Linear Network Codes and Systems of Polynomial Equations

Randall Dougherty, Chris Freiling, and Kenneth Zeger

(*Submitted to ISIT 2008*)

***Abstract*— If β and γ are nonnegative integers and F is a field, then a polynomial collection $\{p_1, \ldots, p_\beta\} \subseteq \mathbf{Z}[\alpha_1, \ldots, \alpha_\gamma]$ is said to be *solvable over* F if there exist $\omega_1, \ldots, \omega_\gamma \in F$ such that for all $i = 1, \ldots, \beta$ we have $p_i(\omega_1, \ldots, \omega_\gamma) = 0$. We say that a network and a polynomial collection are *solvably equivalent* if for each field F the network has a scalar-linear solution over F if and only if the polynomial collection is solvable over F. Koetter and Médard's work implies that for any directed acyclic network, there exists a solvably equivalent polynomial collection. We provide the converse result, namely that for any polynomial collection there exists a solvably equivalent directed acyclic network. (Hence, the problems of network scalar-linear solvability and polynomial collection solvability have the same complexity.) The construction of the network is modeled on a matroid construction using finite projective planes, due to MacLane in 1936.**

A set Ψ of prime numbers is a *set of characteristics* of a network if for every $q \in \Psi$, the network has a scalar-linear solution over some finite field with characteristic q and does not have a scalar-linear solution over any finite field whose characteristic lies outside of Ψ. We show that a collection of primes is a set of characteristics of some network if and only if the collection is finite or co-finite.

Two networks $\mathcal{N}$ and $\mathcal{N}'$ are *ls-equivalent* if for any finite field F, $\mathcal{N}$ is scalar-linearly solvable over F if and only if $\mathcal{N}'$ is scalar-linearly solvable over F. We further show that every network is ls-equivalent to a multiple-unicast matroidal network.

I. Introduction

We first demonstrate a certain equivalence between networks and collections of polynomials. Specifically, we show that associated with every finite collection of polynomials with integer coefficients is a corresponding network which is scalar-linearly solvable precisely over those finite fields where the polynomials have a common root. A consequence is that the complexity of determining whether networks are scalar-linearly solvable over particular finite fields is equivalent to the complexity of determining whether collections of polynomial have common roots over the corresponding fields. Secondly, we show that the collections of prime numbers corresponding to the field characteristics of scalar-linearly solvable network alphabets are precisely those which are finite or co-finite. Finally, we show that for every network, there exists a multiple-unicast network which is matroidal (i.e. obtained from a certain matroid-to-network construction), such that the two networks are scalar-linearly solvable over the same finite fields.

There has been interest in determining the solvability, scalar linear solvability, and vector linear solvability of an arbitrary network with respect to a chosen alphabet (e.g. [4]–[7], [9]–[11], [13], [15], [17], [18]).

For a given finite alphabet, to determine if a network is solvable or scalar-linearly solvable, one can perform a finite exhaustive search of all possible codes for the network. If a vector dimension is also fixed, a finite search can also establish if a network is vector-linearly solvable over that dimension. There is presently no known algorithm for determining the general solvability or vector-linear solvability of an arbitrary network. The existence of an algorithm (which is apparently not computationally efficient) to determine scalar-linear solvability of an arbitrary network follows from work in [12]. Their technique was to construct a finite collection of polynomials from an arbitrary network, such that for each finite field, the polynomials have a common root over the field if and only if the network has a scalar-linear solution over the field.

Throughout, polynomials will have integer coefficients and will use the variables $\alpha_1, \alpha_2, \ldots$. For nonnegative integers β and γ, any finite set $\mathcal{P} = \{p_1, \ldots, p_\beta\} \subseteq \mathbf{Z}[\alpha_1, \ldots, \alpha_\gamma]$ will be called a *polynomial collection*. If F is a field, then a polynomial collection is said to be *solvable over* F if there exist $\omega_1, \ldots, \omega_\gamma \in F$ such that for all $i = 1, \ldots, \beta$ we have $p_i(\omega_1, \ldots, \omega_\gamma) = 0$. We say that a network and a polynomial collection are *solvably*

This work was supported by the Institute for Defense Analyses, the National Science Foundation, and the UCSD Center for Wireless Communications.

