11-7-2016

Sum-networks from incidence structures: construction and capacity analysis

Ardhendu Tripathy
Iowa State University, ardhendu@iastate.edu

Aditya Ramamoorthy
Iowa State University, adityar@iastate.edu

Sum-networks from incidence structures: construction and capacity analysis

Ardhendu Tripathy, *Student Member, IEEE,* and Aditya Ramamoorthy, *Member, IEEE*

Abstract

A sum-network is an instance of a network coding problem over a directed acyclic network in which each terminal node wants to compute the sum over a finite field of the information observed at all the source nodes. Many characteristics of the well-studied multiple unicast network communication problem also hold for sum-networks due to a known reduction between instances of these two problems. In this work, we describe an algorithm to construct families of sum-network instances using *incidence structures*. The computation capacity of several of these sum-network families is characterized. We demonstrate that unlike the multiple unicast problem, the computation capacity of sum-networks depends on the characteristic of the finite field over which the sum is computed. This dependence is very strong; we show examples of sum-networks that have a rate-1 solution over one characteristic but a rate close to zero over a different characteristic. Additionally, a sum-network can have an arbitrary different number of computation capacities for different alphabets. This is contrast to the multiple unicast problem where it is known that the capacity is independent of the network coding alphabet.

Index Terms

network coding, function computation, sum-networks, characteristic, incidence structures

The material in this work has appeared in part at the 52nd Allerton Conference on Communication, Control and Computing, 2014 and the 2015 IEEE International Symposium on Information Theory. The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011 USA. e-mail: {ardhendu@iastate.edu, adityar@iastate.edu}

Sum-networks from incidence structures: construction and capacity analysis

I. Introduction

Applications as diverse as parallel processing, distributed data analytics and sensor networks often deal with variants of the problem of distributed computation. This has motivated the study of various problems in the fields of computer science, automatic control and information theory. Broadly speaking, one can model this question in the following manner. Consider a directed acyclic network with its edges denoting communication links. A subset of the nodes observe certain information, these nodes are called sources. A different subset of nodes, called terminals, wish to compute functions of the observed information with a certain fidelity. The computation is carried out by the terminals with the aid of the information received over the edges. The demand functions and the network topology are a part of the problem instance and can be arbitrary. This framework is very general and encompasses several problems that have received significant research attention.

Prior work [1], [2], [3] concerning information theoretic issues in function computation worked under the setting of correlated information observed at the sources and *simple* network structures, which were simple in the sense that there were edges connecting the sources to the terminal without any intervening nodes or relays. For instance, [2] characterizes the amount of information that a source must transmit so that a terminal with some correlated side-information can reliably compute a function of the message observed at the source and the side-information. Reference [3] considered distributed functional compression, in which two messages are separately encoded and given to a decoder that computes a function of the two messages with an arbitrarily small probability of error.

With the advent of network coding [4], [5], the scope of the questions considered included the setting in which the information observed at the sources is independent and the network topology is more complex. Under this setting, information is sent from a source to a terminal over a path of edges in the directed acyclic network with one or more intermediate nodes in it, these relay nodes have no limit on their memory or computational power. The communication edges are abstracted into error-free, delay-free links with a certain capacity for information transfer and are sometimes referred to as *bit-pipes*. The messages are required to be recovered with zero distortion. The *multicast* scenario, in which the message observed at the only source in the network is demanded by all terminals in the network, is solved in [4], [6], [5]. A sufficient condition for solvability in the multicast scenario is that each terminal has a max-flow from the source that is at least the entropy rate of the message random process [4]. Reference [6] established that *linear* network codes over a sufficiently large alphabet can solve this problem and [5] provided necessary and sufficient conditions for solving a multicast problem instance in an algebraic framework. The work in [5] also gave a simple algorithm to construct a network code that satisfies it.

Unlike the multicast problem, the multiple unicast problem does not admit such a clean solution. This scenario has multiple source-terminal pairs over a directed acyclic network of bit-pipes and each terminal wants to recover the message sent by its corresponding source with the help of the information transmitted on the network. To begin with, there are problem instances where more than one use of the network is required to solve it. To model this, each network edge is viewed as carrying a vector of n alphabet symbols, while each message is a vector of m alphabet symbols. A network code specifies the relationship between the vector transmitted on each edge of the network and the message vectors, and it solves a network coding problem instance if $m = n$. It is shown that linear network codes are in general not sufficient to solve this problem [7]. One can define the notion of *coding capacity* of a network as the supremum of the ratio m/n over all network codes that allow each terminal to recover its desired message; this ratio m/n for a particular network code is called its *rate*. The coding capacity of a network is independent of the alphabet used [8]. While a network code with any rational rate less than the coding capacity exists by definition and zero-padding, a network code with rate equal to coding capacity does not exist for certain networks, even if the coding capacity is rational [9]. The multi-commodity flow solution to the multiple unicast problem is called a routing solution, as the different messages can be interpreted as distinct commodities *routed* through the intermediate nodes. It is well-known that in the case of multicast, network coding can provide a gain in rate over traditional routing of messages that scales with the size of the network [10]. However, evaluating the coding capacity for an arbitrary instance of the network coding problem is known to be hard in general [11], [12], [13], [14].

Expanding the scope of the demands of the terminals, [15] considered *function computation* over directed acyclic networks with only one terminal; the value to be recovered at the terminal was allowed to be a function of the messages as opposed to being a subset of the set of all messages. This computation is performed using information transmitted over the edges by a network code. Analogous to the coding capacity, a notion of *computation capacity* can be defined in this case. A rate-m/n network code that allows the terminal to compute its demand function has the interpretation that the function can be computed by the terminal m times in n uses of the network. Cut-set based upper bounds for the computation capacity of a directed acyclic network with one terminal were given in [15], [16]. A matching lower bound for function computation in tree-networks was given in [15] and the computation capacity of linear and non-linear network codes for different *classes* of demand functions was explored in [17].

A different flavor of the function computation problem, often called the *sum-network* problem, considers directed acyclic networks with multiple terminals, each of which demands the finite-field sum of all the messages observed at the sources [18], [19]. Reference [20] characterized the requirements that sum-networks with two or three sources or terminals must satisfy so that each terminal can recover the sum at unit rate. Similar to the network coding scenario, a sum-network whose terminals are satisfied by a rate-1 network code are called solvable sum-networks. Reference [19] established that deciding whether an arbitrary instance of a sum-network problem instance is solvable is at least as hard as deciding whether a suitably defined multiple unicast instance is solvable. As a result of this reduction the various characteristics of the solvability problem for network coding instances are also true for the solvability problem for sum-networks; this establishes the broadness of the class of sum-networks within all communication problems on directed acyclic networks. For instance, reference [21] showed that if a collection of integer coefficient polynomials have a common root over a field $\mathcal{F}$, then there exists a directed acyclic network that is solved by a linear network code over $\mathcal{F}$ with $m = n = 1$ (also called a *scalar* linear network code over $\mathcal{F}$) and vice versa. Such an implication does not hold for the class of multicast networks, as it is known that every solvable multicast network has a scalar linear network code over an extension field $GF(2^k)$ for an appropriate k [5], while there exist polynomial collections with no root in $GF(2^k)$. However, because of the equivalence in [19], it can be concluded that there exists a directed acyclic sum-network which has a scalar linear solution over $\mathcal{F}$ if and only if a given polynomial collection has a common root in $\mathcal{F}$.

While solvable sum-networks and solvable network coding instances are intimately related, the results in this paper indicate that these classes of problems diverge when we focus on coding/computation capacity, which can be strictly less than one. In [8, Section VI], the coding capacity of networks is shown to be independent of the finite field chosen as the alphabet for the messages and the information transmitted over the edges. We show that an analogous statement is not true for sum-networks by demonstrating infinite families of sum-network problem instances whose computation capacity varies depending on the finite field alphabet. Moreover, the gap in computation capacity on two different finite fields is shown to scale with the network size for certain classes of sum-networks. For two alphabets $\mathcal{F}_1, \mathcal{F}_2$ of different cardinality and a network $\mathcal{N}$, the authors in [8, Theorem VI.5] described a procedure to simulate a rate-m_2/n_2 network code on $\mathcal{F}_2$ for $\mathcal{N}$ using a rate-m_1/n_1 network code on $\mathcal{F}_1$ for the same network, such that $m_2/n_2 \geq (m_1/n_1) - \epsilon$ for any $\epsilon > 0$. We demonstrate that this procedure does not apply for sum-networks. Along the lines of the counterexample given in [20] regarding minimum max-flow connectivity required for solvability of sum-networks with three sources and terminals, we provide an infinite family of counterexamples that mandate certain value of max-flow connectivity to allow solvability (over some finite field) of a general sum-network with more than three sources and terminals. These sum-network problem instances are arrived at using a systematic construction procedure on combinatorial objects called *incidence structures*. Incidence structures are structured set systems and include, e.g., graphs and combinatorial designs [22]. We note here that combinatorial designs have recently been used to address issues such as the construction of distributed storage systems [23], [24] and coded caching systems [25]

This paper is organized as follows. Section II describes previous work related to the problem considered and summarizes the contributions. Section III describes the problem model formally and Section IV describes the construction procedure we use to obtain the sum-network problem instances considered in this work. Section V gives an upper bound on the computation capacity of these sum-networks and Section VI describes a method to obtain linear network codes that achieve the upper bound on rate for several families of the sum-networks constructed. Section VII interprets the results in this paper and outlines the key conclusions drawn in this paper. Section VIII concludes the paper and discusses avenues for further work.

II. Background, Related Work and Summary of Contributions

The problem setting in which we will work is such that the information observed at the sources are independent and uniformly distributed over a finite field alphabet $\mathcal{F}$. The network links are error-free and assumed to have unit-capacity. Each of the possibly many terminals wants to recover the finite field sum of all the messages with zero error. This problem was introduced in the work of [18]. Intuitively, it is reasonable to assume the network resources, i.e., the capacity of the network links and the network structure have an effect on whether the sum can be computed successfully by all the terminals in the network. Reference [20] characterized this notion for the class of sum-networks that have either two sources and/or two terminals. For this class of sum-networks it was shown that if the source messages had unit-entropy, a max-flow of one between each source-terminal pair was enough to solve the problem. It was shown by means of a counterexample that a max-flow of one was not enough to solve a sum-network with three sources and terminals. However, it was also shown that a max-flow of two between each source-terminal pair was sufficient to solve any sum-network with three sources and three terminals.

Reference [26] considered the computation capacity of the class of sum-networks that have three sources and three or more terminals or vice versa. It was shown that for any integer $k \geq 2$, there exist three-source, n-terminal sum-networks (where $n \geq 3$) whose computation capacity is $\frac{k}{k+1}$. The work most closely related to this paper is [27], which gives a construction procedure that for any positive rational number p/q returns a sum-network whose computation capacity is p/q. Assuming that p and q are relatively prime, the procedure described in [27] constructs a sum-network that has $2q - 1 + \binom{2q-1}{2}$ sources and $2q + \binom{2q-1}{2}$ terminals, which can be very large when q is large. The authors asked the question if there exist smaller sum-networks (i.e., with fewer sources and terminals) that have the computation capacity as p/q. Our work in [28] answered

it in the affirmative and proposed a general construction procedure that returned sum-networks with a prescribed computation capacity. The sum-networks in [27] could be obtained as special cases of this construction procedure. Some smaller instances of sum-networks for specific values were presented in [29]. Small sum-network instances can be useful in determining sufficiency conditions for larger networks.

The scope of the construction procedure proposed in [28] was widened in [30], as a result of which, it was shown that there exist sum-network instances whose computation capacity depends rather strongly on the finite field alphabet. This work builds on the contributions in [28], [30]. In particular, we present a systematic algebraic technique that encompasses the prior results. We also include proofs of all results and discuss the implications of our results in depth.

A. Summary of contributions

In this work, we define several classes of sum-networks for which we can explicitly determine the computation capacity. These networks are constructed by using appropriately defined incidence structures. The main contributions of our work are as follows.

- We demonstrate novel techniques for determining upper and lower bounds on the computation capacity of the constructed sum-networks. In most cases, these bounds match, thus resulting in a determination of the capacity of these sum-networks.
- We demonstrate a strong dependence of the computation capacity on the characteristic of the finite field over which the computation is taking place. In particular, for the *same* network, the computation capacity changes based on the characteristic of the underlying field. This is somewhat surprising because the coding capacity for the multiple unicast problem is known to be independent of the network coding alphabet.
- Consider the class of networks where every source-terminal pair has a minimum cut of value at least α, where α is an arbitrary positive integer. We demonstrate that there exists a sum-network (with a large number of sources and terminals) where the rate of computation can be made arbitrarily small. This implies, that the capacity of sum-networks cannot be characterized just by individual source-terminal minimum cuts.

III. Problem formulation and Preliminaries

We consider communication over a directed acyclic graph (DAG) $G = (V, E)$ where V is the set of nodes and $E \subseteq V \times V \times \mathbb{Z}_+$ are the edges denoting the delay-free communication links between them. The edges are given an additional index as the model allows for multiple edges between two distinct nodes. For instance, if there are two edges between nodes u and v, these will be represented as $(u, v, 1)$ and $(u, v, 2)$. Subset $S \subset V$ denotes the source nodes and $T \subset V$ denotes the terminal nodes. The source nodes have no incoming edges and the terminal nodes have no outgoing edges. Each source node $s_i \in S$ observes an independent random process X_i, such that the sequence of random variables $X_{i1}, X_{i2}, \dots$ indexed by time (denoted by a positive integer) are i.i.d. and each X_{ij} takes values that are uniformly distributed over a finite alphabet $\mathcal{F}$. The alphabet $\mathcal{F}$ is assumed to be a finite field with $|\mathcal{F}| = q$ and its characteristic denoted as $\mathrm{ch}(\mathcal{F})$. Each edge represents a communication channel of unit capacity, i.e., it can transmit one symbol from $\mathcal{F}$ per time slot. When referring to a communication link (or edge) without its third index, we will assume that it is the set of all edges between its two nodes. For such a set denoted by (u, v), we define its capacity $\mathrm{cap}(u, v)$ as the number of edges between u and v. We use the notation $\mathrm{In}(v)$ and $\mathrm{In}(e)$ to represent the set of incoming edges at node $v \in V$ and edge $e \in E$. For the edge $e = (u, v)$ let $\mathrm{head}(e) = v$ and $\mathrm{tail}(e) = u$. Each terminal node $t \in T$ demands the sum (over $\mathcal{F}$) of the individual source messages. Let $Z_j = \sum_{\{i:s_i \in S\}} X_{ij}$ for all $j \in \mathbb{N}$ (the set of natural numbers); then each $t \in T$ wants to recover the sequence $Z := (Z_1, Z_2, \dots)$ from the information it receives on its incoming edges, i.e., the set $\mathrm{In}(t)$.

A network code is an assignment of local encoding functions to each edge $e \in E$ (denoted as $\tilde{\phi}_e(\cdot)$) and a decoding function to each terminal $t \in T$ (denoted as $\psi_t(\cdot)$) such that all the terminals can compute Z. The local encoding function for an edge connected to a set of sources has only the messages observed at those particular source nodes as its input arguments. Likewise, the input arguments for the local encoding function of an edge that is not connected to any source are the values received on its incoming edges and the inputs for the decoding function of a terminal are the values received on its incoming edges. As we consider directed acyclic networks, it can be seen that we can also define a global encoding function that expresses the value transmitted on an edge in terms of the source messages in the set $X := \{X_i : s_i \in S\}$. The global encoding function for an edge e is denoted as $\phi_e(X)$.

The following notation describes the domain and range of the local encoding and decoding functions using two natural numbers m and n for a general vector network code. m is the number of i.i.d. source values that are encoded simultaneously by the local encoding function of an edge that emanates from a source node. n is the number of symbols from $\mathcal{F}$ that are transmitted across an edge in the network. Thus for such an edge e whose $\mathrm{tail}(e) = s \in S$, the local encoding function is $\tilde{\phi}_e(X_{s1}, X_{s2}, \dots, X_{sm}) \in \mathcal{F}^n$. All vectors considered in this paper are assumed to be column vectors unless mentioned otherwise. If u is a vector, the u^T represents its transpose.

- Local encoding function for edge $e \in E$.

$$\tilde{\phi}_e : \mathcal{F}^m \to \mathcal{F}^n \quad \text{if tail}(e) \in S,$$
$$\tilde{\phi}_e : \mathcal{F}^{n|\,\mathrm{In}(\mathrm{tail}(e))|} \to \mathcal{F}^n \quad \text{if tail}(e) \notin S.$$

- Decoding function for the terminal $t \in T$.

$$\psi_t : \mathcal{F}^{n|\operatorname{In}(t)|} \to \mathcal{F}^m.$$

A network code is linear over the finite field $\mathcal{F}$ if all the local encoding and decoding functions are linear transformations over $\mathcal{F}$. In particular, in this case the local encoding functions can be represented via matrix products where the matrix elements are from $\mathcal{F}$. For example, for an edge e such that $\operatorname{tail}(e) \notin S$, let $c \in \mathbb{N}$ be such that $c = |\operatorname{In}(\operatorname{tail}(e))|$ and $\operatorname{In}(\operatorname{tail}(e)) = \{e_1, e_2, \ldots, e_c\}$. Also, let each $\phi_{e_i}(X) \in \mathcal{F}^n$ be denoted as a column vector of size n whose elements are from $\mathcal{F}$. Then the value transmitted on e can be evaluated as

$$\phi_e(X) = \tilde{\phi}_e(\phi_{e_1}(X), \phi_{e_2}(X), \ldots, \phi_{e_c}(X)) = M_e \begin{bmatrix}\phi_{e_1}(X)^T & \phi_{e_2}(X)^T & \ldots & \phi_{e_c}(X)^T\end{bmatrix}^T,$$

where $M_e \in \mathcal{F}^{n \times nc}$ is a matrix indicating the local encoding function for edge e. For the sum-networks that we consider, a valid (m, n) fractional network code solution over $\mathcal{F}$ has the interpretation that the component-wise sum over $\mathcal{F}$ of m i.i.d. source symbols can be communicated to all the terminals in n time slots.

