Matroidal Networks

Randall Dougherty, Chris Freiling, and Kenneth Zeger

***Abstract*— We define a class of networks, called matroidal networks, which includes as special cases all scalar-linearly solvable networks, and in particular solvable multicast networks. We then present a method for constructing matroidal networks from known matroids. We specifically construct networks that play an important role in proving results in the literature, such as the insufficiency of linear network coding and the unachievability of network coding capacity. We also construct a new network, from the Vámos matroid, which we call the Vámos network, and use it to prove that Shannon-type information inequalities are in general not sufficient for computing network coding capacities.**

I. Introduction

In this paper, a *network* is a finite, directed, acyclic multigraph with node set ν and edge set ϵ, together with a finite set μ called the *message set*, a *source mapping*

$$S : \nu \to 2^{\mu},$$

and a *receiver mapping*

$$R : \nu \to 2^{\mu}.$$

For every node x, if $S(x)$ is nonempty, then x is called a *source*, and if $R(x)$ is nonempty, then x is called a *receiver*. The elements of $S(x)$ are called the *messages generated by* x and the elements of $R(x)$ are called the *messages demanded by* x. For convenience in definitions of capacity, we will assume that for each message m, every receiver demanding m is reachable from at least one source generating m.

An *alphabet* is a finite set $\mathcal{A}$ with at least two elements. For each network node x, let $\mathrm{In}(x)$ denote the union of the set of messages generated by x with the set of in-edges of x, and let $\mathrm{Out}(x)$ denote the union of the set of messages demanded by x with the set of out-edges of x.

Let k and n be positive integers, called the *source dimension* and the *edge capacity*, respectively. For every node x, fix an ordering of $\mathrm{In}(x)$ such that all messages in the resulting list occur before the edges in the list; the resulting ordered list is called the *input list* of x. For every edge $e = (x, y)$, an *edge function* is a map

$$f_e : (\mathcal{A}^k)^{\alpha} \times (\mathcal{A}^n)^{\beta} \to \mathcal{A}^n,$$

where α and β are the number of messages and edges, respectively, in the input list of x (note that α and β are functions of x, whereas k and n are constants). For every $x \in \nu$ and $m \in R(x)$, a *decoding function* is a map

$$f_{x,m} : (\mathcal{A}^k)^{\alpha} \times (\mathcal{A}^n)^{\beta} \to \mathcal{A}^k,$$

where α and β are the number of messages and edges, respectively, in the input list of x.

Given an alphabet $\mathcal{A}$, a (k, n) *code*[1] for a network is an assignment of edge functions and decoding functions to the network's edges and receivers, respectively. A *message assignment* is a map $a : \mu \to \mathcal{A}^k$. For any (k, n) code and for any message assignment, we recursively define the function

$$c : \epsilon \to \mathcal{A}^n$$

as follows. For every edge $e = (x, y)$, let

$$c(e) = f_e(a(x_1), \ldots, a(x_{\alpha}), c(x_{\alpha+1}), \ldots, c(x_{\alpha+\beta}))$$

where $x_1, \ldots, x_{\alpha}$ are the messages generated by x and $x_{\alpha+1}, \ldots, x_{\alpha+\beta}$ are the in-edges of x. We say that each edge e *carries* the symbol vector $c(e)$.

For a given network, (k, n) code, receiver x, and message m demanded by x, if for every message assignment $a : \mu \to \mathcal{A}^k$ we have

$$f_{x,m}(a(x_1), \ldots, a(x_{\alpha}), c(x_{\alpha+1}), \ldots, c(x_{\alpha+\beta}))=a(m)$$

then we say that *x's demand m is satisfied*. In other words, the receiver x can recover an arbitrary instance of the message m generated by its source. A (k, n) code is said to be a (k, n) *solution* if every demand of every receiver is satisfied.

Special codes of interest include *linear codes*, where the edge functions and decoding functions are linear, and *routing codes*, where the edge functions and decoding functions simply copy input components to output components. Special networks of interest include *multicast* networks, where there is only one source node and every receiver demands all of the source messages, and *multiple-unicast* networks, where each network message is generated by exactly one source node and is demanded by exactly one receiver node. The network coding terminology used here generally follows that of [3].

If a network has a (k, n) solution over some alphabet, then we say the ratio k/n is an *achievable coding rate* for the network. A network is said to be *solvable* if it has a (k, n) solution for the case $k = n = 1$. (Note that any (k, k) coding solution on alphabet $\mathcal{A}$ yields a $(1, 1)$ coding solution on alphabet $\mathcal{A}^k$, so we do not need to distinguish between

This work was supported by the Institute for Defense Analyses, Air Force Office of Scientific Research, 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).

[1]Sometimes called a *fractional code* [11] or simply a *code*.

scalar and vector solvability.) A network is said to be *scalar-linearly solvable* if it has a linear (k, n) solution for the case $k = n = 1$, or *vector-linearly solvable* if it has a linear (k, n) solution for the case $k = n$ (here we do need to distinguish).

An important goal in network coding is to find an achievable coding rate which is as large as possible for a network. The *coding capacity of a network with respect to (*or *over) an alphabet* $\mathcal{A}$ *and a class* $\mathcal{C}$ *of network codes* (a related definition appears in [15, p. 339]) is

$$\sup\{k/n : \exists\ (k, n) \text{ coding solution in } \mathcal{C} \text{ over } \mathcal{A}\}.$$

If $\mathcal{C}$ consists of all network codes, then we simply refer to the above quantity as the *coding capacity* of the network with respect to $\mathcal{A}$. If the class $\mathcal{C}$ of network codes consists of all routing codes or all linear codes, then the coding capacity is referred to as the *routing capacity* or *linear coding capacity*, respectively. (In all cases, if the alphabet $\mathcal{A}$ is not mentioned, the capacity is taken to be the supremum of the capacities over all alphabets $\mathcal{A}$.) The coding capacity of a given network is said to be *achievable* if there is some (k, n) solution for the network for which k/n equals the capacity.

