb{F}$ in the variable D representing a unit-time delay. The ring $\mathbb{F}[[D]]$ of formal power series form a DVR with the unique maximal ideal being generated by D. Involving infinitely many data symbols, a power series cannot be expressed in finitely many data symbols. It is shown in [9] that a finite-state linear time-invariant causal sequential circuit with feedbacks, as depicted in Figure 1.2, realizes the encoder with the impulse response being a *rational power series*, i.e., a function in the form $p(D)/[1–D\cdot q(D)]$, where $p(D)$ and $q(D)$ are polynomials in the polynomial ring $\mathbb{F}[D]$ over the field $\mathbb{F}$ in indeterminate D. For the sake of physical implementability, the data units in convolutional network coding are restricted [30] to be rational power series, of which the ring will be denoted as $\mathbb{F}[(D)]$. A power series is invertible if and only if it is not divisible by D, and then the ring $\mathbb{F}[[D]]$ of formal power series can be regarded as the localization of itself at the prime ideal. Thus, in the finite case, $\mathbb{F}[(D)]$ is the localization of $\mathbb{F}[D]$ at the maximal ideal and hence is also a DVR by itself. A short introduction of related concepts in commutative algebra, such as the localization at a maximal ideal, can be found in Appendix A.

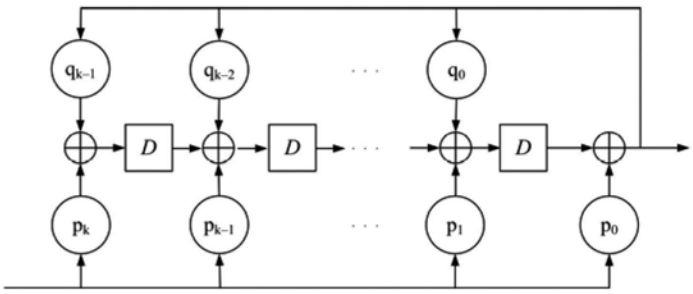


Figure 1.2. A finite-state linear time-invariant causal encoder with the impulse response $p(D)/[1–D\cdot q(D)]$, where $p(D) = p_0 + p_1\cdot D + \ldots + p_k\cdot D^k$, and $q(D) = q_0 + q_1\cdot D + \ldots + p_{k-1}\cdot D^{k-1}$, the polynomials in the polynomial ring $\mathbb{F}[D]$.

The practical application of convolutional network coding, however, is hindered by the difficulty in precise inter-node synchronization. General DVR-based network coding is not restricted to time-multiplexing or even combined time/space/frequency/phase/code/wavelength-multiplexing of data symbols. Generality enhances the potential of applicability. For example, if the uniformizer in the DVR, which generates the unique maximal ideal, represents a shift in some domain other than time, then the network code is insensitive to the aforementioned hindrance of imprecise inter-node synchronization.

1.2 Matroid Theory in Network Coding

Matroid theory is an abstract generalization of linear independence properties among vectors, *e.g.*, the algebraic independence among polynomials, the independence among bipartite matching, and the independence among edges in a forest of an undirected graph. It was first formulated in the seminal paper [39] in 1935 and has lately found rich applications in network coding.

Following the definition in [22], a *multicast network* means an acyclic network with a single source and an affiliated set of eligible receivers (whose maximum flow from the source is no less than the data generating rate). A network code is called a solution for this multicast network if the desired data transmission rate can be achieved via it for every receiver. Thus, the fundamental theorem of linear network coding [31] asserts the existence of a scalar linear solution, *i.e.*, the linear coding operation at intermediate nodes is memoryless, for every multicast network. However, the solvability for a multi-source multicast network becomes much more intricate to deal with. It was once conjectured in the literature [33] that every solvable multi-source

multicast network has a linear solution in the broad sense (*e.g.*, convolutional network codes and vector linear network codes [33][20].) A counterexample for this conjecture was provided in [5]. As revealed in [6], the construction of the solvable network without a linear solution is based on two matroidal networks, where one is established from Fano matroid and the other from non-Fano matroid. A matroid is said to be *representable* if its dependence structure can be translated into the linear dependence among vectors. The Fano and non-Fano matroids are known to be representable over the fields of even and odd characteristics, respectively. Moreover, based on the non-representability of the Vámos matroid (Proposition 6.1.10 in [34]), an acyclic solvable matroidal network is established to show the insufficiency of Shannon-type information inequalities in computing the network coding capacity [6]. As another application of matroids, every collection of polynomials is associated with an acyclic network in [7] such that every solution of this polynomial collection is equivalent to a scalar linear solution on the corresponding network.

Ahead of the above results in network coding theory from matroids, the original proof [31] of the fundamental theorem of linear network coding is in fact closely connected with matroids too. This will be explored in Chapter 3. On a multicast network, a *generic* linear network code is defined and constructed in [31] such that it is further shown to qualify as a scalar linear solution for this network. The design of a generic linear network code is motivated by the construction of a code in which every collection of coding vectors that can possibly be linearly independent is linearly independent. The class of generic linear network codes is further investigated from the network flow perspective in two parallel separate papers [38] and [37], in which the

optimality of generic codes in terms of linear independence among coding vectors is characterized under the same dependence structure of edge-disjoint paths with different descriptions, the former from the graph theoretical point of view but the latter in the language of matroids.

After the concept of generic linear network codes is extended in Chapter 2 to be PID-based and over a cyclic network, Chapter 3 offers a new characterization of it which in turn shows the code optimality in terms of linear independence. First, a *network matroid* is defined on the edge set of a network with possible cycles, via the structure of edge-disjoint paths. Meanwhile, the linear independence among coding vectors of a normal PID-based network code naturally induces a matroid on the edge set. It is shown that every independent set in the matroid so induced is also independent in the network matroid. Moreover, the two matroids coincide with each other if and only if the normal code is a generic one. This in turn reveals the representability of the network matroid. On the other hand, when the network is acyclic, every representation for the network matroid is proved to induce a generic linear network code on the same network. This offers a new characterization of generic linear network codes on an acyclic network in terms of the representation for network matroids. Unfortunately, in the case that the network contains cycles, we show that although a linear network code can still be induced from every representation for the network matroid, the code is not necessarily normal. Since the normality of a code is a prerequisite for generic codes, the representation for the network matroid is not able to fully characterize the class of generic linear network codes on a cyclic network.

1.3 Efficient Network Code Design

Chapter 3 delves into the efficiency issue of code construction. A linear network code that delivers a complete message from the source to all eligible receivers is called a *linear multicast*. The fundamental theorem in conventional linear network coding theory [31] asserts the existence of a linear multicast on an acyclic network when the symbol field is sufficiently large. However, there was not a polynomial time algorithm (which is in terms of number of edges in the network) to construct such an optimal code until a flow path oriented approach was published in two parallel conference papers [19] and [36], which have later been joined into a journal paper [21]. A different approach to construct an optimal network code efficiently is proposed in [16] from the aspect of matrix completion. Both algorithms utilize the upstream-to-downstream node ordering in the acyclic network.

Over a cyclic network, due to the lack of upstream-to-downstream node ordering, the efficient acyclic algorithms can not be directly adapted. In order to efficiently construct optimal codes on cyclic networks, the cycles in the network are classified in [13][3] for different treatments. By the discovery that the subnetwork consisting of the flow paths from the source node to a single receiver has an upstream-to-downstream ordering, a series of conference papers [9], [10], and [11] proposed and analyzed a variant (compared with the one in [21]) flow path oriented approach to construct an optimal linear network code on a cyclic network. All these algorithms, both acyclic and cyclic, will be briefly reviewed in Section 4.1.

In Section 4.2, we propose a unified efficient method to construct linear multicast on cyclic networks. Given a cyclic network, a quadratically large layered acyclic network is established by de-cycling such that every optimal

code on the de-cycled network subject to some straightforward restriction directly induces an optimal code on the cyclic one. In this way, existing algorithms for optimal codes on acyclic networks can be adapted for optimal codes on cyclic networks as well. This method is motivated by the duality theory of matroids. As introduced in Chapter 3, every linear network code on a network naturally induces a matroid on the edge set of the network. In particular, given a cyclic network, an optimal code C_1 on the corresponding layered acyclic network subject to some simple restriction not only induces an optimal code C_2 on the cyclic one, but induces a matroid on the edge set of the cyclic network as well. Then, the two matroids on the edge set of the cyclic network induced from C_1 and C_2 are in fact duals. The relationship will be elaborated in Section 4.4 such that other types of optimal linear network codes can also be constructed via this unified adaptation method. In Section 4.3, we further extend the method to be able to construct causal optimal linear network codes over cyclic networks from two perspectives.

Chapter 2.

Network Coding Theory via Commutative Algebra

This chapter presents a general formulation of scalar linear network coding theory via commutative algebra. This model includes the conventional linear network coding theory [31] and convolutional network coding theory [29] as the special cases. It not only ensures the causal data transmission around a cycle, but potentially alleviates the problem of imprecise inter-node synchronization in convolutional network coding. The rudimentary definitions and propositions in commutative algebra related to this thesis is summarized in Appendix A.

Hereafter throughout the present thesis, only the *linear theory* of network coding will be dealt with over a single-source network with possible cycles. We adopt the following convention throughout the thesis unless otherwise specified.

- The network under consideration is represented by a quadruple (V, E, s, ω), where V is the set of nodes, E the set of directed edges, s the source node with indgree 0, and ω the fixed data generating rate of the source node. An edge represents a communication channel of the unit capacity. There may possibly be cycles in the network.
- For every node v, denote by $\mathrm{In}(v)$ and $\mathrm{Out}(v)$, respectively, the sets of its incoming and outgoing edges. In particular, $\mathrm{Out}(s)$ consists of ω edges, which will be called *data-generating edges*. The set of ω data-generating edges will further be denoted by E_{DG}.
- Every edge e is assumed to be from the node $tail(e)$ to the node $head(e)$.

- An ordered pair (d, e) of edges is called an *adjacent pair* when $head(d) = tail(e)$.
- For every set $\wp$ of non-source nodes, denote by $\text{cut}(\wp)$ the collection of edges that lead from nodes outside $\wp$ to nodes in $\wp$, and by $\text{maxflow}(\wp)$ the *maximum flow* from the source node s to $\wp$.

$$\text{maxflow}(\wp) = \min_{\Im \supseteq \wp} |\text{cut}(\Im)|$$

- Impose a linear order on edges led by data-generating edges. In the special case when the network is acyclic, this linear order is assumed to be in an upstream-to-downstream fashion.
- Adopt the notation $Adj(\cdot)$ for the adjugate of a matrix.
- For any integer k, let I_k denote the $k \times k$ identity matrix.
- Form the $\omega \times |E|$ matrix $J_{\omega,|E|}$ by appending $|E| - \omega$ columns of zeroes to I_ω.
- Let $\mathbb{F}$ denote a field and $\mathbb{F}^\omega$ the vector space consisting of ω-dim column vectors over $\mathbb{F}$.
- Let $\mathbb{P}$ denote a PID and $\mathbb{P}^\omega$ the free $\mathbb{P}$-module consisting of ω-dim column vectors over $\mathbb{P}$.
- Let $\mathbb{Q}$ denote the quotient field of $\mathbb{P}$ and regard $\mathbb{P}$ as a subdomain of $\mathbb{Q}$.

2.1 Normality of PID-Based Network Coding

The model to be formulated in this section is an abstract extension of conventional linear network coding from acyclic to cyclic networks by structuring the ensemble of data units as a PID. Fundamental theory of linear algebra deals with vectors over a PID. Although the PID-based network coding does not guarantee causal data propagation around a cycle, it is a general framework on which all other results in this thesis are developed. The

causality issue of data propagation by network coding will be discussed in Section 2.3.

Definition 2.1. A $\mathbb{P}$-*linear network code* C on the network means the assignment of an element $k_{d,e}$ in $\mathbb{P}$ to every pair (d, e) of edges such that $k_{d,e} = 0$ when (d, e) is not an adjacent pair. The element $k_{d,e}$ is called the *coding coefficient* for the pair (d, e). We shall also adopt the convention $C = (k_{d,e})$. Moreover, denote by K_C the $|E|\times|E|$ matrix $[k_{d,e}]_{d,e\in E}$, where rows and columns are indexed according to the ordering of edges.

Definition 2.2. Given the $\mathbb{P}$-linear network code $C = (k_{d,e})$, a set of *coding vectors* means an assignment of an ω-dimensional column vector f_e over $\mathbb{P}$ to each edge e such that

(1) $\{f_e, e \in E_{DG}\}$ forms the natural basis of the free module $\mathbb{P}^\omega$

(2) $f_e = \sum_{d\in \mathrm{In}(v)} k_{d,e} f_d$ for every node v and every edge $e \in \mathrm{Out}(v)$

The two conditions can be combined in the matrix form into the equation $[f_e]_{e\in E} = [f_e]_{e\in E} \cdot K_C + J_{\omega,|E|}$ with the following two equivalent forms:

(3) $[f_e]_{e\in E} \cdot (I_{|E|} - K_C) = J_{\omega,|E|}$

(4) $\det(I_{|E|} - K_C)\ [f_e]_{e\in E} = J_{\omega,|E|} \cdot Adj(I_{|E|} - K_C)$

The condition (2) amounts to a nonhomogeneous system of $\omega\cdot(|E|-\omega)$ linear equations over $\mathbb{P}$ for the $\omega\cdot(|E|-\omega)$ variables that are entries to the vectors f_e, $e \in E\backslash E_{DG}$. The discriminant of this linear system is $\det(I_{|E|} - K_C)$. When the discriminant is zero, none or multiple solutions exist. On the other hand when it is nonzero, $C = (k_{d,e})$ can be regarded as a $\mathbb{Q}$-linear network code and thereby determines a unique set of coding vectors $f_e \in \mathbb{Q}^\omega$ by the formula

(5) $[f_e]_{e\in E} = \det(I_{|E|} - K_C)^{-1} J_{\omega,|E|} \cdot Adj(I_{|E|} - K_C)$

Thus, when $\det(I_{|E|}-K_C) \neq 0$, the $\mathbb{P}$-linear network code C determines a unique set of coding vectors or none depending whether $\det(I_{|E|}-K_C) \in \mathbb{P}$ divides all entries in the matrix $J_{\omega,|E|} \cdot Adj(I_{|E|}-K_C)$ or not.

Definition 2.3. The *discriminant* of a $\mathbb{P}$-linear network code C is $\det(I_{|E|}-K_C)$. The code is said to be *nonsingular* when it has nonzero discriminant. A nonsingular code is said to be *normal* when it determines a unique set of coding vectors.

Represent the message from the source by a row vector $m^{\mathbf{T}}$, where $m \in \mathbb{P}^{\omega}$. For a normal $\mathbb{P}$-linear network code with coding vectors f_e, the intended data unit for the transmission over an edge e is $m^{\mathbf{T}} \cdot f_e$. According to (2), $m^{\mathbf{T}} \cdot f_e = \sum_{d \in \mathrm{In}(v)} k_{d,e}\, m^{\mathbf{T}} \cdot f_d$ for an edge $e \in \mathrm{Out}(v)$. That is, an outgoing data unit from a node v is a linear combination of incoming data units, where the "linear gains" are the coding coefficients. Normality of a $\mathbb{P}$-linear network code unambiguously identifies the coding vectors and then the data units to be carried on edges, and hence is a prerequisite to the notion of data propagation via the code. A normal GF(3)-linear network code on the *Shuttle Network*, which contains cycles, is depicted in Figure 2.1.

Corollary 2.4. A nonsingular $\mathbb{P}$-linear network code C is normal if and only if $\det(I_{|E|}-K_C)$ divides all entries in the matrix $J_{\omega,|E|} \cdot Adj(I_{|E|}-K_C)$. This, in particular, is the case when $\det(I_{|E|}-K_C)$ is a unit in $\mathbb{P}$. Thus, a nonsingular $\mathbb{P}$-linear network code may be regarded as a normal $\mathbb{Q}$-linear network code.

Corollary 2.5. Assume that the network is acyclic. Then, $\det(I_{|E|}-K_C) = 1$ for every $\mathbb{P}$-linear network code C and, consequently, C is normal.

Proof. According to the convention at the beginning of the section, the edge ordering is from upstream to downstream. Thus, the matrix K_C is strictly upper triangular. ■

Definition 2.6. The *normalization* of a nonsingular $\mathbb{P}$-linear network code $C = (k_{d,e})$ means the $\mathbb{P}$-linear network code $(k'_{d,e})$, where

(6) $k'_{d,e} = \det(I_{|E|} - K_C) \cdot k_{d,e}$ for $d \in E_{DG}$

(7) $k'_{d,e} = k_{d,e}$ for $d \in E \backslash E_{DG}$

Corollary 2.7. The normalization of a nonsingular $\mathbb{P}$-linear network code is normal. Moreover, for a nonsingular $\mathbb{P}$-linear network code C with coding vectors $f_e \in \mathbb{Q}^\omega$, the coding vectors of the normalization are given by:

(8) $f'_e = \det(I_{|E|} - K_C) f_e$ for $e \notin E_{DG}$, while f'_e for $e \in E_{DG}$ still abide by (1).

Proof. Denote the normalization of C by $C' = (k'_{d,e})$. The $|E| \times |E|$ matrices K_C and $K_{C'}$ are

$$\begin{bmatrix} \mathbf{0} & \left[k_{d,e}\right]_{d \in E_{DG}, e \notin E_{DG}} \\ \mathbf{0} & \left[k_{d,e}\right]_{d,\, e \notin E_{DG}} \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{0} & \left[\det(I_{|E|} - K_C) \cdot k_{d,e}\right]_{d \in E_{DG}, e \notin E_{DG}} \\ \mathbf{0} & \left[k_{d,e}\right]_{d,\, e \notin E_{DG}} \end{bmatrix},$$

respectively, where the $\mathbf{0}$'s represent zero matrices containing $k_{d,e} = k'_{d,e} = 0$ for $e \in E_{DG}$. Thus, $\det(I_{|E|} - K_{C'}) = \det(I_{|E|-\omega} - [k_{d,e}]_{d,e \notin E_{DG}}) = \det(I_{|E|} - K_C) \neq 0$. Hence C' is nonsingular. From (4), the vectors $f'_e \in \mathbb{P}^\omega$. We need establish f'_e as the coding vectors of C'. It can be observed that in the two $\omega \times |E|$ matrices $J_{\omega,|E|} \cdot Adj(I_{|E|} - K_C)$ and $J_{\omega,|E|} \cdot Adj(I_{|E|} - K_{C'})$, the entries of the first three columns are identical, and every other $(i,j)^{\text{th}}$ entry in $J_{\omega,|E|} \cdot Adj(I_{|E|} - K_{C'})$ is $(I_{|E|} - K_C)$ times the $(i, j)^{\text{th}}$ entry in $Adj(I_{|E|} - K_C)$. From (5), the juxtaposition of coding vectors (belonging to $\mathbb{Q}^\omega$) of C' can be written as $[[f_e]_{e \in E_{DG}} \mid \det(I_{|E|} - K_C)[f_e]_{e \in E \backslash E_{DG}}]$, which is identical to $[f'_e]_{e \in E}$ from (8).

2.2 Optimal Normal Network Codes

In a normal $\mathbb{P}$-linear network code, coding vectors of data-generating edges generate the free module $\mathbb{P}^{\omega}$, which contains all coding vectors. Denote by $\mathbb{S}$ the $\mathbb{P}$-submodule generated by the coding vectors for any set of edges. According to the *invariant factor theorem of free submodule* (see Appendix A,) $\mathbb{S}$ is also a free-module, in which there exists a set B of vectors in $\mathbb{S}$, called the *basis* for $\mathbb{S}$, such that every nonzero vector in $\mathbb{S}$ can be uniquely written as a $\mathbb{P}$-linear combination of vectors in B. Moreover, the cardinality of B coincides with the rank of $\mathbb{S}$. In particular, coding vectors of incoming edges to a node v generates a free submodule, of which the rank represents the data reception rate of v from the source s. Without the constraint of data ensemble to be a PID, a module generated by a finite set of vectors does not necessarily have a basis. This justifies the PID to be the appropriate algebraic structure to start with for theoretical formulation of linear network coding.

Definition 2.8. A *sink* means a node v with $\mathrm{maxflow}(v) \geq \omega$. A normal $\mathbb{P}$-linear network code with the coding vectors f_e is called a $\mathbb{P}$-*linear multicast* when

(9) $\mathrm{rank}_{\mathbb{P}}(\langle f_e : e \in \mathrm{In}(v)\rangle) = \omega$ for every sink v

The point-to-point rate of information flow, as well as commodity flow, from s to v is bounded by $\mathrm{maxflow}(v)$. Thus a sink means node eligible for receiving data from s at the full rate ω. A linear multicast is an *optimal* network code in the sense of enabling every eligible node to receive at the full rate. It is the weakest sense of optimality in the conventional theory of linear network coding [41]. In its simplest form, the fundamental theorem of network coding [29][31] asserts the existence of an $\mathbb{F}$-linear multicast on an

acyclic network for every sufficiently large field $\mathbb{F}$. The following lemma generalizes this theorem from acyclic networks to all networks, while the proof partially follows the approach in [22] and the lower bound of $|\mathbb{F}|$ for the existence of an $\mathbb{F}$-linear multicast is based on the observation in [17].

Lemma 2.9. There exists an $\mathbb{F}$-linear multicast with all coding coefficients belonging to any subset $F \subseteq \mathbb{F}$ with $|F| > \delta + 1$, where δ is the number of sinks.

Proof. Associate every adjacent pair (d, e) with an indeterminate $x_{d,e}$. Let $\mathbb{F}[*]$ denote the polynomial ring in these indeterminates over $\mathbb{F}$. Write $x_{d,e} = 0$ when (d, e) is not an adjacent pair. As a polynomial in $\mathbb{F}[*]$, $\det(I_{|E|} - [x_{d,e}]_{d,e \in E})$ is nonzero, because its evaluation is equal to 1 at $x_{d,e} = 0$ for all adjacent pairs (d, e). As a consequence, $\det(I_{|E|-\omega} - [x_{d,e}]_{d,e \notin E_{DG}})$ is also a nonzero polynomial in $\mathbb{F}[*]$.

For every sink v, select ω edge-disjoint paths from data-generating edges to ω distinct edges belonging to $\mathrm{In}(v)$. For each edge e among the said ω distinct edges in $\mathrm{In}(v)$, let p_e denote the $(|E|-\omega)$-dim column vector indexed by $E \backslash E_{DG}$ in which the only nonzero entry is indexed by e and equal to 1. Juxtapose these ω vectors p_e into an $(|E|-\omega)\times\omega$ matrix, which will be denoted by P_v, and consider the $|E|\times|E|$ matrix, which will be denoted by M_v, in the form

$$M_v = \begin{bmatrix} [x_{d,e}]_{d \in E_{DG}, e \notin E_{DG}} & 0 \\ I_{|E|-\omega} - [x_{d,e}]_{d,\, e \notin E_{DG}} & P_v \end{bmatrix}.$$

The determinant of M_v is a nonzero polynomial in $\mathbb{F}[*]$, because its evaluation is equal to 1 when we set $x_{d,e}$ to 1 for all adjacent pairs (d, e) on the selected ω edge-disjoint paths and all other $x_{d,e}$ to 0.

When |F| is greater than the highest degree among all indeterminates in the nonzero polynomial $\det(I_{|E|}-[x_{d,e}]_{d,e\in E})\cdot\prod_{v:\text{sink}}\det(M_v)$, there exists values $k_{d,e} \in \mathrm{F}$ such that the evaluation of this nonzero polynomial at $x_{d,e} = k_{d,e}$ is not equal to $0 \in \mathbb{F}$. In this case, it suffices for |F| to be larger than $\delta + 1$, since in every polynomial $\det(M_v)$, all indeterminates $x_{d,e}$ have degrees no larger than 1. Under this evaluation at $x_{d,e} = k_{d,e}$, $\det(I_{|E|}-[x_{d,e}]_{d,e\in E}) \neq 0$ and $\det(M_v) \neq 0$ for all sinks. From Corollary 2.4, the $\mathbb{F}$-linear network code $(k_{d,e})$ is normal. Moreover, because

$$
\begin{aligned}
0 \neq \pm\det\left(\begin{bmatrix} [k_{d,e}]_{d\in E_{DG},e\notin E_{DG}} & \mathbf{0} \\ I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}} & P_v \end{bmatrix}\right)\cdot\det{}^{\omega}(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}}) \\
=\det\left(\begin{bmatrix} [k_{d,e}]_{d\in E_{DG},e\notin E_{DG}} & \mathbf{0} \\ I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}} & P_v \end{bmatrix}\cdot\begin{bmatrix} Adj(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})\cdot P_v & I_{|E|-\omega} \\ -\det(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})I_{\omega} & \mathbf{0} \end{bmatrix}\right) \\
=\det\left(\begin{bmatrix} [k_{d,e}]_{d\in E_{DG},e\notin E_{DG}}\cdot Adj(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})\cdot P_v & [k_{d,e}]_{d\in DGC,e\notin E_{DG}} \\ \mathbf{0} & I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}} \end{bmatrix}\right) \\
=\det([k_{d,e}]_{d\in E_{DG},e\notin E_{DG}}\cdot Adj(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})\cdot P_v)\cdot\det(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}}),
\end{aligned}
$$

the $\omega\times\omega$ matrix $[k_{d,e}]_{d\in E_{DG},e\notin E_{DG}}\cdot Adj(I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})\cdot P_v$ has nonzero determinant and thus its full rank qualifies $(k_{d,e})$ as an $\mathbb{F}$-linear multicast. ■

Theorem 2.10. A $\mathbb{P}$-linear multicast exists with all coding coefficients belonging to any subset $\mathrm{F} \subseteq \mathbb{P}$ with $|\mathrm{F}| > \delta + 1$, where δ is the number of sinks.

Proof. Because of Lemma 2.9, we need consider only the case when $\mathbb{P}$ is not a field. Then, $|\mathbb{P}|$ is infinite since every finite integral domain is a field (See Appendix A.) Applying Lemma 2.9 to $\mathbb{F} = \mathbb{Q}$ and $\mathrm{F} \subseteq \mathbb{P} \subset \mathbb{Q}$, there exists a $\mathbb{Q}$-linear multicast with coding coefficients in $\mathbb{P}$, which may be regarded as a nonsingular $\mathbb{P}$-linear network code. The lemma below shows that the normalization of this nonsingular $\mathbb{P}$-linear network code is a $\mathbb{P}$-linear multicast. ■

Lemma 2.11. Let C be a nonsingular $\mathbb{P}$-linear network code. Then, the normalization of C is a $\mathbb{P}$-linear multicast if and only if C is a $\mathbb{Q}$-linear multicast.