Definition 1: The *rate* of a (m, n) network code is defined to be the ratio m/n. A sum-network is solvable if it has a (m, m) network coding solution for some $m \in \mathbb{N}$.

Definition 2: The *computation capacity* of a sum-network is defined as

$$\sup\left\{\frac{m}{n} : \text{ there is a valid } (m, n) \text{ network code for the given sum-network.}\right\}$$

We use different types of *incidence structures* for constructing sum-networks throughout this paper. We now formally define and present some examples of incidence structures.

Definition 3: Incidence Structure. Let $\mathcal{P}$ be a set of elements called *points*, and $\mathcal{B}$ be a set of elements called *blocks*, where each block is a subset of $\mathcal{P}$. The incidence structure $\mathcal{I}$ is defined as the pair $(\mathcal{P}, \mathcal{B})$. If $p \in \mathcal{P}, B \in \mathcal{B}$ such that $p \in B$, then we say that point p is incident to block B. In general $\mathcal{B}$ can be a multiset, i.e., it can contain repeated elements, but we will not be considering them in our work. Thus for any two distinct blocks B_1, B_2 there is at least one point which is incident to one of B_1 and B_2 and not the other.

We denote the cardinalities of the sets $\mathcal{P}$ and $\mathcal{B}$ by the constants v and b respectively. Thus the set of points and blocks can be indexed by a subscript, and we have that

$$\mathcal{P} = \{p_1, p_2, \ldots, p_v\}, \text{ and } \mathcal{B} = \{B_1, B_2, \ldots, B_b\}.$$

Definition 4: Incidence matrix. The incidence matrix associated with the incidence structure $\mathcal{I}$ is a $(0, 1)$-matrix of dimension $v \times b$ defined as follows.

$$A_{\mathcal{I}}(i, j) := \begin{cases} 1 & \text{if } p_i \in B_j, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, incidence matrices can be viewed as general set systems. For example, a simple undirected graph can be viewed as an incidence structure where the vertices are the points and edges are the blocks (each block of size two). Combinatorial designs [22] form another large and well-investigated class of incidence structures. In this work we will use the properties of t-designs which are defined next.

Definition 5: t-design. An incidence structure $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ is a t-(v, k, λ) design, if

- it has v points, i.e., $|\mathcal{P}| = v$,
- each block $B \in \mathcal{B}$ is a k-subset of the point set $\mathcal{P}$, and
- $\mathcal{P}$ and $\mathcal{B}$ satisfy the *t-design property*, i.e., any t-subset of $\mathcal{P}$ is present in exactly λ number of blocks.

A t-(v, k, λ) design is called *simple* if there are no repeated blocks, i.e., $\mathcal{B}$ is a set and not a multiset. In this work, we work exclusively with simple t-(v, k, λ) designs. These designs have been the subject of much investigation when $t = 2$; in this case they are also called balanced incomplete block designs (BIBDs).

Example 1: A famous example of a 2-design with $\lambda = 1$ is the Fano plane $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ shown in Figure 1. Letting numerals denote points and alphabets denote blocks for this design, we can write:

$$\mathcal{P} = \{1, 2, 3, 4, 5, 6, 7\},\ \mathcal{B} = \{A, B, C, D, E, F, G\},\ \text{where}$$
$$A = \{1, 2, 3\},\ B = \{3, 4, 5\},\ C = \{1, 5, 6\},\ D = \{1, 4, 7\},\ E = \{2, 5, 7\},\ F = \{3, 6, 7\},\ G = \{2, 4, 6\}.$$

The corresponding incidence matrix $A_{\mathcal{I}}$, with rows and columns arranged in numerical and alphabetical order, is shown below.

$$A_{\mathcal{I}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{1}$$

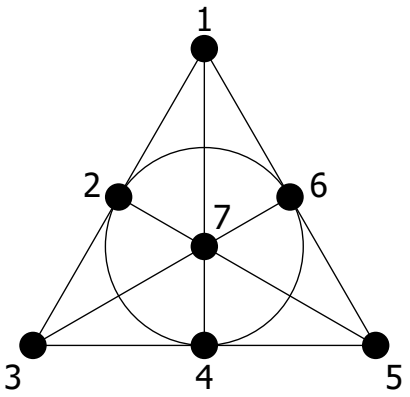


Fig. 1. A pictorial depiction of the Fano plane. The point set $\mathcal{P} = \{1, \ldots, 7\}$. The blocks are indicated by a straight line joining their constituent points. The points $2, 4$ and 6 lying on the circle also depict a block.

It can be verified that every pair of points in $\mathcal{P}$ appears in exactly one block in $\mathcal{B}$.

There are some well-known necessary conditions that need to be satisfied for the existence of a t-(v, k, λ) design (see [22]).

- Let b_i denote the number of blocks that are incident to any i-subset of $\mathcal{P}$ where $i \leq t$. Then,
$$b_i = \lambda \binom{v-i}{t-i} / \binom{k-i}{t-i}, \ \forall i \in \{0, 1, 2, \ldots, t\}. \tag{2}$$
We note that b_0 is simply the total number of blocks denoted by b. Likewise, b_1 represents the number of blocks that each point is incident to; we use the symbol ρ to represent it. Furthermore, $b_t = \lambda$.
It follows that a necessary condition for the existence of a t-(v, k, λ) design is that $\binom{k-i}{t-i}$ divides $\lambda\binom{v-i}{t-i}$ for all $i = 1, 2, \ldots, t$.
- Counting the number of ones in the point-block incidence matrix for a particular design in two different ways, we arrive at the equation $bk = v\rho$.

IV. Construction of a family of sum-networks

Let $[t] := \{1, 2, \ldots, t\}$ for any $t \in \mathbb{N}$. Our construction of sum-networks takes as input a $(0, 1)$-matrix A of dimension $r \times c$. It turns out that the constructed sum-networks have interesting properties when the matrix A is the incidence matrix of appropriately chosen incidence structures. The construction algorithm is presented in Algorithm 1. The various steps in the algorithm that construct components of the sum-network $G = (V, E)$ are described below.

1) *Source node set S and terminal node set T:* Let $\boldsymbol{p}_i$ denote the ith row of A for $i \in [r]$ and $\boldsymbol{B}_j$ denote the jth column of A for $j \in [c]$ [1]. Then S and T both contain $r + c$ nodes, one for each row and column of A. The source nodes are denoted at line 4 as s_{p_i}, s_{B_j} if they correspond to the ith row, jth column respectively. The terminal nodes are also denoted in a similar manner at line 5. They are added to the vertex set V of the sum-network at line 6.
2) *Bottleneck edges:* We add r unit-capacity edges indexed as e_i for $i \in [r]$ in line 2 to the edge set E. Each edge e_i thus corresponds to a row of the matrix A. We also add the required tail and head vertices of these edges to V.
3) *Edges between $S \cup T$ and the bottleneck edges:* For every $i \in [r]$, we connect $\mathrm{tail}(e_i)$ to the source node corresponding to the row $\boldsymbol{p}_i$ and to the source nodes that correspond to all columns of A with a 1 in the ith row. This is described in line 8 of the algorithm. Line 9 describes a similar operation used to connect each $\mathrm{head}(e_i)$ to certain terminal nodes.
4) *Direct edges between S and T:* For each terminal in T, these edges directly connect it to source nodes that do not have a path to that particular terminal through the bottleneck edges. Using the notation for rows and columns of the matrix A, they can be characterized as in lines 12 and 15.

For an incidence structure $\mathcal{I}$, let $A_{\mathcal{I}}$ represent its incidence matrix. The sum-networks constructed in the paper are such that the matrix A used in the SUM-NET-CONS algorithm is either equal to $A_{\mathcal{I}}$ or $A_{\mathcal{I}}^T$ for some incidence structure $\mathcal{I}$. When $A = A_{\mathcal{I}}$, we call the sum-network constructed as the *normal* sum-network for $\mathcal{I}$. Otherwise when $A = A_{\mathcal{I}}^T$, we call the sum-network constructed as the *transpose* sum-network for $\mathcal{I}$. The following definitions are useful for analysis. For every $p \in \mathcal{P}$, we denote the set of blocks that contain the point p as

$$\langle p \rangle := \{B \in \mathcal{B} : p \in B\}, \tag{3}$$

[1]A justification for this notation is that later when we use the incidence matrix ($A_{\mathcal{I}}$) of an incidence structure $\mathcal{I}$ to construct a sum-network, the rows and columns of the incidence matrix will correspond to the points and blocks of $\mathcal{I}$ respectively.

Algorithm 1 SUM-NET-CONS

Input: A. /* A is a $(0,1)$-matrix of size $r \times c$ */
Output: $G = (V, E)$. /* G is the directed acyclic sum-network returned */
1: Initialize $V, E, S, T \leftarrow \phi$.
2: $E \leftarrow \{e_i : i \in [r]\}$. /* these are bottleneck edges */
3: $V \leftarrow \{\text{head}(e_i), \text{tail}(e_i) : i \in [r]\}$.
4: $S \leftarrow \{s_{p_i} : i \in [r]\} \cup \{s_{B_j} : j \in [c]\}$. /* the source-node set */
5: $T \leftarrow \{t_{p_i} : i \in [r]\} \cup \{t_{B_j} : j \in [c]\}$. /* the terminal-node set */
6: $V \leftarrow V \cup S \cup T$.
7: **for all** $i \in [r]$ **do** /* edges connecting the bottleneck edges to source and terminal nodes */
8: $\quad E \leftarrow E \cup \{(s_{B_j}, \text{tail}(e_i)) : A(i,j) = 1; j \in [c]\} \cup \{(s_{p_i}, \text{tail}(e_i))\}$.
9: $\quad E \leftarrow E \cup \{(\text{head}(e_i), t_{B_j}) : A(i,j) = 1; j \in [c]\} \cup \{(\text{head}(e_i), t_{p_i})\}$.
10: **end for**
11: **for all** $i \in [r]$ **do** /* direct edges connecting source nodes to row-terminals */
12: $\quad E \leftarrow E \cup \{(s_{p_j}, t_{p_i}) : i \neq j; j \in [r]\} \cup \{(s_{B_j}, t_{p_i}) : A(i,j) = 0; j \in [c]\}$.
13: **end for**
14: **for all** $j \in [c]$ **do** /* direct edges connecting source nodes to column-terminals */
15: $\quad E \leftarrow E \cup \{(s_{p_i}, t_{B_j}) : A(i,j) = 0; i \in [r]\} \cup \{(s_{B_{j'}}, t_{B_j}) : \boldsymbol{B}_j^T \boldsymbol{B}_{j'} = 0; j' \in [c]\}$.
16: **end for**
17: **return** $G \leftarrow (V, E)$.

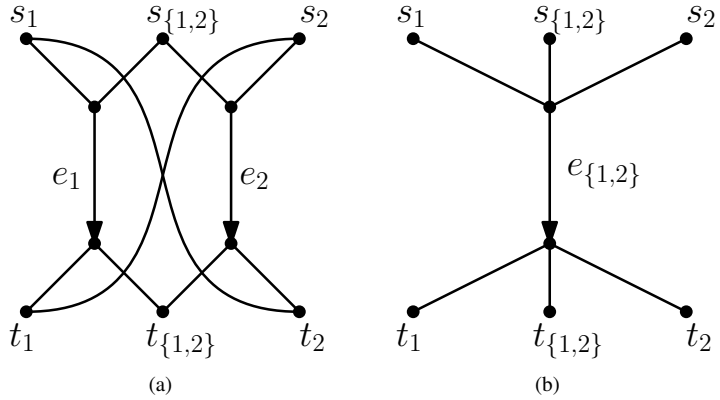


Fig. 2. Two sum-networks obtained from the line graph on two vertices described in Example 2. The source set S and the terminal set T contain three nodes each. All edges are unit-capacity and point downward. The edges with the arrowheads are the bottleneck edges constructed in step 1 of the construction procedure. (a) Normal sum-network, and (b) transposed sum-network.

and for every $B \in \mathcal{B}$, the collection of blocks that have a non-empty intersection with B is denoted by the set

$$\langle B \rangle := \{B' \in \mathcal{B} : B' \cap B \neq \phi\} \tag{4}$$
$$= \{B' \in \mathcal{B} : \boldsymbol{B}^T \boldsymbol{B'} \neq 0\}. \tag{5}$$

The inner product above is computed over the reals. In the sequel, we will occasionally need to perform operations similar to the inner product over a finite field. This shall be explicitly pointed out. The number of edges and nodes added in the SUM-NET-CONS algorithm for a normal sum-network and a transpose sum-network are summarized in Appendix A. We now present some examples of sum-networks constructed using the above technique.

Example 2: Let $\mathcal{I}$ be the unique simple line graph on two vertices, with points corresponding to the vertices and blocks corresponding to the edges of the graph. Denoting the points as natural numbers, we get that $\mathcal{P} = \{1, 2\}$ and $\mathcal{B} = \{\{1, 2\}\}$. Then the associated incidence matrices are as follows.

$$A_{\mathcal{I}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \text{and } A_{\mathcal{I}}^T = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

Following the SUM-NET-CONS algorithm the two sum-networks obtained are as shown in the Figure 2.

Example 3: In this example we construct a sum-network using a simple t-design. Let $\mathcal{I}$ denote the 2-$(3, 2, 1)$ design with its points denoted by the numbers $\{1, 2, 3\}$ and its blocks denoted by the letters $\{A, B, C\}$. For this design we have that

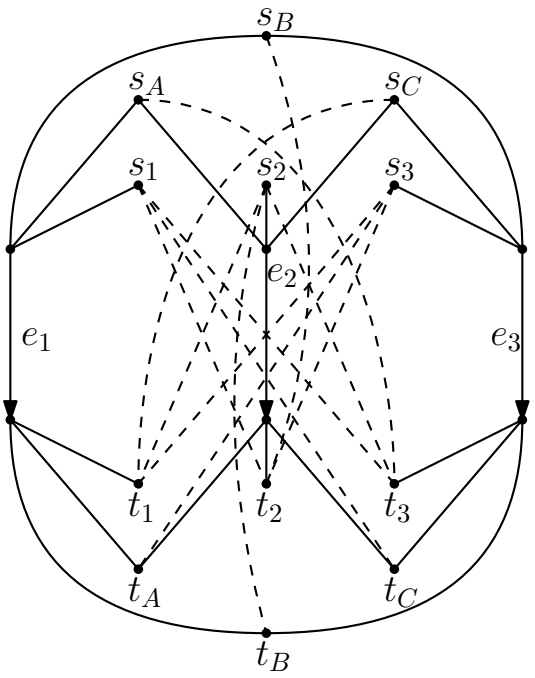


Fig. 3. The normal sum-network obtained for the incidence structure $\mathcal{I}$ described in Example 3. All edges are unit-capacity and directed downward. The edges with the arrowheads are the bottleneck edges, and the edges denoted by dashed lines correspond to the direct edges introduced in step 4 of the construction procedure. For this case, the normal and the transposed sum-network are identical.

$A = \{1, 2\}, B = \{1, 3\}, C = \{2, 3\}$ and its associated incidence matrix under row and column permutations can be written as follows.

$$A_{\mathcal{I}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Note that $A_{\mathcal{I}} = A_{\mathcal{I}}^T$. Hence the normal sum-network and the transposed sum-network are identical in this case. Following the SUM-NET-CONS algorithm, we obtain the sum-network shown in Figure 3.

Remark 1: Note that each edge added in the SUM-NET-CONS algorithm has unit capacity. Proposition 5 in Section VII modifies the SUM-NET-CONS algorithm so that each edge e in the sum-network has $\text{cap}(e) = \alpha$ where $\alpha \in \mathbb{N}, \alpha > 1$.

V. Upper bound on the computation capacity

In this section, we describe an upper bound on the computation capacity of a sum-network obtained from a $(0, 1)$-matrix A of dimension $r \times c$. We assume that there exists a (m, n) fractional network code assignment, i.e., $\tilde{\phi}_e$ for $e \in E$ (and corresponding global encoding functions $\phi_e(X)$) and decoding functions ψ_t for $t \in T$ so that all the terminals in T can recover the sum of all the independent sources.

For convenience of presentation, we will change notation slightly and let the messages observed at the source nodes corresponding to the rows of A as X_{p_i} for $i \in [r]$ and those corresponding to the columns of A as X_{B_j} for $j \in [c]$. Each of the messages is a vector of length m over $\mathcal{F}$. The set of all source messages is represented by X. We let $\phi_e(X)$ denote the n-length vector of symbols from $\mathcal{F}$ that are transmitted by the edge $e \in E$, as it is the value returned by the global encoding function ϕ_e for edge e on the set of inputs denoted by X. As is apparent for our networks, non-trivial encoding functions can only be employed on the bottleneck edges, i.e., e_i for $i \in [r]$ as these are the only edges that have more than one input. For brevity, we set $\phi_i(X) = \phi_{e_i}(X)$. In addition, we define the following set of global encoding functions.

$$\phi_{\text{In}(v)}(X) := \{\phi_e(X) : e \in \text{In}(v)\}, \ \forall v \in V.$$

Let $H(Y)$ be the entropy function for a random variable Y. We use the notation $H(Y_1, Y_2, \ldots, Y_l) = H(\{Y_i\}_1^l)$ for any $l > 1$. The following lemma demonstrates that certain partial sums can be computed by observing subsets of the bottleneck edges.