Ahlswede, Cai, Li, and Yeung [2] exhibited a network whose linear coding capacity is larger than its routing capacity. Li, Yeung, and Cai [12] showed in the special case of a multicast network (i.e. a network with a single source and each receiver demanding all messages), the coding capacity and the linear coding capacity are equal. It was shown in [3] that for all networks the coding capacity is independent of the alphabet size. Clearly the routing capacity is also independent of the alphabet size. However, it was shown in [4] that the linear coding capacity of a network can depend on the alphabet size and the largest linear coding capacity of a network over any finite field alphabet can be smaller than the network's coding capacity. It was also shown in [3] that the routing capacity is always rational, achievable, and computable by an algorithm.

Although the routing capacity of an arbitrary network is always computable, there is no known computationally efficient algorithm for such a task. Unfortunately, it is not even presently known whether or not there exist algorithms that can compute the coding capacity or the linear coding capacity of an arbitrary network. In fact, computing the exact coding capacity or linear coding capacity of even relatively simple networks can be a non-trivial task. At present, very few exact coding capacities have been rigorously derived in the literature. It is also known that the coding capacity might not be achievable [5].

As an alternative to determining exact coding capacities, it can be useful to determine bounds on the coding capacity and linear coding capacity of a network. One approach to obtaining capacity bounds (and possibly exact capacities) is to use information-theoretic entropy arguments. The basic idea is to assume a network's source messages are i.i.d. uniform random variables on some finite alphabet and then to use standard information theory identities and inequalities to derive bounds on the largest possible ratio of the source dimension k to the edge capacity n.

Specifically, we construct a network (from the well-known Vámos matroid) which we call the Vámos network, and demonstrate that no collection of Shannon-type information inequalities can produce an upper bound on the coding capacity which is as small as an upper bound obtainable using a certain non-Shannon-type information inequality. Additionally, for the Vámos network, we compute the exact routing capacity and the exact linear coding capacity over every finite field.

The Vámos network is one of many networks that are closely related to matroids. The field of matroid theory has had many interesting results discovered over the last several decades. We explore the connection between matroids and networks and present a method of constructing networks from matroids. In addition to the Vámos network, we demonstrate that some specific known networks can be constructed from matroids. These include the Butterfly network from [2], and parts of networks used to establish the insufficiency of linear network coding in [4] and the unachievability of network coding capacity in [5].

II. NETWORK FUNDAMENTALS

If a network has nodes n_i and n_j (on diagrams these will usually be marked just i and j), then an edge between them will be written as $e_{i,j}$.

For any node $x \in \nu$ and any $S \subseteq \mathrm{Out}(x)$, we call the ordered pair $(\mathrm{In}(x), S)$ a *dependency* of the network. This terminology reflects the fact that the out-edges and demands of each node are deterministic functions of the in-edges and messages generated at the node. Using these, one can deduce further dependencies. For more on this, see [1], [9], [10].

In order to compute capacity bounds for networks, we will compute various joint entropies, where we take the network messages to be independent uniform random variables. In that case, given a network code, which determines the vectors carried by the edges from the messages, we will write $H(x)$ to denote the joint entropy of any collection x of edges and messages.

III. NETWORKS FROM MATROIDS

In this section, we give a method for building networks from matroids.

If a matroid $\mathcal{M}$ is isomorphic to the vector matroid of a matrix over a field F, then $\mathcal{M}$ is said to be *representable over* F or F*-representable*. A matroid is *representable* if it is representable over some field.

A *geometric depiction* of any particular rank-$(m+1)$ matroid is a diagram in $\mathbf{R}^m$ consisting of nodes and undirected edges, where the nodes are in one-to-one correspondence with the matroid's ground set elements, and a collection of j of the matroid's ground set elements is dependent if and only if it corresponds to points in the diagram that are depicted as lying on a common $(j-2)$-dimensional plane[2]. Geometric depictions will be given to describe matroids in Figures 3, 4, 6, and 8.

[2]where a "plane" is sometimes drawn, by necessity, as a circle or other curved item.

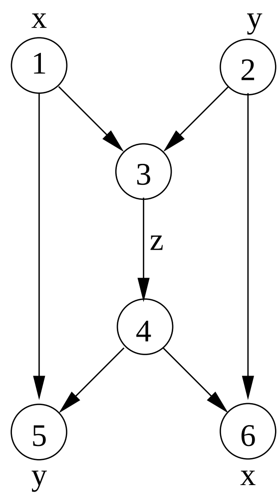


Fig. 1. The Butterfly network has source nodes n_1 and n_2 generating k-dimensional messages x and y, respectively. Receiver nodes n_5 and n_6 demand messages y and x, respectively. The n-dimensional vector carried on edge $e_{3,4}$ is denoted by z.

Definition III.1. 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 r. The network $\mathcal{N}$ is a *matroidal network* associated with $\mathcal{M}$ if there exists a function $f : \mu \cup \epsilon \to \mathcal{S}$ such that the following conditions are satisfied:

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

We call the function f the *network-matroid mapping*. Condition (M1) assigns unique matroid ground set elements to the network messages, and condition (M2) assures that the network messages correspond to an independent set. Condition (M3) reflects the fact that the out-edges of each network node are completely determined by the in-edges and source messages of the node.