Proof. Denote the normalization of C by C'. Assume that either C' is a $\mathbb{P}$-linear multicast or C is a $\mathbb{Q}$-linear multicast. Let $f'_e \in \mathbb{P}^\omega$ and $f_e \in \mathbb{Q}^\omega$, respectively, denote the coding vectors of C' and C. Then, the vectors f'_e relate to f_e according to (8). For every sink v,

$$\begin{aligned}
&\mathrm{rank}_{\mathbb{P}}(\langle f'_e\colon e\in \mathrm{In}(v)\rangle)\\
&= \mathrm{rank}_{\mathbb{Q}}(\langle f'_e\colon e\in \mathrm{In}(v)\rangle)\\
&= \mathrm{rank}_{\mathbb{Q}}(\langle \det(I_{|E|}-K_C)f_e\colon e\in \mathrm{In}(v)\rangle)\\
&= \mathrm{rank}_{\mathbb{Q}}(\langle f_e\colon e\in \mathrm{In}(v)\rangle)
\end{aligned}$$

is equal to ω under either assumption at the beginning of this proof. ■

Example. Figure 2.1 depicts a GF(3)-linear multicast on the *Shuttle Network*[1], which contains cycles, in terms of coding coefficients and coding vectors.

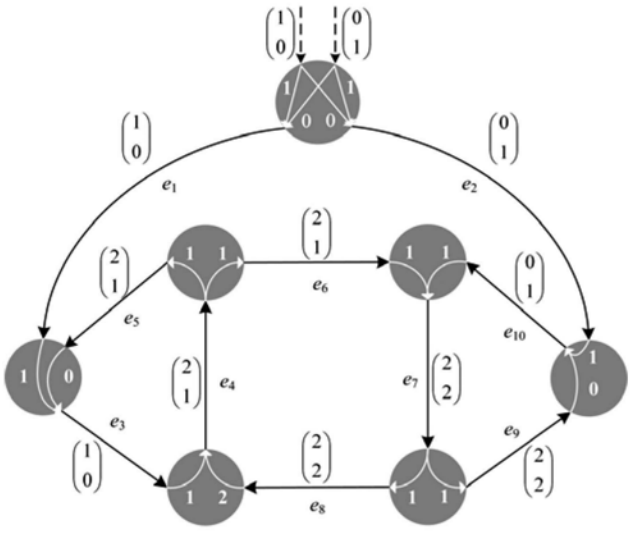


Figure 2.1. A normal GF(3)-linear network code on the *Shuttle Network*, which contains cycles, is given in terms of coding coefficients and coding vectors. It qualifies as a

[1] In all figures hereafter, the source node of a network, which has ω outgoing data-generating edges, will not be depicted.

GF(3)-linear multicast. The two dotted arrows represent data-generating edges from the source node, which is omitted. An adjacent pair is depicted by a white arrow inside.

The next theorem "normalizes" the inverse homomorphic image of a linear multicast into a linear multicast. It is a tool to be employed in Theorem 4.7 of Chapter 4 for efficient construction of an optimal convolutional multicast without the prior information of network transmission delays.

Theorem 2.12. Let μ be a homomorphism from $\mathbb{P}$ to another PID $\mathbb{P}'$ and $C = (k_{d,e})$ a $\mathbb{P}$-linear network code such that the homomorphic image $(\mu(k_{d,e}))$ is a $\mathbb{P}'$-linear multicast. Then, the normalization of C is a $\mathbb{P}$-linear multicast.

Proof. Applying componentwise, the homomorphism μ extends to a mapping from $\mathbb{P}^\omega$ to $(\mathbb{P}')^\omega$ and also to a mapping from matrices over $\mathbb{P}$ to matrices over $\mathbb{P}'$.

Write $C' = (\mu(k_{d,e}))$. Since $\det(I_{|E|}-K_{C'}) \neq 0$ and $\mu(\det(I_{|E|}-K_C)) = \det(I_{|E|}-K_{C'})$, we find that $\det(I_{|E|}-K_C) \neq 0$. Thus C is nonsingular. Let $f'_e \in (\mathbb{P}')^\omega$ and $f_e \in \mathbb{Q}^\omega$, respectively, denote the coding vectors of C' and C. Applying (4) to C and then to C',

$$\begin{aligned}
&\mu(\det(I_{|E|}-K_C)\ [f_e]_{e\in E}) \\
&= \mu(J_{\omega,|E|} \cdot Adj(I_{|E|}-K_C)) \\
&= J_{\omega,|E|} \cdot Adj(I_{|E|}-\mu(K_C)) \\
&= J_{\omega,|E|} \cdot Adj(I_{|E|}-K_{C'}) \\
&= \det(I_{|E|}-K_{C'})\ [f'_e]_{e\in E}
\end{aligned}$$

Hence $\mu(\det(I_{|E|}-K_C)f_e) = \det(I_{|E|}-K_{C'})f'_e$. Let g_e denote the coding vectors of the normalization of C. Then, $g_e = \det(I_{|E|}-K_C)f_e$ for $e \notin E_{DG}$ according to Corollary 2.7. For every sink v,

$$\begin{aligned}
&\mathrm{rank}_{\mathbb{P}}(\langle g_e\colon e\in \mathrm{In}(v)\rangle) \\
&= \mathrm{rank}_{\mathbb{P}}(\langle \det(I_{|E|}-K_C)f_e\colon e\in \mathrm{In}(v)\rangle) \\
&\geq \mathrm{rank}_{\mathbb{P}'}(\langle \mu(\det(I_{|E|}-K_C)f_e)\colon e\in \mathrm{In}(v)\rangle)
\end{aligned}$$

$$= \mathrm{rank}_{\mathbb{P}'}(\langle \det(I_{|E|} - K_{C'}) f'_e : e \in \mathrm{In}(v) \rangle)$$

$$= \mathrm{rank}_{\mathbb{P}'}(\langle f'_e : e \in \mathrm{In}(v) \rangle)$$

$$= \omega \qquad \blacksquare$$

Apart from linear multicast, three more classes of optimal linear network codes have been proposed in [41] over acyclic networks in terms of different strengths of linear independence. The remainder of this section extends them to the PID-based case in the manner analogous to linear multicast.

Definition 2.13. A normal $\mathbb{P}$-linear network code with the coding vectors f_e is called a $\mathbb{P}$-*linear broadcast*, a $\mathbb{P}$-*linear dispersion*, or a *generic* $\mathbb{P}$-linear network code, respectively, if the following statements hold:

(10) $\mathrm{rank}_{\mathbb{P}}(\langle f_e : e \in \mathrm{In}(v) \rangle) = \min(\omega, \mathrm{maxflow}(v))$ for every non-source node v.

(11) $\mathrm{rank}_{\mathbb{P}}(\langle f_e : e \in \mathrm{In}(\wp) \rangle) = \min(\omega, \mathrm{maxflow}(\wp))$ for every collection $\wp$ of non-source nodes.

(12) Let $\{e_1, \ldots, e_m\}$ be an arbitrary set of edges except data-generating edges, where each $e_j \in \mathrm{Out}(v_j)$. Then the vectors $f_{e_1}, \ldots, f_{e_m}$ are linearly independent (and hence $m \le \omega$) provided that

$$\langle f_d : d \in \mathrm{In}(v_j) \rangle \not\subset \langle f_{e_k} : k \ne j \rangle \text{ for } 1 \le j \le m$$

The definition of generic $\mathbb{P}$-linear network codes here is in line with the one in [41]. In a generic code, any collection of coding vectors must be linearly independent unless "obviously otherwise." This motivation will become more transparent when interpreted from the matroid perspective in Chapter 3.

Clearly, (11)$\Rightarrow$(10)$\Rightarrow$(9). Thus, every $\mathbb{P}$-linear dispersion is a $\mathbb{P}$-linear broadcast, and every $\mathbb{P}$-linear broadcast is a $\mathbb{P}$-linear multicast. The existence of an $\mathbb{F}$-linear dispersion and an $\mathbb{F}$-linear broadcast can be shown analogous

to Lemma 2.9. This in turn implies the existence of a $\mathbb{P}$-linear broadcast and a $\mathbb{P}$-linear dispersion in the similar approach to Theorem 2.10.

Proposition 2.14. A $\mathbb{P}$-linear broadcast exists when $|\mathbb{P}| > |V|$.

Proposition 2.15. A $\mathbb{P}$-linear dispersion exists when $|\mathbb{P}| > 2^{|V|-1} + 1$.

Theorem 2.16. A generic $\mathbb{P}$-linear linear network code exists when $|\mathbb{P}| > |\mathcal{B}| + 1$, where $\mathcal{B}$ denotes the family of sets B of ω edges in $E \backslash E_{DG}$ such that there are ω edge-disjoint paths starting from data-generating edges and ending at B.

Proof. First we shall prove that when $\mathbb{P}$ is a field $\mathbb{F}$, a generic $\mathbb{F}$-linear linear network code exists with all coding coefficients belonging to any subset F in $\mathbb{F}$ with $|F| > |\mathcal{B}| + 1$. Then this result can be extended to the existence of a generic $\mathbb{P}$-linear network code from the same approach as the proof of Theorem 2.10.

In the same manner as the one in the proof of Lemma 2.9, associate every adjacent pair (d, e) with an indeterminate $x_{d,e}$. Let $\mathbb{F}[*]$ denote the polynomial ring in these indeterminates over $\mathbb{F}$. Write $x_{d,e} = 0$ when (d, e) is not an adjacent pair. As a polynomial in $\mathbb{F}[*]$, $\det(I_{|E|} - [x_{d,e}]_{d,e \in E})$ is nonzero, because its evaluation is equal to 1 at $x_{d,e} = 0$ for all adjacent pairs (d, e).

We shall next construct a normal $\mathbb{F}$-linear network code subject to the condition:

(13) For every set B in $\mathcal{B}$, the coding vectors for edges in B are linearly independent.

Then we shall further show that the $\mathbb{F}$-linear network code thus constructed qualifies as a generic $\mathbb{F}$-linear network code.

Consider a set B in $\mathcal{B}$ with ω edge-disjoint paths starting from data-generating edges and ending at B. For each edge e in B, let g_e denote the

column e in the matrix $J_{\omega,|E|} \cdot Adj(I_{|E|}-[x_{d,e}]_{d,e\in E})$. Thus g_e is an ω-dim vector over $\mathbb{F}[*]$. Juxtaposes these ω vectors g_e into an $\omega\times\omega$ matrix denoted by M_B. Then, as a polynomial in $\mathbb{F}[*]$, $\det(M_B)$ is nonzero because its evaluation is equal to 1 when we set $x_{d,e}$ to 1 for all adjacent pairs (d, e) on the ω edge-disjoint paths and all other $x_{d,e}$ to 0.

When |F| is greater than the highest degree among all indeterminates in the nonzero polynomial $\det(I_{|E|}-[x_{d,e}]_{d,e\in E})\cdot\prod_{B\in\mathcal{B}}\det(M_B)$, there exist values $k_{d,e} \in$ F such that the evaluation of this nonzero polynomial at $x_{d,e} = k_{d,e}$ is not $0 \in \mathbb{F}$. Under this evaluation, $\det(I_{|E|}-[x_{d,e}]_{d,e\in E}) \neq 0$ and $\det(M_B) \neq 0$ for all sinks. From Corollary 2.4, the $\mathbb{F}$-linear network code $(k_{d,e})$ is normal and is subject to (13).

Let f_e denote the coding vectors of the normal $\mathbb{F}$-linear network code $(k_{d,e})$. For a set $X \subseteq E\backslash E_{DG}$ of edges, denote by C the set of edges with minimum cardinality such that after removal of it from the network, there is no path leading from data-generating edges to edges in X. Let $\wp$ denote the set of nodes that are disconnected with s after the removal of C. Then $X\backslash C \subseteq \mathrm{Out}(\wp)$ and $C = \mathrm{cut}(\wp)$. According to Lemma 2.17 below, $\langle f_e: e\in\mathrm{In}(\wp)\rangle = \langle f_e: e\in C\rangle$. Hence,

$$\langle f_e: e\in X\backslash C\rangle \subseteq \langle f_e: e\in\mathrm{Out}(\wp)\rangle \subseteq \langle f_e: e\in\mathrm{In}(\wp)\rangle = \langle f_e: e\in C\rangle$$

$$\langle f_e: e\in X\rangle \subseteq \langle f_e: e\in C\rangle$$

When X is not contained in any element in $\mathcal{B}$, according to the Max-Flow-Min-Cut Theorem, $|C| < |X|$. Thus,

$$\mathrm{rank}(\langle f_e: e\in X\rangle) \leq \mathrm{rank}(\langle f_e: e\in C\rangle) \leq |C| < |X|$$

In conclusion,

(14) The coding vectors f_e for edges $e \in X$, where X is not contained in any element in $\mathcal{B}$, are linearly dependent.

Consider a set $\{e_1, \ldots, e_m\}$ of edges such that the coding vectors f_{e_j}, $1 \le j \le m$ are *linearly dependent*. Assume that $\langle f_d\colon d \in \mathrm{In}(v_j)\rangle \subset \langle f_{e_k}\colon k \ne j\rangle$ for some j. Then there exists at least one edge $e'_j \in \mathrm{In}(v_j)$ such that the coding vectors for edges in $\{e_1, \ldots, e_m\}\backslash\{e_j\}\cup\{e'_j\}$ are linearly independent. This implies that $\{e_1, \ldots, e_m\}\backslash\{e_j\}\cup\{e'_j\} \in \mathcal{B}$, because otherwise the coding vectors for edges in it are linearly dependent according to (14). Consequently, the set $\{e_1, \ldots, e_m\}$ is also in $\mathcal{B}$, and the coding vectors f_{e_j}, $1 \le j \le m$ are linearly independent via the code construction, a contradiction to the assumption. Therefore, $\langle f_d\colon d \in \mathrm{In}(v_j)\rangle \subseteq \langle f_{e_k}\colon k \ne j\rangle$ for $1 \le j \le m$. The qualification of $(k_{d,e})$ as a generic $\mathbb{F}$-linear network code is proved by contrapositive. ■

Lemma 2.17. Let f_e denote the coding vectors of a normal $\mathbb{P}$-linear network code. Then for every set $\wp$ of non-source nodes,

$$\langle f_e\colon e \in \mathrm{cut}(\wp)\rangle = \langle f_e\colon e \in \mathrm{In}(\wp)\rangle.$$

Proof. Since the $\mathbb{P}$-linear network code is normal, the coding vectors f_e can uniquely be determined via (4). Adopt the abbreviation $E_1 = \mathrm{cut}(\wp)$ and $E_2 = \mathrm{In}(\wp)\backslash\mathrm{cut}(\wp)$. Without loss generality, assume that the edges in E_1 have higher topological order than edges in E_2. Then

$$[[f_e]_{e\in E_1} \mid [f_e]_{e\in E_2}] \cdot \begin{bmatrix} I_{|E_1|} & -(k_{d,e})_{d\in E_1, e\in E_2} \\ \mathbf{0} & I_{|E_2|} - (k_{d,e})_{d,e\in E_2} \end{bmatrix} = [[f_e]_{e\in E_1} \mid \mathbf{0}],$$

$$[f_e]_{e\in E_2} \cdot (I_{|E_2|} - (k_{d,e})_{d,e\in E_2}) = [f_e]_{e\in E_1} \cdot (k_{d,e})_{d\in E_1, e\in E_2},$$

$$\det(I_{|E_2|} - (k_{d,e})_{d,e\in E_2})\ [f_e]_{e\in E_2} = [f_e]_{e\in E_1} \cdot (k_{d,e})_{d\in E_1, e\in E_2} \cdot Adj(I_{|E_2|} - (k_{d,e})_{d,e\in E_2}),$$

where the normality of the $\mathbb{P}$-linear network code ensures that every entry in the matrix on the right hand side is divisible by $\det(I_{|E_2|} - (k_{d,e})_{d,e\in E_2})$. Therefore, every coding vector f_d for edge $d \in \mathrm{In}(\wp)\backslash\mathrm{cut}(\wp)$ is a linear combination of f_e, $e \in \mathrm{cut}(\wp)$. ■

Theorem 2.18. Every generic $\mathbb{P}$-linear network code is a $\mathbb{P}$-linear dispersion.

Proof. This proof follows the same line in one for Theorem 2.29 in [41], which considers the field-based acyclic case but does not utilize the upstream-to-downstream order of the network. The proof for the acyclic case requires a fact stated in Lemma 2.27 in [41], which is extended to the PID-based cyclic case in the previous lemma. The validity of this extension is ensured by the normality of a linear network code. ■

Example. The GF(3)-linear multicast on the Shuttle Network depicted in Figure 2.1 qualifies as a GF(3)-linear broadcast and a GF(3)-linear dispersion as well. However, it is not a generic GF(3)-linear network code, because for $\{e_1, e_3\}, f_{e_1}$ is equal to f_{e_3} but $\langle\{f_{e_1}, f_{e_5}\}\rangle \not\subset \langle f_{e_1}\rangle$, a contradiction to the criterion (12). The GF(3)-linear network code depicted in Figure 2.2 is a generic one.

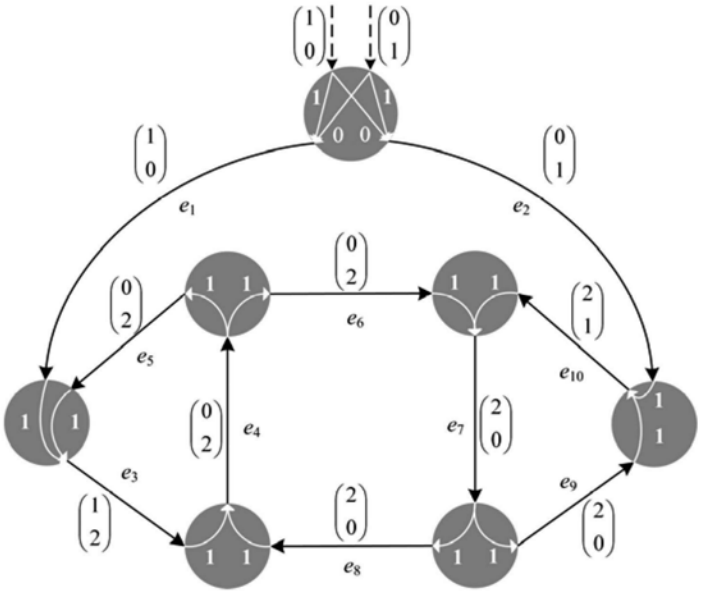


Figure 2.2. A generic GF(3)-linear network code on the Shuttle Network, is given in terms of coding coefficients and coding vectors. Consequently, it qualifies as a linear dispersion, a linear broadcast, and a linear multicast as well.

This section ends in the remark that the connection between a linear multicast and its inverse homomorphic image exposed in Theorem 2.12 can

easily be adapted to the cases of linear broadcast, linear dispersion and generic codes as well. The proofs will be skipped.

2.3 Causal Data Propagation by Network Coding

Represent the message from the source by a row vector $m^{\mathbf{T}}$, where $m \in \mathbb{P}^{\omega}$. If there is a sensible way of data propagation via a normal $\mathbb{P}$-linear network code with coding vectors f_e, then each edge e must carry the data unit $m^{\mathbf{T}} \cdot f_e$. This makes every outgoing data unit from a node a linear combination of incoming data units to that node. However, there is the question on how do edges around a cycle acquire their respective data units for transmission without a deadlock. The question pertains to the algebraic structure of $\mathbb{P}$. To explain this point, we take convolutional network coding as an example. Streams of data symbol propagate through the network. The time-series of data symbols carried by an edge is represented by a power series over the symbol field $\mathbb{F}$. Due to finiteness in physical realization, the data unit is actually restricted to a rational power series. Thus the ensemble of data units is $\mathbb{F}[(D)]$. In causal data propagation by a convolutional network code, the transmission delay is nonzero along every cycle and thus the time-multiplexed deployment of the network can be unfolded into a trellis network, which is acyclic. The transmission of a data unit by every edge is logically equivalent to propagation of symbols through the trellis network. This implies that a causal convolutional network code not only qualifies as a normal $\mathbb{F}[(D)]$-linear network code but also precludes transmission deadlock around a cycle. This causal characteristic stems from the acyclic nature of the trellis network, which in turn is due to the incorporation of time, a unidirectional dimension, inside the domain $\mathbb{F}[(D)]$.

Convention. Hereafter let $\mathbb{D}$ denote a DVR and z the *uniformizer*, that is, the generator of the maximal ideal in $\mathbb{D}$. The uniformizer is unique up to a unit factor. In the particular instance when $\mathbb{D} = \mathbb{F}[(D)]$, the uniformizer z is D.

The algebraic structure of a DVR incorporates a unidirectional attribute similar to time. In fact, all ideals in $\mathbb{D}$ form the infinite strictly descending chain

$$\langle z\rangle \supset \langle z^2\rangle \supset \ldots \supset \langle z^t\rangle \ldots \supset \bigcap_{k=1}^{\infty}\langle D\rangle^k = 0,$$

where the equality is a direct consequence of the *Nakayama Lemma* (See Appendix A.) This unidirectional chain makes $\mathbb{D}$ a suitable domain of data units. In particular, when the DVR is $\mathbb{F}[[D]]$, the ideal $\langle D^k\rangle$ consists of those power series that represent symbol streams to appear at the time k or later. Then the equality $\bigcap_{k=1}^{\infty}\langle D\rangle^k = 0$ reflects the physical axiom that every non-zero symbol stream must start within finite time. Causal transmission of data units via a $\mathbb{D}$-linear linear network code requires the coding coefficient for at least one adjacent pair along every cycle to be divisible by z.

Definition 2.19. A *delay function* on the network is a nonnegative integer function t, defined over the set of adjacent pairs such that, along every cycle, there is at least one pair (d, e) with $t(d, e) > 0$. A $\mathbb{D}$-linear network code is said to be *t-causal* if the coding coefficient for every adjacent pair (d, e) is divisible by $z^{t(d, e)}$. A *causal* $\mathbb{D}$-linear network code means one that is t-causal for some t.

Proposition 2.20. A causal $\mathbb{D}$-linear network code C is normal. In fact, $\det(I_{|E|}–K_C)$ is a unit in $\mathbb{D}$.

Proof. Denote by $m_{d,e}$ the $(d, e)^{\text{th}}$ entry in the matrix $I_{|E|}–K_C$. Thus $\det(I_{|E|}–K_C)$ is a sum of $|E|!$ summands, each being in the form of

$\pm\prod_{d\in E} m_{d,\sigma(d)}$ in correspondence to a permutation σ on edges. When $\sigma(d) = d$, $m_{d,\sigma(d)} = 1$; when $(d, \sigma(d))$ is an adjacent pair, $m_{d,\sigma(d)} = -k_{d,\sigma(d)}$. Else, $m_{d,\sigma(d)} = 0$. In the conventional *cycle representation* of the permutation group, a permutation is uniquely factorized into disjoint *cycles* of length ≥ 2. When $\pm\prod_{d\in E} m_{d,\sigma(d)} \neq 0$, such disjoint cycles correspond to edge-disjoint cycles in the network and $\pm\prod_{d\in E} m_{d,\sigma(d)}$ is the product of coding coefficients for edges on these cycles. Causality of C makes the product of coding coefficients for edges on a cycle divisible by z. Therefore, $\pm\prod_{d\in E} m_{d,\sigma(d)}$ is divisible by z when the cycle representation of σ is not null, that it, when σ is not the identity mapping. On the other hand when σ is the identity mapping, $\prod_{d\in E} m_{d,\sigma(d)} = 1$. This makes $\det(I_{|E|}-K_C)$ a unit in $\mathbb{D}$. Thus C is normal by Corollary 2.4. ■

Theorem 2.21. Given a delay function t, there exists a t-causal $\mathbb{D}$-linear multicast.

Proof. Let m be the largest $t(d, e)$ among all adjacent pairs (d, e). Being a PID but not a field, the DVR $\mathbb{D}$ is infinite in cardinality. The ideal generated by z^m is $z^m\cdot\mathbb{D}$, which shares the same cardinality with $\mathbb{D}$. Applying Lemma 2.9 with $\mathbb{F}$ being the quotient field of $\mathbb{D}$ and $\mathrm{F} = z^m\cdot\mathbb{D} \subseteq \mathbb{F}$, there exists an $\mathbb{F}$-linear multicast C with coding coefficients $k_{d,e} \in z^m\cdot\mathbb{D}$. As a $\mathbb{D}$-linear network code, C is t-causal and hence also normal by Theorem 2.21. Denote the coding vectors by $f_e \in \mathbb{D}^\omega$. For every sink v,

$$\mathrm{rank}_{\mathbb{D}}(\langle f_e\colon e\in \mathrm{In}(v)\rangle) = \mathrm{rank}_{\mathbb{F}}(\langle f_e\colon e\in \mathrm{In}(v)\rangle) = \omega.$$

Therefore, C is a $\mathbb{D}$-linear multicast. ■

Corollary 2.22. Given a delay function t, there exists a t-causal $\mathbb{D}$-linear broadcast, a t-causal $\mathbb{D}$-linear dispersion, and a t-causal generic $\mathbb{D}$-linear network code.

Example. Figure 2.3 depicts a causal $\mathbb{D}$-linear network code on the shuttle network, which qualifies as a generic $\mathbb{D}$-linear network code, a $\mathbb{D}$-linear dispersion, a $\mathbb{D}$-linear broadcast, and a $\mathbb{D}$-linear multicast.

Given a causal $\mathbb{D}$-linear network code, juxtapose the incoming coding vectors f_e to a node v into an $\omega \times |\mathrm{In}(v)|$ matrix $[f_e]_{e\in \mathrm{In}(v)}$. Represent the message from the source by a row vector $m^{\mathbf{T}}$, where $m \in \mathbb{D}^{\omega}$. Thus the data received by the node v is represented by the row vector $m^{\mathbf{T}}\cdot[f_e]_{e\in \mathrm{In}(v)}$. To receive the message $m^{\mathbf{T}}$ means to *decode* it out of the received row vector. An explicit form of decoding is to calculate $z^{\delta}m^{\mathbf{T}}$, for some integer $\delta \geq 0$, from $m^{\mathbf{T}}\cdot[f_e]_{e\in \mathrm{In}(v)}$. Here δ means a "delay measured by the exponent of z" in reflection of the definition of a causal $\mathbb{D}$-linear network code. A prerequisite is the full rank ω of the matrix $[f_e]_{e\in \mathrm{In}(v)}$, because the message vector $m^{\mathbf{T}}$ is arbitrarily given.