Lemma 1: If a network code allows each terminal to compute the demanded sum, then the value $X'_{p_i} := X_{p_i} + \sum_{j:A(i,j)=1} X_{B_j}$ can be computed from $\phi_i(X)$, i.e., $H\left(X'_{p_i} | \phi_i(X)\right) = 0$ for all $i \in [r]$. Similarly for any $j \in [c]$ the value $X'_{B_j} := \sum_{i:A(i,j)=1} X_{p_i} + \sum_{j':B_{j'} \in \langle B_j \rangle} X_{B_{j'}}$ can be computed from the set of values $\{\phi_i(X) : \text{ for } i, \text{ such that } A(i, j) = 1\}$.

Proof: We let for any $i \in [r]$

$$Z_1 = \sum_{i' \neq i} X_{p_{i'}},\ Z_2 = \sum_{j:A(i,j)=1} X_{B_j} \text{ and } Z_3 = \sum_{j:A(i,j)=0} X_{B_j},$$

such that the sum $Z = X_{p_i} + Z_1 + Z_2 + Z_3$ and $X'_{p_i} = X_{p_i} + Z_2$.

By our assumption that each terminal can recover the demanded sum, we know that Z can be evaluated from $\phi_{\mathrm{In}(t_{p_i})}(X)$ for all $i \in [r]$, i.e., $H\left(Z|\phi_{\mathrm{In}(t_{p_i})}(X)\right) = 0$ for all $i \in [r]$. Since $\{X_{p_{i'}} : i' \neq i\}$ and $\{X_{B_j} : A(i,j) = 0\}$ determine the value of Z_1 and Z_3 respectively and also determine the values transmitted on each of the unit-capacity edges that directly connect certain source nodes to t_{p_i}, we get that

$$\begin{aligned}
H\left(Z|\phi_{\mathrm{In}(t_{p_i})}(X)\right) &= H\left(Z|\phi_i(X), \{\phi_{(s_{p_{i'}},t_{p_i})}(X) : i' \neq i\}, \{\phi_{(s_{B_j},t_{p_i})} : A(i,j) = 0\}\right) \\
&\overset{(a)}{\geq} H\left(X_{p_i} + Z_1 + Z_2 + Z_3|\phi_i(X), \{X_{p_{i'}} : i' \neq i\}, \{X_{B_j} : A(i,j) = 0\}\right) \\
&= H\left(X'_{p_i}|\phi_i(X), \{X_{p_{i'}} : i' \neq i\}, \{X_{B_j} : A(i,j) = 0\}\right) \\
&= H\left(X'_{p_i}, \{X_{p_{i'}} : i' \neq i\}, \{X_{B_j} : A(i,j) = 0\}|\phi_i(X)\right) \\
&\quad - H\left(\{X_{p_{i'}} : i' \neq i\}, \{X_{B_j} : A(i,j) = 0\}|\phi_i(X)\right) \\
&= H\left(X'_{p_i}|\phi_i(X)\right) + H\left(\{X_{p_{i'}} : i' \neq i\}, \{X_{B_j} : A(i,j) = 0\}|X'_{p_i}, \phi_i(X)\right) \\
&\quad - H\left(\{X_{p_{i'}} : i' \neq i\}, \{X_{B_j} : A(i,j) = 0\}|\phi_i(X)\right) \\
&\overset{(b)}{=} H\left(X'_{p_i}|\phi_i(X)\right) \qquad (6)
\end{aligned}$$

where inequality (a) follows from the fact that $\phi_{(s_{p_{i'}},t_{p_i})}(X)$ is a function of $X_{p_{i'}}$ for $i' \neq i$ and $\phi_{(s_{B_j},t_{p_i})}(X)$ is a function of $\{X_{B_j} : A(i,j) = 0\}$ and equality (b) is due to the fact that X'_{p_i} is conditionally independent of both $\{X_{p_{i'}} : i' \neq i\}$ and $\{X_{B_j} : A(i,j) = 0\}$ given $\phi_i(X)$. This conditional independence can be checked as follows. Let bold lowercase symbols represent specific realizations of the random variables.

$$\begin{aligned}
&\Pr\left(X'_{p_i} = \boldsymbol{x}'_{p_i}, \{X_{p_{i'}} = \boldsymbol{x}_{p_{i'}} : i' \neq i\}, \{X_{B_j} = \boldsymbol{x}_{B_j} : A(i,j) = 0\}|\phi_i(X) = \phi_i(\boldsymbol{x})\right) \\
&\overset{(a)}{=} \Pr(X'_{p_i} = \boldsymbol{x}'_{p_i}, \phi_i(X) = \phi_i(\boldsymbol{x}))\Pr(\{X_{p_{i'}} = \boldsymbol{x}_{p_{i'}} : i' \neq i\}, \{X_{B_j} = \boldsymbol{x}_{B_j} : A(i,j) = 0\})/\Pr(\phi_i(X) = \phi_i(\boldsymbol{x})) \\
&= \Pr\left(X'_{p_i} = \boldsymbol{x}'_{p_i}|\phi_i(X) = \phi_i(\boldsymbol{x})\right)\Pr(\{X_{p_{i'}} = \boldsymbol{x}_{p_{i'}} : i' \neq i\}, \{X_{B_j} = \boldsymbol{x}_{B_j} : A(i,j) = 0\}) \\
&\overset{(b)}{=} \Pr\left(X'_{p_i} = \boldsymbol{x}'_{p_i}|\phi_i(X) = \phi_i(\boldsymbol{x})\right)\Pr(\{X_{p_{i'}} = \boldsymbol{x}_{p_{i'}} : i' \neq i\}, \{X_{B_j} = \boldsymbol{x}_{B_j} : A(i,j) = 0\}|\phi_i(X) = \phi_i(\boldsymbol{x})),
\end{aligned}$$

where equalities (a) and (b) are due to the fact that the source messages are independent and $\phi_i(\boldsymbol{x})$ is a function of $\boldsymbol{x}_{p_i}$ and messages in the set $\{\boldsymbol{x}_{B_j} : A(i,j) = 1\}$. Since terminal t_{p_i} can compute Z, $H\left(Z|\phi_{\mathrm{In}(t_{p_i})}(X)\right) = 0$ and we get from Eq (6) that $H(X_{p_i} + Z_2|\phi_i(X)) = 0$.

For the second part of the lemma, we argue similarly as follows. We let for any $j \in [c]$

$$Z_1 = \sum_{i:A(i,j)=1} X_{p_i}, Z_2 = \sum_{i:A(i,j)=0} X_{p_i}, Z_3 = \sum_{j':B_{j'} \in \langle B_j \rangle} X_{B_{j'}}, Z_4 = \sum_{j':B_{j'} \notin \langle B_j \rangle} X_{B_{j'}}$$

such that $Z = Z_1 + Z_2 + Z_3 + Z_4$ and $X'_{B_j} = Z_1 + Z_3$. By our assumption, for all $j \in [c]$, $H\left(Z|\phi_{\mathrm{In}(t_{B_j})}(X)\right) = 0$. The sets $\{X_p : p \notin B_j\}$ and $\{X_B : B \notin \langle B_j \rangle\}$ determine the value of Z_2 and Z_4 respectively and also the values transmitted on each of the direct edges that connect a source node to the terminal t_{B_j}. Hence, we have that

$$\begin{aligned}
&H\left(Z|\phi_{\mathrm{In}(t_{B_j})}(X)\right) \\
&= H\left(Z_1 + Z_2 + Z_3 + Z_4|\{\phi_i(X) : A(i,j) = 1\}, \{\phi_{(s_{p_i},t_{B_j})}(X) : A(i,j) = 0\}, \{\phi_{(s_{B_{j'}},t_{B_j})} : B_{j'} \notin \langle B_j \rangle\}\right) \\
&\overset{(a)}{\geq} H\left(Z_1 + Z_2 + Z_3 + Z_4|\{\phi_i(X) : A(i,j) = 1\}, \{X_{p_i} : A(i,j) = 0\}, \{X_{B_{j'}} : B_{j'} \notin \langle B_j \rangle\}\right) \\
&= H\left(X'_{B_j}|\{\phi_i(X) : A(i,j) = 1\}, \{X_{p_i} : A(i,j) = 0\}, \{X_{B_{j'}} : B_{j'} \notin \langle B_j \rangle\}\right) \\
&= H\left(X'_{B_j}, \{X_{p_i} : A(i,j) = 0\}, \{X_{B_{j'}} j' : B_{j'} \notin \langle B_j \rangle\}|\{\phi_i(X) : A(i,j) = 1\}\right) \\
&\quad - H\left(\{X_{p_i} : A(i,j) = 0\}, \{X_{B_{j'}} j' : B_{j'} \notin \langle B_j \rangle\}|\{\phi_i(X) : A(i,j) = 1\}\right) \\
&= H\left(X'_{B_j}|\{\phi_i(X) : A(i,j) = 1\}\right) + H\left(\{X_{p_i} : A(i,j) = 0\}, \{X_{B_{j'}} : B_{j'} \notin \langle B_j \rangle\}|X'_{B_j}, \{\phi_i(X) : A(i,j) = 1\}\right) \\
&\quad - H\left(\{X_{p_i} : A(i,j) = 0\}, \{X_{B_{j'}} : B_{j'} \notin \langle B_j \rangle\}|\{\phi_i(X) : A(i,j) = 1\}\right)
\end{aligned}$$

$$\overset{(b)}{=} H\left(X'_{B_j}|\{\phi_i(X) : A(i,j)=1\}\right).$$

Inequality (a) is due to the fact that $\phi_{(s_{p_i},t_{B_j})}(X)$ is a function of X_{p_i} and similarly for $\phi_{(s_{B_{j'}},t_{B_j})}(X)$. Equality (b) follows from the fact that Z_1+Z_3 is conditionally independent of both $\{X_{p_i} : A(i,j)=0\}$ and $\{X_{B_{j'}} : B_{j'} \notin \langle B_j\rangle\}$ given the set of random variables $\{\phi_i(X) : A(i,j)=1\}$. This can be verified in a manner similar to as was done previously. This gives us the result that $H(X'_{B_j}|\{\phi_i(X) : A(i,j)=1\})=0$. ∎

Next, we show that the fact that the messages observed at the source nodes are independent and uniformly distributed over $\mathcal{F}^m$ imply that the random variables X'_{p_i} for all $i\in[r]$ are also uniform i.i.d. over $\mathcal{F}^m$. To do that, we introduce some notation. For a matrix $N\in\mathcal{F}^{r\times c}$, for any two index sets $\mathcal{R}\subseteq[r],\mathcal{C}\subseteq[c]$, we define the submatrix of N containing the rows indexed by $\mathcal{R}$ and the columns indexed by $\mathcal{C}$ as $N[\mathcal{R},\mathcal{C}]$. Consider two $(0,1)$-matrices N_1,N_2 of dimensions $r_1\times t$ and $t\times c_2$ respectively. Here 1 and 0 indicate the multiplicative and additive identities of the finite field $\mathcal{F}$ respectively. The ith row of N_1 is denoted by the row submatrix $N_1[i,[t]]\in\{0,1\}^t$ and the jth column of N_2 be denoted by the column submatrix $N_2[[t],j]\in\{0,1\}^t$. Then we define a matrix function on N_1N_2 that returns a $r_1\times c_2$ matrix $(N_1N_2)_\#$ as follows.

$$(N_1N_2)_\#(i,j)=\begin{cases}1, & \text{if the inner product } N_1[i,[t]]\,N_2[[t],j] \text{ over } \mathbb{Z} \text{ is positive}\\ 0, & \text{otherwise.}\end{cases}$$

For an incidence structure $\mathcal{I}=(\mathcal{P},\mathcal{B})$ with $r\times c$ incidence matrix A, let $X_p,\ \forall p\in\mathcal{P}$ and $X_B,\ \forall B\in\mathcal{B}$ be m-length vectors with each component i.i.d. uniformly distributed over the elements of a finite field $\mathcal{F}$ of size q. We collect all the independent source random variables in a column vector $\boldsymbol{X}$ having $m(r+c)$ elements from $\mathcal{F}$ as follows

$$\boldsymbol{X}:=\begin{bmatrix}X_{p_1}^T & X_{p_2}^T & \cdots & X_{p_r}^T & X_{B_1}^T & X_{B_2}^T & \cdots & X_{B_c}^T\end{bmatrix}^T.$$

Recall that $\boldsymbol{p}_i$ denotes the ith row and $\boldsymbol{B}_j$ denotes the jth column of the matrix A. For all $i\in[r]$ let $\boldsymbol{e}_i\in\mathcal{F}^r$ denote the vector with 1 in its ith component and zero elsewhere. Then for X'_{p_i},X'_{B_j} as defined in Lemma 1, one can check that ($\otimes$ indicates the Kronecker product of two matrices)

$$X'_{p_i}=\left(\begin{bmatrix}\boldsymbol{e}_i^T & \boldsymbol{p}_i\end{bmatrix}\otimes I_m\right)\boldsymbol{X},\ \text{ for all } i\in[r] \text{ and} \tag{7}$$

$$X'_{B_j}=\left(\begin{bmatrix}\boldsymbol{B}_j^T & (\boldsymbol{B}_j^T\boldsymbol{B}_1)_\# & (\boldsymbol{B}_j^T\boldsymbol{B}_2)_\# & \cdots & (\boldsymbol{B}_j^T\boldsymbol{B}_c)_\#\end{bmatrix}\otimes I_m\right)\boldsymbol{X},\ \text{ for all } j\in[c], \tag{8}$$

where I_m is the identity matrix of size m. By stacking these values in the correct order, we can get the following matrix equation.

$$\begin{bmatrix}X'^T_{p_1} & X'^T_{p_2} & \cdots & X'^T_{p_r} & X'^T_{B_1} & X'^T_{B_2} & \cdots & X'^T_{B_c}\end{bmatrix}^T=(M_A\otimes I_m)\boldsymbol{X} \tag{9}$$

where the matrix $M_A\in\mathcal{F}^{(r+c)\times(r+c)}$ is defined as follows.

$$M_A:=\begin{bmatrix}I_r & A\\ A^T & (A^TA)_\#\end{bmatrix}. \tag{10}$$

Note that the first r rows of M_A are linearly independent. There is a natural correspondence between the rows of M_A and the points and blocks of $\mathcal{I}$ of which A is the incidence matrix. If $1\le i\le r$, then the ith row $M_A[i,[r+c]]$ corresponds to the point $p_i\in\mathcal{P}$ and if $r+1\le j\le r+c$, then the jth row $M_A[j,[r+c]]$ corresponds to the block $B_j\in\mathcal{B}$.

Lemma 2: For a $(0,1)$-matrix A of size $r\times c$, let $X'_{p_i},X'_{B_j}\in\mathcal{F}^m$ be as defined in equations (7), (8) and matrix M_A be as defined in equation (10). Let $r+t:=\operatorname{rank}_{\mathcal{F}}(M_A)$ for some non-negative integer t and index set $\mathcal{S}'\subseteq\{r+1,r+2,\ldots,r+c\}$ be such that $\operatorname{rank}_{\mathcal{F}}(M_A[[r]\cup\mathcal{S}',[r+c]])=r+t$. Let $\mathcal{B}_{\mathcal{S}'}:=\{B_{\mathcal{S}'_1},B_{\mathcal{S}'_2},\ldots,B_{\mathcal{S}'_t}\}\subseteq\mathcal{B}$ be the set of blocks that correspond to the rows of M_A indexed by $\mathcal{S}'$ in increasing order. Then we have

$$\Pr\left(X'_{p_1}=x'_1,\ldots,X'_{p_r}=x'_r,X_{B_{\mathcal{S}'_1}}=y'_1,\ldots,X_{B_{\mathcal{S}'_t}}=y'_t\right)=\prod_{i=1}^{r}\Pr\left(X'_{p_i}=x'_i\right)\prod_{j=1}^{t}\Pr\left(X_{B_{\mathcal{S}'_j}}=y'_j\right),\ \text{and} \tag{11}$$

$$\Pr\left(X'_{p_i}=x'_i\right)=\Pr\left(X_{B_{\mathcal{S}'_j}}=y'_j\right)=\frac{1}{q^m}\ \text{ for all }\ i\in[r],j\in[t].$$

Proof: See Appendix B. ∎

The above lemmas are useful in stating the first major result of this paper.

Theorem 1: The computation capacity of any sum-network constructed using the SUM-NET-CONS algorithm is at most 1.

Proof: Recall that $|\mathcal{F}| = q$ and under a valid (m, n) fractional network code, we have that $H(X'_{p_i}) = m \log_2 q$ bits and $H(\phi_i(X)) \leq n \log_2 q$ bits. We then have that $H(\{\phi_i(X)\}_1^r) \leq rn \log_2 q$ bits and

$$\begin{aligned} H(\{\phi_i(X)\}_1^r) &= H(\{\phi_i(X)\}_1^r, \{X'_{p_i}\}_1^r) - H(\{X'_{p_i}\}_1^r | \{\phi_i(X)\}_1^r) \\ &= H(\{X'_{p_i}\}_1^r) + H(\{\phi_i(X)\}_1^r | \{X'_{p_i}\}_1^r) - H(\{X'_{p_i}\}_1^r | \{\phi_i(X)\}_1^r) \\ &\overset{(a)}{=} rm \log_2 q + H(\{\phi_i(X)\}_1^r | \{X'_{p_i}\}_1^r) - \sum_{i=1}^{r} H\left(X'_{p_i} | \{X'_{p_j}\}_1^{i-1}, \{\phi_j(X)\}_1^r\right) \\ &\overset{(b)}{=} rm \log_2 q + H(\{\phi_i(X)\}_1^r | \{X'_{p_i}\}_1^r) \\ &\leq rn \log_2 q, \end{aligned}$$

where (a) is by Lemma 2 and the chain rule of entropy and (b) is due to Lemma 1. This implies that $m/n \leq 1$. ∎

Next we show that the upper bound on the computation capacity exhibits a strong dependence on the characteristic of the field (denoted $\mathrm{ch}(\mathcal{F})$) over which the computation takes place.