Lemma III.2. *For any matroidal network, the polymatroid upper bound on the capacity is at least* 1.

Theorem III.3. *If a network is scalar-linearly solvable over some finite field, then the network is matroidal. In fact, the network is associated with a representable matroid.*

Theorem III.3 suggests a technique for obtaining a network that has a good chance of not being scalar-linearly solvable. That is, choose a network that is matroidal over a non-representable matroid. The Vámos matroid defined in Section III-F is the smallest example of a non-representable matroid [14, p. 512], providing inspiration to define and study a "Vámos network".

The following corollary follows immediately from Theorem III.3 and the fact that all solvable multicast networks are scalar-linearly solvable over some finite field [12].

Corollary III.4. *All solvable multicast networks are matroidal.*

A. The M-Network

Here, we demonstrate that not all solvable networks are matroidal.

We call the network shown in Figure 2 the *M-network* (due to its shape). The M-network was discussed in [13] as an example of a network with no scalar linear solution, but with a simple vector linear solution.

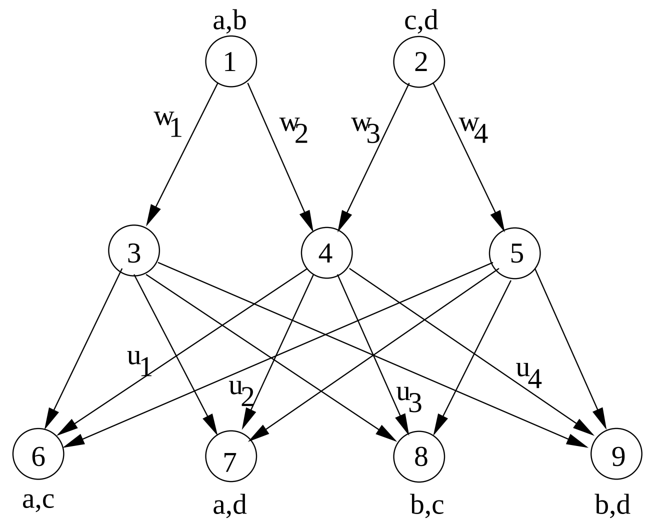


Fig. 2. The M-network. Messages a and b are generated by source n_1 and messages c and d are generated by source n_2. The four messages a, b, c, d are demanded in various pairs at the receivers n_6, n_7, n_8, and n_9. The edges $e_{1,3}$, $e_{1,4}$, $e_{2,4}$, $e_{2,5}$, $e_{4,6}$, $e_{4,7}$, $e_{4,8}$, and $e_{4,9}$, are denoted by w_1, w_2, w_3, w_4, u_1, u_2, u_3, and u_4, respectively.

Theorem III.5. *The M-network is solvable, but is not matroidal.*

The two-dimensional vector-linear solution to the M-network given in [13] is a simple routing solution and easily extends to a vector-linear solution over any even vector dimension. We next show that no other vector dimensions are possible for vector-linear solutions to the M-network.

Theorem III.6. *The M-network does not have any vector-linear solutions of odd vector dimension.*

In particular, the M-network does not have a scalar-linear solution.

B. Method for Constructing Networks from Matroids

We will next describe a method that can be useful for constructing a matroidal network associated with a matroid. Such constructions allow us to transfer various interesting properties of matroids to networks. As matroid theory is a field rich in important results, the goal in constructing matroidal networks is to obtain some analogues for networks.

Let $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ be a matroid with rank function r. Let $\mathcal{N}$ denote the network to be constructed, μ its message set, ν its node set, and ϵ its edge set.

The construction will simultaneously construct the network $\mathcal{N}$, the function

$$f : \mu \cup \epsilon \to \mathcal{S},$$

and an auxiliary function

$$g : \mathcal{S} \to \nu,$$

where for each $x \in \mathcal{S}$, either

(i) $g(x)$ is a source node with message m and $f(m) = x$; or

(ii) $g(x)$ is a node with in-degree 1 and whose in-edge e satisfies $f(e) = x$.

The construction is carried out in 4 stages; each stage can be completed in many ways.

Step 1: Create network source nodes $n_1, n_2, \ldots, n_{r(\mathcal{S})}$ and corresponding messages $m_1, m_2, \ldots, m_{r(\mathcal{S})}$. Choose any base $B = \{b_1, \ldots, b_{r(\mathcal{S})}\}$ for $\mathcal{M}$ and let $f(m_i) = b_i$ and $g(b_i) = n_i$.

Step 2: (to be repeated until it is no longer possible). Find a circuit $\{x_0, \ldots, x_j\}$ in $\mathcal{M}$, such that $g(x_1), \ldots, g(x_j)$ have been already defined, but $g(x_0)$ has not yet been defined. Then we will add the following:

(i) a new node y and edges $e_1, \ldots, e_j$, such that e_i connects $g(x_i)$ to y, and we define $f(e_i) = x_i$.

(ii) another new node n_0 with a single in-edge e_0 connecting y to n_0, and we let $f(e_0) = x_0$ and $g(x_0) = n_0$.

Step 3: (to be repeated as many times as desired). If $\{x_0, \ldots, x_j\}$ is a circuit in $\mathcal{M}$ and $g(x_0)$ is a source node with message m_0, then add to the network a new receiver node y which demands the message m_0 and which has in-edges $e_1, \ldots, e_j$ where e_i connects $g(x_i)$ to y and where $f(e_i) = x_i$.