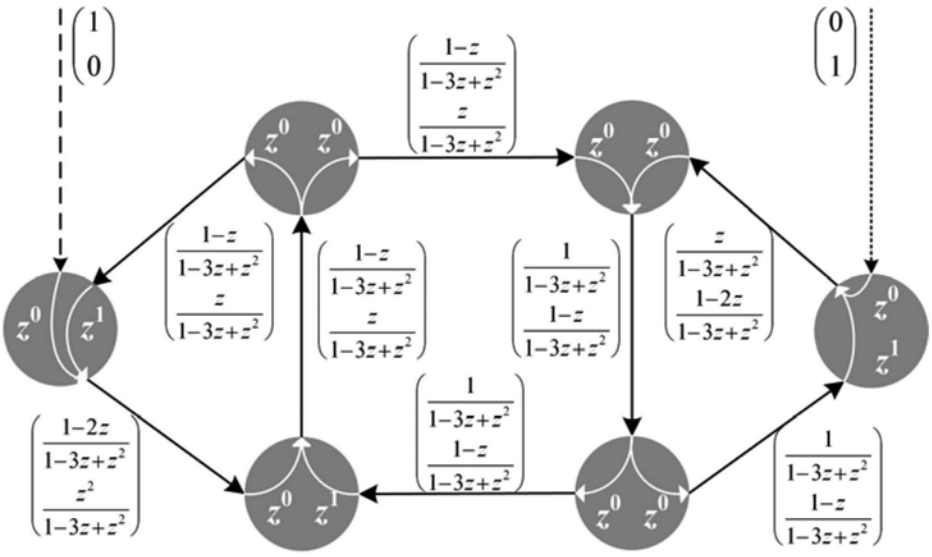


Figure 2.3. A $\mathbb{D}$-linear network code on the Shuttle Network, is given in terms of coding coefficients and coding vectors. It qualifies as a generic $\mathbb{D}$-linear network code, a $\mathbb{D}$-linear dispersion, a $\mathbb{D}$-linear broadcast, and a $\mathbb{D}$-linear multicast. Let t be any delay function subject to the specification of $t(d, e) = n$ when the coding coefficient $k_{d,e} = z^n$. Then, the given optimal code is t-causal.

Definition 2.23. For a causal $\mathbb{D}$-linear network code with the coding vectors f_e, a *decoding matrix* with *delay* $\delta \geq 0$ at a node v means an $|\mathrm{In}(v)| \times \omega$ matrix M over $\mathbb{D}$ such that $[f_e]_{e \in \mathrm{In}(v)} \cdot M = z^{\delta} I_{\omega}$.

Theorem 2.24. For a $\mathbb{D}$-linear multicast, the submodule $\mathbb{S}_v$ of $\mathbb{D}^{\omega}$ spanned by incoming coding vectors to a node v is free. Assume that v is a sink so $\mathrm{rank}_{\mathbb{D}}(\mathbb{S}_v) = \omega$. Denote the invariant factors[1] of $\mathbb{S}_v$ in $\mathbb{D}^{\omega}$ by $z^{i_1}, z^{i_2}, \ldots, z^{i_{\omega}}$, where $i_1 \leq \ldots \leq i_{\omega}$. Let $\delta \geq i_{\omega}$. Then, a decoding matrix with delay δ exists at v.

Proof. According to the *invariant factor theorem of free submodule*, $\mathbb{S}_v$ is a free submodule of $\mathbb{D}^{\omega}$ and there exists a basis $\{u_1, \ldots, u_{\omega}\}$ of $\mathbb{D}^{\omega}$ such that $\{z^{i_1} u_1, \ldots, z^{i_{\omega}} u_{\omega}\}$ is a basis of $\mathbb{S}_v$. Thus, the matrix $[u_j]_{1 \leq j \leq \omega}$ is invertible and the vectors $z^{\delta} u_1, \ldots, z^{\delta} u_{\omega}$ are all $\mathbb{D}$-linear combinations of the coding vectors f_e, $e \in \mathrm{In}(v)$. Translating into the matrix form, there exists an $|\mathrm{In}(v)| \times \omega$ matrix M over $\mathbb{D}$ such that

(15) $z^{\delta} [u_j]_{1 \leq j \leq \omega} = [f_e]_{e \in \mathrm{In}(v)} \cdot M$

Hence $M \cdot ([u_j]_{1 \leq j \leq \omega})^{-1}$ is a decoding matrix with delay δ at v. ■

When the network is acyclic, the presence of the zero delay function renders every $\mathbb{D}$-linear network code causal and the uniformizer z in the DVR superfluous. Treat a field $\mathbb{F}$ as a degenerated DVR with the unique maximal ideal $\{0\}$ and no uniformizer. Then, an $\mathbb{F}$-linear network code on an acyclic network is automatically causal. Thus the assumption of delay-free transmission in linear network coding (see [29], [31] and the pursuing literature) on an acyclic network is causal transmission in a degenerated sense of Definition 2.19.

[1] For the discussion of invariant factors, refer to Appendix A.

Chapter 3.

Linear Network Codes and Matroids

Matroid theory is an abstract generalization of linear dependence properties among vectors. This chapter discusses the inherent connection between PID-based linear network coding theory and the representation of matroids over a network. Moreover, based on the duality theory of matroids, a unified method will be developed in Chapter 4 to construct optimal linear network codes on a cyclic network.

3.1 Basic Definition of Matroids

Let S be a finite set. A matroid $\mathcal{M}$ on the *ground set* S is an ordered pair $(S, \mathcal{I})$, where $\mathcal{I}$ is a family of subsets of S satisfying the following three axioms:

(I1) $\phi \in \mathcal{I}$.

(I2) If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$

(I3) (Augmentation axiom) If I_1 and I_2 are in $\mathcal{I}$ and $|I_1| < |I_2|$, then there is an element $e \in I_2 \setminus I_1$ such that $I_1 \cup e \in \mathcal{I}$.

A subset of S is called an *independent set* if it is in $\mathcal{I}$ and otherwise a *dependent set*. A maximal independent set is called a *base*. All bases of a matroid can be easily shown to share the same cardinality, which is called the *rank* of the matroid (See Appendix B.) The *restriction* of the matroid $(S, \mathcal{I})$ to a subset S' of S is the matroid $(S', \mathcal{I}')$, where $\mathcal{I}' = \{I \cap S' : I \in \mathcal{I}\}$. The rank of this matroid restriction is also called the *rank* of S' in the matroid $(S, \mathcal{I})$, which will be denoted by $r(S')$. Two matroids $(S_1, \mathcal{I}_1)$ and $(S_2, \mathcal{I}_2)$ are *isomorphic* if there is a bijection between S_1 and S_2 that maps independent sets to independent sets.

Besides the first definition of matroids in terms of the family of independent sets presented in [39], there are several other equivalent definitions, e.g., in terms of the family of bases or the rank function on the power set of the ground set. Among them, the equivalent definition in terms of bases will be further introduced in Appendix B. A great variety of algebraic structures can be characterized as matroids. Following are three examples.

- Let S be the set of columns in a matrix over $\mathbb{P}$, then the family of linearly independent subsets of S forms the *vector matroid* of the matrix. The rank of the vector matroid coincides with the rank of the matrix.
- The family of forests of edges in an undirected graph forms a *graphic matroid*.
- Consider a bipartite graph between two vertex sets S and T. A *transversal matroid* is formed by those subsets of S that can be one-to-one mapped into T through edges.

A matroid is said to be *representable* over $\mathbb{P}$ if it is isomorphic to the vector matroid of a matrix over $\mathbb{P}$, and this matrix is called a $\mathbb{P}$-*representation* for the matroid. Graphic matroids and transversal matroids are all representable (See Proposition 5.1.2 in [34] for graphic matroid, and [35] for transversal matroid.) A good introduction to fundamental properties of matroids can be found in [34].

By the linear independence among coding vectors of a *normal* linear network code, a matroid can naturally be defined on the edge set $E\backslash E_{DG}$.

Definition 3.1. The *induced matroid* of a normal $\mathbb{P}$-linear network code C is the matroid $(E\backslash E_{DG}, \mathcal{I}_C)$, where $\mathcal{I}_C$ is the family of subset of edges whose

coding vectors are linearly independent. C is said to *induce* the matroid $(E\backslash E_{DG}, \mathcal{I}_C)$.

The definition above asserts that every normal linear network code is associated with a representable matroid on $E\backslash E_{DG}$. However, not every representable matroid on $E\backslash E_{DG}$ can be an induced matroid of some linear network code.

Example. Consider the rank-2 *uniform* matroid $U_{2,5}$ on $E\backslash E_{DG}$ of the network in Figure 3.1, where the family of bases comprises all two-element subsets of $E\backslash E_{DG}$. It is known that $U_{2,5}$ is representable over the field $\mathbb{F}$ when $|\mathbb{F}| \geq 4$ (Proposition 6.5.2 in [34]). However, no linear network code on the network can induce $U_{2,5}$ because f_{e_4} and f_{e_5} must be linearly dependent.

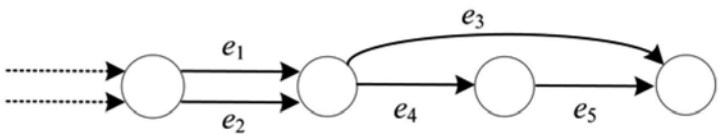


Figure 3.1. On the depicted network, there does not exist a $\mathbb{P}$-linear network code that can induce the uniform matroid $U_{2,5}$, in which all two-element subsets are bases.

3.2 Network Matroids and Linear Network Codes

In this section we reveal that the independence structure of edge-disjoint paths in fact establishes a matroid on the edge set of the network, and this matroid has the maximal independence structure among the ones a normal $\mathbb{P}$-linear network code can induce.

Theorem 3.2. Let $\mathcal{I}$ be the family of subsets $X \subseteq E\backslash E_{DG}$ such that there are $|X|$ edge-disjoint paths leading from data-generating edges to X. Then, $(E\backslash E_{DG}, \mathcal{I})$ is a matroid, which will be called the *network matroid* of the network.

Proof. Define a directed multi-graph with the node set $V' = E$ and the edge set $E' = \{(e_1, e_2) \in V'^2 : head(e_2) = tail(e_1)\}$. Two paths $(d_1, d_2, \ldots, d_m)$ and $(e_1, e_2, \ldots, e_n)$ are edge-disjoint if and only if the two paths $(d_m, d_{m-1}, \ldots, d_1)$ and $(e_n, e_{n-1}, \ldots, e_1)$, which are characterized by nodes, are node-disjoint paths in (V', E'). Let $\mathcal{I}'$ denote the family of subsets $X' \subseteq V'$ such that there are $|X'|$ node-disjoint paths starting from nodes in X' and ending at nodes in $E_{DG} \subseteq V'$. Then, according to the result in [32], $(V', \mathcal{I}')$ is a matroid, which is known to be a *strict gammoid.* Therefore, $(E \backslash E_{DG}, \mathcal{I})$ also forms a matroid, which is isomorphic to the restriction of the strict gammoid $(V', \mathcal{I}_{B_0})$ to $V' \backslash E_{DG}$. ■

Example. In the network matroid of the Shuttle Network in Figure 2.1, all 2-edge sets, except for $\{e_4, e_5\}$, $\{e_4, e_6\}$, $\{e_5, e_6\}$, $\{e_7, e_8\}$, $\{e_7, e_9\}$, and $\{e_8, e_9\}$, are independent. All sets of 3 or more edges are not.

Now we will investigate the relationship between the network matroid and the matroids induced by normal linear network codes.

Proposition 3.3. Every independent set in an induced matroid is an independent set in the network matroid.

Proof. Let X be an independent set in an induced matroid. Denote by C the minimal set of edges such that after removal of it from the network, there is no path leading from data-generating edges to edges in X. According to the Max-Flow-Min-Cut Theorem, the rank $r_{\mathcal{M}}(X)$ of X in network matroid $\mathcal{M}$ is equal to $|C|$. Let $\wp$ denote the set of nodes that are disconnected with s after the removal of C. Then $X \backslash C \subseteq \mathrm{Out}(\wp)$ and $C = \mathrm{cut}(\wp)$. According to Lemma 2.17, $\langle f_e : e \in \mathrm{In}(\wp) \rangle = \langle f_e : e \in C \rangle$. Hence,

$$\langle f_e : e \in X \backslash C \rangle \subseteq \langle f_e : e \in \mathrm{Out}(\wp) \rangle \subseteq \langle f_e : e \in \mathrm{In}(\wp) \rangle = \langle f_e : e \in C \rangle$$

$$\langle f_e : e \in X \rangle \subseteq \langle f_e : e \in C \rangle$$

And thus,

$$|X| = \text{rank}(\langle f_e : e \in X \rangle) \le \text{rank}(\langle f_e : e \in C \rangle) \le |C| = r_{\mathcal{M}}(X) \le |X|.$$

Therefore, $r_{\mathcal{M}}(X) = |X|$, *i.e.*, X is an independent set in the network matroid. ■

3.3 Optimality of Generic Linear Network Codes

In last section, we have already shown that every independent set in an induced matroid is an independent set in the network matroid, but whether there exists a linear network code that can induce the network matroid has not yet been discussed. In this section, we concentrate on the connection between generic linear network codes and the network matroid.

Theorem 3.4. A normal $\mathbb{P}$-linear network code is generic *if and only if* it induces the network matroid.

Proof. (Sufficiency) Suppose that the induced matroid of a normal $\mathbb{P}$-linear network code C is the network matroid $\mathcal{M}$. Consider a set $X = \{e_1, e_2, \ldots, e_m\}$ of edges that satisfies (12). The rank $r_{\mathcal{M}}(X)$ of X in $\mathcal{M}$ is just the maximal number of edge-disjoint paths starting from data-generating edges and ending at edges in X. Then for any e_j,

$$\begin{aligned} &\text{rank}\left(\left\langle \left\{ f_{e_k} : k \ne j \right\} \right\rangle\right) \\ &< \text{rank}\left(\left\langle \left\{ f_d : d \in \text{In}(T_j) \right\} \right\rangle \cup \left\langle \left\{ f_{e_k} : k \ne j \right\} \right\rangle\right) \\ &\Rightarrow r_{\mathcal{M}}\left(\bigcup_{k \ne j} e_k\right) < r_{\mathcal{M}}\left(\text{In}(v_j) \cup \left(\bigcup_{k \ne j} e_k\right)\right). \end{aligned}$$

Let I_1, I_2 be, respectively, the maximal independent set contained in $\bigcup_{k \ne j} e_k$ and $\text{In}(v_j) \cup \left(\bigcup_{k \ne j} e_k\right)$. Then $|I_1| < |I_2|$. According to (I3) of matroid definition, there exists an element e in $I_2 - I_1 \subseteq \text{In}(T_j)$ such that $I_1 \cup e$ is an independent set, which implies that

(16) $r_{\mathcal{M}}(X - e_j) = r_{\mathcal{M}}(X) - 1$.

Assume that X is a dependent set. Then every $(m–1)$-element subset of X is dependent, otherwise, $r_{\mathcal{M}}(X) = m$ by (16), a contradiction. Denote by Y an $(m–1)$-element subset of X. Let I_X and I_Y be the maximal independent sets contained in X and Y respectively. Since $r_{\mathcal{M}}(Y) = r_{\mathcal{M}}(X)–1$, $|I_Y| < |I_X|$. Then according to (I3) of matroid definition, there exists an element $e_i \in I_X–I_Y$ such that $I_Y \cup e_i$ is independent. Since Y is dependent, there exists an element $e_j \in Y–I_Y$ such that $r_{\mathcal{M}}(Y–e_j) = r_{\mathcal{M}}(Y)$. Consider the $X–e_j$ as $(Y–e_j) \cup e_i$, $r_{\mathcal{M}}(X–e_j) = r_{\mathcal{M}}(Y) + 1 = r_{\mathcal{M}}(X)$, which is a contradiction to that $r_{\mathcal{M}}(X) = r_{\mathcal{M}}(X–e_j) + 1$ for $1 \leq j \leq m$. Therefore X is an independent set, and correspondingly the coding vectors for these edges are linearly independent.

(Necessity) Consider a normal $\mathbb{P}$-linear network code C with the coding vectors f_e. According to Proposition 3.3, every independent set in the induced matroid $(E\backslash E_{DG}, \mathcal{I}_C)$ is an independent set in the network matroid. What remains to show is that every independent set in the network matroid satisfies (12), and thus is in $\mathcal{I}_C$. Moreover, because every independent set in a matroid is contained by at least one base and every subset of a base is an independent set, it suffices to show that every base of the network matroid satisfies (12).

Denote by $\mathcal{B}$ the family of bases of the network matroid. For a base B of the network matroid, denote by $l(B)$ the minimum number of edges contained in the ω edge-disjoint paths starting from data-generating edges and ending at edges in B. Index all bases $\{B_1, B_2, \ldots, B_{|\mathcal{B}|}\}$ in $\mathcal{B}$ according to any non-decreasing order of the value $l(\cdot)$.

First, every base B_i with $l(B_i) = 2\omega$ is a subset of $\mathrm{Out}(head(E_{DG}))$. Then for all edges $e \in B_i$, the coding vectors for edges in $B_i\backslash\{e\}$ could not generate the free module $\mathbb{P}^{\omega} = \langle f_d: d \in \mathrm{In}(head(E_{DG}))\rangle = \langle f_d: d \in E_{DG}\rangle$, and thus (12) is held. Assume that criterion (12) holds for all bases B_i, $1 \leq i \leq k$. Consider the base

$B_{k+1} = \{e_1, e_2, \ldots, e_\omega\}$ of the network matroid. If B_{k+1} does not satisfy (12), there is at least one edge e_j in B_{k+1} such that $\langle f_d : d \in \mathrm{In}(v_j)\rangle \subset \langle f_d : d \in B_{k+1}\backslash\{e_j\}\rangle$. Apparently $e_j \notin \mathrm{Out}(s)$. Thus $\mathrm{In}(v_j)$ is not empty and for every edge $e \in \mathrm{In}(v_j)$, $f_e \in \langle f_d : d \in B_{k+1}\backslash\{e_j\}\rangle$. On the other hand, because B_{k+1} is a base of the network matroid, there exists at least one edge e' in $\mathrm{In}(v_j)$ such that $(B_{k+1}-e_j) \cup e'$ is also a base of the network matroid. Since $l((B_{k+1}- e_j) \cup e') = l(B_{k+1})-1$, $(B_{k+1}-e_j) \cup e'$ is in $\{B_1, B_2, \ldots, B_k\}$. By induction assumption, $(B_{k+1}-e_j) \cup e'$ satisfies (12) and thus $f_{e'} \notin \langle f_d : d \in B_{k+1}-e_j\rangle$, which leads to a contradiction. Therefore, condition (12) holds for B_{k+1} too. ■

Corollary 3.5. The network matroid has the maximum family of independent sets in the family of induced matroids.

The corollary above describes the optimality of generic linear network codes with regard to linear independence. Every generic linear network code induces a representation for the network matroid, and the existence of a generic $\mathbb{P}$-linear network code shown in Theorem 2.16 implies that the network matroid is representable over sufficiently large $\mathbb{P}$.

Corollary 3.6. The network matroid is representable over $\mathbb{P}$ when $|\mathbb{P}| > |\mathcal{B}| + 1$, where $\mathcal{B}$ is the collection of bases.

Every $\mathbb{P}$-representation for the network matroid can be regarded as a set of coding vectors for a $\mathbb{Q}$-linear network code, which is normal and generic when the network is acyclic. This offers a new characterization of generic linear network codes, in terms of the representation for network matroids.

Proposition 3.7. Assume that the network is acyclic. Every $\mathbb{P}$-representation for the network matroid is the set of coding vectors for a generic $\mathbb{Q}$-linear network code.

Proof. A $\mathbb{P}$-representation for the network matroid $(E\backslash E_{DG}, \mathcal{I})$ associates every edge e with an ω-dim column vector $f(e)$, which can be regarded as a vector over $\mathbb{Q}$, such that for every subset X of $E\backslash E_{DG}$, $f(e)$, $e \in X$, are linearly independent if and only if $X \in \mathcal{I}$. For every non-source node v and any $e \in \mathrm{Out}(v)$, since $\{\mathrm{In}(v) \cup e\}$ is not a member in $\mathcal{I}$,

$$f(e) \in \langle f(d), d \in \mathrm{In}(v) \rangle \subseteq \mathbb{Q}^{\omega}.$$

Thus, when $f(e)$, $e \in E_{DG}$ are defined to form the natural basis of $\mathbb{Q}^{\omega}$, $\{f(e), e \in E\}$ is a set of coding vectors for a $\mathbb{Q}$-linear network code $(k_{d,e})$ by Definition 2.2. When the network is acyclic, Corollary 2.5 asserts that $(k_{d,e})$ is normal. Therefore, $(k_{d,e})$ is a generic $\mathbb{Q}$-linear network code by Theorem 3.4.■

When the network contains cycles, the $\mathbb{P}$-representation for the network matroid is not necessarily a set of coding vectors of a generic linear network code. Although it can still be regarded as a set of coding vectors for a $\mathbb{Q}$-linear network code $(k_{d,e})$ subject to conditions (1) and (2), the normality of the code thus generated can not be guaranteed. Therefore, representation for the network matroid can not fully characterize the class of generic linear network codes on a cyclic network.

Example. Figure 3.2 depicts a network with a cycle and a set of vectors from a GF(5)-representation for the network matroid on $E\backslash E_{DG}$, in which every 2-edge set is a base. It also depicts a GF(5)-linear network code C in terms of the coding coefficients induced from the set of vectors in the representation. Since $\det(I_8 - K_C) = 0$, C is not normal.

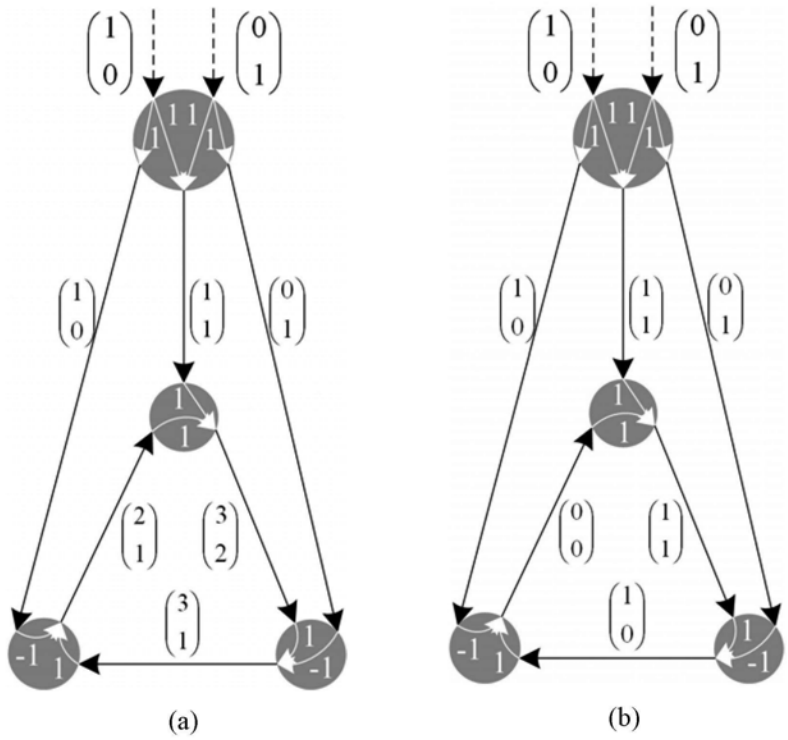


Figure 3.2. (a) Over the network with a cycle, the set of vectors associated with the edge set forms a GF(5)-representation for the network matroid on $E\backslash E_{DG}$, in which every 2-edge set is a base. Correspondingly, the coding coefficients of a GF(5)-linear network code C are induced based on conditions (1) and (2). Since $\det(I_8 - K_C) = 0$, C is not normal. (b) The network code C can determine another set of coding vectors subject to (1) and (2), from which the matroid induced is not the network matroid.

Based on the independence structure in the network matroid, besides generic linear network codes, other optimal linear network codes — linear multicast, linear broadcast, and linear dispersion defined can also be interpreted without the notation of maxflow.

Definition 3.8. Let $\mathcal{M}_N$ be the network matroid. A normal $\mathbb{P}$-linear network code C is qualified as a *linear multicast,* a *linear broadcast* and a *linear dispersion* network code, respectively, if the following conditions for the induced matroid $\mathcal{M}_C$ of C hold:

- $r_{\mathcal{M}_C}(\mathrm{In}(v)) = \omega$ for every non-source node v such that $r_{\mathcal{M}_N}(\mathrm{In}(v)) = \omega$.
- $r_{\mathcal{M}_C}(\mathrm{In}(v)) = r_{\mathcal{M}_N}(\mathrm{In}(v))$ for every non-source node v.
- $r_{\mathcal{M}_C}(\mathrm{In}(\wp)) = r_{\mathcal{M}_N}(\mathrm{In}(\wp))$ for every collection $\wp$ of non-source nodes.

Chapter 4.

Efficient Construction of Optimal Network Codes

This chapter delves into the efficiency issue of code construction. Only *centralized deterministic* algorithms will be considered. In the literature, there has been a variety of polynomial-time[1] algorithms for the construction of linear multicast over an acyclic network, such as [21] and [16]. However, it is not straightforward to adapt these algorithms to a network with cycles due to the lack of a topological order in the network. Hence the code design over a cyclic network meets more trouble. Several polynomial-time algorithms, including [9] and [3], have been proposed over a cyclic network with the special treatment to network cycles. We start this chapter by reviewing these acyclic and cyclic algorithms in Section 4.1. From Section 4.2, we introduce a unified method and its applications to adapt existing acyclic algorithms to cyclic networks.

4.1 Existing Algorithms

4.1.1 Acyclic Case

In this subsection, assume that the network (V, E, s, ω) in consideration is acyclic. Two approaches for the construction of an $\mathbb{F}$-linear multicast will be reviewed. One is the flow path approach exemplified in [21], which requires the cardinality of $\mathbb{F}$ no less than the number of sinks. The other one is the matrix completion approach exemplified in [15], which requires the

[1] Polynomial in the number of edges in the network.

cardinality of $\mathbb{F}$ larger than the number of sinks. We shall first interpret a common initial procedure of all algorithms discussed in this chapter for constructing an $\mathbb{F}$-linear multicast.

Common Initial Procedure. We need first prescribe the coding vectors for data-generating edges to form the natural basis of $\mathbb{F}^{\omega}$. This is subject to condition (1) of a linear network code in Definition 2.1. Then for each sink v, a collection of ω edge-disjoint paths leading from data-generating edges to edges belonging to $\mathrm{In}(v)$, which will be denoted by $\wp_v$, can be identified first. This step can be realized by iteratively searching a flow-augmenting path from s to each v for ω times (See Chapter 26 in [4].) Since finding one augmenting path requires the computational complexity $O(|E|)$, the total computation complexity for this initial step is $O(\delta\omega|E|)$, where δ is the number of sinks. Once $\wp_v$ is established, we can delete the edges that are not on any of the paths in $\cup_{v:\mathrm{sink}} \wp_v$ from the network. The remaining task for constructing an $\mathbb{F}$-linear multicast is just to assign coding coefficients for all adjacent pairs on each path in $\cup_{v:\mathrm{sink}} \wp_v$.

The *flow path* approach exemplified by [21]. The assignment process of coding coefficients is by dealing with one node at a time from upstream to downstream in the network. Through the assignment process, a sink v is always associated with a set of ω edges chosen from different paths in $\wp_v$. This set, which will be denoted by B_v, is initially set to be E_{DG}. During the process of traversing the node set, the edges in B_v are successively changed at the same time until B_v becomes a subset of $\mathrm{In}(v)$ in the end. The key criterion of this approach to assign coding coefficients is just to maintain the coding vectors for edges in B_v linear independent.