Theorem 2: Let A be a $(0,1)$-matrix of dimension $r \times c$ and suppose that we construct a sum-network corresponding to A using the SUM-NET-CONS algorithm. The matrix M_A is as defined in equation (10). If $\mathrm{rank}_{\mathcal{F}}(M_A) = r + c$, i.e., M_A has full rank over $\mathcal{F}$, the upper bound on computation capacity of the sum-network is $r/(r+c)$. Furthermore, $\mathrm{rank}_{\mathcal{F}}(M_A) = r+c$ if and only if $\det_{\mathcal{F}}(M_A) = \det_{\mathcal{F}}\left[(A^TA)_{\#} - A^TA\right] \neq 0$.

Proof: From Lemma 1, we have that $H\left(X'_{p_i} | \phi_i(X)\right) = 0,\ \forall i \in [r]$ and $H\left(X'_{B_j} | \{\phi_i(X) : A(i,j) = 1\}\right) = 0,\ \forall j \in [c]$. Hence, from the information transmitted over all the bottleneck edges, i.e., $\{\phi_i(X) : i \in [r]\}$, we can recover the value of

$$\boldsymbol{t} := \begin{bmatrix} X'^T_{p_1} & X'^T_{p_2} & \cdots & X'^T_{p_r} & X'^T_{B_1} & X'^T_{B_2} & \cdots & X'^T_{B_c} \end{bmatrix}^T = (M_A \otimes I_m)\, \boldsymbol{X}, \tag{12}$$

where the second equality is due to equation (9). We have that $\det(M_A \otimes I_m) = (\det(M_A))^m$. Furthermore,

$$M_A = \begin{bmatrix} I_r & A \\ A^T & (A^TA)_{\#} \end{bmatrix} = \begin{bmatrix} I_r & \mathbf{0} \\ A^T & I_c \end{bmatrix} \begin{bmatrix} I_r & \mathbf{0} \\ \mathbf{0} & (A^TA)_{\#} - A^TA \end{bmatrix} \begin{bmatrix} I_r & A \\ \mathbf{0} & I_c \end{bmatrix} \tag{13}$$

and hence $\det_{\mathcal{F}}(M_A) = \det_{\mathcal{F}}\left[(A^TA)_{\#} - A^TA\right]$. Note that the matrix product A^TA is over the elements of the finite field $\mathcal{F}$. If M_A has full rank then the value of each source message can be obtained by the operation $(M \otimes I_m)^{-1}\, \boldsymbol{t}$. Each source message can take any of q^m different values and the maximum number of different values that can be transmitted across any bottleneck edge is q^n. Since all $r + c$ different source messages are recovered with zero-error based on the information transmitted across the r bottleneck edges, we get

$$(q^m)^{r+c} \leq (q^n)^r \implies \frac{m}{n} \leq \frac{r}{r+c}.$$

Note that M_A is a $(0,1)$-matrix, and $\mathrm{rank}_{\mathcal{F}}\, M_A = r+c$ implies that $\mathrm{ch}(\mathcal{F}) \nmid \det_{\mathbb{Z}}(M_A)$, where $\det_{\mathbb{Z}}$ indicates the determinant of the matrix with its elements interpreted as 0 or 1 in $\mathbb{Z}$ as opposed to being elements of $\mathcal{F}$. ∎

Corollary 1: The computation capacity of a sum-network constructed using a $(0,1)$-matrix A of dimension $r \times c$ is at most r/x where $x := \mathrm{rank}_{\mathcal{F}}\, M_A$ for the matrix M_A as defined in equation (10).

Proof: Note that $r \leq x \leq (r + c)$ as the first r rows of M are necessarily linearly independent. We choose a submatrix $M_A[[r] \cup \mathcal{S}', [r+c]]$ of the matrix M_A where index set $\mathcal{S}'$ is as defined in Lemma 2. $M_A[[r] \cup \mathcal{S}', [r+c]]$ is a $x \times (r+c)$ matrix consisting of x linearly independent rows of M_A and the first r rows are the same as the first r rows of M_A, i.e., they are the matrix $[I_r\ A]$. We consider the following matrix equation, similar to equation (12), with M_A replaced by $M_A[[r] \cup \mathcal{S}', [r+c]]$.

$$\boldsymbol{t}_{\mathcal{S}'} := \begin{bmatrix} X'^T_{p_1} & \cdots & X'^T_{p_r} & X'^T_{B_{\mathcal{S}'_1}} & \cdots & X'^T_{B_{\mathcal{S}'_t}} \end{bmatrix}^T = (M_A[[r] \cup \mathcal{S}', [r+c]] \otimes I_m)\boldsymbol{X}.$$

By Lemma 2, all mx components of $\boldsymbol{t}_{\mathcal{S}'}$ are uniform over the elements of $\mathcal{F}$. The total number of symbols from $\mathcal{F}$ communicated over the r bottlenecks is rn. Hence by the zero-error criterion, we must have that $(q^n)^r \geq q^{mx} \implies m/n \leq r/x$. ∎

Example 4: Consider the normal sum-network obtained from using the Fano plane for which the incidence matrix $A_{\mathcal{I}}$ is as defined in equation (1), so that $r = c = 7$. It can be verified that $\mathrm{rank}_{GF(2)}\, M_{A_{\mathcal{I}}} = 7$, i.e., $M_{A_{\mathcal{I}}}$ is rank deficient over $GF(2)$. Hence the upper bound in Theorem 2 is not applicable to this sum-network, however Corollary 1 gives an upper bound of 1 for the computation capacity. In fact, there is a rate-1 network code that satisfies all terminals in the normal sum-network obtained using the Fano plane as described later in Proposition 3.

We can obtain a different upper bound on the computation capacity by considering submatrices of M_A that do not necessarily contain all the initial r rows. To do this we define an new index set $\mathcal{S}''$ with respect to any index set $\mathcal{S} \subseteq [r]$ as follows.

$$\mathcal{S}'' \subseteq \{r+1, r+2, \ldots, r+c\} \text{ such that } \forall i \in \mathcal{S}'', A^T[i-r, [r]] \in \mathrm{Span}\{I_r[j, [r]] : j \in \mathcal{S}\}. \tag{14}$$

Here Span indicates the vector space mapped out by linear combinations of vectors in a set. The submatrix of M_A that contains all the rows indexed by numbers in $\mathcal{S} \cup \mathcal{S}''$ is $M[\mathcal{S} \cup \mathcal{S}'', [r+c]]$.

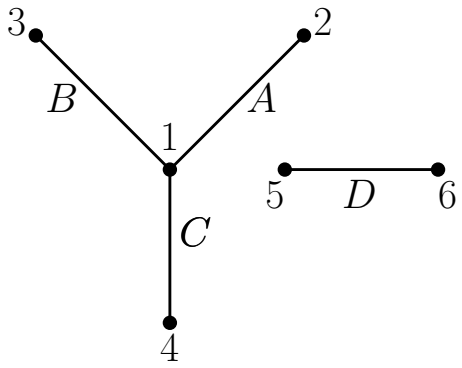


Fig. 4. A simple undirected graph G with two connected components. It has 6 vertices and 4 edges.

Theorem 3: Let A be a $(0,1)$-matrix of dimension $r \times c$ and suppose that we construct a sum-network corresponding to A using the SUM-NET-CONS algorithm. For any (m,n)-network code that enables all the terminals to compute the sum, we must have that

$$\frac{m}{n} \leq \min_{\mathcal{S} \subseteq [r]} \left\{ \frac{|\mathcal{S}|}{x_{\mathcal{S}}} \right\},$$

where $x_{\mathcal{S}} := \operatorname{rank}_{\mathcal{F}} M_A[\mathcal{S} \cup \mathcal{S}^{''}, [r+c]]$ and $\mathcal{S}^{''}$ is as defined in equation (14).

Proof: Note that for the choice $\mathcal{S} = [r]$, the index set $\mathcal{S}^{''}$ is the same as the index set $\mathcal{S}'$ defined in Lemma 2 and $x_{\mathcal{S}} = \operatorname{rank}_{\mathcal{F}} M_A$, thus recovering the $r/\operatorname{rank}_{\mathcal{F}} M_A$ upper bound on the computation capacity from Corollary 1. For $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{S}|}\} \subset [r]$, we can obtain an index set $\mathcal{T} \subseteq \mathcal{S}^{''}$ such that

$$x_{\mathcal{S}} = \operatorname{rank}_{\mathcal{F}} M_A[\mathcal{S} \cup \mathcal{S}^{''}, [r+c]] = \operatorname{rank}_{\mathcal{F}} M_A[\mathcal{S} \cup \mathcal{T}, [r+c]] = |\mathcal{S}| + |\mathcal{T}|.$$

We collect the blocks indexed in increasing order by $\mathcal{T}$ in the set $\mathcal{B}_{\mathcal{T}} = \{B_{\mathcal{T}_1}, \ldots, B_{\mathcal{T}_y}\} \subseteq \mathcal{B}$, where $y := |\mathcal{T}|$. The matrix equation similar to equation (12) for this case is

$$\boldsymbol{t}_{\mathcal{T}} := \begin{bmatrix} X'^{T}_{p_{\mathcal{S}_1}} & \cdots & X'^{T}_{p_{\mathcal{S}_{|\mathcal{S}|}}} & X'^{T}_{B_{\mathcal{T}_1}} & \cdots & X'^{T}_{B_{\mathcal{T}_y}} \end{bmatrix}^T = \left(\begin{bmatrix} M_A[\mathcal{S}, [r+c]] \\ M_A[\mathcal{T}, [r+c]] \end{bmatrix} \otimes I_m \right) \boldsymbol{X}.$$

Consider the set of terminals $T' := \{t_{p_i} : i \in \mathcal{S}\} \cup \{t_B : B \in \mathcal{B}_{\mathcal{T}}\}$. By the definition of $\mathcal{S}^{''}$, none of the terminals in T' are connected to any of the bottleneck edges $\{e_j : j \notin \mathcal{S}\}$. Moreover, by Lemma 1, we get that each element of the vector $\boldsymbol{t}_{\mathcal{T}}$ can be computed by some terminal in T' based on only the information received over the bottleneck edges. Specifically, $(M_A[i, [r+c]] \otimes I_m)\boldsymbol{X}$ for some $i \in \mathcal{S}$ can be recovered by terminal t_{p_i} and $(M_A[j, [r+c]] \otimes I_m)\boldsymbol{X}$ for some $j \in \mathcal{T}$ can be recovered by a terminal t_B where block B corresponds to the jth row of M_A. Following a procedure similar to the proof of Lemma 2, we get that all $mx_{\mathcal{S}}$ components of $\boldsymbol{t}_{\mathcal{T}}$ are uniform i.i.d. over $\mathcal{F}$. They are exactly recovered from the $n|\mathcal{S}|$ symbols transmitted over the bottleneck edges in the set $\{e_j : j \in \mathcal{S}\}$. Hence we must have that $q^{n|\mathcal{S}|} \geq q^{mx_{\mathcal{S}}} \implies m/n \leq |\mathcal{S}|/x_{\mathcal{S}}$. The same reasoning is valid for any choice of $\mathcal{S} \subseteq [r]$ and that gives us the result. ∎

Example 5: Consider the transposed sum-network corresponding to the undirected graph G shown in Figure 4. One can check that the matrix $M_{A_G^T}$ when the rows and columns of the incidence matrix A_G^T are arranged in increasing alphabetical and numeric order is as follows.

$$M_{A_G^T} = \left[\begin{array}{cccc|cccccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

We choose our finite field alphabet to be $GF(3)$ in this example. Then $\operatorname{rank}_{GF(3)} M_{A_G^T} = 5$ and Corollary 1 gives that the computation capacity is at most $4/5$. However, Theorem 3 gives a tighter upper bound in this case. Specifically, if $\mathcal{S} = \{1,2,3\}$ then $\mathcal{S}^{''} = \{5,6,7,8\}$ and $\operatorname{rank}_{GF(3)} M_{A_G^T}[\mathcal{S} \cup \mathcal{S}^{''}, [10]] = 4$. Hence Theorem 3 states that the computation capacity of the transposed sum-network for the graph G is at most $3/4$.

We apply the above theorems to obtain characteristic-dependent upper bounds on the computation capacity for some infinite families of sum-networks constructed using the given procedure.

Corollary 2: Let $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ be an incidence structure obtained from a simple undirected graph where $\mathcal{P}$ denotes the set of vertices and $\mathcal{B}$ consists of the 2-subsets of $\mathcal{P}$ corresponding to the edges. Let $\deg(p) \in \mathbb{Z}$ represent the degree of vertex $p \in \mathcal{P}$. The incidence matrix $A_{\mathcal{I}}$ has dimension $|\mathcal{P}| \times |\mathcal{B}|$. The computation capacity of the normal sum-network constructed using $A_{\mathcal{I}}$ is at most $\frac{|\mathcal{P}|}{|\mathcal{P}|+|\mathcal{B}|}$ for any finite field $\mathcal{F}$.

Let $\mathcal{F}$ be the finite field alphabet of operation and define $\mathcal{P}' \subseteq \mathcal{P}$ as

$$\mathcal{P}' := \{p : \mathrm{ch}(\mathcal{F}) \nmid (\deg(p) - 1), p \in \mathcal{P}\}.$$

Consider the set of edges $\mathcal{B}' := \{B : B \in \langle p \rangle \text{ for some } p \in \mathcal{P}', B \in \mathcal{B}\}$. The computation capacity of the transposed sum-network is at most $\frac{|\mathcal{B}'|}{|\mathcal{B}'|+|\mathcal{P}'|}$.

Proof: Recall that $\boldsymbol{B}_i^T$ is the ith row of $A_{\mathcal{I}}^T$ for all $i \in [|\mathcal{B}|]$. Then, the inner product over $\mathcal{F}$ between two rows is

$$\boldsymbol{B}_i^T \boldsymbol{B}_j = \begin{cases} 2 \mod \mathrm{ch}(\mathcal{F}) & \text{if } i = j \\ 1 & \text{if the edges indexed by } i \text{ and } j \text{ have a common vertex,} \\ 0 & \text{otherwise.} \end{cases}$$

It can be observed that the matrix of interest, i.e., $(A_{\mathcal{I}}^T A_{\mathcal{I}})_{\#} - A_{\mathcal{I}}^T A_{\mathcal{I}} = -I_{|\mathcal{B}|}$ has full rank over every finite field. The transposed sum-network for $\mathcal{I}$ is obtained by applying the SUM-NET-CONS algorithm on the $|\mathcal{B}| \times |\mathcal{P}|$ matrix $A_{\mathcal{I}}^T$, so that the parameters $r = |\mathcal{B}|, c = |\mathcal{P}|$. We apply Theorem 3 by choosing the index set $\mathcal{S} \subseteq [|\mathcal{B}|]$ such that $\mathcal{S} = \{j : B_j \in \mathcal{B}'\}$. Defined this way, $|\mathcal{S}| = |\mathcal{B}'|$ and $\mathcal{S}^{''}$ is obtained from $\mathcal{S}$ using equation (14). We collect all the points corresponding to the rows in the submatrix $M_{A_{\mathcal{I}}^T}[\mathcal{S}^{''}, [c+r]]$ in a set $\mathcal{P}_{\mathcal{S}^{''}} \subseteq \mathcal{P}$. Note that $\mathcal{P}_{\mathcal{S}^{''}}$ depends on the set of edges $\mathcal{B}'$. By definitions of $\mathcal{B}'$ and $\mathcal{S}^{''}$, we have that $\mathcal{P}' \subseteq \mathcal{P}_{\mathcal{S}^{''}}$. This is true because $\mathcal{B}'$ consists of all the edges that are incident to at least one point in $\mathcal{P}'$ while indices in the set $\mathcal{S}^{''}$ correspond to all points that are not incident to any edge outside $\mathcal{B}'$. For instance, in Example 5 above, as $\mathcal{F} = GF(3)$, $\mathcal{P}' = \{1\}$. Then $\mathcal{B}' = \{A, B, C\}$ and $\mathcal{P}_{\mathcal{S}^{''}} = \{1, 2, 3, 4\}$.