Step 4: (to be repeated as many times as desired). Choose a base $B = \{x_1, \ldots, x_{r(\mathcal{S})}\}$ of $\mathcal{M}$ and create a receiver node y that demands all of the network messages, and such that y has in-edges $e_1, \ldots, e_{r(\mathcal{S})}$ where e_i connects $g(x_i)$ to y. Let $f(e_i) = x_i$.

Note that after each step above, the network constructed so far is matroidal with respect to $\mathcal{M}$.

It is clear that after Step 2, the function g has been completely determined. This is because for each $x \in \mathcal{S}$, one can always create a circuit containing x and some subset of the starting base B.

It is possible that some circuits cannot be used in Step 3 since they have no element which is mapped by g to a source message. Hence, after this stage of the construction there may be dependencies in $\mathcal{M}$ which are not reflected in the properties of the network $\mathcal{N}$. The final stage (Step 4), however, can at least assure us that all of the independencies in $\mathcal{M}$ are reflected in the properties of $\mathcal{N}$.

C. The Butterfly Network

The Butterfly network in Figure 1 is matroidal associated with the rank-2 uniform matroid $U_{2,3}$ geometrically depicted in Figure 3. The network-matroid mapping (from the network sources and edges to the matroid) constructed along with the network has been partially given[3] in Figure 1. This network is known to have a linear solution over any ring alphabet (by taking $z = x + y$). One can easily check that the conditions (M1)–(M3) hold.

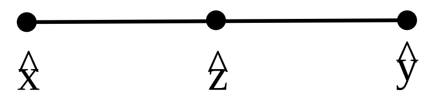
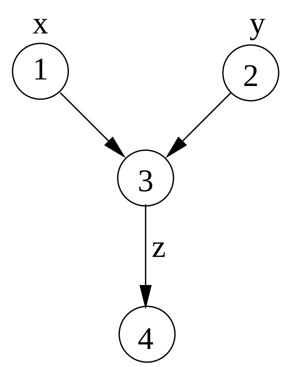


Fig. 3. Geometric depiction of the rank-2 uniform matroid $U_{2,3}$, which can be used to construct the Butterfly network. The matroid has ground set $\{\hat{x}, \hat{y}, \hat{z}\}$ and a set is independent if and only if it does not have three collinear points in the figure (i.e. iff it has size at most 2).

To illustrate the construction of a network from a matroid, we next show the steps from Section III-B involved in the construction of the Butterfly network.

Step 1: We choose a matroid base $B = \{\hat{x}, \hat{y}\}$ and network messages x and y, and we assign $f(x) = \hat{x}$ and $f(y) = \hat{y}$, and $g(\hat{x}) = n_1$ and $g(\hat{y}) = n_2$.

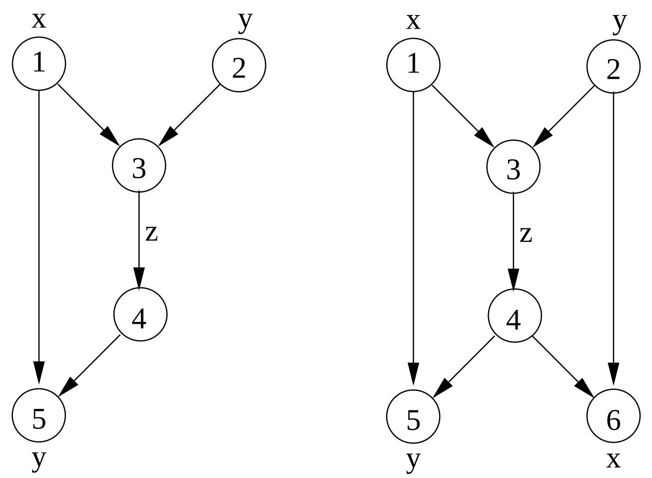
Step 2: The only circuit in the matroid is $\{\hat{x}, \hat{y}, \hat{z}\}$, and $g(\hat{x})$ and $g(\hat{y})$ have already been defined, but $g(\hat{z})$ has not yet been defined. We add a new node n_3 and edges $e_{1,3}$ and $e_{2,3}$, and we define $f(e_{1,3}) = \hat{x}$ and $f(e_{2,3}) = \hat{y}$. We add another new node n_4 with a single in-edge $e_{3,4}$ and we let $f(e_{3,4}) = \hat{z}$ and $g(\hat{z}) = n_4$.



Step 3: The only circuit in the matroid is $\{\hat{x}, \hat{y}, \hat{z}\}$ and $g(\hat{x}) = n_1$ is a source node with message m_1. We add a new receiver node n_5 which demands the message m_2 and has in-edges $e_{1,5}$ and $e_{4,5}$. We repeat this step once more with the same circuit $\{\hat{x}, \hat{y}, \hat{z}\}$, but this time using the source node $g(\hat{y}) = n_2$ with message m_2. We add a new receiver node n_6 which demands the message m_1 and has in-edges $e_{2,6}$ and $e_{4,6}$. The result is the Butterfly network.

Table I lists the dependencies in the uniform matroid $U_{2,3}$ which are directly reflected in the Butterfly network.

[3] Any edge coming from a node with only one input will not be labeled in diagrams, and it can be assumed that any such label equals the label of the unique input to the node.