R. Dougherty is with the Center for Communications Research, 4320 Westerra Court, San Diego, CA 92121-1969 (rdough@ccrwest.org).

C. Freiling is with the Department of Mathematics, California State University, San Bernardino, 5500 University Parkway, San Bernardino, CA 92407-2397 (cfreilin@csusb.edu).

K. Zeger is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 (zeger@ucsd.edu).

equivalent if for each field F the network has a scalar-linear solution over F if and only if the polynomial collection is solvable over F.

We present an algorithm in Section II for constructing a network from any polynomial system. Our main results are that: the network is scalar-linearly solvable over the same fields as those for which the polynomials have common roots (Theorem I.2), the constructed network is always matroidal (Theorem I.3), every network is scalar-linearly solvable on the same set of fields as a multiple-unicast matroidal network (Corollary I.8), and the collections of prime numbers corresponding to the field characteristics of scalar-linearly solvable network alphabets are characterized as either finite or co-finite (in Theorem I.9).

Let Ψ be an arbitrary collection of integers of the form q^i, where q is prime and $i \geq 1$. We say that Ψ is the *solvability set* for a network (respectively, polynomial collection) if for every finite field F, the network is scalar-linearly solvable (respectively, polynomial collection is solvable) if and only if $|F| \in \Psi$. The set of primes q, such that q^i lies in the solvability set for some $i \geq 1$, is called the *set of characteristics*[1] for a network (respectively, polynomial collection).

The following theorem leads to an algorithm (via Gröbner bases [2]) for determining whether a network has a scalar-linear solution. No such algorithm is presently known for determining whether a network has a general nonlinear solution.

Theorem I.1. *(follows from Koetter-Médard [12])*
Every directed acyclic network has a solvably equivalent polynomial collection.

In this paper, we provide the following converse result.

Theorem I.2. *(Converse to Theorem I.1)*
Any polynomial collection has a solvably equivalent directed acyclic network.

Furthermore, the solvably equivalent network in Theorem I.2 is given constructively and is matroidal, as stated in the next theorem.

Theorem I.3. *If a polynomial collection $\mathcal{P}$ is solvable over some finite field, then any network constructed, as in Section II, from $\mathcal{P}$ is matroidal.*

The next definition is taken from [8] ("CSLS" stands for "coding solvability, linear solvability").

Definition I.4. Two networks $\mathcal{N}$ and $\mathcal{N}'$ are *CSLS-equivalent* if the following two conditions hold:

1) For any finite alphabet $\mathcal{A}$, $\mathcal{N}$ is solvable over $\mathcal{A}$ if and only if $\mathcal{N}'$ is solvable over $\mathcal{A}$.
2) For any finite field F and any positive integer k, $\mathcal{N}$ is vector-linearly solvable over F in dimension k if and only if $\mathcal{N}'$ is vector-linearly solvable over F in dimension k.

The following definition gives a type of equivalence that is weaker than CSLS (the acronym "ls" stands for "(scalar) linear solvability").

Definition I.5. Two networks $\mathcal{N}$ and $\mathcal{N}'$ are *ls-equivalent* if for any finite field F, $\mathcal{N}$ is scalar-linearly solvable over F if and only if $\mathcal{N}'$ is scalar-linearly solvable over F.

Theorem I.6. *(see [8, Theorem II.1])*
Any network is CSLS-equivalent to a multiple-unicast network.

The next theorem shows that if Theorem I.6 is applied to a matroidal network, then the resulting multiple-unicast network can also be taken to be matroidal.