We now show that $\mathrm{rank}_{\mathcal{F}} M_A[\mathcal{S} \cup \mathcal{S}^{''}] = |\mathcal{B}'| + |\mathcal{P}'|$ and that gives us the result using Theorem 3. Recall that $\boldsymbol{p}_i$ denotes the ith row of $A_{\mathcal{I}}$, which corresponds to the vertex p_i for all $i \in [|\mathcal{P}|]$. It follows that the inner product between p_i, p_j over $\mathcal{F}$ is

$$\boldsymbol{p}_i \boldsymbol{p}_j^T = \begin{cases} \deg(p_i) \mod \mathrm{ch}(\mathcal{F}) & \text{if } i = j \\ 1 & \text{if } \{i, j\} \in \mathcal{B} \\ 0 & \text{otherwise.} \end{cases}$$

Because of the above equation, all the off-diagonal terms in the matrix $(A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#} - A_{\mathcal{I}} A_{\mathcal{I}}^T$ are equal to zero. We focus on the submatrix $M[\mathcal{S} \cup \mathcal{S}^{''}, [r+c]]$ obtained from equation (13), letting $\mathcal{S}^{''}_{|\mathcal{B}|} = \{j - |\mathcal{B}| : j \in \mathcal{S}^{''}\}$ we get that

$$M[\mathcal{S} \cup \mathcal{S}^{''}, [r+c]] = \begin{bmatrix} I_{|\mathcal{B}|}[\mathcal{S}, \mathcal{S}] & \mathbf{0} \\ A_{\mathcal{I}}[\mathcal{S}^{''}_{|\mathcal{B}|}, \mathcal{S}] & I_{|\mathcal{P}|}\left[\mathcal{S}^{''}_{|\mathcal{B}|}, \mathcal{S}^{''}_{|\mathcal{B}|}\right] \end{bmatrix} \begin{bmatrix} I_{|\mathcal{B}|}[\mathcal{S}, [|\mathcal{B}|]] & \mathbf{0} \\ \mathbf{0} & \left((A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#} - A_{\mathcal{I}} A_{\mathcal{I}}^T\right)\left[\mathcal{S}^{''}_{|\mathcal{B}|}, [|\mathcal{P}|]\right] \end{bmatrix} \begin{bmatrix} I_{|\mathcal{B}|} & A_{\mathcal{I}}^T \\ \mathbf{0} & I_{|\mathcal{P}|} \end{bmatrix}.$$

By definition of $\mathcal{P}'$ the points in the set $\mathcal{P}_{\mathcal{S}^{''}} \setminus \mathcal{P}'$ are such that $\deg(p_i) - 1 \equiv 0 \mod \mathrm{ch}(\mathcal{F})$, i.e., the diagonal entry corresponding to those points in $(A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#} - A_{\mathcal{I}} A_{\mathcal{I}}^T$ in the above equation is zero. Thus, the middle matrix has exactly $|\mathcal{B}'| + |\mathcal{P}'|$ rows which are not equal to the all-zero row vector. The first and third matrices are invertible, and hence we get that $\mathrm{rank}_{\mathcal{F}} M_A[\mathcal{S} \cup \mathcal{S}^{''}, [r+c]] = |\mathcal{B}'| + |\mathcal{P}'|$. ∎

Corollary 3: Let $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ be a 2-$(v, k, 1)$ design. For the normal sum-network constructed using the $|\mathcal{P}| \times |\mathcal{B}|$ incidence matrix $A_{\mathcal{I}}$, the computation capacity is at most $\frac{|\mathcal{P}|}{|\mathcal{P}|+|\mathcal{B}|}$ if $\mathrm{ch}(\mathcal{F}) \nmid (k-1)$. For the transposed sum-network constructed using $A_{\mathcal{I}}^T$, the computation capacity is at most $\frac{|\mathcal{B}|}{|\mathcal{P}|+|\mathcal{B}|}$ if $\mathrm{ch}(\mathcal{F}) \nmid \frac{v-k}{k-1}$.

Proof: We first describe the case of the transposed sum-network. From equation (2) each point in a 2-$(v, k, 1)$ design is incident to $\rho = \frac{v-1}{k-1}$ blocks. Moreover any two points occur together in exactly one block. Thus, we have the inner product over $\mathcal{F}$ as

$$\boldsymbol{p}_i \boldsymbol{p}_j^T = \begin{cases} \frac{v-1}{k-1} \mod \mathrm{ch}(\mathcal{F}), & \text{if } j = i \\ 1, & \text{otherwise.} \end{cases}$$

This implies that $A_{\mathcal{I}} A_{\mathcal{I}}^T - (A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#} = \left[\left(\frac{v-1}{k-1} - 1\right)\right] I_v = \left[\frac{v-k}{k-1}\right] I_v$ and setting this determinant as non-zero gives the result.

For the normal sum-network, we argue as follows. Note that $\boldsymbol{B}_i^T \boldsymbol{B}_i = k \mod \mathrm{ch}(\mathcal{F})$ for any i. Since any two points determine a unique block, two blocks can either have one point or none in common. Hence, for $i \neq j$, the inner product over $\mathcal{F}$ is

$$\boldsymbol{B}_i^T \boldsymbol{B}_j = \begin{cases} 1 & \text{if } B_i \cap B_j \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Then $A_{\mathcal{I}}^T A_{\mathcal{I}} - (A_{\mathcal{I}}^T A_{\mathcal{I}})_{\#} = [(k-1)]\, I_b$ and setting its determinant as non-zero gives the result. ∎

Corollary 4: Let $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ be a t-(v, k, λ) design, for $t \geq 2$. From equation (2), each point is present in $\rho := \lambda\binom{v-1}{t-1}/\binom{k-1}{t-1}$ blocks and the number of blocks incident to any pair of points is given by $b_2 := \lambda\binom{v-2}{t-2}/\binom{k-2}{t-2}$. Consider the transposed sum-network constructed using the incidence matrix $A_{\mathcal{I}}^T$ which has dimension $|\mathcal{B}| \times |\mathcal{P}|$. The computation capacity of the transposed sum-network is at most $\frac{|\mathcal{B}|}{|\mathcal{B}|+|\mathcal{P}|}$ if

$$\mathrm{ch}(\mathcal{F}) \nmid [\rho - b_2 + v(b_2-1)](\rho - b_2)^{v-1}.$$

Proof: By definition, we have that the inner product over $\mathcal{F}$ between two rows is

$$\boldsymbol{p}_i \boldsymbol{p}_j^T = \begin{cases} \rho \mod \mathrm{ch}(\mathcal{F}), & \text{if } j = i \\ b_2 \mod \mathrm{ch}(\mathcal{F}), & \text{otherwise.} \end{cases}$$

It follows that $A_{\mathcal{I}} A_{\mathcal{I}}^T - (A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#}$ has the value $(\rho - 1)$ on the diagonal and $(b_2 - 1)$ elsewhere. That is,

$$A_{\mathcal{I}} A_{\mathcal{I}}^T - (A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#} = [(\rho - b_2) \mod \mathrm{ch}(\mathcal{F})]\, I_v + [(b_2 - 1) \mod \mathrm{ch}(\mathcal{F})]\, J_v,$$

where J_v denotes the square all ones matrix of dimension v. Then we have that

$$\det\left[A_{\mathcal{I}} A_{\mathcal{I}}^T - (A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#}\right] = \det \begin{bmatrix} \rho-1 & b_2-1 & \dots & b_2-1 \\ b_2-1 & \rho-1 & b_2-1 & \dots \\ \vdots & & \ddots & \vdots \\ b_2-1 & \dots & b_2-1 & \rho-1 \end{bmatrix} \mod \mathrm{ch}(\mathcal{F}).$$

Subtracting from each row (except the last) its succeeding row and adding each column (except the first and last) to its preceding column, we get that

$$\det \begin{bmatrix} \rho-1 & b_2-1 & \dots & b_2-1 \\ b_2-1 & \rho-1 & b_2-1 & \dots \\ \vdots & & \ddots & \vdots \\ b_2-1 & \dots & b_2-1 & \rho-1 \end{bmatrix} = \det \begin{bmatrix} \rho-b_2 & b_2-\rho & 0 & \dots & 0 \\ 0 & \rho-b_2 & b_2-\rho & 0 & \dots \\ \vdots & \dots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \rho-b_2 & b_2-\rho \\ b_2-1 & \dots & b_2-1 & b_2-1 & \rho-1 \end{bmatrix},$$

$$= \det \begin{bmatrix} \rho-b_2 & 0 & 0 & \dots & 0 \\ 0 & \rho-b_2 & 0 & \dots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ 0 & \dots & 0 & \rho-b_2 & b_2-\rho \\ b_2-1 & 2(b_2-1) & \dots & (v-1)(b_2-1) & \rho-1 \end{bmatrix},$$

and the determinant of matrix can be evaluated to be equal to $[\rho - b_2 + v(b_2-1)](\rho-b_2)^{v-1} \mod \mathrm{ch}(\mathcal{F})$. ∎

Corollary 5: Let $\mathcal{D} = (\mathcal{P}, \mathcal{B})$ be a t-$(v, t+1, \lambda)$ design with $\lambda \neq 1$ and incidence matrix $A_{\mathcal{D}}$. We define a *higher* incidence matrix $A_{\mathcal{D}'}$ of dimension $\binom{|\mathcal{P}|}{t} \times |\mathcal{B}|$ such that each row corresponds to a distinct t-subset of $\mathcal{P}$ and each column corresponds to a block in $\mathcal{B}$. $A_{\mathcal{D}'}$ is a $(0,1)$-matrix such that for any $i \in \left[\binom{v}{t}\right], j \in [|\mathcal{B}|]$, its entry $A_{\mathcal{D}'}(i,j) = 1$ if each of the points in the t-subset corresponding to the ith row is incident to the block $B_j \in \mathcal{B}$ and zero otherwise. The computation capacity of the normal sum-network constructed using $A_{\mathcal{D}'}$ is at most $\frac{v}{v+|\mathcal{B}|}$ if $\mathrm{ch}(\mathcal{F}) \nmid t$. The computation capacity of the transposed sum-network constructed using $A_{\mathcal{D}'}^T$ is at most $\frac{|\mathcal{B}|}{|\mathcal{B}|+v} = \frac{\lambda}{\lambda+t+1}$ if $\mathrm{ch}(\mathcal{F}) \nmid (\lambda - 1)$.

Proof: The incidence matrix $A_{\mathcal{D}'}$ is a $(0,1)$ matrix of dimension $\binom{v}{t} \times \frac{\lambda}{t+1}\binom{v}{t}$. Let $\boldsymbol{p}_i, \boldsymbol{B}_u$ denote the ith row and uth column respectively of $A_{\mathcal{D}'}$ for $i \in \left[\binom{v}{t}\right], u \in \left[\frac{\lambda}{t+1}\binom{v}{t}\right]$. Each row of $A_{\mathcal{D}'}$ corresponds to a distinct t-subset of $\mathcal{P}$. By t-design criterion, any set of t points belongs to exactly λ blocks. Since the columns have a one-to-one correspondence with the blocks in $\mathcal{B}$, each row of $A_{\mathcal{D}'}$ has exactly λ 1's. Two rows will have a 1 in the same column if the block corresponding to the column is incident to both the t-subsets corresponding to the two rows. Since each block has $t+1$ points, there cannot be more than one block incident to two different t-subsets. Hence, for the inner product over $\mathcal{F}$, we have that $\boldsymbol{p}_i \boldsymbol{p}_i^T = \lambda \mod \mathrm{ch}(\mathcal{F})$ and for all $i \neq j; i, j, \in \left[\binom{v}{t}\right]$,

$$\boldsymbol{p}_i \boldsymbol{p}_j^T = \begin{cases} 1, & \text{if the union of the } t\text{-subsets corresponding to the } i\text{th and } j\text{th rows determines a unique block in } \mathcal{B} \\ 0, & \text{otherwise.} \end{cases}$$

Then $A_{\mathcal{D}'} A_{\mathcal{D}'}^T - (A_{\mathcal{D}'} A_{\mathcal{D}'}^T)_{\#} = [(\lambda - 1) \mod \mathrm{ch}(\mathcal{F})]\, I_{\binom{v}{t}}$ and that gives the result for the transposed sum-network.

For the normal sum-network, we look at the columns of $A_{\mathcal{D}'}$ in a similar manner. Each column of $A_{\mathcal{D}'}$ corresponds to a block in $\mathcal{B}$. Since the size of each block is $t+1$, each column has exactly $\binom{t+1}{t} = t+1$ elements as 1. Also, two different

blocks can have at most t points in common, and only when that happens, will the two columns have a 1 in the same row. Hence, for the inner product over $\mathcal{F}$, we have that $\boldsymbol{B}_u^T \boldsymbol{B}_u = (t+1) \mod \mathrm{ch}(\mathcal{F})$ and for all $u \neq v; u, v \in \left[\binom{v}{t}\right]$,

$$\boldsymbol{B}_u^T \boldsymbol{B}_v = \begin{cases} 1, & \text{if the } u\text{th and } v\text{th blocks have } t \text{ points in common} \\ 0, & \text{otherwise.} \end{cases}$$

Then we have that $A_{\mathcal{D}'}^T A_{\mathcal{D}'} - (A_{\mathcal{D}'}^T A_{\mathcal{D}'})_{\#} = t \mod \mathrm{ch}(\mathcal{F}) I_{\frac{\lambda}{t+1}\binom{v}{t}}$ and Theorem 2 gives the result. ■

VI. Linear network codes for constructed sum-networks

In this section, we propose linear network codes for the sum-networks constructed using the SUM-NET-CONS algorithm. Recall that the algorithm takes a $(0,1)$-matrix A that has r rows and c columns as its input. In Section V, we demonstrated that the incidence matrix of certain incidence structures result in sum-networks whose capacity can be upper bounded (*cf.* Corollaries 2, 3, 5). We now demonstrate that under certain conditions, we can obtain network codes whose rate matches the corresponding upper bound. Thus, we are able to characterize the capacity of a large family of sum-networks.

We emphasize that random linear network codes that have been used widely in the literature for multicast code constructions are not applicable in our context. In particular, it is not too hard to argue that a random linear network code would result in the each terminal obtaining a different linear function or subspace. Thus, constructing codes for these sum-networks requires newer ideas. We outline the key ideas by means of the following example.

Example 6: Consider the sum-network shown in Figure 2(a). The matrix $A_{\mathcal{I}}$ used in its construction is of dimension $r \times c$ where $r = 2, c = 1$ and is described in Example 2. It can be observed that $A_{\mathcal{I}}^T A_{\mathcal{I}} - \left(A_{\mathcal{I}}^T A_{\mathcal{I}}\right)_{\#} = 1$. Then Theorem 2 states that the computation capacity of this sum-network is at most $2/3$. We describe a network code with $m = 2, n = 3$. The global encoding functions for the two bottleneck edges are shown in Table I. Using the values transmitted, all three terminals can

TABLE I

The function values transmitted across e_1, e_2 in Figure 2(a) for a network code with rate $= 2/3$. Each message $X_1, X_2, X_{\{1,2\}}$ is a vector with 2 components, and $\phi_1(X), \phi_2(X)$ are vectors with 3 components each. A number within square brackets adjoining a vector indicates a particular component of the vector.

Component	$\phi_1(X)$	$\phi_2(X)$
1	$X_1[1] + X_{\{1,2\}}[1]$	$X_2[1] + X_{\{1,2\}}[1]$
2	$X_1[2] + X_{\{1,2\}}[2]$	$X_2[2] + X_{\{1,2\}}[2]$
3	$X_{\{1,2\}}[1]$	$X_{\{1,2\}}[2]$

recover the sum in the following manner. t_1 receives the value of X_2 from the direct edge (s_2, t_1) while t_2 receives the value of X_1 from the direct edge (s_1, t_2). Then t_1 recovers the sum using the first two components of $\phi_1(X)$ while t_2 recovers the sum using the first two components of $\phi_2(X)$. Additionally, $t_{\{1,2\}}$ receives both $\phi_1(X), \phi_2(X)$ and can carry out the operation $(X_1 + X_{\{1,2\}}) + (X_2 + X_{\{1,2\}}) - X_{\{1,2\}}$. Thus, each terminal is satisfied.

The network code in the example has the following structure. For each bottleneck edge, the first r components of the global encoding vector are the sum of all messages that are incident to that bottleneck. The remaining c components of the encoding vectors transmit selected components of messages observed at source nodes that correspond to columns in the matrix $A_{\mathcal{I}}$. In the example, $t_{\{1,2\}}$ received the first component of $X_{\{1,2\}}$ from $\phi_1(X)$ and the second component from $\phi_2(X)$. Thus it was able to recover the value of $X_{\{1,2\}}$, which it used in computing the demanded sum.

Our construction of network codes for sum-networks will have this structure, i.e., the first r components on a bottleneck edge will be used to transmit a *partial* sum of the messages observed at the sources that are connected to that bottleneck edge and the remaining c components will transmit portions of certain sources in an uncoded manner. For a given incidence matrix A , our first step is to identify (if possible) a corresponding non-negative integral matrix D of the same dimensions with the following properties.

- $D(i,j) = 0$ if $A(i,j) = 0$.
- Each row in D sums to r.
- Each column in D sums to c.

Under certain conditions on the incidence matrix A, we will show that D can be used to construct suitable network codes for the sum-networks under consideration.

The existence of our proposed network codes are thus intimately related to the existence of non-negative integral matrices that satisfy certain constraints. The following theorem [31, Corollary 1.4.2] is a special case of a more general theorem in [32] that gives the necessary and sufficient conditions for the existence of non-negative integral matrices with constraints on their row and column sums. We give the proof here since we use some ideas from the proof in the eventual network code assignment.