Step	Variables	Nodes	Type
1	$\{x, y\}$	n_1, n_2	message
2	$\{x, y, z\}$	n_3, n_4	circuit
3	$\{x, y, z\}$	n_5	circuit
3	$\{x, y, z\}$	n_6	circuit
4	none	not used	

TABLE I

DEPENDENCIES IN THE UNIFORM MATROID $U_{2,3}$ THAT ARE REFLECTED IN THE BUTTERFLY NETWORK. THE SECOND COLUMN INDICATES SETS OF VARIABLES IN THE BUTTERFLY NETWORK CORRESPONDING TO DEPENDENT SETS IN THE $U_{2,3}$ MATROID. THE THIRD COLUMN INDICATES AT WHICH NODES IN THE BUTTERFLY NETWORK THE CORRESPONDING DEPENDENCY IS ENFORCED.

D. The Fano Network

Figure 4 is a geometric depiction of the well-known Fano matroid [14]. The network shown in Figure 5, which we call the *Fano network*, is a matroidal network associated with the Fano matroid and is constructed using the technique described in Section III-B. The network-matroid mapping is partially shown in Figure 5 where the mapping on the unlabeled edges is given by the usual convention. The network-matroid mapping is the identity function on the network source messages a, b, and c. It is easy to see that there exists a dependency between any three network variables if and only if the corresponding three matroid elements are dependent. Table II lists the dependencies in the Fano matroid which are directly reflected in the Fano network.

The Fano matroid is known to be F-representable over a finite field F if and only if F has characteristic two [14]. Correspondingly, the Fano network was shown in [5], to be solvable if and only if the alphabet size is an integer power of two. It, in fact, has a linear solution over any finite field of characteristic two (by taking $w = a+b$, $x = a+c$, $y = b+c$, and $z = a+b+c$). The Fano network was used as a building block to construct a network whose coding capacity cannot be achieved by the network. The Fano network was also used as a building block in [4] to construct a solvable network that is not linearly solvable (in a very general sense).

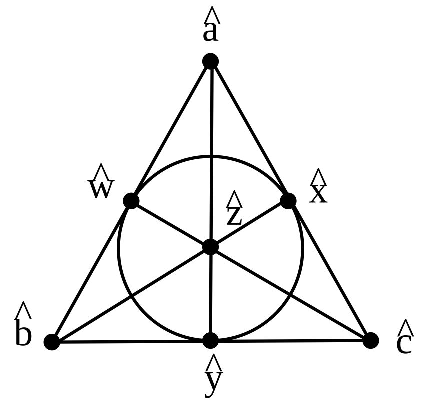


Fig. 4. Geometric depiction of the Fano matroid. The matroid has ground set $\{\hat{a}, \hat{b}, \hat{c}, \hat{w}, \hat{x}, \hat{y}, \hat{z}\}$ and has rank 3. Any three elements of the ground set are dependent if and only if they are collinear in the diagram (where we pretend that points on the drawn circle are also "collinear").

Step	Variables	Nodes	Type
1	$\{a, b, c\}$	n_1, n_2, n_3	message
2	$\{a, b, w\}$	n_4, n_6	circuit
2	$\{b, c, y\}$	n_5, n_7	circuit
2	$\{w, x, y\}$	n_8, n_{10}	circuit
2	$\{c, w, z\}$	n_9, n_{11}	circuit
3	$\{a, c, x\}$	n_{12}	circuit
3	$\{b, x, z\}$	n_{13}	circuit
3	$\{a, y, z\}$	n_{14}	circuit
4	none	not used	

TABLE II

DEPENDENCIES IN THE FANO MATROID THAT ARE REFLECTED IN THE FANO NETWORK. THE SECOND COLUMN INDICATES SETS OF VARIABLES IN THE FANO NETWORK CORRESPONDING TO DEPENDENT SETS IN THE FANO MATROID. THE THIRD COLUMN INDICATES AT WHICH NODES IN THE FANO NETWORK THE CORRESPONDING DEPENDENCY IS ENFORCED.

E. The Non-Fano Network

Figure 6 is a geometric depiction of the well-known non-Fano matroid [14]. The network shown in Figure 7, which we call the *non-Fano network*, is a matroidal network associated with the non-Fano matroid and is constructed using the technique described in Section III-B. The network-matroid mapping is partially shown in Figure 7 where the mapping on the unlabeled edges is given by the usual convention.

Table III lists the dependencies in the non-Fano matroid which are directly reflected in the non-Fano network.

The non-Fano matroid is known [14] to be F-representable over a finite field F if and only if F has odd characteristic. Correspondingly, the non-Fano network was shown in [5], to be solvable if and only if the alphabet size is odd. It, in fact, has a linear solution over any alphabet of odd cardinality (by taking $w = a+b$, $x = a+c$, $y = b+c$, and $z = a+b+c$). The non-Fano network was used as a building block to construct a network whose coding capacity cannot be achieved by the network. The non-Fano network was also used as a building block in [4] to construct a solvable network that is not