In detail, assume that a node n is being considered by the algorithm, and for all sinks v, the coding vectors f_d for edges d in B_v are linearly independent. Without loss of generality, for an edge $e \in \mathrm{Out}(n)$, let $\{v_1, \ldots, v_m\}$ denote the set of sinks such that e is on a path in $\wp_{v_r}$, and by $\{d_1, \ldots, d_m\}$ the subset of $\mathrm{In}(n)$ such that d_r is on the same path as e in $\wp_{v_r}$. Then we can observe

$$\begin{aligned}
&\dim_{\mathbb{F}}(\langle f_{d_i}\colon 1 \le i \le m\rangle \cap \langle f_d\colon d \in B_{v_r}\backslash d_r\rangle)\\
&= \dim_{\mathbb{F}}(\langle f_{d_i}\colon 1 \le i \le m\rangle) + \dim_{\mathbb{F}}(\langle f_d\colon d \in B_{v_r}\backslash d_r\rangle)\\
&\quad -\dim_{\mathbb{F}}(\langle f_{d_i}\colon 1 \le i \le m\rangle \cup \langle f_d\colon d \in B_{v_i}\backslash d_i\rangle)\\
&= \dim_{\mathbb{F}}(\langle f_{d_i}\colon 1 \le i \le m\rangle) + \dim_{\mathbb{F}}(\langle f_d\colon d \in B_{v_r}\backslash d_r\rangle)\\
&\quad -\dim_{\mathbb{F}}(\langle f_d\colon d \in B_{v_r} \cup \{d_1, \ldots, d_m\}\rangle)\\
&= \dim_{\mathbb{F}}(\langle f_{d_i}\colon 1 \le i \le m\rangle) - 1
\end{aligned}$$

Adopt the abbreviation $\dim_{\mathbb{F}}(\langle f_{d_i}\colon 1 \le i \le m\rangle) = \eta$. The number of non-zero vectors in $\langle f_{d_r}\colon 1 \le i \le m\rangle$ and $\langle f_{d_i}\colon 1 \le i \le m\rangle \cap (\cup_{v_i}\langle f_d\colon d \in B_{v_r}\backslash d_r\rangle)$ are then, respectively, $|\mathbb{F}|^{\eta}-1$ and $m\cdot|\mathbb{F}|^{\eta-1}-1$. As a result, when $|\mathbb{F}| > m$, there always exists an assignment of $k_{d_i,e}$, $1 \le i \le m$ such that

(17) $\dim_{\mathbb{F}}(\langle f_e \cup \{f_d\colon d \in B_{v_r}\backslash d_r\}\rangle) = \omega$ for all v_r, where $f_e = \sum_i k_{d_i,e} f_{d_i}$.

In fact, the algorithm introduced in [21] to assign $k_{d_r,e}$ subject to (17) for edge e is performed just under the condition $|\mathbb{F}| \ge m$. It assigns $k_{d_r,e}$ iteratively from 1 to m such that when the $(j+1)^{th}$ one is to be assigned, the subspace $\langle(\sum_{i\le j} k_{d_i,e} f_{d_i}) \cup \{f_d\colon d \in B_{v_r}\backslash d_r\}\rangle$ has full rank ω for all v_r, $r < j+1$. Corresponding to a sink v_r, $r < j+1$, it is shown that there is only one element q_r in $\mathbb{F}$ such that

$$\dim_{\mathbb{F}}\langle(q_r f_{d_{j+1}} + \sum_{i\le j} k_{d_i,e} f_{d_i}) \cup \{f_d\colon d \in B_{v_r}\backslash d_r\}\rangle) = \omega - 1.$$

Therefore, when $k_{d_r,e}$ is assigned to be any value in $\mathbb{F}\backslash\{q_1, \ldots, q_j\}$, $\langle(\sum_{i\le j+1} k_{d_i,e} f_{d_i}) \cup \{f_d\colon d \in B_{v_{j+1}}\backslash d_{j+1}\}\rangle$ has full rank ω. This requires $|\mathbb{F}| \ge j+1$. Since j is upper bounded by $m-1$ and hence it suffices when $|\mathbb{F}| \ge m$. Such iterative

process takes computational complexity $O(\delta^2\omega+\delta\omega^2)$, where δ is the number of sinks, the upper bound of m. Therefore, the total complexity to construct an $\mathbb{F}$-linear multicast takes the computational complexity $O((\delta^2\omega+\delta\omega^2)|E|)$ with the condition $|\mathbb{F}| \geq \delta$.

The *improved flow path* approach exemplified by [24]. As a pre-processing step, the network is transformed to an auxiliary network at the computational complexity $O(\delta^2\omega|E|)$[1], such that every $\mathbb{F}$-linear multicast on the auxiliary network yields an $\mathbb{F}$-linear multicast on the original one. On this equivalent auxiliary network, the number of edges is upper-bounded by $O(\delta^2\omega^3)$, and hence an $\mathbb{F}$-linear multicast can be constructed via the flow path approach in [21] with the computational complexity $O(\delta^4\omega^4 + \delta^3\omega^5)$. In total, it takes $O(\delta^2\omega|E| + \delta^4\omega^4 + \delta^3\omega^5)$ to construct an $\mathbb{F}$-linear multicast on the original network.

The *matrix completion* approach exemplified by [15]. Associate each adjacent pair (d, e) with an indeterminate $x_{d,e}$. Let $\mathbb{F}[*]$ denote the polynomial ring in these indeterminates over the field $\mathbb{F}$, $|\mathbb{F}| > \delta$. Moreover, associate every sink v with an $(|E|-\omega)\times\omega$ matrix P_v and an $|E|\times|E|$ matrix M_v over $\mathbb{F}[*]$. In P_v, the rows and columns are, respectively, indexed by $E\backslash E_{DG}$ and E_{DG}, and all entries that can be indexed by row d and column e such that $d \in \mathrm{In}(v)$ and edges e and d are on a same path in $\wp_v$, are 1s. The matrix M_v is in the form

$$\begin{bmatrix} \left[x_{d,e}\right]_{d\in E_{DG}, e\notin E_{DG}} & 0 \\ I_{|E|-\omega} - \left[x_{d,e}\right]_{d,\, e\notin E_{DG}} & P_v \end{bmatrix},$$

[1] To be more precise, the computation complexity for this transformation is $O(\delta^2\omega^2|V|)$. However, for more transparent complexity comparison with other algorithms in terms of the same variables δ, ω, and $|E|$, $O(\delta^2\omega|E|)$ will be adopted in this thesis.

in which every entry is one of the indeterminates or a scalar. Such a matrix is called a *mixed matrix*. As a polynomial in $\mathbb{F}[*]$, $\det(M_v)$ is nonzero because its evaluation is 1 when we set $x_{d,e}$ to 1 for all adjacent pairs (d, e) on one of the paths in $\wp_v$ and all other $x_{d,e}$ to 0. Thus, M_v has full rank ω. After the initial setup, the algorithm iteratively replaces an indeterminate by a scalar in such a way that the full rank of the matrix M_v is preserved for every sink v. The whole process takes the computational complexity $O(\delta\eta^3 \log \eta)$, where δ is the number of sinks. After all indeterminates are replaced by scalars, the full rank of a matrix can be shown to imply the full rate of data transmission from the source to a sink. To view this, assume that the indeterminates $x_{d,e}$ have been assigned values $k_{d,e} \in \mathbb{F}$ such that for every sink v

$$\det\left(\begin{bmatrix} \left[k_{d,e}\right]_{d\in E_{DG}, e\notin E_{DG}} & \mathbf{0} \\ I_{|E|-\omega} - \left[k_{d,e}\right]_{d,\, e\notin E_{DG}} & P_v \end{bmatrix}\right) \neq 0.$$

Since the network under consideration is *acyclic*, the square matrix $[k_{d,e}]_{d,e\notin E_{DG}}$ is strictly upper triangular and thus

$$\det(I_{|E|-\omega} - [k_{d,e}]_{d,e\notin E_{DG}}) = \det((I_{|E|-\omega} - [k_{d,e}]_{d,e\notin E_{DG}})^{-1}) = 1.$$

Adopt the abbreviation of

$$A = [k_{d,e}]_{d\in E_{DG}, e\notin E_{DG}} \text{ and } B = I_{|E|-\omega} - [k_{d,e}]_{d,e\notin E_{DG}}.$$

Observe that

$$\begin{bmatrix} A & \mathbf{0} \\ B & P_v \end{bmatrix} \cdot \begin{bmatrix} B^{-1}\cdot P_v & B^{-1} \\ -I_\omega & \mathbf{0} \end{bmatrix} = \begin{bmatrix} A\cdot B^{-1}\cdot P_v & \mathbf{0} \\ \mathbf{0} & I_{|E|-\omega} \end{bmatrix},$$

and hence

$$\det(A\cdot B^{-1}\cdot P_v) = \det\left(\begin{bmatrix} A & \mathbf{0} \\ B & P_v \end{bmatrix} \cdot \begin{bmatrix} B^{-1}\cdot P_v & B^{-1} \\ -I_\omega & \mathbf{0} \end{bmatrix}\right) = \pm \det\left(\begin{bmatrix} A & \mathbf{0} \\ B & P_v \end{bmatrix}\right) \neq 0,$$

in other words, the matrix $([k_{d,e}]_{d\in E_{DG},\, e\notin E_{DG}} \cdot (I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})^{-1} \cdot P_v)$ has the full rank ω. On the other hand, the matrix form of the condition (2) in Definition 2.2 for the coding vectors f_e of a linear network code gives

$$[f_e]_{e\notin E_{DG}} \cdot (I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}}) = [k_{d,e}]_{d\in E_{DG},\, e\notin E_{DG}}.$$

For each sink v,

$$\begin{aligned} & \omega \\ & \geq \dim_{\mathbb{F}}(\langle f_e : e\in \mathrm{In}(v)\rangle) \\ & \geq \dim_{\mathbb{F}}([k_{d,e}]_{d\in E_{DG},\, e\notin E_{DG}} \cdot (I_{|E|-\omega}-[k_{d,e}]_{d,e\notin E_{DG}})^{-1} \cdot P_v) \\ & = \omega \end{aligned}$$

When a $\mathbb{D}$-linear multicast is to be constructed, the two algorithms introduced above can easily be adapted via constructing a $\mathbb{Q}$-linear multicast with coding coefficients selected in $\mathbb{D}$, where $\mathbb{Q}$ is the quotient field of $\mathbb{D}$. If a delay function t is given, then a t-causal $\mathbb{D}$-linear multicast can be constructed via constructing a $\mathbb{Q}$-linear multicast with coding coefficients $k_{d,e}$ selected in $z^{t(d,\, e)} \cdot \mathbb{D}$.

4.1.2 Cyclic Case

The upstream-to-downstream node ordering in a network is necessitated by the polynomial time algorithms introduced in the previous subsection to deterministically construct an optimal network code. Over a cyclic network, however, due to the lack of such an order, the efficient optimal code design becomes more challengeable.

Now assume that the network under consideration contains cycles, and the collection $\wp_v$ of ω edge-disjoint *simple*[1] paths leading from E_{DG} to edges

[1] A simple path is the one in which an edge appears at most once.

belonging to In(v) for each sink v has been identified. One approach [13][3] for efficient design of optimal linear network codes is to classify the network cycles for separate treatment. In [3], the cycles are divided into three classes: link cycles, flow cycles, and flow knots.

- A *link cycle* simply means a path $(e_1, e_2, \ldots, e_k, e_1)$.
- A *flow cycle* means a link path $(e_1, e_2, \ldots, e_k, e_1)$ such that each adjacent pair (e_i, e_j) in it belongs to a path in $\wp_v$ for some sink v.
- A *flow knot* is a union of intersected flow cycles.

It is easy to see that in a link cycle which is not a flow cycle, a partial ordering of edges exists. Thus all algorithms over acyclic networks can be adapted to cyclic networks without flow cycles as well. In a cyclic network with flow cycles or flow knots, on the other hand, a partial order of edges does not exist any more. To deal with both classes of cycles, more complex techniques are adopted in [3] such that a t-causal GF(2)[(D)]-linear multicast is claimed to be constructed in polynomial time when the delay function t is prescribed as $t(d, e) = 1$ for all adjacent pairs (d, e).

Although the network may contain flow cycles or knot cycles, there is no flow cycle in each $\wp_v$. By this finding, a series of conference papers [9], [10], and [11] proposed and analyzed a variant (compared with the one in [21]) flow path approach to construct an optimal linear network code on a cyclic network. The key idea is to iteratively deal with the subnetwork consisting of all paths in $\wp_v$ corresponding to each sink v by traversing edges in it in an upstream-to-downstream fashion. An edge is possible to be traversed several times corresponding to different sinks. Given a delay function t, this approach is claimed to construct a causal $\mathbb{F}[(D)]$-linear multicast over any field $\mathbb{F}$ with the computation complexity $O(\delta^3 \omega |E|^3)$ where δ is the number of sinks

4.2 Algorithm Adaptation from Acyclic to Cyclic Networks

4.2.1 A General PID-based Theorem

When a given network $\mathcal{N}$ contains cycles, we shall associate with it a quadratically large acyclic network $\mathcal{N}'$ so that a $\mathbb{P}$-linear multicast on $\mathcal{N}'$ over a PID directly induces a $\mathbb{P}$-linear multicast on $\mathcal{N}$. In this way, efficient algorithms in the literature for the construction of $\mathbb{F}$-linear multicast on an *acyclic* network can be adapted for a cyclic network.

Hereafter let $\mathcal{N}$ denote the network (V, E, s, ω). Nodes of the corresponding acyclic network $\mathcal{N}'$ are on five layers, labeled 0 to 4 from upstream to downstream, as exemplified by Figure 4.1[1]:

- Layer 0 consists of just the source node which generates $|E|-\omega$ outgoing data-generating edges.
- On layer 1, there is a node e_1 corresponding to every edge $e \in E\backslash E_{DG}$.
- On layers 2 and 3, there are nodes e_2 and e_3, respectively, corresponding to every edge $e \in E$.
- Layer 4 consists of a node v_4 corresponding to each node v in $\mathcal{N}$ that is either the source or a sink.

Corresponding to every $e \in E\backslash E_{DG}$, there is the data-generating edge $e_{(1)}$ from s terminating at e_1, and the edge $e_{(2)}$ from e_1 to e_2. Corresponding to every adjacent pair (d, e) in $\mathcal{N}$, there is the edge $\overline{de}$ from e_1 to d_2. Corresponding to every $e \in E$, there is the edge $e_{(3)}$ from e_2 to e_3. Finally, an edge is installed from every layer-3 node e_3, $e \in E\backslash E_{DG}$ to the layer-4 node s_4, and incoming edges to other layer-4 nodes v_4 are prescribed in two steps:

(i). Arbitrarily take ω edge-disjoint paths in $\mathcal{N}$ that lead from s to the node v.

[1] Same as other figures, the layer-0 source node will not be shown.

(ii). For every $e \in E$, install an edge from e_3 to v_4 *unless* $e \in \text{In}(v)$ and is an edge on these paths.

Altogether, the number of edges in $\mathcal{N}'$ is bounded by $|E|+(|E|+|E|^2)+|E|+|E|\cdot|V| \approx |E|^2$.

The construction of the network $\mathcal{N}'$ from $\mathcal{N}$ would appear more transparent when two intermediate steps are presented in subsection 4.2.2 ahead of the theorem proof. Appendix C depicts a slightly larger example to show the connection between $\mathcal{N}$ and $\mathcal{N}'$.

Theorem 4.1. Let C' be a $\mathbb{P}$-linear multicast on $\mathcal{N}'$ subject to the constraint that:

(18) The coding coefficient $k'_{x,y} = 1$ when $(x, y) = (e_{(1)}, e_{(2)})$ or $(e_{(2)}, e_{(3)})$ for some $e \in E\backslash E_{DG}$.

Let C denote the induced $\mathbb{P}$-linear network code $(k_{d,e})$ on $\mathcal{N}$ via

$$(19)\quad k_{d,e} = \begin{cases} k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } d \in E_{DG} \\ -k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } d \in E\backslash E_{DG} \end{cases}$$

Then, the normalization of C is a $\mathbb{P}$-linear multicast.

As will be shown in the next subsection, every layer-4 node in $\mathcal{N}'$ qualifies to be a sink. Based on this fact, a high level understanding of Theorem 4.1 is as follows. On $\mathcal{N}'$, a $\mathbb{P}$-linear multicast C' ensures each layer-4 node to achieve the maximum data reception rate $|E|-\omega$. Moreover, the additional condition (18) for C' and the maximum data reception rate $|E|-\omega$ of s_4 guarantees the nonsingularity of the $\mathbb{P}$-linear network code C on $\mathcal{N}$ induced via the relation (19). The maximum data reception rate $|E|-\omega$ of v_4 in $\mathcal{N}'$ implies the maximum data reception rate ω of sink v in $\mathcal{N}$. The motivation of such a connection is from the matroid duality structures. This motivation will be further developed in Section 4.4.

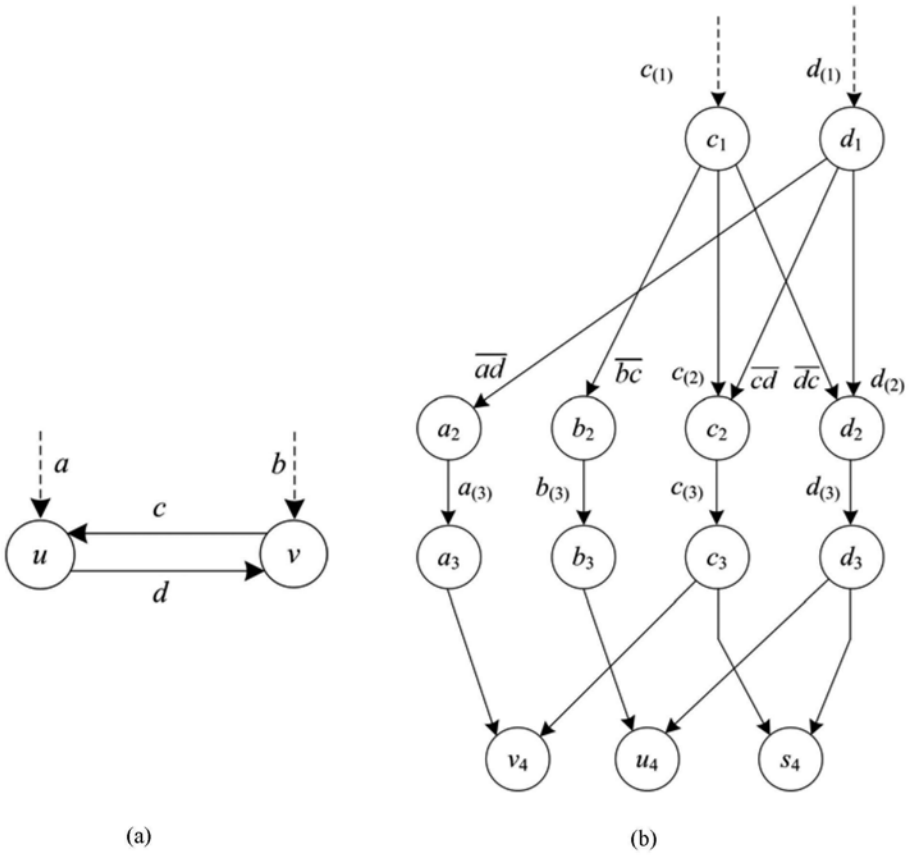


Figure 4.1. (a) The given network $\mathcal{N}$ contains a cycle. (b) The associated acyclic network $\mathcal{N}'$ consists of five layers of nodes (with the top layer source node not shown), while edges are between adjacent layers. The two networks are so related that a linear multicast on $\mathcal{N}'$ subject to a straightforward condition induces a linear multicast on $\mathcal{N}$.

4.2.2 Theorem Proof

The present section is dedicated to proving Theorem 4.1. For enhanced transparency of the connection between $\mathcal{N}$ and $\mathcal{N}'$, we shall first break down the construction of $\mathcal{N}'$ from $\mathcal{N}$ into three steps. This necessitates the notion of a *bipartite network* below, of which the formulation does not totally conform to the formulation of a *network* in Chapter 2.

Definition 4.2. A *bipartite network* is a finite directed graph with the following attributes.

- Nodes are partitioned into *squares* and *triangles*.

- There are two kind of directed links between nodes: An *arrow* connects from a square to a triangle, and an *edge* connects from a triangle to a square. There is exactly one incoming edge to each square and exactly one outgoing edge from each triangle.
- The terminating square of a data-generating edge is called a *source*. There may be multiple sources.

There is a bipartite version of every network. The bipartite version of $\mathcal{N}$, to be denoted by $\mathcal{N}_B$, is constructed by the following "node dilation" process.

- Corresponding to every data-generating edge e in $\mathcal{N}$, there are the square e_2 and the data-generating edge e toward e_2 in $\mathcal{N}_B$.
- Corresponding to every $e \in E \backslash E_{DG}$, there are the triangle e_1, the square e_2, and the edge e connecting from e_1 to e_2 in $\mathcal{N}_B$.
- Corresponding to every adjacent pair (d, e) in $\mathcal{N}$, there is an arrow de connecting from d_2 to e_1.

In summary, every edge e in $\mathcal{N}$ remains an edge in $\mathcal{N}_B$ while every node v in $\mathcal{N}$ is dilated into a subnetwork in $\mathcal{N}_B$ that consists of $|\text{In}(v)|$ squares, $|\text{Out}(v)|$ triangles, and $|\text{In}(v)| \cdot |\text{Out}(v)|$ arrows. The bipartite version of the network in Figure 4.1(a) is depicted by Figure 4.2(a).

There is an acyclic counterpart to every bipartite network. The acyclic version of $\mathcal{N}_B$, to be denoted by $\mathcal{N}'_B$, is constructed by the following "de-cycle" process.

- All squares become triangles and vice versa. Thus the notation e_1 becomes for a square and e_2 for a triangle in $\mathcal{N}'_B$.
- The ω data-generating edges are removed. In replacement, $|E|-\omega$ data-generating edges are created, each toward a different square.

- As a result of the square/triangle inversion, an edge e from e_1 to e_2 in $\mathcal{N}_B$ becomes an arrow in $\mathcal{N}'_B$, which will be denoted as $e_{(2)}$.
- Meanwhile, we reverse the orientation of arrows in $\mathcal{N}_B$ so that they remain arrows in $\mathcal{N}'_B$. Thus an arrow de from d_2 to e_1 in $\mathcal{N}_B$ becomes an arrow from e_1 to d_2 in $\mathcal{N}'_B$, which will be denoted by $\overline{de}$.
- Corresponding to every triangle e_2 in $\mathcal{N}_B$, a new square e_3 and a new edge $e_{(3)}$ from e_2 to e_3 are created for $\mathcal{N}'_B$.

The acyclic counterpart to the bipartite network in Figure 4.2(a) is depicted by Figure 4.2(b). When the square/triangle and arrow/edge distinctions are ignored, the acyclic bipartite network $\mathcal{N}'_B$ becomes a subnetwork of $\mathcal{N}'$. In fact, $\mathcal{N}'_B$ is $\mathcal{N}'$ minus nodes and edges at the top and bottom layers. In summary, the construction of $\mathcal{N}'$ is in three steps: first from $\mathcal{N}$ to the bipartite version $\mathcal{N}_B$, then to the acyclic counterpart $\mathcal{N}'_B$, and finally to $\mathcal{N}'$ by the appendage of the top and bottom layers.

Definition 4.3. A *sink group* in a bipartite network means a collection of squares such that there are disjoint paths tracing from all data-generating edges to all members of the group.

For the bipartite network in Figure 4.2(a), two of the sink groups are $\{a_2, c_2\}$, $\{b_2, d_2\}$. They correspond respectively to the two sink groups $\{b_3, d_3\}$, $\{a_3, c_3\}$ of the acyclic counterpart in Figure 4.2(b). When this acyclic bipartite network expands into the network in Figure 4.1(b), a layer-4 node is created at the downstream of each of these two sink groups.

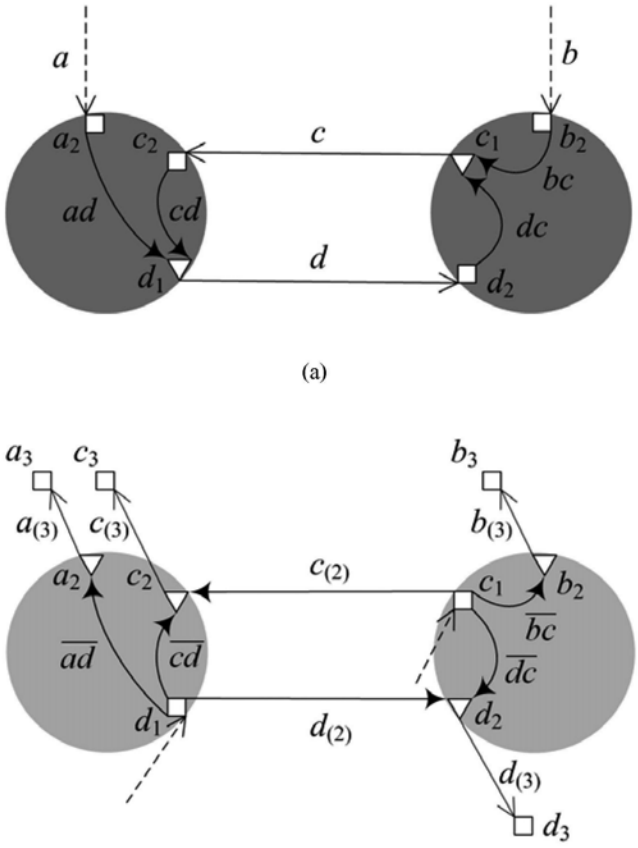


Figure 4.2. Through the "node dilation" process, the network in Figure 4.1(a) is converted into the *bipartite network* of (a). The acyclic counterpart of this bipartite network appears in (b), which is the network in Figure 4.1(b) minus nodes and edges at the top and bottom layers.

Lemma 4.4. Every layer-4 node in $\mathcal{N}'$ is a sink.

Proof. This will be merged to the proof in next subsection as a whole. Fix a layer-4 node v_4 in $\mathcal{N}'$. When it corresponds to the source s in $\mathcal{N}$, the $|E|-\omega$ edge-disjoint paths can simply be prescribed as $\{e_1e_2e_3: e\in E\backslash E_{DG}\}$. When it corresponds to a sink v in $\mathcal{N}$, incoming edges to v_4 are prescribed by the steps (i) and (ii) in last subsection. Thus there are ω layer-3 nodes in $\mathcal{N}'$ that are *not* adjacent to v_4. Define the node set

(20) $P_v = \{e_2$: The corresponding node e_3 is *not* adjacent to v_4 in $\mathcal{N}'\}$.