Theorem 4: Let $R = (r_1, r_2, \ldots, r_m)$ and $S = (s_1, s_2, \ldots, s_n)$ be non-negative integral vectors satisfying $r_1 + \ldots + r_m = s_1 + \ldots + s_n$. There exists an $m \times n$ nonnegative integral matrix D such that

$$\begin{aligned} &0 \leq D(i,j) \leq c_{ij}, \ \ \forall i \in [m], \forall j \in [n], \\ &\sum_{j=1}^{n} D(i,j) = r_i, \ \ \forall i \in [m], \text{ and} \\ &\sum_{i=1}^{m} D(i,j) = s_j, \ \ \forall j \in [n] \end{aligned}$$

if and only if for all $I \subseteq [m]$ and $J \subseteq [n]$, we have that

$$\sum_{i \in I} \sum_{j \in J} c_{ij} \geq \sum_{j \in J} s_j - \sum_{i \notin I} r_i. \tag{15}$$

Proof: Consider a capacity-limited flow-network modelled using a bipartite graph on $m+n$ nodes. The left part has m nodes denoted as $x_i, \forall i \in [m]$ and the right part has n nodes denoted as $y_j, \forall j \in [n]$. For all i, j there is a directed edge (x_i, y_j) with capacity c_{ij}. There are two additional nodes in the flow-network, the source node S and terminal node T. There are directed edges (S, x_i) with capacity r_i for all $i \in [m]$ and directed edges (y_j, T) with capacity s_j for all $j \in [n]$. Let x_I be the set of all nodes in the left part whose indices are in I and let $y_{\bar{J}}$ be the set of all nodes in the right path whose indices are *not* in J. Consider a cut separating nodes in $\{S\} \cup x_I \cup y_{\bar{J}}$ from its complement. Let f^* be the value of the maximum S-T flow in this network. Then we must have that for all possible choice of subsets $I \subseteq [m], J \subseteq [n]$,

$$\sum_{i \notin I} r_i + \sum_{(i,j): i \in I, j \in J} c_{ij} + \sum_{j \notin J} s_j \geq f^*. \tag{16}$$

In particular, suppose that $f^* = \sum_{j \in [n]} s_j$ in the flow-network. Substituting this in equation (16), we get the condition that for all possible subsets $I \subseteq [m], J \subseteq [n]$,

$$\sum_{i \in I} \sum_{j \in J} c_{ij} \geq \sum_{j \in J} s_j - \sum_{i \notin I} r_i. \tag{17}$$

Note that by choosing all possible subsets I, J, we are considering every possible S-T cut in the network. Then by the mincut-maxflow theorem, the set of conditions of the form of equation (17) for all I, J are not only necessary but also sufficient for the existence of a flow of value $f^* = \sum_{j \in [n]} s_j$ in the network.

A feasible flow with this value can be used to arrive at the matrix D as follows. We set the value of element $D(i,j)$ in the matrix to be equal to the value of the feasible flow on the edge (x_i, y_j) for all $i \in [m], j \in [n]$. It is easy to verify that the matrix D satisfies the required conditions. ∎

Using the existence theorem for nonnegative integral matrices, we can obtain network codes for sum-networks constructed from certain incidence structures. The following theorem describes a set of sufficient conditions that, if satisfied by an incidence structure, allow us to construct a linear network code that has the same rate as the computation capacity of that sum-network. The proof of the theorem is constructive and results in an explicit network code.

Theorem 5: Let $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ be an incidence structure and let $A_{\mathcal{I}}$ denote the corresponding incidence matrix of dimension $v \times b$. Suppose that the following conditions are satisfied.

- $A_{\mathcal{I}}^T A_{\mathcal{I}} - (A_{\mathcal{I}}^T A_{\mathcal{I}})_{\#} = \text{diag}(\mu_1, \mu_2, \ldots, \mu_b) \mod \text{ch}(\mathcal{F})$, where $\mu_i, \forall i$ is a non-zero element of $\mathcal{F}$.
- There exists a matrix $D_{\mathcal{I}}$ of the same dimension as $A_{\mathcal{I}}$ whose entries satisfy

$$D_{\mathcal{I}}(i,j) = 0, \text{ if } A_{\mathcal{I}}(i,j) = 0, \tag{18}$$

$$\sum_{i=1}^{v} D_{\mathcal{I}}(i,j) = v, \text{ and} \tag{19}$$

$$\sum_{j=1}^{b} D_{\mathcal{I}}(i,j) = b. \tag{20}$$

Then, the computation capacity of the sum-network constructed using $A_{\mathcal{I}}$ via the SUM-NET-CONS algorithm is $\frac{v}{v+b}$. This rate can be achieved by linear network codes.

Proof: Note that $A_{\mathcal{I}}^T A_{\mathcal{I}} - (A_{\mathcal{I}}^T A_{\mathcal{I}})_{\#}$ has full rank by assumption, Theorem 2 states that the computation capacity of the sum-network is at most $v/(v+b)$. We construct a (m,n) linear network code with $m = v, n = v + b$ using the matrix $D_{\mathcal{I}}$. Since $m = v$, each message vector has v components. For a vector $t \in \mathcal{F}^v$, the notation $t[l_1 : l_2]$ for two positive integers $l_1, l_2 \in [v]$ denotes a $(l_2 - l_1 + 1)$ length vector that contains the components of t with indices in the set $\{l_1, l_1 + 1, \ldots, l_2\}$ in order. We need to specify the global encoding vectors $\phi_i(X)$ only for the bottleneck edges $e_i, i \in [v]$ as all the other edges

in the network act as repeaters. The linear network code is such that the first v components of the vector transmitted along $e_i\ \forall i \in [v]$ is

$$\phi_i(X)[1:v] = X_{p_i} + \sum_{j:A_{\mathcal{I}}(i,j)=1} X_{B_j}.$$

By construction, each $t_{p_i} \forall i \in [v]$ is connected to the source nodes in $\{s_{p_{i'}} : i' \neq i\} \cup \{s_{B_j} : A_{\mathcal{I}}(i,j) = 0\}$ by direct edges. t_{p_i} can then compute the following value from the information received on the direct edges.

$$\sum_{i' \neq i} X_{p_i} + \sum_{j:A_{\mathcal{I}}(i,j)=0} X_{B_j}.$$

Adding the above value to $\phi_i(X)[1:v]$ enables t_{p_i} to compute the required sum. In what follows, we focus on terminals of the form $t_{B_j} \forall j \in [b]$.

Since $n = v + b$, each vector $\phi_i(X) \in \mathcal{F}^n$ has b components that haven't been specified yet. We describe a particular assignment for the b components on every $\phi_i(X) \forall i \in [v]$ using the matrix $D_{\mathcal{I}}$ that enables each $t_{B_j} \forall j \in [b]$ to compute the sum.

Recall the bipartite flow network constructed in the proof of Theorem 4 for demonstrating the existence of $D_{\mathcal{I}}$. The nodes in the left part are denoted as $p_i \forall i \in [v]$ and the nodes in the right part are denoted as $B_j \forall j \in [b]$. There is an edge (p_i, B_j) if and only if $A_{\mathcal{I}}(i,j) = 1$. The flow on the edge (p_i, B_j) is denoted as $f(p_i, B_j)$ and its value is determined by $D_{\mathcal{I}}(i,j)$, i.e., $f(p_i, B_j) := D_{\mathcal{I}}(i,j)$.

By constraints on the row and column sums of $D_{\mathcal{I}}$, we conclude that the value of the flow through any $p_i \forall i \in [v]$ is b and the value of the flow through any $B_j \forall j \in [b]$ is v. Without loss of generality, assume that $B_j = \{p_1, p_2, \cdots, p_{|B_j|}\}$. We can partition the v components of message vector X_{B_j} into $|B_j|$ parts such that the ith partition contains $f(p_i, B_j)$ distinct components of X_{B_j}. Such a partitioning can be done for all message vectors $X_{B_j}, j \in [b]$. Then the flow $f(p_i, B_j)$ indicates that the vector $\phi_i(X)$ transmits $f(p_i, B_j)$ components of X_{B_j} in an uncoded manner. Assigning such an interpretation to every edge in the flow-network is possible as the total number of components available in each $\phi_i(X)$ is b and that is also equal to the flow through the point p_i.

By construction, terminal t_{B_j} is connected to all bottleneck edges in the set $\{e_i : A_{\mathcal{I}}(i,j) = 1\}$. From the assignment based on the flow, t_{B_j} receives $f(p_i, B_j)$ distinct components of X_{B_j} from $\phi_i(X)$ for all $\{i : A_{\mathcal{I}}(i,j) = 1\}$. Since $\sum_{i=1}^{v} f(p_i, B_j) = v$, it can recover all v components of X_{B_j} in a piecewise fashion.

By adding the first v components transmitted on all the bottleneck edges that are connected to t_{B_j}, it can recover

$$\begin{aligned}\sum_{i:A_{\mathcal{I}}(i,j)=1} \phi_i(X)[1:v] &= \sum_{i:A_{\mathcal{I}}(i,j)=1} X_{p_i} + \sum_{i:A_{\mathcal{I}}(i,j)=1} \sum_{B_l \in \langle B_j \rangle} X_{B_l}, \\ &= \sum_{i:A_{\mathcal{I}}(i,j)=1} X_{p_i} + \sum_{B_l \in \langle B_j \rangle} \boldsymbol{B}_j^T \boldsymbol{B}_l X_{B_l}.\end{aligned}$$

Because of the condition that $A_{\mathcal{I}}^T A_{\mathcal{I}} - (A_{\mathcal{I}}^T A_{\mathcal{I}})_{\#} = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_b)$, one can verify that

$$\sum_{B_l \in \langle B_j \rangle} \boldsymbol{B}_j^T \boldsymbol{B}_l X_{B_j} = (\mu_j + 1) X_{B_j} + \sum_{B_l \in \langle B_j \rangle \setminus B_j} X_{B_l}.$$

By the flow-based assignment, each t_{B_j} obtains the value of X_{B_j} in a piecewise manner. It can then carry out the following operation

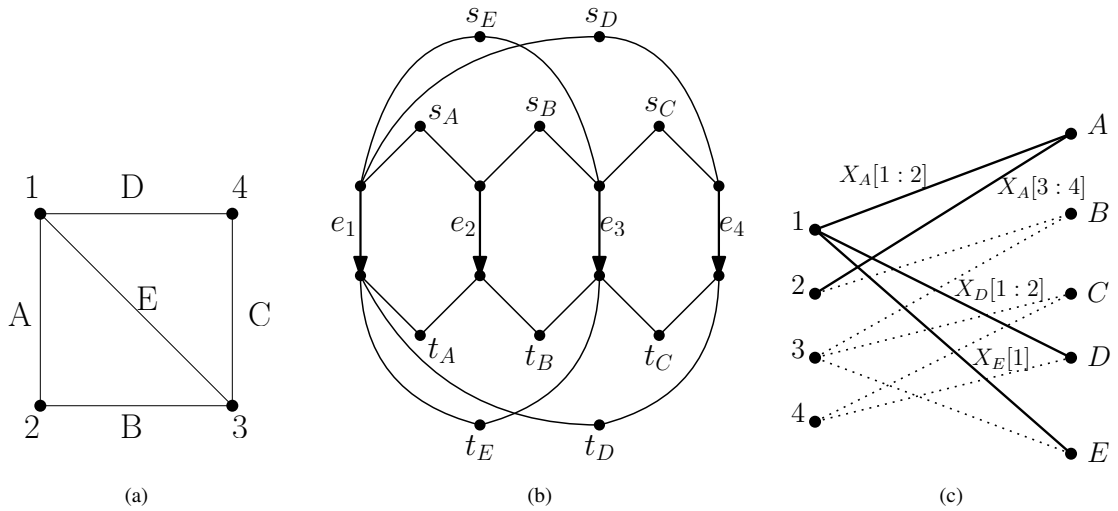
$$\begin{aligned}\sum_{i:A_{\mathcal{I}}(i,j)=1} \phi_i(X)[1:v] - \mu_j X_{B_j} &= \sum_{i:A_{\mathcal{I}}(i,j)=1} X_{p_i} + (\mu_j + 1) X_{B_j} + \sum_{B_l \in \langle B_j \rangle \setminus B_j} X_{B_l} - \mu_j X_{B_j}, \\ &= \sum_{p \in B_j} X_{B_j} + \sum_{B_l \in \langle B_j \rangle} X_{B_l}.\end{aligned}$$

The messages not present in this partial sum, i.e., $\{X_p : p \notin B_j\} \cup \{X_B : B \notin \langle B_j \rangle\}$ are available at t_{B_j} through direct edges by construction. Hence, terminals that correspond to a column of $A_{\mathcal{I}}$ are also able to compute the required sum. ∎

We illustrate the linear network code proposed above by means of the following example.

Example 7: Consider the normal sum-network obtained from the undirected simple graph G shown in Figure 5(a). A part of the sum-network is shown in Figure 5(b). The 4×5 incidence matrix A_G satisfies the condition of Theorem 4 and therefore has an associated matrix D_G with row-sum as 5 and column-sum 4 as shown below. The rows and columns of A_G are arranged in increasing numeric and alphabetical order.

$$A_G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad D_G = \begin{bmatrix} 2 & 0 & 0 & 2 & 1 \\ 2 & 3 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 3 \\ 0 & 0 & 3 & 2 & 0 \end{bmatrix}.$$



(a) (b) (c)

Fig. 5. (a) Undirected graph considered in Example 7. (b) Part of the corresponding normal sum-network constructed for the undirected graph in (a). The full normal sum-network has nine nodes each in the source set S and the terminal set T. However, for clarity, only the five sources and terminals that correspond to the columns of the incidence matrix of the graph are shown. Also, the direct edges constructed in Step 4 of the construction procedure are not shown. All edges are unit-capacity and point downward. The edges with the arrowheads are the bottleneck edges constructed in step 1 of the construction procedure. (c) Bipartite flow network as constructed in the proof of Theorem 4 for this sum-network. The message values corresponding to the flow on the solid lines are also shown.

TABLE II

THE FUNCTION VALUES TRANSMITTED ACROSS e_1, e_2, e_3, e_4 IN FIGURE 5(B) FOR A NETWORK CODE WITH RATE $= 4/9$. EACH MESSAGE X_A, X_B, X_C, X_D, X_E IS A VECTOR WITH 4 COMPONENTS, AND $\phi_1(X), \phi_2(X), \phi_3(X), \phi_4(X)$ ARE VECTORS WITH 9 COMPONENTS EACH. THE NUMBER INSIDE SQUARE BRACKETS ADJOINING A VECTOR INDICATES A PARTICULAR COMPONENT OF THE VECTOR.

Component	$\phi_1(X)$	$\phi_2(X)$	$\phi_3(X)$	$\phi_4(X)$
1 to 4	$X_1 + X_A + X_D + X_E$	$X_2 + X_A + X_B$	$X_3 + X_B + X_C + X_E$	$X_4 + X_C + X_D$
5	$X_A[1]$	$X_A[3]$	$X_B[4]$	$X_C[2]$
6	$X_A[2]$	$X_A[4]$	$X_C[1]$	$X_C[3]$
7	$X_D[1]$	$X_B[1]$	$X_E[2]$	$X_C[4]$
8	$X_D[2]$	$X_B[2]$	$X_E[3]$	$X_D[3]$
9	$X_E[1]$	$X_B[3]$	$X_E[4]$	$X_D[4]$

Using the matrix D_G, one can construct a structured linear network code with rate $= v/(v+b) = 4/9$ as shown in Table II. One can check that it enables all the terminals to compute the required sum. The flow-network corresponding to D_G is shown in Figure 5(c), and the messages corresponding to the flow on the solid edges is shown alongside the respective edge.

We can also consider the transposed sum-network for the same graph G. Corollary 2 gives an upper bound on the computation capacity that depends on $\mathcal{F}$. If $\mathcal{F} = GF(2)$, then the subset of points $\mathcal{P}' = \{2, 4\}$ and the upper bound is $4/6$. Note that Theorem 5 is not applicable here as the matrix $A_G^T A_G - (A_G^T A_G)_\#$ does not have all its diagonal elements as non-zero over $GF(2)$. Proposition 2 gives a condition for the existence of a network code for transposed sum-networks obtained using irregular graphs. We apply that condition to the transposed sum-network of the graph G considered here in Example 8.

In the following proposition we show that certain infinite families of incidence structures satisfy the requirements stated in Theorem 5. In particular, the incidence structures considered in Corollaries 2, 3 and 5 satisfy the conditions and hence the computation capacity of the associated sum-networks can be calculated.

Proposition 1: The following incidence structures and their transposes satisfy condition (ii) in Theorem 5, i.e., if their incidence matrix of dimension $v \times b$ is denoted by $A_{\mathcal{I}}$, there exists a corresponding non-negative integral matrix $D_{\mathcal{I}}$ that satisfies the conditions in equations (18) – (20).

1) Incidence structures derived from a regular graph or a biregular bipartite graph.
2) t-(v, k, λ) designs with $\lambda = 1$.
3) The higher incidence structure of a t-$(n, t+1, \lambda)$ design with $\lambda \neq 1$ obtained using the procedure described in Corollary 5.

Proof: The existence of $D_{\mathcal{I}}$ with row-sums as v and column-sums b is the same as the existence of $D_{\mathcal{I}}^T$ with row-sums as b and column-sums v. Thus, it suffices to argue for $D_{\mathcal{I}}$.

To check the validity of the condition we first choose the bounds on the elements of the matrix $D_{\mathcal{I}}$. We set $r_i = b$ and

$s_j = v$ for all $i \in [v], j \in [b]$ and

$$c_{ij} = \begin{cases} 0, & \text{if } A_{\mathcal{I}}(i,j) = 0, \\ \infty, & \text{if } A_{\mathcal{I}}(i,j) = 1. \end{cases}$$

By this choice the condition in inequality (15) is trivially satisfied whenever I, J are chosen such that there is a point in I which is incident to some block in J, i.e., there exist $i \in I, j \in J$ such that $A_{\mathcal{I}}(i,j) = 1$. Hence we restrict our attention to choices of I and J such that none of the points in I are incident to any block in J. Under this restriction, the L.H.S. of inequality (15) is 0 and the condition is equivalent to $(v - |I|)b \geq |J|v$. We will assume that

$$\exists I \subseteq [v], J \subseteq [b] \text{ such that } A_{\mathcal{I}}(i,j) = 0 \ \forall i \in I, j \in J, \text{ and } (v - |I|)b < |J|v, \tag{21}$$

and show that it leads to a contradiction for each of the three incidence structures considered.