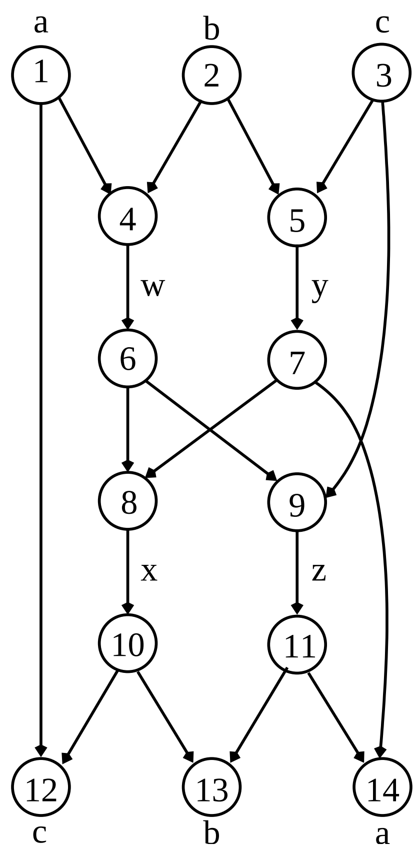


Fig. 5. The Fano network. Messages a, b, and c are emitted by sources n_1, n_2, and n_3, respectively, and are demanded by receivers n_{14}, n_{13}, and n_{12}, respectively. The edges $e_{4,6}$, $e_{5,7}$, $e_{8,10}$, and $e_{9,11}$ are labeled according to the network-matroid mapping by their corresponding ground set elements in the Fano matroid shown in Figure 4.

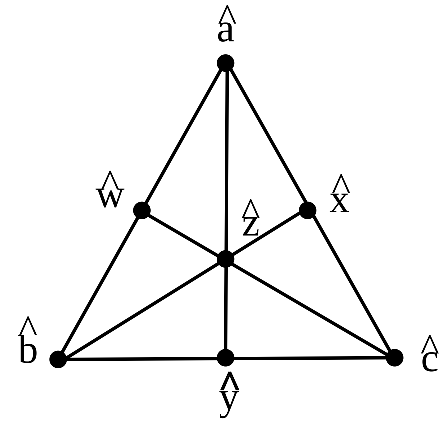


Fig. 6. Geometric depiction of the non-Fano matroid. The matroid has ground set $\{\hat{a}, \hat{b}, \hat{c}, \hat{w}, \hat{x}, \hat{y}, \hat{z}\}$ and has rank 3. Any three elements of the ground set are dependent if and only if they are collinear in the diagram.

Step	Variables	Nodes	Type
1	$\{a, b, c\}$	n_1, n_2, n_3	message
2	$\{a, b, c, z\}$	n_4, n_5	circuit
2	$\{a, b, w\}$	n_6, n_9	circuit
2	$\{a, c, x\}$	n_7, n_{10}	circuit
2	$\{b, c, y\}$	n_8, n_{11}	circuit
3	$\{c, w, x\}$	n_{12}	circuit
3	$\{b, x, z\}$	n_{13}	circuit
3	$\{a, y, z\}$	n_{14}	circuit
4	$\{w, x, y\}$	n_{15}	independent set

TABLE III

DEPENDENCIES IN THE NON-FANO MATROID THAT ARE REFLECTED IN THE NON-FANO NETWORK. THE SECOND COLUMN INDICATES SETS OF VARIABLES IN THE NON-FANO NETWORK CORRESPONDING TO DEPENDENT SETS IN THE NON-FANO MATROID. THE THIRD COLUMN INDICATES AT WHICH NODES IN THE NON-FANO NETWORK THE CORRESPONDING DEPENDENCY IS ENFORCED.

linearly solvable (in a very general sense).

F. The Vámos Network

The *Vámos matroid* is an 8-element rank-4 matroid $(\mathcal{S}, \mathcal{I})$ with

$$\mathcal{S} = \{\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{w}, \hat{x}, \hat{y}, \hat{z}\},$$

and whose dependent sets are the 4-element sets which are coplanar in the three-dimensional drawing in Figure 8 (i.e. precisely $\{\hat{b}, \hat{c}, \hat{x}, \hat{y}\}$, $\{\hat{a}, \hat{c}, \hat{w}, \hat{y}\}$, $\{\hat{a}, \hat{b}, \hat{w}, \hat{x}\}$, $\{\hat{c}, \hat{d}, \hat{y}, \hat{z}\}$, and $\{\hat{b}, \hat{d}, \hat{x}, \hat{z}\}$) and all subsets of $\mathcal{S}$ of cardinality at least 5. Note that $\{\hat{a}, \hat{d}, \hat{w}, \hat{z}\}$ is not considered a coplanar set in Figure 8.

One of the interesting properties of the Vámos matroid is the following.

Theorem III.7. *[14, p. 170] The Vámos matroid is not representable.*

We call the network shown in Figure 9 the *Vámos network*; it is a matroidal network associated with the Vámos matroid and constructed using the technique described in Section III-B. The network has 17 nodes and 4 message variables. Nodes $n_9, \ldots, n_{13}$ are receiver nodes, each demanding one source message, except for n_{11}, which demands two source messages. The network has 4 hidden source nodes, each generating exactly one of the messages a, b, c, d. As depicted in Figure 9, source messages are carried on hidden edges from their hidden source to various other network nodes (e.g. message c is carried by hidden edges from its hidden source to nodes n_1, n_5, n_7, n_{10}, and n_{12}).

The network-matroid mapping $f : \mu \cup \epsilon \to \mathcal{S}$ defined along with the network from the matroid in Figure 9 is determined by: $f(u) = \hat{u}$ for all $u \in \{a, b, c, d, w, x, y, z\}$. Table IV lists the dependencies in the Vámos matroid which are directly reflected in the Vámos network.

<u>Note 1</u>: As depicted in Figure 9, several of the message variables a, b, c, d appear above some of the nodes. This is simply a convenience that makes the depiction easier to draw. When this happens, it is understood that there is an unshown edge from the appropriate source node to the node in question. So, for example, node n_1 actually has four in-edges (not shown), one from each source node (also not shown).