Theorem I.7. *(see [7, Corollary VII.8])*
Any matroidal network is CSLS-equivalent to a multiple-unicast matroidal network.

The next corollary follows from our main result in Theorem I.2 together with several previous results. It demonstrates that, when considering which finite fields arbitrary networks are scalar-linearly solvable over, it suffices to consider to the subclass of networks which are simultaneously multiple-unicast and matroidal.

Corollary I.8.
Any network is ls-equivalent to a multiple-unicast matroidal network.

Theorem I.9. *A set of prime numbers is the set of characteristics of some network if and only if the set is finite or co-finite.*

Theorem I.1 and our Theorem I.2 together indicate that determining the scalar-linear solvability of a directed acyclic network over a field F is computationally equivalent to determining whether a collection of polynomials has a common root over F. Given any algorithm for determining scalar-linear network solvability, our result gives an algorithm for determining polynomial solvability. This is a "many-to-one reduction" (i.e., it converts a single instance of the polynomial solvability problem to a single instance of the network scalar-linear solvability problem with the same answer). The reduction causes at most a linear blowup in input size, in the following sense: the number of nodes and edges in the resulting network is at most a linear function of the number

[1] This terminology is taken from [1].

of steps (variable retrievals and arithmetic operations) needed to compute the values of the polynomials in the collection. In terms of bit representations, it is at most an $O(n \ln n)$ blowup. This many-to-one reduction has the additional property that, given a scalar-linear solution to the network, we can directly reconstruct a solution to the polynomial collection.

It can be shown (via Gröbner bases) that the sets of characteristics of polynomial collections are precisely the sets of primes which are finite or co-finite. In contrast, there has been no known characterization of the sets of characteristics or the solvability sets of networks. If Ψ is the solvability set of a network and $n \in \Psi$, then $n^i \in \Psi$ for all positive integers i While there are an uncountable number of sets of powers of primes closed under exponentiation, there are only a countably infinite number of solvability sets since there are only a countable number of networks and polynomial collections.

A fundamental problem is to determine which sets of integers can be solvability sets and which can be sets of characteristics for networks. Theorem I.1 shows that every network solvability set is also a polynomial collection solvability set. Our Theorem I.2 shows that every polynomial collection solvability set is also a network solvability set. Thus, the network solvability sets are the same as the polynomial collection solvability sets. Our Theorem I.9 shows that a set of primes is the set of characteristics of a network if and only if the set of primes is finite or co-finite.

II. Network Construction from Polynomial System

In this section we present an algorithm for constructing a directed acyclic network $\mathcal{N}$ from a finite polynomial collection $\mathcal{P} = \{p_1, \ldots, p_\beta\} \subseteq \mathbf{Z}[\alpha_1, \ldots, \alpha_\gamma]$, for $i = 1, \ldots, \beta$. The network will be built piece by piece from eight building block components, $\mathcal{C}_0, \ldots, \mathcal{C}_7$, which are shown in Figures 1 and 2 (using Table I). The messages will be a, b, and c. Certain nodes of the network will be labeled by x_q, y_q, u_q, or z_q, where for each such node, q is some polynomial in $\mathbf{Z}[\alpha_1, \ldots, \alpha_\gamma]$. For example, the sources for a, b, c will be nodes x_0, x_1, y_1, respectively. During the construction, we will label various nodes with polynomials and will later demonstrate a connection between these polynomials and the alphabet symbols carried by these nodes. It will be demonstrated that this construction algorithm produces a network such that for any field F, the network has a scalar-linear solution over F if and only if the polynomial collection $\mathcal{P}$ has a solution over F.