If $\mathcal{I}$ corresponds to a d-regular simple graph, then $b = dv/2$. Consider the point-block incidence matrix $A_{\mathcal{I}}$, which is a $(0,1)$-matrix of size $v \times b$. For the chosen I in equation (21), we look at the submatrix $A_{\mathcal{I}}[I]$ of size $|I| \times b$ that consists of the rows of $A_{\mathcal{I}}$ indexed by the points in I and all the columns. Let l_1 be the number of columns with a single 1 in $A_{\mathcal{I}}[I]$ and l_2 be the number of columns with two 1s in $A_{\mathcal{I}}[I]$. By counting the total number of 1s in $A_{\mathcal{I}}[I]$ in two ways, we get that

$$d|I| = l_1 + 2l_2 \leq 2(l_1 + l_2) \implies l_1 + l_2 \geq \frac{d|I|}{2}.$$

Since the number of edges incident to at least one point in I is $l_1 + l_2$, any subset J of the edges that has no incidence with any point in I satisfies $|J| \leq b - d|I|/2$. Using these in equation (21) we get that

$$(v - |I|)b < |J|v \implies (v - |I|)\frac{dv}{2} < \left(\frac{dv}{2} - \frac{d|I|}{2}\right)v,$$

which is a contradiction.

Now suppose that $\mathcal{I}$ corresponds to a biregular bipartite graph, with L vertices having degree d_L in the left part and R vertices having degree d_R in the right part. Then $b = Ld_L = Rd_R$. Consider a subset $I_L \cup I_R$ of its vertices. Let E_L (resp. E_R) be the set of edges which are incident to some vertex in I_L (resp. I_R) but not incident to any vertex in I_R (resp. I_L). The number of edges that are not incident to any vertex in $I_L \cup I_R$ is equal to $(L - |I_L|)d_L - |E_R| = (R - |I_R|)d_R - |E_L|$. Suppose the choice of I in equation (21) is such that $I = I_L \cup I_R$ for some I_L, I_R. Then we have that

$$\begin{aligned}
&(v - |I|)b < |J|v, \\
&\implies (L + R - (|I_L| + |I_R|))\frac{Ld_L + Rd_R}{2} < \frac{(L - |I_L|)d_L - E_R + (R - |I_R|)d_R - |E_L|}{2}(L + R), \\
&\implies 1 - \frac{|I_L| + |I_R|}{L + R} < \frac{(L - |I_L|)d_L - |E_R| + (R - |I_R|)d_R - |E_L|}{Ld_L + Rd_R} = 1 - \frac{|I_l|d_L + |I_R|d_R + |E_L| + |E_R|}{Ld_L + Rd_R}, \\
&\implies (Ld_L + Rd_R)(|I_L| + |I_R|) - (L + R)(I_L d_L + I_R d_R) > (L + R)(|E_L| + |E_R|), \\
&\implies (L + R)(|E_L| + |E_R|) < (L - R)|I_L|d_L + (R - L)|I_R|d_R = (L - R)(|E_L| - |E_R|).
\end{aligned}$$

If $L > R$ or $|E_L| > |E_R|$, then we have a contradiction. That leaves the case when $L < R$ and $|E_L| < |E_R|$, which implies $(L + R)(|E_L| + |E_R|) < (R - L)(|E_R| - |E_L|)$ and that is also a contradiction.

Next, consider a BIBD on v points, b blocks such that repetition degree of each point is ρ and the number of points in each block is k, then we have that $bk = v\rho$. With the I of equation (21), we employ a similar procedure as for the case of the d-regular graph. We choose the submatrix $A_{\mathcal{I}}[I]$ of size $|I| \times b$ that corresponds to the rows indexed by the points in I and let $l_i, \forall i \in [k]$ denote the number of columns with exactly i 1s in $A_{\mathcal{I}}[I]$. We count the total number of 1s in $A_{\mathcal{I}}[I]$ in two ways, yielding

$$\rho|I| = l_1 + 2l_2 + \cdots + (k-1)l_{k-1} + kl_k \leq k\sum_{i=1}^{k} l_i \implies \sum_{i=1}^{k} l_i \geq \frac{\rho|I|}{k} = \frac{b|I|}{v}.$$

The number of blocks that are incident to at least one point in I is equal to $\sum_{i=1}^{k} l_i$. Hence any subset J of blocks that has no incidence with any point in I satisfies $|J| \leq b - |I|b/v$. Using this in equation (21) we get that

$$(v - |I|)b < |J|v \implies (v - |I|)b < \left(b - \frac{|I|b}{v}\right)v,$$

which is a contradiction.

If $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ is the higher incidence structure obtained from a t-$(n, t+1, \lambda)$ design as described in Corollary 5, then we have that $|\mathcal{P}| = \binom{n}{t}$ and $|\mathcal{B}| = \frac{\lambda}{t+1}\binom{n}{t}$. By definition of t for the original design, we have that each of the points in $\mathcal{P}$ are incident to exactly λ blocks. Also, each block in $\mathcal{B}$ consists of $\binom{t+1}{t} = t + 1$ points. For the submatrix $A_{\mathcal{I}}[I]$ whose rows

TABLE III

THE FUNCTION VALUES TRANSMITTED ACROSS THE BOTTLENECK EDGES OF THE TRANSPOSED SUM-NETWORK CORRESPONDING TO THE GRAPH SHOWN IN FIGURE 5(A) FOR A RATE-$4/6$ NETWORK OVER $GF(2)$. EACH MESSAGE X_2, X_4 IS A VECTOR WITH 4 COMPONENTS, AND $\phi_A(X), \phi_B(X), \phi_C(X), \phi_D(X), \phi_E(X)$ ARE VECTORS WITH 6 COMPONENTS EACH. THE NUMBER INSIDE SQUARE BRACKETS ADJOINING A VECTOR INDICATES A PARTICULAR COMPONENT OF THE VECTOR. A DASH INDICATES THAT THE VALUE TRANSMITTED ON THAT COMPONENT IS NOT USED IN DECODING BY ANY TERMINAL.

Component	$\phi_A(X)$	$\phi_B(X)$	$\phi_C(X)$	$\phi_D(X)$	$\phi_E(X)$
1 to 4	$X_1+X_2+X_A$	$X_2+X_3+X_B$	$X_3+X_4+X_C$	$X_1+X_4+X_D$	$X_1+X_3+X_E$
5	$X_2[1]$	$X_2[3]$	$X_4[1]$	$X_4[3]$	–
6	$X_2[2]$	$X_2[4]$	$X_4[2]$	$X_4[4]$	–

correspond to the points in I from Condition 21, we let $l_i, \forall i \in [t+1]$ denote the number of columns that have exactly i 1s in them. By counting the total number of 1s in $= A_{\mathcal{I}}[I]$ in two ways we get that

$$\lambda|I| = \sum_{i=1}^{t+1} i l_i \leq (t+1)\sum_{i=1}^{t+1} l_i \implies \sum_{i=1}^{t+1} l_i \geq \frac{\lambda|I|}{t+1}.$$

The total number of blocks incident to at least one point in I is $\sum_{i=1}^{t+1} l_i$. Then the number of blocks $|J|$ that are not incident to any point in I satisfy $|J| \leq |\mathcal{B}| - |I|\lambda/(t+1)$. Using these we get that

$$(v-|I|)b < |J|v \implies \left[\binom{n}{t} - |I|\right]\frac{\lambda}{t+1}\binom{n}{t} < \frac{\lambda}{t+1}\left[\binom{n}{t} - |I|\right]\binom{n}{t},$$

which is a contradiction. Thus in all the three kinds of incidence structures considered, we have shown that they admit the existence of the associated matrix $D_{\mathcal{I}}$ under the stated qualifying conditions. This enables us to apply Theorem 5 and obtain a lower bound on the computation capacity of these sum-networks. ∎

For an undirected graph $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ that is not regular, Proposition 1 is not applicable. Theorem 5 describes a sufficient condition for the existence of a linear network code that achieves the upper bound on the computation capacity of normal sum-networks constructed from undirected graphs that are not necessarily regular. The upper bound on the capacity of the transposed sum-network constructed using the incidence matrix $A_{\mathcal{I}}^T$ however can be different from $\frac{|\mathcal{B}|}{|\mathcal{B}|+|\mathcal{P}|}$ depending on the finite field $\mathcal{F}$ used for communication (*cf.* Corollary 2) and Theorem 5 needs to be modified to be applicable in that case. The following example illustrates this.

Example 8: Consider the transposed sum-network for the irregular graph G described in Example 7. Corollary 2 gives an upper bound of $4/6$ on the computation capacity when $\mathcal{F} = GF(2)$, as for that case $\mathcal{P}' = \{2, 4\}$ and $\mathcal{B}' = \{A, B, C, D\}$. We show the submatrix $A_G^T[\mathcal{B}', \mathcal{P}']$ in the equation below and also an associated matrix D_G whose support is the same as that of $A_G^T[\mathcal{B}', \mathcal{P}']$ and whose row-sum $= 6 - 4 = 2$ and column-sum $= 4$. The rows and columns are arranged in increasing alphabetical and numeric order.

$$A_G^T[\mathcal{B}', \mathcal{P}'] = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, D_G = \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 0 & 2 \\ 0 & 2 \end{bmatrix}.$$

Using D_G we can construct a rate-$4/6$ linear network code, shown in Table III, that achieves the computation capacity for $\mathcal{F} = GF(2)$ of the transposed sum-network constructed using the irregular graph G shown in Figure 5(a). In particular, terminals t_1, t_3 don't need any information other than the partial sums obtained over their respective bottleneck edges to compute the sum. Terminals t_2, t_4 need the value X_2, X_4 respectively, and that is transmitted in a piecewise fashion according to the matrix D_G over the bottleneck edges.

For an undirected graph $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ that is not regular, let $\mathcal{P}', \mathcal{B}'$ be the set of points and edges as chosen in the statement of Corollary 2. We describe a condition on the submatrix $A_{\mathcal{I}}^T[\mathcal{B}', \mathcal{P}']$ which consists of the rows and columns of $A_{\mathcal{I}}^T$ corresponding to the blocks and points in the sets $\mathcal{B}', \mathcal{P}'$ respectively. This condition, which is a restatement of Theorem 5 for this submatrix, allows us to construct a capacity-achieving linear network code for the transposed sum-network.

Proposition 2: For an undirected graph $\mathcal{I} = (\mathcal{P}, \mathcal{B})$, let $|\mathcal{P}'| = v', |\mathcal{B}'| = b'$, where $\mathcal{P}', \mathcal{B}'$ are subsets of points and blocks as defined in Corollary 2 and let $A_{\mathcal{I}}^T[\mathcal{B}', \mathcal{P}'](i,j)$ indicate an element of the submatrix for indices $i \in [b'], j \in [v']$. Suppose there exists a matrix $D_{\mathcal{I}}$ of dimension $b' \times v'$ such that

$$D_{\mathcal{I}}(i,j) = 0, \text{ if } A_{\mathcal{I}}^T[\mathcal{B}', \mathcal{P}'](i,j) = 0,$$

$$\sum_{i=1}^{b'} D_{\mathcal{I}}(i,j) = b', \text{ for all } j \in [v'], \text{ and}$$

$$\sum_{j=1}^{v'} D_{\mathcal{I}}(i,j) = v', \text{ for all } i \in [b'].$$

Then, there is linear network code of rate $\frac{b'}{b'+v'}$ that solves the transposed sum-network constructed using $\mathcal{I}$.

Proof: We describe a rate-$b'/(b'+v')$ network code that enables each terminal to compute the sum. Then by Corollary 2 we know that this is a capacity-achieving code. Since this is a transposed sum-network, the bottleneck edges in the sum-network correspond to the blocks in the undirected graph $\mathcal{I}$. The first b' components transmitted over each bottleneck is obtained by the following equation.

$$\phi_i(X)[1:b'] = X_{B_i} + \sum_{p \in B_i} X_p, \text{ for all } B_i \in \mathcal{B}.$$

We show that this partial sum satisfies all the terminals in the set $\{t_{B_i} : B_i \in \mathcal{B}\} \cup \{t_{p_j} : p_j \notin \mathcal{P}'\}$. Terminals in $\{t_{B_i} : B_i \in \mathcal{B}\}$ can recover the sum as all messages not present in the partial sum are available to t_{B_i} through direct edges. For terminals in the set $\{t_p : p \notin \mathcal{P}'\}$, it carries out the following operation as part of its decoding procedure.

$$\sum_{i:B_i \in \langle p \rangle} \phi_i(X)[1:b'] = \sum_{i:B_i \in \langle p \rangle} \left(X_{B_i} + \sum_{j:p_j \in B_i} X_{p_j} \right), \tag{22}$$

$$= \sum_{i:B_i \in \langle p \rangle} X_{B_i} + \sum_{j:\{p,p_j\} \in \mathcal{B}} \boldsymbol{p}\boldsymbol{p}_j^T X_{p_j} + \deg(p) X_p. \tag{23}$$

For $p_j \neq p$, we have that $\boldsymbol{p}\boldsymbol{p}_j^T = 1$ if $\{p, p_j\} \in \mathcal{B}$. Also by condition on the points that are not in $\mathcal{P}'$, we have that $\deg(p) \equiv 1 \mod \mathrm{ch}(\mathcal{F})$, and hence all the coefficients in the above partial sum are 1. The messages in the set $\{X_B : B \notin \langle p \rangle\} \cup \{X_{p_j} : \{p_j, p\} \notin \mathcal{B}\}$ are available to t_p through direct edges and hence it can recover the sum.

The remaining v' components available on the bottleneck edges $\{e_i : B_i \in \mathcal{B}'\}$ are used to transmit information that enables the terminals in the set $\{t_p : p \in \mathcal{P}'\}$ to compute the sum. Specifically, we construct a flow on a bipartite graph whose one part corresponds to the points in $\mathcal{P}'$ and the other part corresponds to the blocks in $\mathcal{B}'$, with incidence being determined by the submatrix $A_{\mathcal{I}}^T[\mathcal{B}', \mathcal{P}']$. Since there exists a matrix $D_{\mathcal{I}}$ with specified row and column sums, we can use it to construct a flow on the bipartite graph such that the messages in the set $\{X_{p_i} : p_i \in \mathcal{P}'\}$ are transmitted in a piecewise fashion over the bottleneck edges $\{e_j : B_j \in \mathcal{B}'\}$ in a manner similar to the proof of Theorem 5. Arguing in the same way, one can show that the network code based on the flow solution allows each t_p $\forall p \in \mathcal{P}'$ to obtain the value of X_p from the information transmitted over the bottleneck edges in the set $\{e_i : B_i \in \langle p \rangle\}$. Terminal t_p computes the sum in equation (22) as a part of its decoding procedure. Since $\deg(p) \not\equiv 1 \mod \mathrm{ch}(\mathcal{F})$, every term in the RHS of equation (23) except X_p has its coefficient as 1. But since t_p knows the value of X_p it can recover the relevant partial sum. The messages not present in this partial sum are available to t_p through direct edges and hence it can also compute the value of the sum. ■

Proposition 1 describes families of incidence structures for which the sum-networks constructed admit capacity-achieving linear network codes. The upper bound on the computation capacity of these sum-networks is obtained from Corollaries 2, 3 and 5. We now describe a rate-1 linear network code for the sum-networks when their corresponding incidence structures do not satisfy the qualifying conditions for the upper bounds. By Theorem 1, the computation capacity of any sum-network obtained using the SUM-NET-CONS algorithm is at most 1.

Proposition 3: For an incidence structure $\mathcal{I} = (\mathcal{P}, \mathcal{B})$ and a finite field $\mathcal{F}$ used for communication, there exists a rate-1 linear network code that satisfies the following sum-networks. If

- $\mathcal{I}$ is a 2-$(v, k, 1)$ design:
  - the normal sum-network with $\mathrm{ch}(\mathcal{F}) \mid k-1$,
  - the transpose sum-network with $\mathrm{ch}(\mathcal{F}) \mid \frac{v-k}{k-1}$,
- $\mathcal{I}$ is a t-$(v, t+1, \lambda)$ design:
  - the normal sum-network obtained using the higher incidence matrix with $\mathrm{ch}(\mathcal{F}) \mid t$,
  - the transpose sum-network obtained using the higher incidence matrix with $\mathrm{ch}(\mathcal{F}) \mid \lambda - 1$.

Proof: Suppose we construct a sum-network using the SUM-NET-CONS algorithm on a $(0,1)$-matrix A of dimension $r \times c$. Then if $A^T A = (A^T A)_{\#}$, then the following rate-1 linear network code satisfies every terminal in the sum-network.

$$\phi_i(X) = X_{p_i} + \sum_{j:B_j \in \langle p_i \rangle} X_{B_j}, \ \forall\ i \in [r].$$

A terminal t_{p_i}, $\forall i \in [r]$ receives all the messages not present in the partial sum transmitted along e_i through direct edges, and hence it can compute the sum. A terminal t_B, $\forall B \in \mathcal{B}$ can carry out the following operation.

$$\sum_{i:p_i \in B_j} \phi_i(X) = \sum_{p_i \in B} X_{p_i} + \sum_{p_i \in B} \sum_{B_j \in \langle p_i \rangle} X_{B_j} = \sum_{p_i \in B} X_{p_i} + \sum_{l:B_l \in \langle B_j \rangle} \boldsymbol{B}_l^T \boldsymbol{B}_j X_{B_l}.$$

Since $A^T A = (A^T A)_{\#}$, all the coefficients in the above sum are 1 and $\sum_{i:p_i \in B_j} \phi_i(X)$ is equal to the sum of all the messages in the set $\{X_{p_i} : p_i \in B_j\} \cup \{X_B : B \in \langle B_j \rangle\}$. All the messages that are not present in this set are available to t_{B_j} through direct edges.