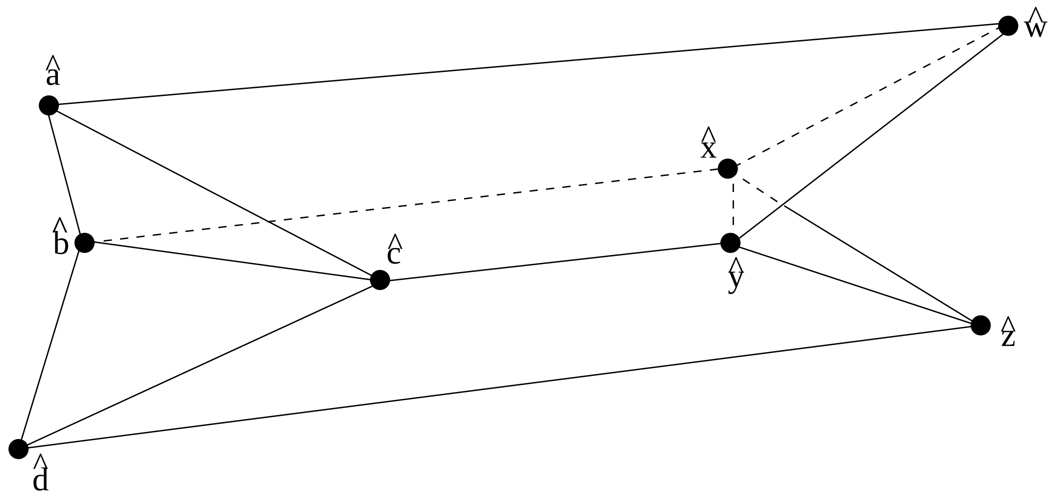


Fig. 8. A 3-dimensional geometric depiction of the Vámos matroid.

Step	Variables	Nodes	Type
1	$\{a, b, c, d\}$	hidden	message
2	$\{a, b, c, d, w\}$	n_1, n_2	circuit
2	$\{a, b, x, w\}$	n_3, n_4	circuit
2	$\{b, c, x, y\}$	n_5, n_6	circuit
2	$\{c, d, y, z\}$	n_7, n_8	circuit
3	$\{b, d, x, z\}$	n_9	circuit
3	$\{a, b, c, d, z\}$	n_{10}	circuit
3	$\{a, b, c, d, y\}$	n_{12}	circuit
3	$\{a, c, w, y\}$	n_{13}	circuit
4	$\{a, d, w, z\}$	n_{11}	independent set

TABLE IV

DEPENDENCIES IN THE VÁMOS MATROID THAT ARE REFLECTED IN THE VÁMOS NETWORK. THE SECOND COLUMN INDICATES SETS OF VARIABLES IN THE VÁMOS NETWORK CORRESPONDING TO DEPENDENT SETS IN THE VÁMOS MATROID. THE THIRD COLUMN INDICATES AT WHICH NODES IN THE VÁMOS NETWORK THE CORRESPONDING DEPENDENCIES ARE ENFORCED.

Theorem III.8. *The Vámos network has routing capacity* $2/5$*, linear coding capacity* $5/6$ *over every finite field, and coding capacity at most* $10/11$.

Corollary III.9. *Shannon-type information inequalities and the network entropy conditions are in general insufficient for determining the coding capacity of a network.*

IV. CREATING MULTIPLE-UNICAST MATROIDAL NETWORKS

In [6], a technique was given for converting arbitrary networks into multiple-unicast networks. The conversion procedure preserves the solvability and linear solvability properties of the original network. In this section, we show that the conversion process also preserves the property of a network being matroidal. This conversion technique was used in [7] to create a multiple-unicast variation of the Vámos network which witnesses the insufficiency of using Shannon-type information inequalities for computing the coding capacity of a multiple-unicast network.

Definition IV.1. A *multiple-unicast version* of a network $\mathcal{N}$ is a network constructed from $\mathcal{N}$ by eliminating multiple sources as and then assuring every message is demanded by exactly one node, as described in [6].

Theorem IV.2. *Every multiple-unicast version of a matroidal network is matroidal.*

The following definition was given in [6]. ("CSLS" stands for "coding solvability and linear solvability".)

Definition IV.3. Two networks $\mathcal{N}$ and $\mathcal{N}'$ are *CSLS-equivalent* if the following two conditions hold:

1) For any 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 solvable over F in dimension k if and only if $\mathcal{N}'$ is vector solvable over F in dimension k.

Lemma IV.4. *[6] Every multiple-unicast version of a network is CSLS-equivalent to that network.*

Corollary IV.5. *Every matroidal network is CSLS-equivalent to a multiple-unicast matroidal network.*

Theorem IV.6. *The Multiple-Unicast Vámos network is matroidal and has coding capacity at most* $12/13$.

In [8], an algorithm is presented for constructing networks from collections of polynomials. It was shown that if a polynomial collection $\mathcal{P}$ is solvable over some finite field, then any network constructed as in [8] from $\mathcal{P}$ is matroidal. 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. It is further shown in [8] that any network is ls-equivalent to a multiple-unicast matroidal network.