The network construction process consists of the steps:

Step (1): Start with component $\mathcal{C}_0$ which creates nodes x_0, x_1, y_1, z_0, and z_∞. (See Figure 1)

Step (2): If $\gamma > 0$, then add components $\mathcal{C}_1(1), \ldots, \mathcal{C}_1(\gamma)$, creating nodes $x_{\alpha_1}, \ldots x_{\alpha_\gamma}$. Each of these components is adjoined to the network at the nodes x_0, x_1, z_∞, which have already been created at Step (1). (See Figure 2 and Table I)

Step (3): Repeatedly add components $\mathcal{C}_2, \ldots, \mathcal{C}_7$ to create nodes $x_{p_1(\alpha_1,\ldots,\alpha_\gamma)}, \ldots, x_{p_\beta(\alpha_1,\ldots,\alpha_\gamma)}$. Steps (3a)-(3d) describe the creation of $x_{p_1(\alpha_1,\ldots,\alpha_\gamma)}, \ldots, x_{p_\beta(\alpha_1,\ldots,\alpha_\gamma)}$ as well as many intermediate nodes. (See Figure 2 and Table I)

> **Step (3a)**: *For any positive integer n, to create a node labeled x_n*: First, add component $\mathcal{C}_4(1)$ to create node u_1. Then, for $i = 1, \ldots, n-1$, add component $\mathcal{C}_2(i)$ to create node z_i and add component $\mathcal{C}_6(i, 1)$ to create node x_{i+1}. This is possible since x_1, z_0, z_∞ have already been created.
>
> **Step (3b)**: *For any positive integer n, to create a node labeled x_{-n}*: First, add component $\mathcal{C}_2(1)$ to create node z_1, and add component $\mathcal{C}_5(1)$ to create node u_{-1}. Then, for $i = 0, \ldots, n-1$, add component $\mathcal{C}_6(-i, -1)$ to create node x_{-i-1} and add component $\mathcal{C}_2(-i-1)$ to create node z_{-i-1}.
>
> **Step (3c)**: *For any positive integer n and any $\alpha \in \{\alpha_1, \ldots, \alpha_\gamma\}$ to create a node labeled x_{α^n}*: First, add component $\mathcal{C}_3(\alpha)$ to create node y_α. Then, for $j = 1, \ldots, n-1$, add component $\mathcal{C}_2(\alpha^j)$ to create node z_{α^j} and add component $\mathcal{C}_7(\alpha^j, \alpha)$ to create node $x_{\alpha^{j+1}}$.
>
> **Step (3d)**: *To create nodes labeled by an arbitrary polynomial in $\mathbf{Z}[\alpha_1, \ldots, \alpha_\gamma]$*: Add various instances of components $\mathcal{C}_6$ and $\mathcal{C}_7$ to create nodes labeled by sums and products of labels of existing nodes created above. (Some instances of components $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ may also have to be added in order to use $\mathcal{C}_6$ and $\mathcal{C}_7$.)

Step (4): Force each of the nodes $x_{p_1(\alpha_1,\ldots,\alpha_\gamma)}, \ldots, x_{p_\beta(\alpha_1,\ldots,\alpha_\gamma)}$ to demand message a.

To construct nodes labeled by arbitrary polynomials in $\mathbf{Z}[\alpha_1, \ldots, \alpha_\gamma]$ in Step (3) of the algorithm, one can use Step (3a) to create all positive integer coefficients of the polynomials, use Step (3b) to create all negative integer coefficients of the polynomials, use Step (3c) to create all variable powers occurring in the polynomials, and finally use Step (3d) to combine the existing network nodes to create the desired polynomials.

This algorithm converts a single instance of the polynomial solvability problem to a single instance of the network scalar-linear solvability problem with the same answer. The procedure above is not the most efficient method to create the network $\mathcal{N}$ from the polynomial collection $\mathcal{P}$. A smaller network can in general be constructed whose size is linear in the size of the representation of the polynomial collection.