P_v consists of ω layer-2 nodes, triangles, or squares, respectively, when interpreted as a node set in $\mathcal{N}'$, $\mathcal{N}'_B$, or $\mathcal{N}_B$. To establish v_4 as a sink, it suffices to construct $|E|-\omega$ edge-disjoint paths in $\mathcal{N}'$ from layer-0 source node to v_4. In

view of the special structure of $\mathcal{N}'$, such paths would be equivalent to a matching between all layer-1 nodes and all those layer-2 nodes outside P_v. Over $\mathcal{N}'_B$, that would mean a matching between all terminating squares of data-generating edges and all those triangles outside P_v. It remains to construct such a matching in $\mathcal{N}'_B$.

The aforementioned step (i) specifies ω edge-disjoint paths in $\mathcal{N}$. Translating into the bipartite version $\mathcal{N}_B$ of $\mathcal{N}$, they become ω disjoint paths leading from data-generating edges to squares in P_v. From the nature of $\mathcal{N}_B$ as a bipartite network, a triangle e_1 is on one of these disjoint paths if and only if the corresponding square e_2 is on the same path. Hence there is a matching between all the $|E|-\omega$ triangles in $\mathcal{N}_B$ with all squares outside P_v via:

- If a triangle e_1 is not on these disjoint paths, then e_1 matches with the square e_2 through the edge e.
- If an arrow de is on one of these paths, then the triangle e_1 matches with the square d_2 through de.

This matching in $\mathcal{N}_B$ translates into a desired matching in the acyclic counterpart $\mathcal{N}'_B$ of $\mathcal{N}_B$. This completes the proof of Lemma 4.4. ■

Now we are ready to prove Theorem 4.1 in subsection 4.2.1. Follow the proof of Lemma 4.4 till the equation (20). Fix a sink v in $\mathcal{N}$, which corresponds to the layer-4 node v_4 in $\mathcal{N}'$and is also associated with a set P_v of ω layer-2 nodes in $\mathcal{N}'$. Let f'_d denote the coding vector of C' for an edge $d_{(3)}$ in $\mathcal{N}'$, which corresponds to the edge d in $\mathcal{N}$. Since v_4 is a sink of $\mathcal{N}'$ by Lemma 4.4 and C' is a linear multicast,

(21) the coding vectors f'_d of C' such that $d_2 \notin P_v$ are linearly independent.

When the sink v is substituted by the source s, the associated set of layer-2 nodes in $\mathcal{N}'$corresponds to data-generating edges in $\mathcal{N}$. Analogous to (21),

(22) the coding vectors f'_d of C' such that $d \in E\backslash E_{DG}$ are linearly independent.

Let $\{u_e\colon e\in E\backslash E_{DG}\}$ denote the natural basis of $\mathbb{P}^{|E|-\omega}$ so that the juxtaposition $[u_e]_{e\in E\backslash E_{DG}}$ is the identity matrix $I_{|E|-\omega}$. Let

$$k'_{\overline{de}} = \begin{cases} k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } (d,\ e) \text{ is an adjacent pair in } \mathcal{N} \\ 0, & \text{otherwise} \end{cases}.$$

Then,

$$f'_d = \begin{cases} \sum_{e\in E\backslash In(s)} k'_{\overline{de}}\, u_e, & \text{when } d \in E_{DG} \\ u_d + \sum_{e\in E\backslash \mathrm{In}(s)} k'_{\overline{de}}\, u_e, & \text{when } d \in E\backslash E_{DG} \end{cases}$$

In the matrix form,

(23) $[f'_d]_{d\in E_{DG}} = [\,k'_{\overline{de}}\,]_{e\in E\backslash E_{DG},\ d\in E_{DG}} = \left([k'_{\overline{de}}]_{d\in E_{DG},\,e\in E\backslash E_{DG}}\right)^{\mathbf{T}}$

(24) $[f'_d]_{d\in E\backslash\, E_{DG}} = I_{|E|-\omega} + [k'_{\overline{de}}]_{d,\,e\in E\backslash E_{DG}} = I_{|E|-\omega} + \left([k'_{\overline{de}}]_{d,\,e\in E\backslash E_{DG}}\right)^{\mathbf{T}}$

In terms of $k'_{\overline{de}}$, formula (19) in Theorem 4.1 for the $\mathbb{P}$-linear network code $C = (k_{d,e})$ on $\mathcal{N}$ becomes

$$k_{d,e} = \begin{cases} k'_{\overline{de}}, & \text{when } d \in E_{DG} \\ -k'_{\overline{de}}, & \text{when } d \in E\backslash E_{DG} \end{cases}$$

The $|E|\times|E|$ matrix $I_{|E|}-K_C$ can be expressed as

$$I_{|E|}-K_C = \begin{bmatrix} I_\omega & -[k_{d,e}]_{d\in E_{DG},\,e\in E\backslash E_{DG}} \\ \mathbf{0} & I_{|E|-\omega} - [k_{d,e}]_{d,\,e\in E\backslash E_{DG}} \end{bmatrix} = \begin{bmatrix} I_\omega & -[k'_{\overline{de}}]_{d\in DGC,\,e\in E\backslash E_{DG}} \\ \mathbf{0} & I_{|E|-\omega} + [k'_{\overline{de}}]_{d,\,e\in E\backslash E_{DG}} \end{bmatrix}$$

Thus, from (23) and (24),

$$I_{|E|}-K_C = \begin{bmatrix} I_\omega & -\left([f'_d]_{d\in E_{DG}}\right)^{\mathbf{T}} \\ \mathbf{0} & \left([f'_d]_{d\in E\backslash E_{DG}}\right)^{\mathbf{T}} \end{bmatrix}$$

This implies that

(25) $\det(I_{|E|}-K_C) = \det([f'_d]_{d\in E\backslash E_{DG}})$

From (22), $\det([f'_d]_{d\in E\setminus E_{DG}}) \neq 0$. The equation (25) then proves the nonsingularity of the $\mathbb{P}$-linear network code C on $\mathcal{N}$. To prove Theorem 4.1 by Lemma 2.11, it remains to establish C as a $\mathbb{Q}$-linear multicast. So we shall verify:

(26) The coding vectors of C over $\mathbb{Q}$ for edges e such that $e_2 \in P_v$ are linearly independent.

Denote by $f_e \in \mathbb{Q}^\omega$ the coding vectors of C on $\mathcal{N}$. The matrix form of the equation (2) in Definition 2.2 for the coding vectors f_e of a $\mathbb{P}$-linear network code gives

$$[f_e]_{e\in E\setminus E_{DG}} \cdot (I_{|E|-\omega} - [k_{d,e}]_{d,e\in E\setminus E_{DG}}) = [k_{d,e}]_{d\in E_{DG},\, e\in E\setminus E_{DG}}$$

Through (23) and (24), this can be rewritten as

$$[f_e]_{e\in E\setminus E_{DG}} \cdot ([f'_d]_{d\in E\setminus E_{DG}})^{\mathbf{T}} = ([f'_d]_{d\in E_{DG}})^{\mathbf{T}}$$

The matrix $[f'_d]_{d\in E\setminus E_{DG}}$ over $\mathbb{Q}$ is invertible because of (22). Adopt the abbreviation of

$$\text{(27)} \qquad \begin{cases} A = ([f'_d]_{d\in E\setminus E_{DG}})^{\mathbf{T}}, \\ B = ([f'_d]_{d\in E_{DG}})^{\mathbf{T}}, \text{ and} \\ M = B \cdot A^{-1}. \end{cases}$$

Thus, $[f_e]_{e\in E\setminus E_{DG}} \cdot A = B$ and hence $[f_e]_{e\in E\setminus E_{DG}} = M$. Equivalently,

(28) $[f_e]_{e\in E} = [I_\omega \mid M]$

Note that M is an $\omega\times(|E|-\omega)$ matrix, in which rows are indexed by E_{DG} and columns by $E\setminus E_{DG}$. Thus columns in $[I_\omega \mid M]$ are indexed by E. The statement (26) to be verified for the proof of Theorem 4.1 asserts the linear independence over $\mathbb{Q}$ among those columns e in the matrix $[f_e]_{e\in E}$ that correspond to $e_2 \in P_v$. Applying the matroid duality theorem as stated in Lemma 4.5 below to $\eta = |E|-\omega$ and $\Lambda = P_v$, this assertion is equivalent to the linear independence among those columns d in the $(|E|-\omega)\times|E|$ matrix $[M^{\mathbf{T}} \mid$

$I_{|E|-\omega}]$ that correspond to $d_2 \notin P_v$. Note that columns in $[M^{\mathbf{T}} \mid I_{|E|-\omega}]$ are also indexed by E. By definition,

$$\begin{aligned}
&[M^{\mathbf{T}} \mid I_{|E|-\omega}] \\
&= [(B \cdot A)^{\mathbf{T}} \mid I_{|E|-\omega}] \\
&= [(A^{-1})^{\mathbf{T}} \cdot B^{\mathbf{T}} \mid I_{|E|-\omega}] \\
&= (A^{-1})^{\mathbf{T}} \cdot [B^{\mathbf{T}} \mid A^{\mathbf{T}}] \\
&= (A^{-1})^{\mathbf{T}} \cdot [[f'_d]_{d \in E_{DG}} \mid [f'_d]_{d \in E \setminus E_{DG}}] \\
&= (A^{-1})^{\mathbf{T}} \cdot [f'_d]_{d \in E}
\end{aligned}$$

From (21), those columns d in the matrix $[f'_d]_{d \in E}$ that correspond to $d_2 \notin P_v$ are linearly independent. Hence the same collection of columns in the matrix $(A^{-1})^{\mathbf{T}} \cdot [f'_d]_{d \in E} = [M^{\mathbf{T}} \mid I_{|E|-\omega}]$ are also linearly independent. This completes the proof of Theorem 4.1.

Lemma 4.5. (Duality of vector matroids) Let M be an $\omega \times \eta$ matrix over $\mathbb{P}$. Label the columns in both matrices $[I_\omega \mid M]$ and $[M^{\mathbf{T}} \mid I_\eta]$ by an ordered set of the cardinality $\omega + \eta$. For any set Λ of ω labels, the columns labelled by Λ in one of the matrices are linearly independent vectors if and only if the columns *not* labelled by Λ in the other matrix are linearly independent vectors.

Proof. Consider the $(\omega+\eta)\times(\omega+\eta)$ matrix, which will be denoted by A, over $\mathbb{P}$ in the form

$$\begin{array}{c} \\ 1, \ldots, \omega \\ 1 \\ \ldots, \\ \omega \\ \omega+1 \\ \ldots, \\ \omega+\eta \end{array}
\begin{array}{c}
\begin{array}{cc} \quad 1, \ldots, \omega & \quad \omega+1, \ldots, \omega+\eta \end{array} \\
\left(\begin{array}{cc} I_\omega & M \\ \mathbf{0} & I_\eta \end{array} \right)
\end{array}$$

The matrix $[I_\omega \mid M]$, consists of the first ω rows of A, and the transpose of the matrix $[M^{\mathbf{T}} \mid I_\eta]$ consists of the last η columns in A. Label the rows and columns in A by $\{1, 2, \ldots, \omega+\eta\}$. The theorem is equivalent to say that for any

set Λ of ω labels, the columns labelled by Λ in $[I_\omega \mid M]$ are linearly independent vectors if and only if the rows not labelled by Λ in $[M^{\mathbf{T}} \mid I_\eta]^{\mathbf{T}}$ are linearly independent vectors. When we exchange the rows and columns in A indexed by any two labels $a_1, a_2 \leq \omega$ (and the labels are correspondingly exchanged at the same time,) it does not change the linear independence among rows in $[M^{\mathbf{T}} \mid I_\eta]^{\mathbf{T}}$ and columns in $[I_\omega \mid M]$ indexed by any set of labels. Similarly, when we exchange the columns and rows in A indexed by any two labels $\omega < a_1, a_2 \leq \omega+\eta$ (and the labels are correspondingly exchanged at the same time,) it does not change the linear independence among columns in $[I_\omega \mid M]$ and rows in $[M^{\mathbf{T}} \mid I_\eta]^{\mathbf{T}}$ indexed by any set of labels. Therefore, without loss of generality, we may assume that any set of ω labels can be rearranged to start from label k+1 to label $k+\omega$ for some $k \leq \omega$. Then the matrix A can be decomposed as

$$
\begin{array}{c}
 \\
1, \ldots, k \\
k+1, \ldots, \omega \\
\omega+1, \ldots, \omega+k \\
\omega+k+1, \ldots, \omega+\eta
\end{array}
\begin{array}{c}
\begin{array}{cccc}
1, \ldots, k & k+1, \ldots, \omega & \omega+1, \ldots, \omega+k & \omega+k+1, \ldots, \omega+\eta
\end{array} \\
\begin{pmatrix}
I_k & 0 & M_1 & M_2 \\
0 & I_{\omega-k} & M_3 & M_4 \\
 & & I_k & 0 \\
 & \mathbf{0} & & \\
 & & 0 & I_{\eta-k}
\end{pmatrix}
\end{array}
$$

If the columns in $[I_\omega \mid M]$ indexed by $\{k+1, \ldots, k+\omega\}$ are linearly independent vectors, then the submatrix M_1 of M has full rank k. Since the concatenation of rows in $[M^{\mathbf{T}} \mid I_\eta]^{\mathbf{T}}$ indexed by $\{1, \ldots, k\} \cup \{\omega+k+1, \ldots, \omega+\eta\}$ forms the matrix

$$\begin{pmatrix} M_1 & M_2 \\ \mathbf{0} & I_{n-k} \end{pmatrix},$$

whose determinant is equal to $\det(M_1)\cdot\det(I_{n-k}) \neq 0$, the said rows are linearly independent vectors. Similarly, if the rows in $[M^{\mathbf{T}} \mid I_\eta]^{\mathbf{T}}$ indexed by $\{1, \ldots, k\}\cup\{\omega+k+1, \ldots, \omega+\eta\}$, then the M_1 has full rank k, which implies that the matrix

$$\begin{pmatrix} \mathbf{0} & M_1 \\ I_{\omega-k} & M_3 \end{pmatrix},$$

has the full rank ω. This matrix is exactly the concatenation of columns vectors in $[I_\omega \mid M]$ indexed by $\{k+1, \ldots, k+\omega\}$, which are in turn linearly independent. ■

4.2.3 Adaptation of Existing Algorithms

In subsection 4.1.1, we have reviewed two of the most popular deterministic polynomial time algorithms to construct an $\mathbb{F}$-linear multicast on an acyclic network when $|\mathbb{F}|$ is no less than δ and $\delta + 1$, respectively, where δ is the number of sinks. Via both algorithms, efficient construction of a $\mathbb{P}$-linear multicast subject to (18) for sufficiently large $\mathbb{P}$ can also be achieved via establishing a $\mathbb{Q}$-linear multicast with coding coefficients in $\mathbb{P}$. The present section will be devoted to demonstrating how they can be easily adapted to construct an $\mathbb{F}$-linear multicast on $\mathcal{N}'$ under the constraint (18) in Theorem 4.1. In fact, we shall impose one further constraint on such construction:

- When there is a unique incoming edge x to a node u in an acyclic network, the coding coefficient for every adjacent pair in the form (x, y) can be multiplied into immediately downstream coding coefficients. Thus the

coding coefficient for (x, y) can be fixed to 1. This fact applies to every layer-1 or -3 node in $\mathcal{N}'$.

This constraint and (18) on the construction of $\mathbb{F}$-linear multicast on $\mathcal{N}'$ can be combined into:

(29) Coding coefficients need be calculated only for adjacent pairs in the form of $(\overline{de}, d_{(3)})$. Such adjacent pairs in $\mathcal{N}'$ are in one-to-one correspondence with adjacent pairs in $\mathcal{N}$.

Next we demonstrate that an $\mathbb{F}$-linear multicast subject to (29) on the special topology of $\mathcal{N}'$ can easily be constructed by any efficient acyclic algorithms in the literature via two of the most popular approaches which have been introduced in subsection 4.1.1. Since a common initial procedure for both algorithms is to identify edge-disjoint paths leading from data-generating edges to each sink, we can prescribe that over the special topology of $\mathcal{N}'$,

- For the sink s_4, all $|E|-\omega$ edge-disjoint paths leading from data-generating edges to s_4 can be chosen to be in the form of $(e_{(1)}, e_{(2)}, e_{(3)}, e_3s_4)$, where $e \in E\backslash E_{DG}$.

It should be noted that when $\mathcal{N}$ contains cycles, the paths prescribed above are not the unique choice. However, when an $\mathbb{F}$-linear multicast on $\mathcal{N}$ has nonzero coding coefficients for adjacent pairs on the paths $(e_{(1)}, e_{(2)}, e_{(3)}, e_3s_4)$, $e \in E\backslash E_{DG}$, the normality of the corresponding $\mathbb{F}$-linear network code on $\mathcal{N}$ can be guaranteed. Under this prescription, the existence of an $\mathbb{F}$-linear multicast subject to (29) above on $\mathcal{N}'$ can be shown in the same manner as the proof of Theorem 2.10.

The *flow path* approach exemplified by [21]. In dealing with the node e_1 during the assignment process, since the in-degree of e_1 is 1, the adjacent pair

$(e_{(1)}, e_{(2)})$ can naturally be assigned with the coding coefficient 1. In dealing with the node e_2, where $e \in E\backslash E_{DG}$, we can prescribe to always first assign the coefficient for the adjacent pair $(e_{(2)}, e_{(3)})$. Thus the coefficient 1 can be ensured to assign to $(e_{(2)}, e_{(3)})$ by the algorithms.

The *improved flow path* approach exemplified by [24]. As the pre-processing step, the original cyclic network $\mathcal{N}$ can be transformed to the equivalent auxiliary network with the number of edges bounded by $O(\delta^2\omega^3)$. Next, the acyclic network $\mathcal{N}'$ is constructed based on the auxiliary network instead of the original one, and then the flow path approach can be applied on $\mathcal{N}'$.

The *matrix completion* approach exemplified by [15]. Associate each adjacent pair in the form $(\overline{de}, d_{(3)})$ with an indeterminate $x_{d,e}$. These indeterminates can be used as coding coefficients, together with pre-assigned 0-1 coding coefficients for all other adjacent pairs, for an $\mathbb{F}[*]$-linear network code C' subject to (29) on $\mathcal{N}'$. According to the general treatment suggested in [15], each layer-4 sink is associated with an $O(|E|^2)$ by $O(|E|^2)$ mixed matrix over $\mathbb{F}[*]$ with full rank, and hence an $\mathbb{F}$-linear multicast can routinely be constructed. On the other hand, due to the special topology of $\mathcal{N}'$, we can in fact associated each layer-4 sink with a $(|E|-\omega)\times(|E|-\omega)$ mixed matrix over $\mathbb{F}[*]$ with full rank as follows. Denote by f'_e the coding vector of the $\mathbb{F}[*]$-linear network code C' for the edge $e_{(3)}$ corresponding to each edge e in $\mathcal{N}$. For every sink v_4 of $\mathcal{N}'$, juxtapose the coding vectors f'_e, where e_3 is adjacent to v_4, into an $(|E|-\omega)\times(|E|-\omega)$ matrix over $\mathbb{F}[*]$. In the current adaptation of the matrix completion approach, this is the matrix associated with the sink v_4. It is an $(|E|-\omega)\times(|E|-\omega)$ matrix. The matrix is nonsingular, because the guaranteed existence of an $\mathbb{F}$-linear multicast subject to (29) on $\mathcal{N}'$ says that there is a way to replace each indeterminate $x_{d,e}$ by a scalar $k_{d,e}$

such that each resulting $(|E|-\omega)\times(|E|-\omega)$ matrix over $\mathbb{F}$ is nonsingular. The current adaptation of the matrix completion approach is the iterative process starting with the δ+1 $(|E|-\omega)\times(|E|-\omega)$ matrices over $\mathbb{F}[*]$, where δ is the number of sinks in $\mathcal{N}$.

4.2.4 Analysis of Computational Complexity

According to Theorem 4.1, once a $\mathbb{P}$-linear multicast on $\mathcal{N}'$ subject to (18) is constructed, a $\mathbb{P}$-linear multicast on $\mathcal{N}$ can be immediately induced via (19) with the computational complexity $O(|E|^2)$. Although the number of edges in $\mathcal{N}'$ is approximately quadratically larger than the one in $\mathcal{N}$, as what has been addressed in (29) in the subsection 4.2.3, the actual number of coding coefficients to be determined is exactly the same as the one in $\mathcal{N}$. The only difference to adapt acyclic algorithms to the cyclic case via (19) is that the coding vectors to deal with have the dimension $|E|-\omega$ instead of ω. In the remaining of the present subsection, we shall calculate the complexities to construct a $\mathbb{P}$-linear multicast subject to (18) over the special topology of $\mathcal{N}'$ via both the flow path approach and the matrix completion approach, and then compare them with the construction complexity via the cyclic algorithm claimed in [9].

Recall that each layer-4 node in $\mathcal{N}'$ qualifies as a sink. As a common initial procedure, $|E|-\omega$ edge-disjoint paths shall be identified from the source node to each layer-4 node v_4. Instead of finding the edge-disjoint paths on $\mathcal{N}'$ directly for each sink v_4, the collection $\wp_v$ of ω edge-disjoint paths on $\mathcal{N}$ leading from s to v can be first established with the complexity $O(\delta\omega|E|)$, where δ is the number of sinks. Then, as justified in the proof of Lemma 4.4

in subsection 4.2.2, we can immediately establish $|E|-\omega$ edge-disjoint paths leading from data-generating edges to v_4 in the form

- $(e_{(1)}, e_{(2)}, e_{(3)}, e_3v_4)$, where e is an edge in $\mathcal{N}$ but not on any path in $\wp_v$.
- $(e_{(1)}, \overline{de}, d_{(3)}, d_3v_4)$, where (d, e) is an adjacent pair in $\mathcal{N}$ and is on a same path in $\wp_v$.

Also, the sink s_4 in $\mathcal{N}'$ can be associated with the paths $(e_{(1)}, e_{(2)}, e_{(3)}, e_3s_4)$, $e \in E\backslash E_{DG}$, which is a prerequisite to construct a linear multicast subject to (18).

When the flow path approach is applied in a given *acyclic* network, the computational complexity is $O(\eta\delta\omega(\delta+\eta))$ to construct a $\mathbb{P}$-linear multicast, where δ is the number of sinks in the given acyclic network and η is the number of edges. Specific to the acyclic $\mathcal{N}'$ constructed from $\mathcal{N}$, since the number of coding coefficients to be determined is exactly the one in $\mathcal{N}$, this approach can construct a $\mathbb{P}$-linear multicast subject to (29) on $\mathcal{N}'$ at the computational complexity of $O(|E|\cdot(\delta+1)\cdot(|E|-\omega)\cdot(\delta+1+|E|-\omega)) = O(|E|^3\cdot\delta)$, where δ is the number of sinks in $\mathcal{N}$. If the acyclic network $\mathcal{N}'$ is no longer constructed from $\mathcal{N}$ but from its equivalent auxiliary network as proposed by [24] at the construction complexity $O(|E|\delta^2\omega)$ instead, then the computational complexity to construct a $\mathbb{P}$-linear multicast subject to (29) on $\mathcal{N}'$ becomes $O(\delta^2\omega^3\cdot(\delta+1)\cdot(\delta^2\omega^3-\omega)\cdot(\delta+1+\delta^2\omega^3-\omega)) = O(\delta^7\omega^9)$.

When the matrix completion approach is applied in a given *acyclic* network, the computational complexity is $O(\delta\eta^3\log\eta)$ to construct a $\mathbb{P}$-linear multicast, where δ is the number of sinks in the given acyclic network and η is the number of edges. Here $O(\eta)$ also refers to the dimension of the mixed matrix associated with each sink. Specific to the acyclic $\mathcal{N}'$ constructed from $\mathcal{N}$, as asserted at the end of last paragraph, each matrix can be associated with a mixed matrix to be completed with dimension $|E|-\omega$ instead of $O(|E|^2)$, the

total number of edges in $\mathcal{N}'$. Hence, this approach can construct a $\mathbb{P}$-linear multicast subject to (29) on $\mathcal{N}'$ at the computational complexity of $O((\delta+1)\cdot(|E|-\omega)^3\log(|E|-\omega)) = O(\delta\cdot|E|^3\log|E|)$

Below is a comparison among various deterministic algorithms to construct a linear multicast in both acyclic and cyclic case.

	Acyclic case	Cyclic case
Flow path approach [21]	$O((\delta^2\omega+\delta\omega^2)\|E\|)$	$O(\delta\|E\|^3)$ (via Theorem 4.6)
Improved flow path approach [24]	$O(\|E\|\delta^2\omega+\omega^4\delta^3(\delta+\omega))$	$O(\|E\|\delta^2\omega + \omega^9\delta^7)$ (via Theorem 4.6)
Matrix completion approach [15]	$O(\delta\cdot\|E\|^3\log\|E\|)$	$O(\delta\cdot\|E\|^3\log\|E\|)$ (via Theorem 4.6)
Flow path cyclic algorithm [9]	—	$O(\delta^3\omega\cdot\|E\|^3)$

Table 4.1. Computational complexities to construct a $\mathbb{P}$-linear multicast on the network $\mathcal{N} = (V, E, \omega, s)$. δ is the number of sinks in $\mathcal{N}$.

4.3 Construction of Causal Network Codes

The main theorem in this chapter deals with efficient code construction in the PID-based general case. The present section focuses on the construction of DVR-based causal linear multicast under the same framework and an interesting property specific to convolutional network codes.

When a delay function t on $\mathcal{N}$ is given before the optimal code construction, we can correspondingly define a delay function t' on $\mathcal{N}'$ via

(30) $t'(x, y) = t(d, e)$ if $x = \overline{de}$ and $y = d_{(3)}$ for some adjacent pair (d, e) in $\mathcal{N}$.

Else, $t'(x, y) = 0$.

Thus, a counterpart of Theorem 4.1 in subsection 4.2.1 for causal $\mathbb{D}$-linear multicast can be developed.

Theorem 4.6. Given a delay function t on $\mathcal{N}$ and the corresponding delay function t' on $\mathcal{N}'$ prescribed by (30), let C' be a t'-causal $\mathbb{D}$-linear multicast on $\mathcal{N}'$ subject to

(31) The coding coefficient $k'_{x,y} = 1$ when $(x, y) = (e_{(1)}, e_{(2)})$ or $(e_{(2)}, e_{(3)})$ for some $e \in E\backslash E_{DG}$.

and C the corresponding $\mathbb{D}$-linear network code via

$$(32)\quad k_{d,e} = \begin{cases} k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } d \in E_{DG} \\ -k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } d \in E\backslash E_{DG} \end{cases}$$

Then, C is a t-causal $\mathbb{D}$-linear multicast.