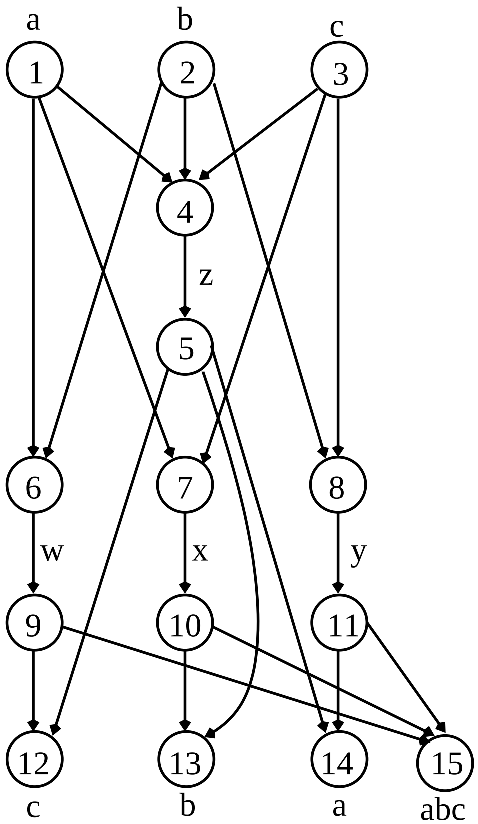


Fig. 7. The non-Fano network. Messages a, b, and c are emitted by sources n_1, n_2, and n_3, respectively, and are demanded by receivers n_{14}, n_{13}, and $\{n_{12}, n_{15}\}$, respectively. The edges $e_{4,5}$, $e_{6,9}$, $e_{7,10}$, and $e_{8,11}$ are labeled according to the network-matroid mapping by their corresponding ground set elements in the non-Fano matroid shown in Figure 6.

REFERENCES

[1] M. Adler, N. J. A. Harvey, K. Jain, R. D. Kleinberg, and A. Rasala Lehman. "On the capacity of information networks", *ACM-SIAM Symposium on Discrete Algorithms (SODA 06)*, Miami, FL, January 2006.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow", *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[3] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity", *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 777-788, March 2006.

[4] 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.

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

[6] R. Dougherty and K. Zeger, "Nonreversibility and equivalent constructions of multiple-unicast networks", *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5067-5077, November 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, C. Freiling, and K. Zeger, "Linear network codes and systems of polynomial equations", *IEEE Transactions on Information Theory*, submitted March 15, 2007 (currently under revision).

[9] N. J. A. Harvey, R. D. Kleinberg, and A. Rasala Lehman. "On the capacity of information networks", joint issue of *IEEE Transactions on Information Theory & IEEE/ACM Transactions on Networking*, vol. 52, no. 6, pp. 2345-2364, June 2006.

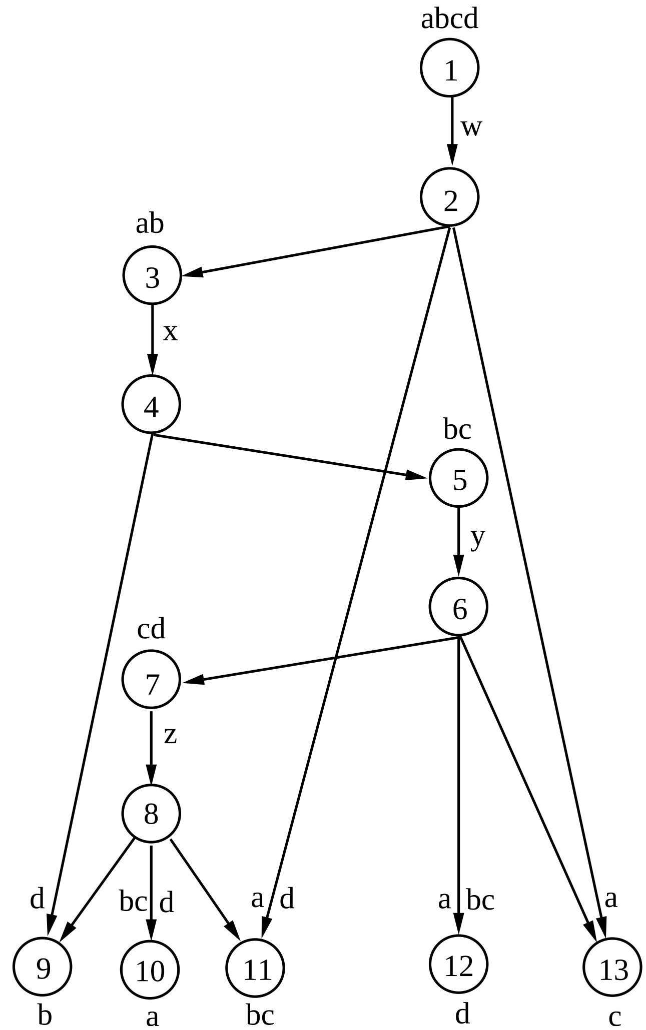


Fig. 9. The Vámos network. A message variable a, b, c, or d labeled above a node indicates an in-edge (not shown) from the source node (not shown) generating the message. Demand variables are labeled below the receivers n_9–n_{13} demanding them. The edges $e_{1,2}$, $e_{3,4}$, $e_{5,6}$, and $e_{7,8}$ are denoted by w, x, y, and z, respectively.

[10] G. Kramer and S. A. Savari, "Edge-cut bounds on network coding rates", *Journal of Network and Systems Management*, vol. 14, no. 1, pp. 49-67, March 2006.

[11] A. Rasala Lehman and E. Lehman, "Network information flow: Does the model need tuning?," *Symposium on Discrete Algorithms (SODA)*, pp. 499–504, Vancouver, British Columbia, Canada, January 2005.

[12] 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.

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

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

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

Note: This paper appears in the proceedings of the 45th Annual Allerton Conference on Communication, Control, and Computing, held September 26-28, 2007.