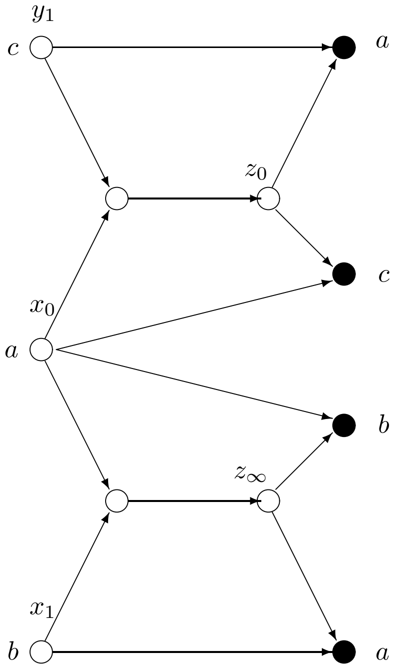

Fig. 1. Network component $\mathcal{C}_0$. The leftmost three nodes are sources, generating messages c, a, and b from top to bottom, respectively. The rightmost four nodes are receivers and demand messages a, c, b, and a, respectively. Five of the nodes are labeled by x_0, x_1, y_1, z_0, or z_∞.

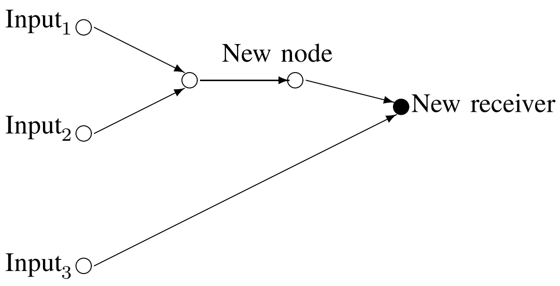

Fig. 2. A generic network component $\mathcal{C}_i$ for $1 \leq i \leq 7$. Input$_1$, Input$_2$, and Input$_3$ are existing nodes in the network and the remaining nodes and edges in component $\mathcal{C}_i$ are new. The rightmost node, "New receiver", demands one message. Table I lists seven different instantiations of this generic network component that are used in a network construction.

Comp.	Input$_1$	Input$_2$	Input$_3$	New node	New demand
$\mathcal{C}_1(i)$	x_0	x_1	z_∞	x_{α_i}	a
$\mathcal{C}_2(q)$	z_0	z_∞	x_q	z_q	c
$\mathcal{C}_3(q)$	x_q	z_1	z_0	y_q	a
$\mathcal{C}_4(q)$	x_q	z_0	z_∞	u_q	c
$\mathcal{C}_5(q)$	x_0	z_q	z_∞	u_{-q}	c
$\mathcal{C}_6(q,r)$	z_q	u_r	z_∞	x_{q+r}	a
$\mathcal{C}_7(q,r)$	z_q	y_r	z_∞	x_{qr}	a

TABLE I
INSTANTIATIONS OF THE GENERIC NETWORK COMPONENT SHOWN IN FIGURE 2. EACH LINE IN THE TABLE GIVES THE FIVE VALUES THAT ARE USED TO FORM A SPECIFIED COMPONENT.

III. MATROIDALITY OF CONSTRUCTED NETWORKS

First we review the concepts of matroids, matroidal networks, and the finite projective plane, each of which will be used in what follows.

A *matroid* $\mathcal{M}$ (e.g. see [16]) is an ordered pair $(\mathcal{S}, \mathcal{I})$, where $\mathcal{S}$ is a finite set and $\mathcal{I}$ is a set of subsets of $\mathcal{S}$ satisfying the following three conditions:

(I1) $\varnothing \in \mathcal{I}$.
(I2) If $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$.
(I3) If $I, J \in \mathcal{I}$ and $|J| < |I|$, then $\exists e \in I - J$ such that $J \cup \{e\} \in \mathcal{I}$.