Proof. Applying Theorem 4.1 to $\mathbb{P} = \mathbb{D}$, the normalization of C is a $\mathbb{D}$-linear multicast on $\mathcal{N}$. Then, C is a $\mathbb{Q}$-linear multicast on $\mathcal{N}$ by Lemma 2.11 in Section 2.2 where $\mathbb{Q}$ in this case is the quotient field of $\mathbb{D}$. In view of the close relationship between the two formulas (30) and (31), one can see that the t'-causality of C' implies the t-causality of C. Thus, C is also a normal $\mathbb{D}$-linear network code by Proposition 2.20. Denote the coding vectors of C by $f_e \in \mathbb{D}^\omega$. For every sink v,

$$\text{rank}_{\mathbb{D}}(\langle f_e\colon e \in \text{In}(v)\rangle) = \text{rank}_{\mathbb{Q}}(\langle f_e\colon e \in \text{In}(v)\rangle) = \omega \qquad \blacksquare$$

Recall that deterministic design of a $\mathbb{P}$-linear multicast on a given network can be achieved via existing algorithms by constructing a $\mathbb{Q}$-linear multicast with all coding coefficients in $\mathbb{P}$ when $\mathbb{P}$ is sufficiently large. Similarly, a t-causal $\mathbb{D}$-linear multicast can also be designed on the given network via existing algorithms by constructing a $\mathbb{Q}$-linear multicast, here $\mathbb{Q}$ being the quotient field of $\mathbb{D}$, with each coding coefficients $k_{d,e}$ chosen from $z^{t(d,\,e)} \cdot \mathbb{D} \subset \mathbb{D} \subset \mathbb{Q}$. Since the ideal $z^{t(d,\,e)} \cdot \mathbb{D}$ contains infinite elements, a t-causal $\mathbb{D}$-linear multicast can always be established.

In practice when a linear multicast is to be designed on a network, we may encounter the scenario that the network delay function is not evaluated, and the only information is the network topology. Thus Theorem 4.6 does no longer work. Next we provide a solution to this task when the special case of DVR-based network codes — convolutional network codes — is considered.

Theorem 4.7. Let C' be an $\mathbb{F}$-Linear multicast on $\mathcal{N}'$ subject to

(33) The coding coefficient $k'_{x,y} = 1$ when $(x, y) = (e_{(1)}, e_{(2)})$ or $(e_{(2)}, e_{(3)})$ for some $e \in E\backslash E_{DG}$

and $C = (k_{d,e})$ the corresponding $\mathbb{F}$-linear network code via

$$(34)\quad k_{d,e} = \begin{cases} k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } d \in E_{DG} \\ -k'_{e_{(1)},\overline{de}} \cdot k'_{\overline{de},d_{(3)}}, & \text{when } d \in E\backslash E_{DG} \end{cases}$$

Given an arbitrary delay function t on $\mathcal{N}$, a t-causal $\mathbb{F}[(D)]$-linear multicast $H = (h_{d,e})$ can be constructed by:

$$(35)\quad h_{d,e} = k_{d,e} \cdot D^{t(d,e)}$$

Proof. Applying Theorem 4.1 to $\mathbb{P} = \mathbb{F}$, the normalization of the $\mathbb{F}$-linear network code C is an $\mathbb{F}$-linear multicast on $\mathcal{N}$. Applying Lemma 2.11 in Section 2.2 to $\mathbb{P} = \mathbb{F} = \mathbb{Q}$, we find C itself an $\mathbb{F}$-linear multicast. Let μ be the mapping from $\mathbb{F}[(D)]$ to $\mathbb{F}$ that preserves $\mathbb{F}$ and maps D to 1. Thus, $\mu(h_{d,e}) = k_{d,e}$. Since $\mu(a + b) = \mu(a) + \mu(b)$ and $\mu(ab) = \mu(a)\mu(b)$ for any two elements $a, b \in \mathbb{F}[(D)]$, μ is a homomorphism from $\mathbb{F}[(D)]$ to $\mathbb{F}$. Substituting $\mathbb{P}$, $\mathbb{P}'$, and C in Theorem 2.12 in Section 2.2 with $\mathbb{F}[(D)]$, $\mathbb{F}$, and H, respectively, we find the normalization of H an $\mathbb{F}[(D)]$-linear multicast. Clearly, H is t-causal, and thus normal by Proposition 2.20. Let f_e and g_e, respectively, denote the coding vectors of H and its normalization. Then, $g_e = \det(I_{|E|} - K_C)f_e$ for $e \notin E_{DG}$ according to Corollary 2.7. For every sink v,

$$\text{rank}_{\mathbb{F}[(D)]}(\langle f_e : e \in \text{In}(v)\rangle) = \text{rank}_{\mathbb{F}[(D)]}(\langle g_e : e \in \text{In}(v)\rangle) = \omega \qquad \blacksquare$$

An $\mathbb{F}$-*convolutional network code* means an $\mathbb{F}[(D)]$-linear network code, and an $\mathbb{F}$-*convolutional multicast* means an $\mathbb{F}[(D)]$-linear multicast. Given a delay function t on $\mathcal{N}$, Theorem 4.7 adapts an $\mathbb{F}$-linear multicast on $\mathcal{N}'$ under the constraint (33) into a t-causal $\mathbb{F}$-convolutional multicast on $\mathcal{N}$ through the simple formula (35). The adaptation algorithm is *independent* of the delay function t except for the plain appearance of t in (35). Therefore, part of the coding coefficients of a convolutional multicast can be designed ahead of the knowledge of delay function and it can be adapted with an arbitrary one.

Definition 4.8. An $\mathbb{F}$-*convolutional network code* means an $\mathbb{F}[(D)]$-linear network code $(k_{d,e})$. It is qualified as a *delay invariant* $\mathbb{F}$-*convolutional multicast* if for an arbitrary delay function t, the t-causal $\mathbb{F}$-convolutional network code $(h_{d,e}) = (k_{d,e} \cdot D^{t(d,e)})$ is an $\mathbb{F}[(D)]$-linear multicast.

Proposition 4.9. An $\mathbb{F}$-linear multicast is a delay invariant $\mathbb{F}$-convolutional multicast.

Proposition 4.9 asserts that every $\mathbb{F}$-linear multicast, which can also be regarded as an $\mathbb{F}$-convolutional multicast (but not causal), is naturally delay invariant. Then the construction method in the previous section provides one possible way to construct a delay invariant convolutional multicast. However, an $\mathbb{F}$-linear multicast may not exist when $\mathbb{F}$ is not sufficiently large whereas an $\mathbb{F}[(D)]$-linear multicast always exists for any field $\mathbb{F}$ due to the infinite cardinality of $\mathbb{F}[(D)]$. Thus an interesting question, which has not been solved yet, arises:

- Does there exist a delay invariant $\mathbb{F}$-convolutional multicast over *any* field $\mathbb{F}$?

We conjecture that such an optimal code exists over every field $\mathbb{F}$ because of the following two reasons. Firstly, the coding coefficients in such a code can be chosen from an infinitely large data ensemble $\mathbb{F}[(D)]$. Secondly, we note an interesting fact, which will be illustrated in the next example, that when a delay invariant $\mathbb{F}$-convolutional multicast is regarded as an $\mathbb{F}[(D)]$-linear network code, it may not be normal at all. This property further relaxes the choice of coding coefficients for a delay invariant convolutional multicast and thus potentially provides a new direction for the study of code existence problem.

Example. Let $\mathbb{F}$ be an arbitrary field. Figure 4.3(a) depicts an $\mathbb{F}$-convolutional network code $(k_{d,e})$ on the shuttle network in terms of coding coefficients. It is not normal, but is qualified as a delay invariant convolutional multicast. The coding coefficients and coding vectors of the $\mathbb{F}$-convolutional multicast $(k_{d,e} \cdot D^{t(d,e)})$ are illustrated in Figure 4.3(b), where the delay function t associates to each adjacent pair an indeterminate.

Now that a delay invariant $\mathbb{F}$-convolutional multicast is not necessarily normal, we may attempt to design a method to construct a t-causal $\mathbb{F}$-convolutional multicast $(k_{d,e} \cdot D^{t(d,e)})$ with a prescribed delay function t such that with any other delay function t', $(k_{d,e} \cdot D^{t'(d,e)})$ is a t'-causal $\mathbb{F}$-convolutional multicast. As reviewed in section 4.1.2, given a delay function t, the approach proposed in [9] is able to efficiently construct a t-causal $\mathbb{F}$-convolutional multicast $(k_{d,e} \cdot D^{t(d,e)})$ over any field $\mathbb{F}$, from which one may naturally think whether the $\mathbb{F}$-convolutional network code $(k_{d,e})$ is qualified as a delay invariant convolutional multicast. Unfortunately, we next offer an example to show that induced from a t-causal $\mathbb{F}$-convolutional multicast $(k_{d,e} \cdot D^{t(d,e)})$, the

$\mathbb{F}$-convolutional network code ($k_{d,e}$) is not necessarily a delay invariant convolutional multicast.

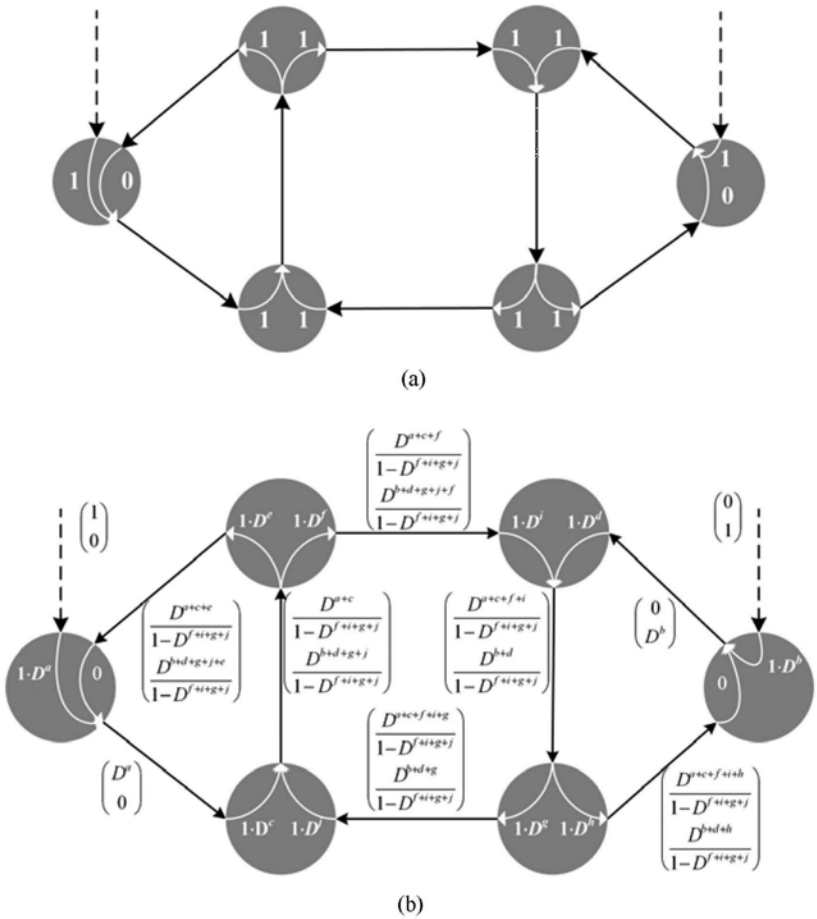


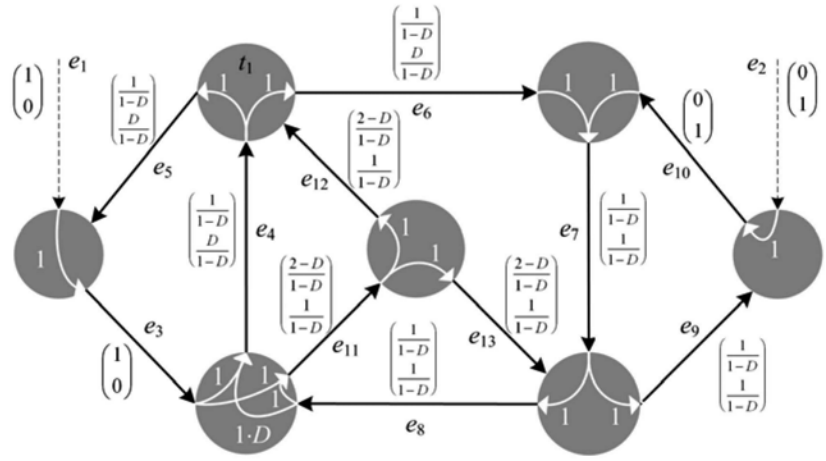
Figure 4.3. Let $\mathbb{F}$ be an arbitrary field. An $\mathbb{F}$-convolutional network code ($k_{d,e}$) on the shuttle network is shown in (a) in terms of coding coefficients. It is *not* normal, but qualifies as a delay invariant convolutional multicast. The coding coefficients and coding vectors of the $\mathbb{F}$-convolutional multicast ($k_{d,e}\cdot D^{t(d,\,e)}$) are illustrated in (b), where the delay function t associates each adjacent pair an indeterminate.

Example. Let $\mathbb{F}$ be an arbitrary field. Figure 4.4(a) depicts a t-causal $\mathbb{F}$-convolutional multicast ($k_{d,e}\cdot D^{t(d,e)}$) on the *extended shuttle network* in terms of nonzero coding coefficients and coding vectors, where t is the prescribed delay function which associates 1 to (e_8, e_4) and adjacent pairs with zero coding coefficients, and 0 to all other adjacent pairs. This convolutional multicast can be constructed via the approach in [9]. However, the $\mathbb{F}$-convolutional network code ($k_{d,e}$) does not qualify as a delay invariant convolutional multicast. As depicted in Figure 4.4(b), when the delay

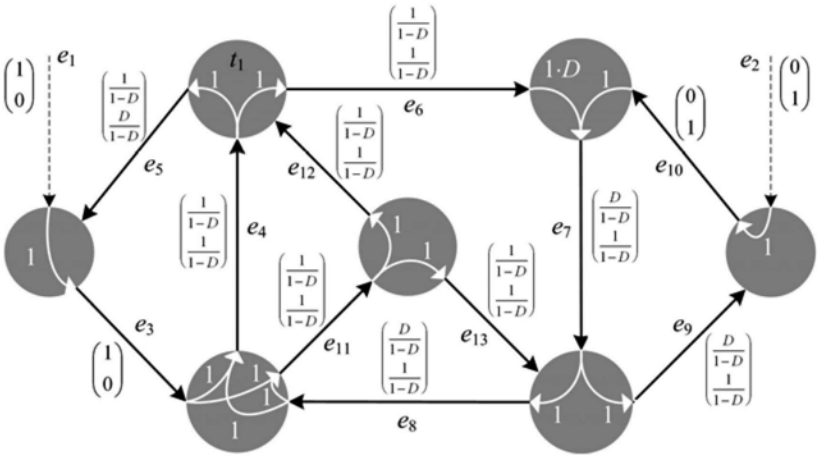
function t is changed to have $t(e_6, e_7) = 1$, $t(e_8, e_4) = 0$, and to keep others the same, the normal $\mathbb{F}$-convolutional network code $(k_{d,e} \cdot D^{t(d,e)})$ is no longer a convolutional multicast. The coding vectors of $(k_{d,e} \cdot D^{t(d,e)})$ for edges e_4, e_{12} into the sink t_1 become

$$f_{e_4} = f_{e_{12}} = \left(\frac{1}{1-D} \quad \frac{1}{1-D}\right)^{\mathrm{T}},$$

which are linearly dependent.



(a)



(b)

Figure 4.4. Let $\mathbb{F}$ an arbitrary field. (a) depicts a t-causal $\mathbb{F}$-convolutional multicast $(k_{d,e} \cdot D^{t(d,e)})$ on the *extended shuttle network* in terms of the nonzero coding coefficients and coding vectors, where t is the prescribed delay function which associates 1 to (e_8, e_4) and adjacent pairs with zero coding coefficients, and 0 to all other adjacent pairs. It can be constructed via the approach in [9]. However, the $\mathbb{F}$-convolutional network code $(k_{d,e})$ does not qualify as a delay invariant convolutional multicast. As depicted in (b), when the delay function t is changed to have $t(e_6, e_7) = 1$, $t(e_8, e_4) = 0$, and to keep others the same, the normal $\mathbb{F}$-convolutional network code $(k_{d,e} \cdot D^{t(d,e)})$ is not a convolutional multicast. The coding vectors for edges e_4, e_{12} into the sink t_1 are linearly dependent.

4.4 Matroid Duality and Theorem Generalization

Matroid duals are the motivation to connect the cyclic network $\mathcal{N}$ with the layered acyclic one $\mathcal{N}'$ such that the relationship between the $\mathbb{P}$-linear multicasts on $\mathcal{N}$ and $\mathcal{N}'$ them can be established as asserted in Theorem 4.1. Unless otherwise specified, the matroid discussed in this section is prescribed on the edge set E of $\mathcal{N}$. Besides the collection $\mathcal{I}$ of independent sets, a matroid on E can be equivalently characterized by the collection $\mathcal{B}$ of bases, which are the maximal independent sets. Such equivalence is reviewed in Appendix B.

Theorem 4.10. (Matroid Duals) Let $\mathcal{B}$ be the collection of bases of a matroid $\mathcal{M}$ on E. Then $\mathcal{B}^* = \{S–B\colon B \in \mathcal{B}\}$ is the collection of bases of another matroid $\mathcal{M}^*$ on E, called the *dual* of $\mathcal{M}$.

Proof. See Appendix B. ■

As interpreted at the beginning of subsection 4.2.3, without loss of generality, assume that a given $\mathbb{P}$-linear network code C' on $\mathcal{N}'$ has the following property:

- For every layer-1 node e_1, $e \in E\backslash E_{DG}$, the coding coefficients for (x, y), $x \in \mathrm{In}(e_1)$ and $y \in \mathrm{Out}(e_1)$, are 1.

Analogous to the connection between two linear multicast on $\mathcal{N}$ and $\mathcal{N}'$ established in Theorem 4.1, when C' is under the constraints

(36) The coding coefficient $k'_{x,y} = 1$ when $(x, y) = (e_{(2)}, e_{(3)})$ for some $e \in E\backslash E_{DG}$

(37) The coding vectors for $e_{(3)}$, $e \in E\backslash E_{DG}$, are linearly independent.

let C denote the induced $\mathbb{P}$-linear network code $(k_{d,e})$ on $\mathcal{N}$ via

$$(38)\quad k_{d,e} = \begin{cases} k'_{\overline{de},d_{(3)}}, & \text{when } d \in E_{DG} \\ -k'_{\overline{de},d_{(3)}}, & \text{when } d \in E\backslash E_{DG} \end{cases}.$$

Lemma 4.11. The $\mathbb{P}$-linear network code C is nonsingular.

Proof. Let f'_d denote the coding vector of C' for an edge $d_{(3)}$ in $\mathcal{N}'$, which corresponds to the edge d in $\mathcal{N}$. Then in the matrix form

- $[f'_d]_{d\in E_{DG}} = [\,k'_{\overline{de}}\,]_{e\in E\setminus E_{DG},\, d\in E_{DG}} = \left([k'_{\overline{de}}]_{d\in E_{DG},\, e\in E\setminus E_{DG}}\right)^{\mathbf{T}}$
- $[f'_d]_{d\in E\setminus E_{DG}} = I_{|E|-\omega} + [k'_{\overline{de}}]_{d,\, e\in E\setminus E_{DG}} = I_{|E|-\omega} + \left([k'_{\overline{de}}]_{d,\, e\in E\setminus E_{DG}}\right)^{\mathbf{T}}$

Since the $|E|\times|E|$ matrix $I_{|E|}-K_C$ can be expressed as

$$I_{|E|}-K_C = \begin{bmatrix} I_\omega & -[k_{d,e}]_{d\in \mathrm{In}(s),\, e\in E\setminus E_{DG}} \\ \mathbf{0} & I_{|E|-\omega} - [k_{d,e}]_{d,\, e\in E\setminus E_{DG}} \end{bmatrix} = \begin{bmatrix} I_\omega & -[k'_{\overline{de}}]_{d\in \mathrm{In}(s),\, e\in E\setminus E_{DG}} \\ \mathbf{0} & I_{|E|-\omega} + [k'_{\overline{de}}]_{d,\, e\in E\setminus E_{DG}} \end{bmatrix},$$

- $\det(I_{|E|}-K_C) = \det([f'_d]_{d\in E\setminus E_{DG}})$.

From (38), $\det([f'_d]_{d\in E\setminus E_{DG}}) \neq 0$, which implies that $\det(I_{|E|}-K_C) \neq 0$. Hence C is nonsingular. ■

Now that the C is nonsingular, the normalization of C (Corollary 2.7) can determine a unique set of linear network codes over $\mathbb{P}$. Let $\mathcal{M}$ and $\mathcal{M}'$, respectively, denote the matroids on E induced by the linear independence among coding vectors of normalization of C for edges in E and coding vectors of C' for edges $d_{(3)}$, $d \in E$. Then we have the following relation.

Theorem 4.12. $\mathcal{M}$ and $\mathcal{M}'$ are matroid duals.

Proof. Let f'_d denote the coding vector of C' for an edge $d_{(3)}$ in $\mathcal{N}'$, and $f_e \in \mathbb{Q}^\omega$ the coding vectors of C on $\mathcal{N}$. The matrix form of the equation (2) in Definition 2.2 for the coding vectors f_e of a $\mathbb{P}$-linear network code gives

$$[f_e]_{e\in E\setminus E_{DG}} \cdot (I_{|E|-\omega} - [k_{d,e}]_{d,e\in E\setminus E_{DG}}) = [k_{d,e}]_{d\in E_{DG},\, e\in E\setminus E_{DG}}.$$

Equivalently,

$$[f_e]_{e\in E\setminus E_{DG}} \cdot ([f'_d]_{d\in E\setminus E_{DG}})^{\mathbf{T}} = ([f'_d]_{d\in E_{DG}})^{\mathbf{T}}.$$

The matrix $[f'_d]_{d\in E\setminus E_{DG}}$ over $\mathbb{Q}$ is invertible because of (38). Adopt the abbreviation of $A = ([f'_d]_{d\in E\setminus E_{DG}})^{\mathbf{T}}$, $B = ([f'_d]_{d\in E_{DG}})^{\mathbf{T}}$. Then $[f'_d]_{d\in E} = [B^{\mathbf{T}} \mid A^{\mathbf{T}}]$ and $[f_e]_{e\in E} = [I_\omega \mid B\cdot A^{-1}]$. According to Lemma 4.5 in subsection 4.2.2, the

vector matroids $[I_\omega \mid B \cdot A^{-1}]$ and $[(B \cdot A^{-1})^{\mathbf{T}} \mid I_{|E|-\omega}]$ are duals. Thus the isomorphisim between the vector matroids $[B^{\mathbf{T}} \mid A^{\mathbf{T}}]$ and $(A^{-1})^{\mathbf{T}} \cdot [B^{\mathbf{T}} \mid A^{\mathbf{T}}] = [(B \cdot A^{-1})^{\mathbf{T}} \mid I_{|E|-\omega}]$ completes the proof. ■

Lemma 4.4 in subsection 4.2.2 asserts that corresponding to a set B_v of ω edges in $\mathcal{N}$ such that it is a prescribed subset of In(v) for a sink v and there are ω edge-disjoint paths from E_{DG} to B_v, the set B^*_v of $|E|-\omega$ edges $e_{(3)}$, $e \in E \backslash B_v$ in $\mathcal{N}'$ can be traced back to the source node s_0 by $|E|-\omega$ edge-disjoint paths as well. Recall that in the network matroid $(E \backslash E_{DG}, \mathcal{I})$ of $\mathcal{N}$, which is defined in Theorem 3.2, a set B of ω edges is a base if and only if there are ω edge-disjoint paths leading from E_{DG} to B. Thus B_v is a base in the network matroid. The assertion of Lemma 4.4 can in fact be completed into the following proposition so as to be adaptable to all bases in the network matroid.

Proposition 4.13. Let B denote a set of ω edges in $\mathcal{N}$ and B^* the corresponding set of $|E|-\omega$ edges $e_{(3)}$, $e \in E \backslash B$ in $\mathcal{N}'$. Then there are ω edge-disjoint paths leading from E_{DG} to B *if and only if* there are $|E|-\omega$ edge-disjoint paths leading from $\{e_{(1)}: e \in E \backslash B\}$ to B^*.

Proof. (Necessity) Analogous to the proof in Lemma 4.4.

(Sufficiency) Let $\wp^*$ be a family of $|E|-\omega$ edge-disjoint paths in $\mathcal{N}'$ from In(s_0) to B^*. Based on it, we can construct ω edge-disjoint paths $P_1, \ldots, P_\omega$, which are initialized to be empty, from E_{DG} to B in $\mathcal{N}$ as follows. For the i^{th} edge $d \in E_{DG}$, insert it to the path P_i. If $d \in B$, then the construction of path P_i is completed. Otherwise, insert an edge e to P_i where $\overline{de}$ is in a path in $\wp^*$. Such an edge e always exists because $d_{(3)} \in B^*$. Set $d = e$ and repeat the procedure to add another edge e such that (d, e) is an adjacent pair in $\mathcal{N}$ and

$\overline{de}$ is in a path in $\wp^*$ until $d \in B$, which implies $d_{(3)} \notin B^*$. In this way, the path P_i will always ends with an edge in B and thus the proof is completed. ■

If we plan to construct a normal $\mathbb{P}$-linear network code such that the coding vectors for a set B of ω edges (which of course must be a base in the network matroid) are linearly independent, then a layer-4 node can correspondingly be created in $\mathcal{N}'$ with edges connected from nodes e_3, $e \notin B$. Justified by Theorem 4.12 and Proposition 4.13, a $\mathbb{P}$-linear broadcast, a $\mathbb{P}$-linear dispersion and a generic $\mathbb{P}$-linear network code can all be constructed via (38) from a $\mathbb{P}$-linear multicast subject to (36) on $\mathcal{N}'$ with respect to different set of layer-4 nodes (which have been designed to be sinks.) Consequently, analogous to Theorem 4.6 and Theorem 4.7 in Section 4.3, a causal linear broadcast, linear dispersion and generic linear network code can also be constructed via two different approaches. The details will be skipped.

Chapter 5. Summary and Future Work

5.1 Summary

Linear network coding [31] structures data symbols as a finite field. Invertibility of a nonsingular matrix over a field translates into decodability of data transmission via a good network code. The algebraic structure of a field though does not support the notion of causal transmission around a cycle. Thus the theory has been formulated over just acyclic networks. Meanwhile, practical applications are not so restricted. What makes the difference is the time-multiplexed deployment of the transmission medium. When every channel in a cyclic network transmits a time series of data symbols, it may be viewed as the transmission of a data symbol on every channel in the trellis network that unfolds the time multiplexing. As time is unidirectional, this trellis network is acyclic. On the other hand, a time series of data symbols is represented by a power series in convolutional network coding [30]. In fact, the power series is normally restricted to just rational power series for finite implementability [12]. The practical application of convolutional network coding, however, is hindered by the difficulty in precise inter-node synchronization.

Chapter 2 of this thesis formulates a network code with every channel transmitting a data unit belonging to a PID. On a cyclic network, the discriminant of a linear network code may possibly be 0. Such *singularity* indicates degeneracy in the code design, which is to be avoided. A nonsingular network code can be *normalized*. A node receiving full-rate data via a *normal* network code can decode the message from the source. Under

the PID-based ensemble of data units, the conventional four types of field-based optimal linear network codes — linear multicast, broadcast, dispersion, and generic linear network codes — are extended to be defined over a cyclic network. To solve the problem of deadlock in cyclic transmission, the PID of data units needs be equipped with additional structure so as to carry a "unidirectional attribute" for breaking the cyclic deadlock. A DVR means a PID with a unique maximal ideal. All ideals in a DVR form a strictly descending chain, which makes a good candidate for the needed "unidirectional attribute." Thus the notion of a *causal* network code is formulated when the PID of data units is restricted to a DVR. A field is a PID with the unique maximal ideal 0 and hence can be regarded as a degenerated DVR, so the conventional field-based network coding becomes a degenerated case of DVR-based network coding. Meanwhile, convolutional network coding becomes the special case when the DVR consists of rational power series over a finite field. General DVR-based network coding is not restricted to time-multiplexing or even combined time/space/frequency/phase/code/wavelength-multiplexing of data symbols. Generality enhances the potential of applicability. For example, if the *uniformizer* in the DVR represents a shift in some domain other than time, then the network code is insensitive to the aforementioned hindrance by imprecise inter-node synchronization. The linear network coding in commutative algebra formulated in this chapter is the framework for the development of Chapter 3 and Chapter 4.

Chapter 3 studies the normal linear network codes from the perspective of matroids, which generalize the independence structures of linear independence among vectors. Over a network with possible cycles, a *network*

matroid is defined on the edge set via the structure of edge-disjoint paths. Meanwhile, the linear independence among coding vectors of a normal code naturally induces a matroid on the edge set. It is shown that every independent set in the matroid so induced is also independent in the network matroid. Moreover, the two matroids coincide with each other if and only if the normal code is a generic one. This shows the optimality of generic codes in terms of linear independence, and in turn reveals the representability of the network matroid. On the other hand, when the network is acyclic, every representation for the network matroid is proved to induce a generic linear network code on the same network. This offers a new characterization of generic codes in terms of the representation for network matroids.

Chapter 4 delves into the efficiency issue of code construction based on the duality theory of matroids. Given a cyclic network, a quadratically large acyclic network is established by de-cycling such that every linear multicast on the de-cycled network subject to some straightforward restriction directly induces a linear multicast on the cyclic one. In this way, existing construction algorithms for linear multicast on acyclic networks can be adapted for optimal codes on cyclic networks as well. Moreover, two approaches are proposed to induce a causal linear multicast on a cyclic network from a linear multicast on the corresponding acyclic network. Furthermore, via this unified adaptation method, when the receivers in the associated acyclic network are properly modified, a linear broadcast, a linear dispersion, and a generic linear network code on the cyclic network can also be induced from a linear multicast on the acyclic one.

5.2 Future Work

One of the results in this thesis is that an $\mathbb{F}$-linear multicast $(k_{d,e})$ is able to directly induce a t-causal $\mathbb{F}$-convolutional multicast $(k_{d,e}\cdot D^{t(d,e)})$ with respect to an arbitrary delay function t. However, an $\mathbb{F}$-linear multicast on a network only exists when $\mathbb{F}$ is sufficiently large. It is of great interest to study whether over every $\mathbb{F}$ there always exists a (not necessarily normal) $\mathbb{F}$-convolutional network code $(k_{d,e})$ such that with every delay function t, the t-causal $\mathbb{F}$-convolutional network code $(k_{d,e}\cdot D^{t(d,e)})$ qualifies as a convolutional multicast. This problem can further be extended to be over a DVR.

General DVR-based formulation of network coding strengthens the axiomatic underpinning of the theory and enhances the potential of applicability. This provides a new direction for future research.

The duality theory of matroids enables the code construction algorithms over acyclic networks adaptable for cyclic ones. An interesting research topic in the future is to attempt to transform any linear network coding problem over a cyclic network to an equivalent problem over an acyclic one via ample powerful tools in matroid theory.

Appendix A.
Preliminaries on Commutative Algebra

Commutative algebra studies commutative rings, their ideals, and modules over such rings. The present thesis, especially, deals with the algebraic structure of principal ideal domain and discrete valuation ring, which are regarded as a generalization of the field-based ensemble of data units in conventional network coding theory. This appendix is devoted to a brief introduction of various algebraic structures that have appeared in this thesis and their relations respective properties. A good detailed introduction of related topics can be found in [8]. To be self-consistent, the algebraic structure of group will be reviewed first.

A.1. Basic algebraic structures

Group. A set $\mathbb{G}$ together with a binary operation $+ : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ such that the following axioms are satisfied:

- (Closure) If $a, b \in \mathbb{G}$, then $a + b \in \mathbb{G}$.
- (Associativity) $(a + b) + c = a + (b + c)$
- (Identity element) There exists an element e in $\mathbb{G}$ such that $a + e = e + a = a$ for all $a \in \mathbb{G}$.
- (Inverse element) For each $a \in \mathbb{G}$, there exists an element $(-a) \in \mathbb{G}$ such that $a + (-a) = (-a) + a = e$.

Abelian Group. A group $\mathbb{G}$ that is *commutative*, i.e., $a + b = b + a$, $\forall a, b \in \mathbb{G}$.

Ring. A set $\mathbb{R}$ together with two binary operations, which comprise the *addition* $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, and the *multiplication* $\cdot : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, such that the following axioms are satisfied:

- $\mathbb{R}$ forms an abelian group under addition with identity $0 \in \mathbb{R}$.

- (Closure) If $a, b \in \mathbb{R}$, then $a \cdot b \in \mathbb{R}$.
- (Associativity) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- (Multiplicative identity) There exists an element 1 in $\mathbb{G}$ such that $a \cdot 1 = 1 \cdot a = a$ for all $a \in \mathbb{G}$.
- (Distribution Law) $a \cdot (b + c) = a \cdot b + b \cdot c$.

Commutative Ring. A ring that is commutative under multiplication.

Domain. A ring that has no *zero divisors*, i.e., for any two elements a and b, if $a \cdot b = 0$ then either $a = 0$ or $b = 0$.

Integral Domain. A commutative domain.

Ideal. A subset $\mathbb{J}$ of a commutative ring $\mathbb{R}$ under the same operations as $\mathbb{R}$ such that $i + j \in \mathbb{J}$ and $i \cdot r \in \mathbb{J}$ for all $i, j \in \mathbb{J}$ and $r \in \mathbb{R}$. Any subset A of $\mathbb{R}$ can generate an ideal

$$\langle A \rangle = A \cdot \mathbb{R} = \{a_1 \cdot r_1 + a_2 \cdot r_2 + \ldots + a_n \cdot r_n : a_i \in A, r_i \in \mathbb{R}, n > 0\}.$$

Below are 4 important types of ideals, three of which appeared in the thesis.

- A *maximal ideal* is a proper ideal (i.e., not $\mathbb{R}$ itself) that is not contained by any other proper ideal.
- A *prime ideal* is a proper ideal such that for any elements $a, b \in \mathbb{R}$, if $a \cdot b \in \mathbb{R}$ then either a or b is in this ideal. Every maximal ideal is naturally prime.
- An *irreducible ideal* is an ideal that can not be written as an intersection of two other ideals properly containing it.
- A *principal ideal* is an ideal that can be generated by a single element. Every maximal ideal is naturally principal.

***Principal Ideal Domain* (PID).** An integral domain in which every ideal is principal.

***Discrete Valuation Ring* (DVR).** A principal ideal domain that is also a *local ring*, i.e., has a unique maximal ideal.

Field. A commutative ring $\mathbb{F}$ that has a multiplication inverse a^{-1} for every nonzero element a in $\mathbb{F}$, i.e. $a \cdot a^{-1} = a^{-1} \cdot a = 1$. In other words, $\mathbb{F}\backslash\{0\}$ forms an abelian group under the multiplication. One way to generate a field from an integral domain $\mathbb{R}$ is as follows:

- A *quotient field* $\mathbb{Q}$ of $\mathbb{R}$ is the set $\{a / b: a, b \in \mathbb{R}\}$ with the addition operation $a / b + c / d = (a \cdot d + b \cdot c) / (b \cdot d)$ and with the multiplication operation $(a / b) \cdot (c / d) = (a \cdot c) / (b \cdot d)$.

Fact A.1. The relation between different classes of commutative rings can be characterized in the following chain.

$$\text{Commutative rings} \supset \text{integral domains} \supset \text{PID} \supset \text{DVR}$$

Moreover, a field can be regarded as a degenerated DVR with the unique maximal ideal $\{0\}$.

In the remaining part of the appendix, the notation $\mathbb{R}$ will always refer to a commutative ring. Below are several examples of different commutative rings:

- The ring $\mathbb{Z}$ of integers is a PID.
- The *quotient ring* $\mathbb{Z}/6\mathbb{Z} = \{a \bmod 6: a \in \mathbb{Z}\}$ of $\mathbb{Z}$ is a commutative ring but *not* an integral domain, because $2 \cdot 3 = 3 \cdot 4 = 0$.
- The polynomial ring $\mathbb{Z}[x]$ over the integer ring $\mathbb{Z}$ in indeterminate x is a integral domain but *not* PID. For instance, the ideal generated by $\{2, x\}$ is not principal.
- The polynomial ring $\mathbb{F}[x]$ over a field $\mathbb{F}$ in indeterminate x is a PID, but *not* a DVR. For instance, both $\langle x \rangle$ and $\langle x+2 \rangle$ are maximal ideals.

- The ring $\mathbb{F}[[x]]$ of formal power series is a DVR. Its unique maximal ideal is $\langle x \rangle$.
- The set $\{a/b: a \in \mathbb{Z}, b \in \mathbb{Z} \setminus (3 \cdot \mathbb{Z})\}$ is a DVR under the same addition and multiplication operations as in $\mathbb{Z}$. Its unique maximal ideal is $\langle 3 \rangle$.
- The quotient field of $\mathbb{Z}$ is the field of rational numbers.

Fact A.2. Every finite integral domain $\mathbb{I}$ is a field.

Proof. Consider an arbitrary nonzero element a in $\mathbb{I}$. It suffices to show that a has a multiplicative inverse. Let i be the finite cardinality of $\mathbb{I}$. Since $\mathbb{I}$ is an integral domain, $a, a^2, \ldots, a^i$ are all nonzero elements in $\mathbb{I}$. If $a^j = a^k$ for some $k < j \leq i$, then $a^k \cdot (a^{j-k}-1) = 0$, a contradiction to the definition of an integral domain that does not contain zero divisors. Therefore, $\{a, a^2, \ldots, a^i\}$ is in one-to-one correspondence with the set of nonzero elements in $\mathbb{I}$, and thus there exists $a^j = 1$ for some $j \leq i$. Since $a \cdot a^{j-1} = 1$, a^{j-1} is the multiplicative inverse of a. ■

Fact A.3. In a PID, every nonzero prime ideal is maximal.

Remark. This does not hold for an integral domain that is not a PID. For instance, in $\mathbb{Z}[x]$, the ideal $\langle x \rangle$ is prime, but it is not maximal, because it is contained in the ideal $\langle x, 2 \rangle$.

A.2. Localization

Localization is a systematic technique to add the multiplicative inverse of a subset of elements in a ring. Specific to our interest, we intend to *localizing* a PID $\mathbb{P}$ *at* any prime, or equivalently, maximal ideal $\mathbb{M} = z \cdot \mathbb{P}$, i.e.,

$$\mathbb{D} = \{p/q: p \in \mathbb{P} \text{ and } q \in \mathbb{P} \setminus \mathbb{M}\}.$$

The algebraic structure $\mathbb{D}$ thus defined under the same addition and multiplication operations as $\mathbb{P}$ forms a DVR due to the following:

- The multiplicative identity $1 \in \mathbb{P}$ is not contained in $\mathbb{M}$, and it is also the multiplicative identity in $\mathbb{D}$.
- If q_1, q_2 are two elements outside the maximal ideal $\mathbb{M}$, then $q_1 \cdot q_2$ is not in $\mathbb{M}$ as well. Thus it is easy to check that $\mathbb{D} \supseteq \mathbb{P}$ indeed forms a PID.
- Because every element e outside $z \cdot \mathbb{D}$ is a unit (with a multiplicative inverse) in $\mathbb{D}$, which generate the whole $\mathbb{D}$, the ideal $z \cdot \mathbb{D}$ is a maximal one, and no other *proper* ideals not contained by it exist. Thus, $z \cdot \mathbb{D}$ is the unique maximal ideal in $\mathbb{D}$.

The localization $\mathbb{D}$ of a PID $\mathbb{P}$ at a prime ideal $z \cdot \mathbb{P}$ can be regarded as a subdomain of the quotient field $\mathbb{Q}$ of $\mathbb{P}$. In fact, $\mathbb{Q}$ can also be regarded as localization of $\mathbb{P}$ at the prime ideal 0. For instance,

- The ring $\mathbb{F}[(D)]$ of rational power series is acquired by localizing the polynomial ring $\mathbb{F}[D]$ at the prime ideal $D \cdot \mathbb{F}[D]$, and it is a subring of the rational field $\mathbb{F}(D)$.
- The ring $\mathbb{F}[[D]]$ of formal power series is a DVR with the unique maximal ideal $D \cdot \mathbb{F}[[D]]$. It can also regarded at localizing $\mathbb{F}[[D]]$ itself at the prime ideal $D \cdot \mathbb{F}[[D]]$.

$\mathbb{F}[(D)]$ can be regarded as the intersection between $\mathbb{F}(D)$ and $\mathbb{F}[[D]]$. The relationships among $\mathbb{F}[D]$, $\mathbb{F}(D)$, $\mathbb{F}[(D)]$ and $\mathbb{F}[[D]]$ is illustrated in the diagram below.

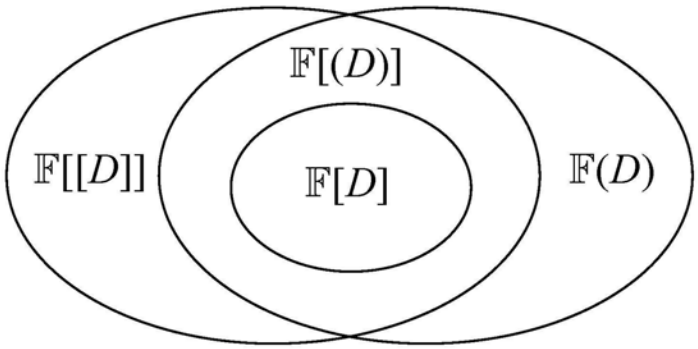


Figure A.1. Relationships among the polynomial ring $\mathbb{F}[D]$, the ring $\mathbb{F}[[D]]$ of formal power series, the quotient field $\mathbb{F}(D)$ and the ring $\mathbb{F}[(D)]$ of rational power series over a same a finite field $\mathbb{F}$ in indeterminate D.

A.3. Modules

A *module* $\mathbb{M}$ over a commutative ring $\mathbb{R}$ (or, an $\mathbb{R}$-module) consists of an abelian group $(\mathbb{M}, +)$ and a binary operation $\mathbb{R} \times \mathbb{M} \to \mathbb{M}$, called *scalar multiplication*, such that for all $r, s \in \mathbb{R}$ and $a, b \in \mathbb{M}$,

- (Closure) $ra \in \mathbb{M}$;
- $(r + s)a = ra + sa$;
- $r(a + b) = ra + rb$;
- $(r \cdot s)a = r(sa)$;
- $1a = a$.

The $\mathbb{R}$-module is a generalization of vector space over a field. Given a commutative ring $\mathbb{R}$, $\mathbb{R}$ itself and any ideal of $\mathbb{R}$ are two of the simplest examples of $\mathbb{R}$-module. As another example, the quotient ring $\mathbb{Z}/2\mathbb{Z}$ of the integer ring $\mathbb{Z}$, which maps every integer i to either 0 or 1 depending on the congruence of i modulo 2, is a $\mathbb{Z}$-module. Moreover, let $\mathbb{M}$ and $\mathbb{M}'$ be modules over a commutative ring $\mathbb{R}$, and define $\mathrm{Hom}_{\mathbb{R}}(\mathbb{M}, \mathbb{M}')$ be the set of all $\mathbb{R}$-module homomorphisms from $\mathbb{M}$ into $\mathbb{M}'$. Then $\mathrm{Hom}_{\mathbb{R}}(\mathbb{M}, \mathbb{M}')$ itself forms an $\mathbb{R}$-module. $\mathrm{Hom}_{\mathbb{R}}(\mathbb{M}, \mathbb{M})$ further forms a ring with the identity

element defined as the identity mapping from $\mathbb{M}$ onto $\mathbb{M}$, and multiplication defined as function composition, *i.e.*, for two homomorphisms ψ and φ, $\psi \times \varphi$ is the mapping $\psi\varphi$.

Below are a few more definitions with respect to an $\mathbb{R}$-module $\mathbb{M}$:

- A set A of elements in $\mathbb{M}$ is said to be $\mathbb{R}$-*linearly independent* if the unique solution to the equation $\sum_{a \in A} r_a a = 0$ is $r_a = 0 \in \mathbb{R}$, $a \in A$.
- The *rank* of $\mathbb{M}$ is the maximum cardinality of an $\mathbb{R}$-linearly independent set.
- $\mathbb{M}$ is said to be *free* if there exists a subset B of $\mathbb{M}$, called a basis of $\mathbb{M}$, such that every element $m \in \mathbb{M}$ can be *uniquely* written as an $\mathbb{R}$-linear combination of B. B is said to *generate* the free module $\mathbb{M}$. Moreover, the cardinality of B coincides with the rank of $\mathbb{M}$.

  Equivalently, a *free* $\mathbb{R}$-module of *rank n* is isomorphic to a direct sum $\mathbb{R}^n$ of n copies of the scalar ring $\mathbb{R}$. It behaves very much like vector spaces over a field.
- A *submodule* $\mathbb{S}$ of $\mathbb{M}$ means an additive subgroup of $\mathbb{M}$ such that for all $r \in \mathbb{R}$ and $s \in \mathbb{S}$, $rs \in \mathbb{S}$.

Fact A.4. A module is not necessarily free. Moreover, a submodule of a free module is not necessarily free. For instances,

- The $\mathbb{Z}$-module $\mathbb{Z}/2\mathbb{Z}$ is not free because $1 \in \mathbb{Z}/2\mathbb{Z}$ can be written as $(2a+1)\cdot 1 \bmod 2$ for all $a \in \mathbb{Z}$.
- Consider $\mathbb{Z}[x]$ itself as a $\mathbb{Z}[x]$-module. It is apparently free of rank 1 since it has the basis $\{1\}$. On the other hand, the ideal $\langle x, 2 \rangle$ of $\mathbb{Z}[x]$ can be regarded as a submodule of $\mathbb{Z}[x]$, in which there does not exist a basis.

Although a submodule of a free module is not necessarily free, it is the true when the free module is restricted to be over a PID. This is ensured by the *invariant factor theorem of free submodule over a PID*, which further asserts that it is possible to choose generators for the two modules related in a simple way.

Invariant factor theorem of free submodule over PID

Let $\mathbb{M}$ be a free module over a PID $\mathbb{P}$ of finite rank n and let $\mathbb{S}$ be a submodule of $\mathbb{M}$. Then,

- $\mathbb{S}$ is free of rank m, $m \leq n$ and
- There exists a basis $\{y_1, y_2, \ldots, y_n\}$ of $\mathbb{M}$ so that $\{a_1y_1, a_2y_2, \ldots, a_my_m\}$ is a basis of $\mathbb{S}$, where $a_1, a_2, \ldots, a_m$ are nonzero elements of $\mathbb{P}$ with the divisibility relations

$$a_1 \mid a_2 \mid \ldots \mid a_m$$

The elements $a_1, a_2, \ldots, a_m \in \mathbb{P}$ are called the *invariant factors*[1] of $\mathbb{S}$, and are known to be *unique* up to multiplication by units in $\mathbb{P}$. (See Theorem 4, Chapter 12 in [8].)

Given a submodule $\mathbb{S}$ of a free $\mathbb{P}$-module $\mathbb{M}$ with invariant factors $a_1, a_2, \ldots, a_m$, not every basis $\{y_1, y_2, \ldots, y_n\}$ of $\mathbb{M}$ has the property that $\{a_1y_1, a_2y_2, \ldots, a_my_m\}$ is a basis of $\mathbb{S}$. The theorem guarantees that there exists one such basis of $\mathbb{M}$. For instance, in the free $\mathbb{Z}$-module $\mathbb{Z}^2$, where $\mathbb{Z}$ means the PID of integers, the submodule generated by $\{(3\ 0)^{\mathbf{T}}, (0\ 2)^{\mathbf{T}}\}$ has the unique set $\{1, 6\}$ of invariant factors. If we choose $\{(3\ 2)^{\mathbf{T}}, (2\ 1)^{\mathbf{T}}\}$ as the basis of $\mathbb{Z}^2$,

[1] The term of invariant factors for a free submodule in this thesis borrows from the one in commutative algebra used for invariant factor decomposition of a finitely generated module over a PID (See Theorem 5, Chapter 12 in [8].)

then the $\mathbb{Z}$-submodule generated by $\{(3\ 2)^{\mathbf{T}}, (12\ 6)^{\mathbf{T}}\}$ is same as the one generated by $\{(3\ 0)^{\mathbf{T}}, (0\ 2)^{\mathbf{T}}\}$. On the other hand, it is easy to see that derived from another basis $\{(1\ 0)^{\mathbf{T}}, (0\ 1)^{\mathbf{T}}\}$ of $\mathbb{Z}^2$, the $\mathbb{Z}$-submodule generated by $\{(1\ 0)^{\mathbf{T}}, (0\ 6)^{\mathbf{T}}\}$ is not same as the one generated by $\{(3\ 0)^{\mathbf{T}}, (0\ 2)^{\mathbf{T}}\}$. Moreover, besides $\{1, 6\}$, for any other set $\{a_1, a_2\}$ of integers with $a_1 \mid a_2$, the $\mathbb{Z}$-module generated by $\{(3\ 0)^{\mathbf{T}}, (0\ 2)^{\mathbf{T}}\}$ can not be generated by $\{(a_1\ 0)^{\mathbf{T}}, (0\ a_2)^{\mathbf{T}}\}$ as well.

A.4. Nakayama Lemma [40]

As discussed in Section 2.3, the infinite strictly descending chain

$$\langle z\rangle \supset \langle z^2\rangle \supset \ldots \supset \langle z^l\rangle \ldots \supset \bigcap_{k=1}^{\infty}\langle z\rangle^k = 0$$

of all ideals in a DVR $\mathbb{D}$ serves to break the deadlock in cyclic data transmission of a DVR-based network code. The equality in this chain is a direct consequence of the *Nakayama Lemma* below by applying it to $\mathbb{M} = \bigcap_{k=1}^{\infty}\langle z\rangle^k$ [1] and $\mathbb{J} = \langle z\rangle$.

Nakayama Lemma Let $\mathbb{M}$ be a finitely generated $\mathbb{R}$-module and $\mathbb{J} \subseteq \mathbb{R}$ an ideal. If $\mathbb{J}\mathbb{M} = \mathbb{M}$, then

- there exists $r \in \mathbb{R}$ such that $r\mathbb{M} = 0$ and $r \equiv 1 \pmod{\mathbb{J}}$.
- In addition, if $\mathbb{J} \subseteq \cap_{\text{maximal ideal } \mathbb{I}}\, \mathbb{I}$, then $\mathbb{M} = 0$.

Proof. In order to prove the first part of the lemma, it suffices to show that

(∗) For every $\mathbb{R}$-module endomorphism[2] φ from $\mathbb{M}$ to $\mathbb{J}\mathbb{M}$, where $\mathbb{J}$ is an ideal of $\mathbb{R}$, there exists a polynomial $\varphi^n + c_{n-1}\varphi^{n-1} + \ldots + c_1\varphi^{n-1} + c_0 = 0$, where n is the rank of $\mathbb{M}$ and $c_0, \ldots, c_{n-1}$ are elements in $\mathbb{J}$.

---

[1] Every ideal in $\mathbb{D}$ is principal, so when regarded as a $\mathbb{D}$-module, $\bigcap_{k=1}^{\infty}\langle z\rangle^k$ is finitely generated.

[2] A homomorphism from a structure to itself.

This is because when the above proposition is applied to set φ as the identity mapping,

$$\varphi^n + c_{n-1}\varphi^{n-1} + \ldots + c_1\varphi^{n-1} + c_0 = 1 + c_{n-1} + \ldots + c_1 + c_0 = 0$$

If r is set to be equal to $1 + c_{n-1} + \ldots + c_1 + c_0$, then $r\mathbb{M} = 0$ and $r - 1 \in \mathbb{J}$. On the other hand, the proposition (∗) be regarded as a direct consequence of the *Caylay-Hamilton Theorem* for an $n\times n$ square matrix A over a commutative ring $\mathbb{R}$, which asserts that $p(A) = 0$, where $p(\lambda) = \det(\lambda I_n - A)$ is the *characteristic polynomial* of A, and will be proved in the last part of this appendix. To see this, let $(m_1, m_2, \ldots, m_n)$ be the row vector of generators of $\mathbb{M}$. Then every $m \in \mathbb{M}$ can be represented as a column vector $(k_1, k_2, \ldots, k_n)^{\mathbf{T}}$ such that $m = (m_1, m_2, \ldots, m_n)\cdot(k_1, k_2, \ldots, k_n)^{\mathbf{T}}$. Next, represent the endomorphism φ as an $n\times n$ matrix $A = [a_{ij}]$ over $\mathbb{J}$, *i.e.*,

$$\varphi(m_i) = \sum_j a_{ij} m_j.$$

According to the Caylay-Hamilton Theorem, $p(A) = 0$. Moreover, since all entries in A belong to $\mathbb{J}$, so are the coefficients in $p(A)$. Therefore, the characteristic polynomial of A is the desired polynomial for (∗).

Now assume that $\mathbb{J} \subseteq \cap_{\text{maximal ideal } \mathbb{I}} \mathbb{I}$. Based on the first part of Nakayama Lemma, there exits $r \in \mathbb{R}$ such that $r\mathbb{M} = 0$ and $r \equiv 1 \pmod{\mathbb{J}}$. It suffices show that r is a unit since $\mathbb{M} = r^{-1}r\mathbb{M} = 0$. If r is not a unit, then r is contained in some maximal ideal and thus in $\mathbb{J}$. However, since $r = a + 1$ for some $a \in \mathbb{J}$, $1 = r - a \in \mathbb{J}$, a contradiction to $\mathbb{J} \subset \mathbb{R}$. ■

Caylay-Hamilton Theorem Let A be an $n\times n$ square matrix over a commutative ring $\mathbb{R}$ and $p(\lambda)$ the characteristic polynomial of A, i.e., $p(\lambda) = \det(\lambda I_n - A)$. Then, $p(A) = 0$.