The set $\mathcal{S}$ is called the *ground set*, the members of $\mathcal{I}$ are called *independent sets*, and any subset of $\mathcal{S}$ not in $\mathcal{I}$ is called a *dependent set*. For any matroid $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ and any $X \subseteq \mathcal{S}$, let $\mathcal{I}|X = \{I \subseteq X : I \in \mathcal{I}\}$, and let $\mathcal{M}|X = (X, \mathcal{I}|X)$. Then $\mathcal{M}|X$ is a matroid and the *rank* of X, denoted $\rho(X)$, is the (unique) size of a maximal independent set of $\mathcal{M}|X$. The rank of the matroid $\mathcal{M}$ is defined to be $\rho(\mathcal{S})$.

Let $\mathcal{N}$ be a network with message set μ, node set ν, and edge set ϵ. Let $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ be a matroid with rank function ρ. The network $\mathcal{N}$ is a *matroidal network* (see [7]) associated with $\mathcal{M}$ if there exists a function $f : \mu \cup \epsilon \to \mathcal{S}$ such that the following conditions hold:

(M1) f is one-to-one on μ.
(M2) $f(\mu) \in \mathcal{I}$.
(M3) $\rho(f(\mathrm{In}(x))) = \rho(f(\mathrm{In}(x) \cup \mathrm{Out}(x)))$, $\forall x \in \nu$.

It was shown in [7] that many interesting networks are matroidal, including all networks that are scalar-linearly solvable over a finite field (e.g. solvable multicast networks). The matroid used is a vector space over the finite field (with dimension the number of messages); the function f maps the messages to elementary vectors (vectors which are all 0 except for a single 1) and maps the edges to the corresponding "global coding vectors" (see, e.g., [11]) for the given scalar-linear code.

In [7], a method was presented for constructing, from given matroids, (matroidal) networks which reflect some of the matroids' properties. This construction was used to obtain networks used to prove various results in the literature [5], [6], [8]. For example, in [7], a network was constructed from the Vámos matroid that demonstrates the insufficiency of using Shannon-type information inequalities to compute network coding capacity. In what follows, we will prove that if a network is constructed from a solvable polynomial collection as in Section II, then the network is matroidal.

The network construction algorithm given in Section II was inspired by the 1936 work of Saunders MacLane in [14].

For any positive integer n, a *projective plane* (e.g. see [3]) comprises a set of points, a set of lines, and an incidence relation between points and lines satisfying:

(P1) Any two points are incident to exactly one line.
(P2) Any two lines are incident to exactly one point.
(P3) There exist 4 points, no 3 of which are incident to the same line.

Every finite projective plane induces a rank-three matroid as follows. Let $\mathcal{S}$ be the set of all points in the projective plane, let $\mathcal{I}$ be the collection of subsets of $\mathcal{S}$ of cardinality at most 3 that do not contain 3 collinear points, and let $\mathcal{M} = (\mathcal{S}, \mathcal{I})$. It is easy to see that $\mathcal{M}$ satisfies (I1) and (I2). Suppose $I, J \in \mathcal{I}$ where $|I| > |J|$. Then $|J| \in \{0, 1, 2\}$. If $|J| < 2$, then for any $v \in I - J$, we trivially have $J \cup \{v\} \in \mathcal{I}$. If $|J| = 2$ and if for each $v \in I - J$ we have $J \cup \{v\} \notin \mathcal{I}$, then the 3 points in I are collinear, contradicting $I \in \mathcal{I}$. Thus, $\mathcal{M}$ also satisfies (I3), and therefore $\mathcal{M}$ is a rank-3 matroid.

For any field F, one can construct a projective plane Π^F (of order $|F|$ if F is finite) as follows. Let $\Pi^F = (F \times F) \cup F \cup \{\infty\}$ where two points (a, b) and (c, d) in $F \times F$ are said to have *slope* $s \in F$ if $a \neq c$ and $s = (d - b)(c - a)^{-1}$, and slope $s = \infty$ if $a = c$. A *line* in Π^F consists of an element s of $F \cup \{\infty\}$ (called a *point at infinity*) together with a maximal set of points in $F \times F$ such that every two of them have slope $-1/s$ (where we make the convention that $v/0 = \infty$ and $v/\infty = 0$, for all nonzero $v \in F$). The set of all points at infinity is also considered a line and its point at infinity is ∞. It can be verified that axioms (P1)–(P3) hold for Π^F.

MacLane [14] (see also [19, pp. 18–21]) used this construction as follows. Let $\mathcal{P}$ be a polynomial collection and let K be a finite field such that $\mathcal{P}$ has a solution over K. Then MacLane constructs a matroid $\mathcal{M}$ that is representable over K and such that, for any finite field F, if $\mathcal{M}$ is representable over F, then $\mathcal{P}$ has a solution over F. However, it is not necessarily true that, if $\mathcal{P}$ has a solution over F, then $\mathcal{M}$ is representable over F. Such an if-and-only-if result is not attainable in general for matroids; for instance, it is known that, if a matroid is representable over the 2-element field and the 3-element field, then it is representable over all finite fields [16, Theorem 6.6.3]. The extra flexibility of networks allows us to construct a network solvably equivalent to any given polynomial collection.

REFERENCES

[1] R. Baines and P. Vámos, "An algorithm to compute the set of characteristics of a system of polynomial equations over the integers", *Journal of Symbolic Computation*, vol. 35, pp. 269-279, 2003.

[2] T. Becker, V. Weispfenning, and H. Kredel, *Gröbner Bases: A Computational Approach to Commutative Algebra*, Springer-Verlag, New York, 1993.

[3] L.M. Blumenthal, *A Modern View of Geometry*, Dover Publications, Inc., New York, 1961.

[4] R. Dougherty, C. Freiling, and K. Zeger, "Linearity and solvability in multicast networks", *IEEE Transactions on Information Theory* vol. 50, no. 10, pp. 2243-2256, October 2004.

[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, August 2005.

[6] R. Dougherty, C. Freiling, and K. Zeger, "Unachievability of network coding capacity", *IEEE Transactions on Information Theory & IEEE/ACM Transactions on Networking* (joint issue), vol. 52, no. 6, pp. 2365-2372, June 2006.

[7] 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, June 2007.

[8] R. Dougherty and K. Zeger, "Nonreversibility and equivalent constructions of multiple-unicast networks", *IEEE Trans. on Info. Theory*, vol. 52, no. 11, pp. 5067-5077, November 2006.

[9] M. Feder, D. Ron, and A. Tavory, "Bounds on linear codes for network multicast," *Electronic Colloquium on Computational Complexity (ECCC)*, Report 33, pp. 1-9, 2003.

[10] T. Ho, D. Karger, M. Médard and R. Koetter, "Network coding from a network flow perspective," *IEEE International Symposium on Information Theory*, Yokohama, Japan, p. 441, June 2003.

[11] 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, June 2005.

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

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

[14] S. MacLane, "Some interpretations of abstract linear dependence in terms of projective geometry", *American Journal of Mathematics*, vol. 58, no. 1, pp. 236-240, January 1936.

[15] M. Médard, M. Effros, T. Ho, D. Karger, "On coding for non-multicast networks", *41st Annual Allerton Conf. on Communication Control and Computing*, Monticello, Illinois, October 2003.

[16] J. G. Oxley, *Matroid Theory*, Oxford University Press, New York, 1992.

[17] A. Rasala Lehman and E. Lehman, "Complexity classification of network information flow problems", *41st Annual Allerton Conference on Communication Control and Computing*, Monticello, Illinois, October 2003.

[18] S. Riis, "Linear versus non-linear boolean functions in network flow", *38th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, March 2004.

[19] N. L. White (editor), *Combinatorial Geometries*, Encyclopedia of Mathematics and its Applications, Cambridge Univ. Press, 1987.

[20] R. W. Yeung, *A First Course in Information Theory*, Kluwer, 2002.