Proof. By Cramer's Rule,

$$\det(\lambda I_n - A)I_n = (\lambda I_n - A)\cdot Adj(\lambda I_n - A).$$

Write $Adj(\lambda I_n - A) = \lambda^{n-1}B_{n-1} + \ldots + \lambda B_1 + B_0$, where $B_{n-1}, \ldots, B_0$ are $n\times n$ square matrices over $\mathbb{R}$. Then

$$\begin{aligned}
&\lambda^n B_{n-1} + \lambda^{n-1}(B_{n-2} - A\cdot B_{n-1}) + \ldots + \lambda(B_0 - A\cdot B_1) - A\cdot B_0 \\
&= (\lambda I_n - A)\cdot Adj(\lambda I_n - A) \\
&= \det(\lambda I_n - A)I_n \\
&= \lambda^n I_n + \lambda^{n-1}p_{n-1}I_n + \ldots + \lambda p_1 I_n + p_0 I_n
\end{aligned}$$

Equivalently,

$$\begin{aligned}
B_{n-1} &= I_n && \ldots (0) \\
B_{n-2} - A\cdot B_{n-1} &= p_{n-1}I_n && \ldots (1) \\
&\ldots \\
B_0 - A\cdot B_1 &= p_1 I_n && \ldots (n-1) \\
-A\cdot B_0 &= p_0 I_n && \ldots (n)
\end{aligned}$$

Via summing above equations as

$$(n) + A(n-1) + \ldots + A^{n-1}(1) + A^n(0),$$

we get $A^n + p_{n-1}A^{n-1} + \ldots + p_1 A + p_0 I_n = 0$. ■

Appendix B. Preliminaries on Matroids

Besides the initial definition of matroids in terms of independent sets given at the beginning of Section 3.1, the present appendix will review another equivalent definition, because it is closely related to the duality theorem of matroids. Moreover, several examples of matroid duality, which motivates the reduction of a cyclic network to an acyclic one for efficient code construction in Chapter 4, will also be given. A good detailed introduction of the related topic on matroids can be found in [34]. To be self-consistent, we will next repeat the first definition of matroids, which is in terms of independent set.

Definition. Let S be a finite set. A matroid $\mathcal{M}$ on the *ground set* S is an ordered pair $(S, \mathcal{I})$, where $\mathcal{I}$ is a family of subsets of S satisfying the following three axioms:

(I1) $\phi \in \mathcal{I}$.

(I2) If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$

(I3) (Augmentation axiom) If I_1 and I_2 are in $\mathcal{I}$ and $|I_2| > |I_1|$, then there is an element $e \in I_2 - I_1$ su ch that $I_1 \cup \{e\} \in \mathcal{I}$.

Below are more notations and simple properties with respect to a matroid $(S, \mathcal{I})$.

- A subset of S is called an *independent set* if it is in $\mathcal{I}$ and otherwise a *dependent set*.
- A maximal independent set is called a *base*. All bases of a matroid share the same cardinality.

  To show this, assume that there are two maximal independent sets B_1 and B_2 with $|B_2| > |B_1|$. Then according to augmentation axiom of a matroid,

there exists an element $b \in B_2 - B_1$ such that $B_1 \cup \{b\}$ is also an independent set, a contradiction to the maximality of B_1.

- The *rank* of the matroid means the cardinality of the base.
- A minimal dependent set is called a *cycle*. Given a base B, there is a unique cycle $C \in B \cup \{e\}$ for every $e \in S - B$.

Theorem B.1. Let S be a finite set, and $\mathcal{B}$ a family of subsets of S satisfying the following two axioms:

(B1) $\mathcal{B}$ is nonempty.

(B2) (Substitution axiom) If B_1 and B_2 are in $\mathcal{B}$, then for each $e \in B_1 \setminus B_2$, there exists an element $f \in B_2 - B_1$ such that $(B_1 - \{e\}) \cup \{f\} \in \mathcal{B}$.

Let $\mathcal{I}$ be the collection of subsets of S that are contained in some member of $\mathcal{B}$. Then $(S, \mathcal{I})$ is a matroid having $\mathcal{B}$ as its collection of bases.

Proof. Given a matroid $(S, \mathcal{I})$, the family of its bases naturally satisfy (B1) and (B2). On the other hand, consider a family $\mathcal{B}$ of subsets of a finite set S satisfying (B1) and (B2), and $\mathcal{I} = \{I: I \subseteq B, \forall B \in \mathcal{B}\}$. Since (B1) holds for $\mathcal{B}$, (I1) holds for $\mathcal{I}$. Moreover, by the construction of $\mathcal{I}$, (I2) holds for $\mathcal{I}$. Then, it suffices to show that (I3) holds for the collection $\mathcal{I}$.

Assume that there are two members I_1 and I_2 in $\mathcal{I}$ with $|I_2| > |I_1|$ such that, for all $e \in I_2 - I_1$, $I_1 \cup \{e\} \notin \mathcal{I}$. Let B_1, B_2 be two members in $\mathcal{B}$ that contain I_1, I_2, respectively, such that $|B_2 - (I_2 \cup B_1)|$ is minimal. Since for all $e \in I_2 - I_1$, $I_1 \cup \{e\} \notin \mathcal{I}$, $I_1 \cup \{e\}$ is not contained by any member in $\mathcal{B}$. Thus,

(39) $I_2 - B_1 = I_2 - I_1$

Now suppose that $B_2 - (I_2 \cup B_1)$ is non-empty. Let x be an element in this set. By (B2), there is an element y of $B_1 - B_2$ such that $(B_2 - \{x\}) \cup \{y\} \in \mathcal{B}$. But then $|((B_2 - \{x\}) \cup \{y\}) - (I_2 \cup B_1)| < |B_2 - (I_2 \cup B_1)|$, a contradiction to the minimality of $|B_2 - (I_2 \cup B_1)|$. Hence $B_2 - (I_2 \cup B_1)$ is empty and so $B_2 - B_1 = I_2 - B_1$. Thus, via (39),

(40) $B_2–B_1 = I_2–I_1$

Next we shall show that $B_1–(I_1 \cup B_2)$ is empty. If not, then there is an element x in this set and an element y in $B_2–B_1$ such that $(B_1–\{x\})\cup\{y\} \in \mathcal{B}$. In this way, $I_1\cup\{y\} \subseteq (B_1–\{x\})\cup\{y\}$ so $I_1\cup\{y\} \in \mathcal{I}$. Since $y \in B_2–B_1$, $y \in I_2–I_1$ by (40). This leads to a contradiction to the assumption that there is no element y in $I_2–I_1$ such that $I_1\cup\{y\} \in \mathcal{I}$. Therefore, we conclude that $B_1–(I_1\cup B_2)$ is empty. Hence $B_1–B_2 = I_1–B_2$. Since $I_1–B_2$ is contained in $I_1–I_2$,

(41) $B_1–B_2 \subseteq I_1–I_2$.

As will be shown next, $|B_1| = |B_2|$. Then $|B_1–B_2| = |B_2–B_1|$. By (40) and (41), $|I_1–I_2| \geq |I_2–I_1|$, so $|I_1| \geq |I_2|$, a contradiction to $|I_2| > |I_1|$. This implies that (I3) holds for the collection $\mathcal{I}$.

What remains to show is that any two sets in $\mathcal{B}$ share the same cardinality. Let B_1, B_2 be two elements in $\mathcal{B}$ with $|B_1| > |B_2|$ such that $|B_1–B_2|$ is minimal. Clearly $B_1–B_2 \neq \phi$. Choose an arbitrary element $x \in B_1–B_2$. According to (B2), there exists an element $y \in B_2–B_1$ such that $(B_1–\{x\})\cup\{y\} \in \mathcal{B}$. Because

$$|(B_1–\{x\})\cup\{y\}| = |B_1| > |B_2|, \text{ but}$$

$$|((B_1–\{x\})\cup\{y\})–B_2| < |B_1–B_2|,$$

A contradiction to the assumption that $|B_1–B_2|$ is minimal. ■

Based on the matroid definition in terms of bases, the theorem of matroid duals becomes easier to be expressed

Theorem B.2. (Matroid Duals) Let $\mathcal{M} = (S, \mathcal{I})$ be a matroid with $\mathcal{B}$ being the collection of its bases. Then $\mathcal{B}^* = \{S–B: B \in \mathcal{B}\}$ is the collection of bases of another matroid $\mathcal{M}^*$ on S, called the *dual* of $\mathcal{M}$.

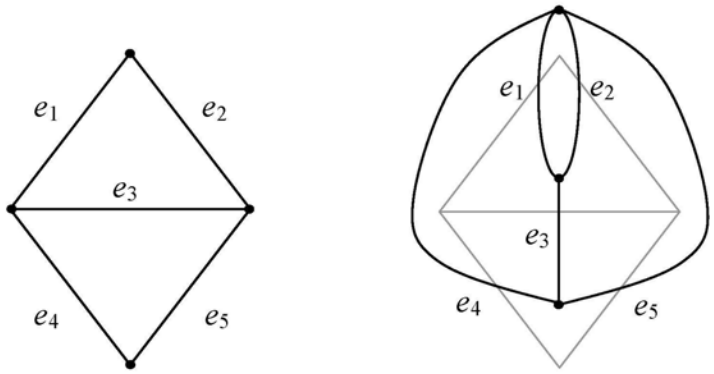
Proof. Since Theorem 2 provides an equivalent definition of matroid, it suffices to show that (B1) and (B2) hold for $\mathcal{B}^*$. Since $\mathcal{B}$ is non-empty, $\mathcal{B}^*$ is also non-empty. Hence $\mathcal{B}^*$ satisfies (B1). Let B_1^* and B_2^* be two members of

$\mathcal{B}^*$ and $x \in B_1{}^* - B_2{}^*$. Then in order to show that (B2) holds for $\mathcal{B}^*$, it suffices to show that there exists an element $y \in B_2{}^* - B_1{}^*$ such that $(B_1{}^* - \{x\}) \cup \{y\} \in \mathcal{B}^*$. Write $B_1 = S - B_1{}^*$ and $B_2 = S - B_2{}^*$. Note that $B_1 - B_2 = B_2{}^* - B_1{}^*$, $B_2 - B_1 = B_1{}^* - B_2{}^*$, and $x \in B_2 - B_1$. Thus, it is equivalent to show that there exists an element $y \in B_1 - B_2 = B_2{}^* - B_1{}^*$ such that $(B_1 - \{y\}) \cup \{x\} \in \mathcal{B}$. Since $x \notin B_1$, there exists a unique cycle $C \subseteq B_1 \cup \{x\}$ with $x \in C$. Since a cycle cannot be contained in a base, there exists an element $y \in C - B_2 \subseteq B_1$. Moreover, since C is the unique cycle in $B_1 \cup \{x\}$, $(B_1 - \{y\}) \cup \{x\}$ is an independent set and thus a member in $\mathcal{B}$. This completes the proof. ■

Given a matroid $\mathcal{M}$, the term "dual" of the matroid $\mathcal{M}^*$ is used because of the fact that $(\mathcal{M}^*)^* = \mathcal{M}$.

One example of matroid duals is the pair of vector matroids $[I_\omega \mid M]$ and $[M^{\mathrm{T}} \mid I_\eta]$, where M is an $\omega \times \eta$ matrix over $\mathbb{P}$. This is proved in Lemma 4.5.

As another example, in an undirected graph G, a *graphic matroid* $\mathcal{M}(G)$ on the edge set E has the collection of independent sets consisting of the set of edges that does not contain a cycle. The dual $\mathcal{M}^*(G)$ of $\mathcal{M}(G)$ is called a *cographic* matroid. As illustrated in the figure below, if the graph G is planar, i.e., it can be drawn on the plane without edge intersections except for endpoints, then the cographic matroid $\mathcal{M}^*(G)$ is isomorphic to the graphic matroid $\mathcal{M}(G^*)$ on the edge set in G's dual graph G^*, in which there is a vertex for each plane region of G, and an edge for each edge in G joining two neighboring regions when G has been drawn without edge intersections except the endpoints.

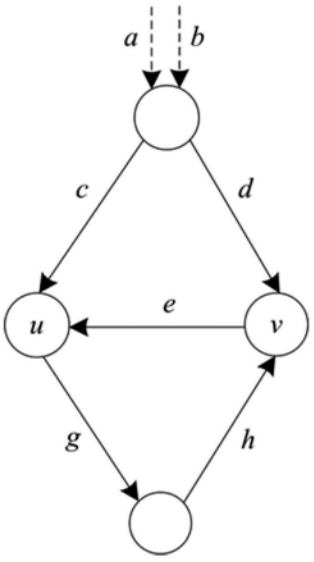


(a) (b)

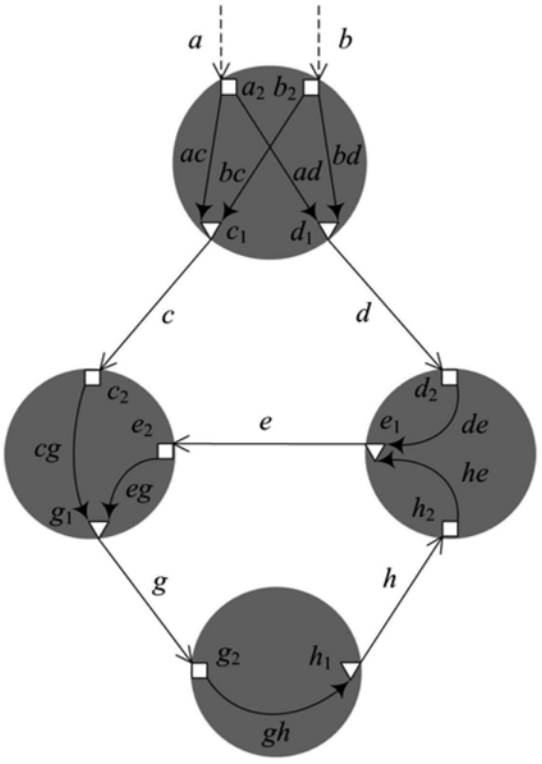
Figure B.1. (a) depicts a planar undirected graph G. The graphic matroid $\mathcal{M}$ of G on the edge set has the collection of bases $\{\{e_1, e_2, e_4\}, \{e_1, e_2, e_5\}, \{e_1, e_3, e_4\}, \{e_1, e_3, e_5\}, \{e_1, e_4, e_5\}, \{e_2, e_3, e_4\}, \{e_2, e_3, e_5\}, \{e_2, e_4, e_5\}\}$. The dual $\mathcal{M}^*$ has the collection of bases $\{\{e_3, e_5\}, \{e_3, e_4\}, \{e_2, e_5\}, \{e_2, e_3\}, \{e_1, e_5\}, \{e_1, e_4\}, \{e_1, e_3\}\}$. (b) depicts the graphic dual G^* of G. $\mathcal{M}^*$ is isomorphic to the graphic matroid of G^*.

Appendix C.
An Example on Network De-cycling

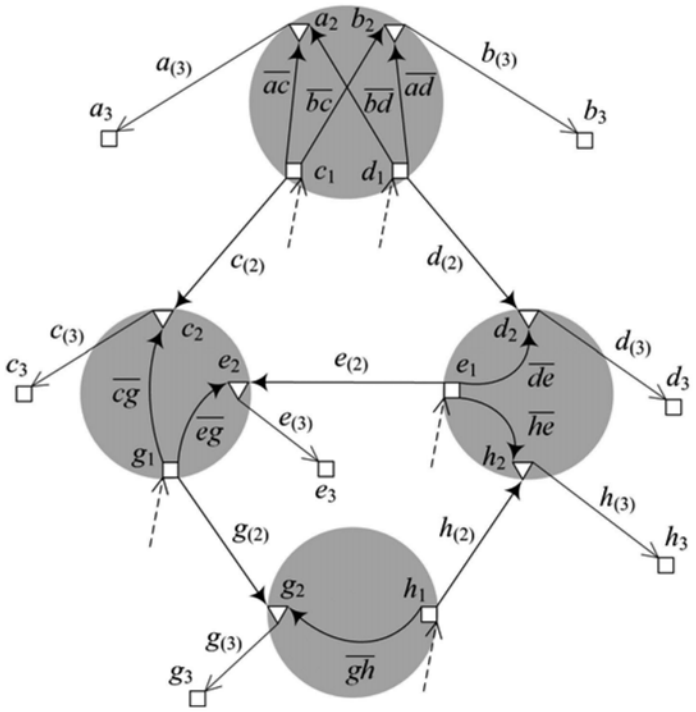
In section 4.2.2, there has been an example of the network de-cycling process to construct the acyclic counterpart of a cyclic network. This appendix provides a slightly larger example to further elucidate this process. Figure (a) depicts the given network $\mathcal{N}$ which contains a cycle. By "node dilation" process, the bipartite version $\mathcal{N}_B$ of $\mathcal{N}$ is constructed in figure (b). The two types of nodes are classified as squares and triangles. The acyclic bipartite counterpart $\mathcal{N}'_B$ of $\mathcal{N}_B$ appears in (c). Based on the acyclic $\mathcal{N}'_B$ without the distinction of squares and triangles, $\mathcal{N}'$ is constructed just by creating the bottom layer sinks corresponding to either a sink or the source node in $\mathcal{N}$.



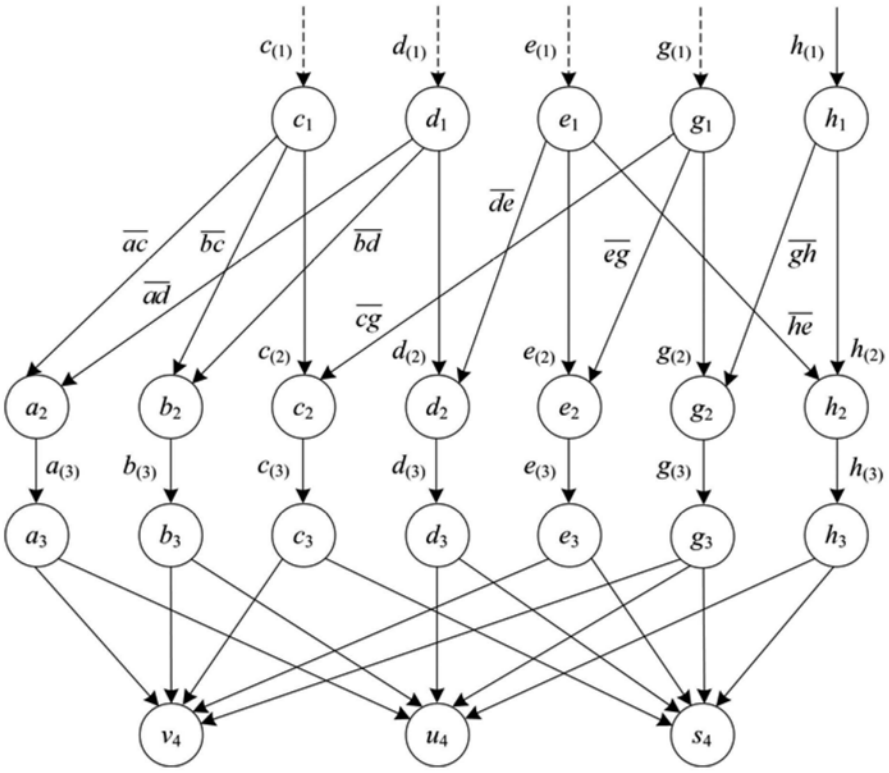
(a)



(b)


a_2 b_2
$a_{(3)}$
$b_{(3)}$
a_3
b_3
$\overline{ac}$ $\overline{bc}$ $\overline{bd}$ $\overline{ad}$
c_1 d_1
$c_{(2)}$ $d_{(2)}$
$c_{(3)}$ c_2 d_2 $d_{(3)}$
c_3 e_2 $e_{(2)}$ e_1 d_3
$\overline{cg}$ $\overline{eg}$ $\overline{de}$ $\overline{he}$
$e_{(3)}$
g_1 h_2
e_3
$h_{(3)}$
$g_{(2)}$ $h_{(2)}$
h_3
g_2 h_1
$g_{(3)}$
$\overline{gh}$
g_3

(c)


$c_{(1)}$ $d_{(1)}$ $e_{(1)}$ $g_{(1)}$ $h_{(1)}$
c_1 d_1 e_1 g_1 h_1
$\overline{de}$
$\overline{ac}$ $\overline{bc}$ $\overline{bd}$ $\overline{eg}$ $\overline{gh}$
$\overline{ad}$ $\overline{cg}$ $\overline{he}$
$c_{(2)}$ $d_{(2)}$ $e_{(2)}$ $g_{(2)}$ $h_{(2)}$
a_2 b_2 c_2 d_2 e_2 g_2 h_2
$a_{(3)}$ $b_{(3)}$ $c_{(3)}$ $d_{(3)}$ $e_{(3)}$ $g_{(3)}$ $h_{(3)}$
a_3 b_3 c_3 d_3 e_3 g_3 h_3
v_4 u_4 s_4

(d)

Bibliography

[1] R. Ahlswede, N. Cai, and R. W. Yeung, "Network information flow," Proceedings of the *IEEE International Symposium on Information Theory*, Aug., 1998.

[2] R. Alshwede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, Feb., 2000.

[3] Á. Barbero, Ø.Ytrehus, "An efficient centralized binary multicast network coding algorithm for any cyclic network," http://arxiv.org/abs/0705.0085v1, 2008.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, Mass.: The MIT Press. 2001.

[5] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745-2759, Aug., 2005.

[6] R. Dougherty, C. Freiling, and K. Zeger, "Networks, matroids, and non-Shannon information inequalities," *IEEE Transactions on Information Theory*, vol. 53, no. 6, pp. 1949-1969, Jun., 2007.

[7] R. Dougherty, C. Freiling, and K. Zeger, "Linear network codes and systems of polynomial equations," *IEEE Transactions on Information Theory,* vol. 54, no. 5, pp. 2303-2316, May, 2008.

[8] D. S. Dummit, R. M. Foote, *Abstract Algebra*, 2nd ed. New York: John Wiley & Sons, Inc. 1999.

[9] E. Erez and M. Feder, "Convolutional network coding," Proceedings of the *IEEE International Symposium on Information Theory*, pp. 146, Chicago, United States, June, 2004.

[10] E. Erez and M. Feder, "Convolutional network codes for cyclic networks," Proceedings of the *First Workshop on Network Coding, Theory, and Applications,* Riva del Garda, Italy, Apr., 2005.

[11] E. Erez and M. Feder, "Efficient network codes for cyclic networks," Proceedings of the *IEEE International Symposium on Information Theory*, pp. 1982-1986, Adelaide, Australia, Sept., 2005.

[12] G. D. Forney, Jr., "Convolutional codes I: algebraic structure," *IEEE Transactions on Information Theory*, vol. 16, No. 2, pp. 720-736, Nov., 1970.

[13] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Transactions on Information Theory*, vol. 52, No. 3, pp. 829-848, Mar., 2006.

[14] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*, Hanover, MA: Now Publishers, 2007.

[15] N. J. A. Harvey, D. R. Karger, and K. Murota, "Deterministic network coding by matrix completion," *Annual ACM-SIAM Symposium on Discrete Algorithm*, 2005.

[16] N. J. A. Harvey, "Deterministic network coding by matrix completion," *Master's Thesis*, Massachusetts Institute of Technology, 2005.

[17] T. Ho, D. R. Karger, M. Médard, and R. Koetter, "Network Coding from a network flow perspective," Proceedings of the *IEEE International*

Symposium on Information Theory, pp. 441, Yokohama, Japan, Jun., 2003.

[18] A. W. Ingleton, M. J. Piff, "Gammoids and transversal matroids," *J. Combinatorial Theory*, Ser. B 15, pp. 51-68, 1973.

[19] S. Jaggi. P. A. Chou, and K. Jain, "Low complexity algebraic network multicast codes," Proceedings of the *IEEE International Symposium on Information Theory*, Yokohama, Japan, June, 2003.

[20] S. Jaggi, M. Effros, T. Ho, and M. Médard, "On linear network coding," *42nd Allerton Conference on Communication, Control and Computing*, Monticello, IL, Sep., 2004.

[21] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973-1982, Jun., 2005.

[22] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, No. 5, pp. 782-795, Oct., 2003.

[23] M. Langberg, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *Joint special issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transaction on Networking*, vol. 52, pp. 2386-2397, 2006.

[24] M. Langberg, A. Sprintson, and J. Bruck, "Network coding: a computational perspective," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 147-157, Jan., 2009.

[25] E. Lawler, *Combinatorial optimization: networks and matroids*, Mineola, N.Y.: Dover Publications, 2001.

[26] S.-Y. R. Li and S. T. Ho, "Ring-theoretic foundation of convolutional network coding," *Workshop on Network Coding, Theory, and Applications*, Hong Kong, Jan., 2008.

[27] S.-Y. R. Li and Q. T. Sun, "Network coding theory via commutative algebra," *Workshop on Network Coding, Theory, and Applications*, Lausanne, Switzerland, June, 2009.

[28] S.-Y. R. Li and Q. T. Sun, "Network coding theory via commutative algebra," submitted to *IEEE Transactions on Information Theory*, Mar., 2009.

[29] S.-Y. R. Li and R. W. Yeung, "Network multicast flow via linear coding," Proceedings of *International Symposium on Operations Research and its Applications* (ISORA'98), pp. 197-211, Kunming, China, Aug., 1998.

[30] S.-Y. R. Li and R. W. Yeung, "On convolutional network coding," Proceedings of the *IEEE International Symposium on Information Theory*, pp.1743-1747, Seattle, USA, July, 2006.

[31] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371-381, Feb., 2003.

[32] J. Mason, "On a class of matroids arising from paths in graphs," *Proceedings of the London Mathematical Society (3)*, 25, pp. 55-74, 1972.

[33] M. Médard, M. Effros, T. Ho, and D. Karger, "On coding for nonmulticast networks," *41st Annual Allerton Conference on Communication Control, and Computing*, Monticello, IL, Oct., 2003.

[34] J. G. Oxley, *Matroid Theory*. New York: Oxford Univ. Press, 1992.

[35] M. J. Piff, D. J. A. Welsh, "On the vector representation of matroids," *Journal of the London Mathematical Society (2)*, 2, pp. 284-288, 1970.

[36] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," Proceedings of *15th ACM Symposium on Parallel Algorithms and Architectures*, 2003.

[37] Q. Sun, S. T. Ho, and S.-Y. R. Li, "On network matroids and linear network codes," Proceedings of the *IEEE International Symposium on Information Theory*, pp. 1833-1837, Toronto, Canada, July, 2008.

[38] M. Tan, R. W. Yeung, and S. T. Ho, "A unified framework for linear network codes," Proceedings of the *Workshop on Network Coding, Theory, and Applications*, Hong Kong, Jan., 2008.

[39] H. Whitney, "On the abstract properties of linear dependence," *American Journal of Mathematics*, pp. 509-533, 1935.

[40] Wikipedia, http://en.wikipedia.org/wiki/Nakayama_lemma.

[41] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*. Boston, MA: Now Publishers, 2006.