Such a rate-1 linear network code gives us our proposition in the following manner. Let $A_{\mathcal{I}}$ be the $v \times \frac{v-1}{k-1}$ incidence matrix for a 2-$(v,k,1)$ design and let $A'_{\mathcal{I}}$ be the higher incidence matrix as defined in Corollary 3 for a t-$(v,t+1,\lambda)$ design with $\lambda \neq 1$. Then we know that (from the proof of the Corollaries 3, 5)

$$\begin{aligned} A_{\mathcal{I}}^T A_{\mathcal{I}} - (A_{\mathcal{I}}^T A_{\mathcal{I}})_{\#} &= (k-1)I, \\ A_{\mathcal{I}} A_{\mathcal{I}}^T - (A_{\mathcal{I}} A_{\mathcal{I}}^T)_{\#} &= \frac{v-k}{k-1} I, \\ A_{\mathcal{I}}^{'T} A'_{\mathcal{I}} - (A_{\mathcal{I}}^{'T} A'_{\mathcal{I}})_{\#} &= tI, \\ A'_{\mathcal{I}} A_{\mathcal{I}}^{'T} - (A'_{\mathcal{I}} A_{\mathcal{I}}^{'T})_{\#} &= (\lambda-1)I. \end{aligned}$$

Thus whenever any of the above matrices is a zero matrix, we have a scalar linear network code that achieves the computation capacity of the associated sum-network. ∎

VII. Discussion and comparison with prior work

The discussion in Sections V and VI establishes the computation capacity for sum-networks derived from several classes of incidence structures. We now discuss the broader implications of these results by appealing to existence results for these incidence structures. BIBDs have been the subject of much investigation in the literature on combinatorial designs. In particular, the following two theorems are well-known.

Theorem 6: [22, Theorem 6.17] There exists a $(v,3,1)$-BIBD (also known as a Steiner triple system) if and only if $v \equiv 1,3 \pmod 6; v \geq 7$.

Theorem 7: [22, Theorem 7.31] There exists a $(v,4,1)$-BIBD if and only if $v \equiv 1,4 \pmod{12}; v \geq 13$.

In particular, these results show that there are an infinite family of Steiner triple systems and BIBDs with block size 4 and $\lambda = 1$. Since $k = 3$ for any Steiner triple system, we can demonstrate the existence of sum-networks whose computation capacity is greatly affected by the choice of the finite field $\mathcal{F}$ used for communication.

Proposition 4: Consider the normal sum-network constructed using a 2-$(v,3,1)$ design. If $\mathrm{ch}(\mathcal{F}) = 2$, then the computation capacity of the sum-network is 1. For odd $\mathrm{ch}(\mathcal{F})$, the computation capacity is $\frac{6}{5+v}$. For the normal sum-network constructed using a $(v,4,1)$-BIBD, the computation capacity is 1 if $\mathrm{ch}(\mathcal{F}) = 3$ and $\frac{12}{11+v}$ otherwise.

Proof: The number of blocks in a 2-$(v,3,1)$ design is equal to $v(v-1)/6$. From Corollary 3, if $\mathrm{ch}(\mathcal{F})$ is odd, then the computation capacity of the sum-network constructed using a Steiner triple system is at most $\frac{v}{v+v(v-1)/6} = \frac{6}{5+v}$. Moreover by Proposition 1, we can construct a linear network code with rate equal to the upper bound. On the other hand, if $\mathrm{ch}(\mathcal{F}) = 2$, then the computation capacity of the same sum-network is 1 by Proposition 3.

The number of blocks in a 2-$(v,4,1)$ design is $v(v-1)/12$. We can recover the result for the computation capacity of a normal sum-network constructed using it in a manner similar to the previous case. ∎

Thus, this result shows that for the *same* network, computing the sum over even characteristic has capacity 1, while the capacity goes to zero as $O(1/v)$ for odd characteristic. Moreover, this dichotomy is not unique to the prime number 2. Similar results hold for sum-networks derived from higher incidence structures (*cf.* Corollary 5).

Theorem 8: [33] For two integers t, v such that $v \geq t+1 > 0$ and $v \equiv t \pmod{(t+1)!^{2t+1}}$, a t-$(v, t+1, (t+1)!^{2t+1})$ design with no repeated blocks exists.

The number of blocks in a t-$(v, t+1, (t+1)!^{2t+1})$ design can be evaluated to be $\binom{v}{t}\frac{(t+1)!^{2t+1}}{t+1}$. We consider the normal sum-network obtained using the higher incidence matrix of this t-design. If $\mathrm{ch}(\mathcal{F}) \nmid t$, then by Corollary 5 and Proposition 1, we have that the computation capacity of this sum-network is

$$\frac{\binom{v}{t}}{\binom{v}{t} + \binom{v}{t}\frac{(t+1)!^{2t+1}}{t+1}} = \frac{1}{1 + t!^2 (t+1)!^{2t-1}}.$$

On the other hand, if $\mathrm{ch}(\mathcal{F})$ is a divisor of t, then by Theorem 1 and Proposition 3 we have that the computation capacity of the normal sum-network constructed using the higher incidence matrix is 1. Thus for the same sum-network, computing the sum over a field whose characteristic divides the parameter t can be done at rate $= 1$. However, if the field characteristic does not divide t, zero-error computation of the sum can only be done at a rate which goes to zero as $O\left(\left(\frac{t}{e}\right)^{-t^2}\right)$.

Theorem 6 describes an infinite family of BIBDs with $k = 3$ and $\lambda = 1$. There are further existence results for BIBDs with $\lambda = 1$ and $k \neq 3$. In particular, for $\lambda = 1, k \leq 9$ there exist BIBDs with value of v as given in Table 3.3 in [34, Section II.3.1]. As an example, if $k = 5$, then there exists a 2-$(v,5,1)$ design whenever $v \equiv 1,5 \mod 20$. For any choice of a BIBD from this infinite family, we can construct a corresponding normal sum-network, whose computation capacity for a particular finite field can be found using Corollary 3 and Proposition 1. Even though Theorem 5 states the existence of t-designs for v, t that satisfy the qualifying conditions, explicit constructions of such t-designs with $t \geq 6$ are very rare.

For a transposed sum-network obtained from an undirected graph that is not regular, the computation capacity can show a more involved dependence on the finite field alphabet as the following example demonstrates.

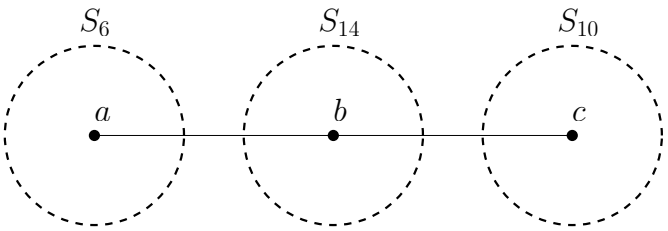


Fig. 6. The schematic shown represents an undirected graph with three components: S_6, S_{14} and S_{10}. S_t denotes the star graph on $t+1$ vertices, with only one vertex having degree t while the rest have degree 1. The vertices with the maximum degree in the three star graphs are a, b, c respectively. In addition, a is connected to b and b is connected to c, such that $\deg(a) = 7, \deg(b) = 16, \deg(c) = 11$.

Example 9: Consider the transposed sum-network obtained by applying the SUM-NET-CONS algorithm on the undirected graph $\mathcal{I}$ shown in Figure 6. Corollary 2 gives us an upper bound on the computation capacity of the transposed sum-network based on the finite field alphabet $\mathcal{F}$. The upper bound for three different choices of $\mathcal{F}$ is as follows.

- $\mathcal{F} = GF(2)$: Then $\mathcal{P}' = \{b\}$, so the upper bound is $16/(16+1) = 16/17$.
- $\mathcal{F} = GF(3)$: Then $\mathcal{P}' = \{c\}$, so the upper bound is $11/(11+1) = 11/12$.
- $\mathcal{F} = GF(5)$: Then $\mathcal{P}' = \{a\}$, so the upper bound is $7/(7+1) = 7/8$.

We use Proposition 2 to check if we can construct a linear network code in each case that has the same rate as the respective upper bound. To do that, we focus on the appropriate submatrix of $A_{\mathcal{I}}$ for each case and see if it satisfies the required condition on row and column sums. The rows of $A_{\mathcal{I}}$ corresponding to the vertices a, b, c (in order) are shown below.

$$\begin{bmatrix} \mathbf{1}_6 & 1 & 0 & \cdots & 0 \\ \mathbf{0}_6 & 1 & \mathbf{1}_{14} & 1 & \mathbf{0}_{10} \\ 0 & \cdots & 0 & 1 & \mathbf{1}_{10} \end{bmatrix},$$

where $\mathbf{1}, \mathbf{0}$ indicate all-one and all-zero row vectors of size specified by their subscripts. Using this, one can verify that the appropriate submatrix for each of the three choices of $\mathcal{F}$ satisfies the conditions of Proposition 2 and hence we can construct a capacity-achieving linear network code in each case.

Thus, as the previous example demonstrates, the computation capacity of a particular sum-network need not take just one of two possible values, and can have a range of different values based on the finite field chosen. We can generalize the example to obtain sum-networks that have arbitrary different possible values for their computation capacity. The above results that demonstrate the strong dependence of the computation capacity on the underlying alphabet are somewhat surprising in light of the known results about multiple unicast. In particular, for multiple unicast, it is well known [8, Section VI] that the capacity of a network is independent of the underlying alphabet, though the linear network coding capacity may depend on the alphabet. In Appendix C we show that the reduction presented in [8] (that demonstrates that coding capacity of a network is independent of the alphabet) cannot be applied to sum-networks by means of a simple example. Thus, function computation over networks exhibits a structurally different behavior.

Our constructed sum-networks have a unit maximum flow between any source and any terminal. It is not too hard to modify our construction so that each edge in the network has a capacity of $\alpha > 1$. Specifically, the following result can be shown.

Proposition 5: Let $\mathcal{N}$ denote the sum-network obtained by applying the SUM-NET-CONS algorithm on a matrix A of dimension $r \times c$. For an integer $\alpha > 1$, let $\mathcal{N}_\alpha$ denote the sum-network obtained by modifying the SUM-NET-CONS algorithm such that $\mathcal{N}_\alpha$ has the same structure as $\mathcal{N}$ but each edge e_α in $\mathcal{N}_\alpha$ has $\text{cap}(e_\alpha) = \alpha > 1$. Then if A satisfies the qualifying conditions in Theorems 2 and 5, then the computation capacity of $\mathcal{N}_\alpha$ is $\frac{\alpha r}{r+c}$.

Proof: Since A satisfies the conditions in Theorem 5, there exists a (m, n) vector linear network code with $m = r, n = r+c$. For every unit-capacity edge in $\mathcal{N}$, we have α unit-capacity edges between the same tail and head in $\mathcal{N}_\alpha$. At the tail of every edge in $\mathcal{N}_\alpha$, we can apply the same network code except now we have α distinct edges on which we can transmit the encoded value. Thus we need transmit only $\frac{r+c}{\alpha}$ symbols on each of those edges. If $\frac{r+c}{\alpha}$ is not an integer, one can appropriately multiply both m, n with a constant. This modified network code has rate $= \frac{\alpha r}{r+c}$. Since A also satisfies the conditions in Theorem 2, we have that an upper bound on the computation capacity of $\mathcal{N}$ is $r/(r+c)$. Applying the same argument on $\mathcal{N}_\alpha$, we get that an upper bound on the computation capacity of $\mathcal{N}_\alpha$ is $\frac{\alpha r}{r+c}$. This matches the rate of the modified vector linear network code described above. ∎

This result can be interpreted as follows. Consider the class of sum-networks where the maximum flow between any source-terminal pair is at least α. Our results indicate, that for any α, we can always demonstrate the existence of a sum-network, where the computation capacity is strictly smaller than 1. Once again, this indicates the crucial role of the network topology in function computation.

A. Comparison with prior work

The work of Rai and Das [27] is closest in spirit to our work. In [27], the authors gave a construction procedure to obtain a sum-network with computation capacity equal to p/q, where p, q are any two co-prime natural numbers. The procedure involved first constructing a sum-network whose capacity was $1/q$. Each edge in this sum-network had unit-capacity. By inflating the capacity of each edge in the sum-network to $p > 1$, the modified sum-network was shown to have computation capacity as p/q.

Our work is a significant generalization of their work. In particular, their sum-network with capacity $1/q$ can be obtained by applying the SUM-NET-CONS algorithm to the incidence matrix of a complete graph on $2q - 1$ vertices [28]. Moreover, we provide a systematic procedure for constructing these sum-networks for much larger classes of incidence structures.

In [27], the authors also posed the question if *smaller* sum-networks (with lesser sources and terminals) with capacity as p/q existed. Using the procedure described in this paper, we can answer that question in the affirmative.

Example 10: The normal sum-network for the undirected graph in Figure 5(a) has computation capacity $= 4/9$ and has nine sources and terminals. To obtain a sum-network with the same computation capacity using the method described in [27] would involve constructing the normal sum-network for a complete graph on 17 vertices, and such a sum-network would have 153 source nodes and terminal nodes each.

In [20], it was shown by a counter-example that for the class of sum-networks with $|S| = |T| = 3$, a maximum flow of 1 between each source-terminal pair was not enough to guarantee solvability (i.e., no network code of rate 1 exists for the counterexample). It can be observed that their counter-example is the sum-network shown in Figure 2(a). Our characterization of computation capacity for a family of sum-networks provides significantly more general impossibility results in a similar vein. In particular, note that for the α-capacity edge version of a sum-network, the maximum flow between any source-terminal pair is at least α. Then suppose we consider the class of sum-networks with $|S| = |T| = x = \beta(\beta + 1)/2$ for some $\beta \in \mathbb{N}$. Consider a complete graph $K_t = (V, E)$ on β vertices; then $|V| + |E| = x$. Consider the sum-network obtained by applying the procedure on K_β, with each edge added having capacity as α. Then the computation capacity of this sum-network is $\alpha\beta/x$, which is less than 1 if $\alpha < (\beta + 1)/2$. This implies that a max-flow of $(\beta + 1)/2$ between each source-terminal pair is a necessary condition for ensuring all sum-networks with $|S| = |T| = x$ are solvable. When x cannot be written as $\beta(\beta + 1)/2$ for some β, a similar argument can be made by finding an undirected graph $G = (V, E)$ (whose incidence matrix A_G satisfies the condition in Theorem 5) such that $|V|$ is minimal and $|V| + |E| = x$.

VIII. Conclusions and Future Work

Sum-networks are a large class of communication problems over directed acyclic networks. The notion of computation capacity is central in function computation problems, and various counterexamples and problem instances have been used by different authors to obtain a better understanding of solvability and computation capacity of general networks. We provide an algorithm to systematically construct sum-network instances using combinatorial objects called incidence structures. We propose novel upper bounds on the computation capacity in most cases, matching achievable schemes that leverage results on the existence of non-negative integer matrices with prescribed row and column sums. We demonstrate that the dependence of computation capacity on the underlying field characteristic can be rather strong.

There are several opportunities for future work. Our proposed linear network codes for the constructed sum-networks require the correspondence incidence structures to have a specific property. In particular, our techniques only work in the case when $A^T A - (A^T A)_\#$ is a diagonal matrix. It would be interesting to find capacity achieving network codes in cases when $A^T A - (A^T A)_\#$ is not diagonal. More generally, it would be interesting to obtain achievability schemes and upper bounds for sum-networks with more general topologies.

References

[1] J. Korner and K. Marton, "How to encode the modulo-2 sum of binary sources," *IEEE Trans. on Info. Th.*, vol. 25, no. 2, pp. 219–221, 1979.

[2] A. Orlitsky and J. Roche, "Coding for computing," *IEEE Trans. on Info. Th.*, vol. 47, no. 3, pp. 903–917, Mar 2001.

[3] V. Doshi, D. Shah, M. Medard, and M. Effros, "Functional compression through graph coloring," *IEEE Trans. on Info. Th.*, vol. 56, no. 8, pp. 3901–3917, Aug 2010.

[4] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. on Info. Th.*, vol. 46(4), pp. 1204–1216, 2000.

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

[6] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Info. Th.*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[7] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. on Info. Th.*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[8] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity," *IEEE Trans. on Info. Th.*, vol. 52, no. 3, pp. 777–788, Mar. 2006.

[9] R. Dougherty, C. Freiling, and K. Zeger, "Unachievability of network coding capacity," *IEEE Trans. on Info. Th.*, vol. 52, no. 6, pp. 2365–2372, June 2006.

[10] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. on Info. Th.*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[11] S. Huang and A. Ramamoorthy, "An achievable region for the double unicast problem based on a minimum cut analysis," *IEEE Trans. on Comm.*, vol. 61(7), pp. 2890–2899, 2013.

[12] ——, "On the multiple unicast capacity of 3-source, 3-terminal directed acyclic networks," *IEEE/ACM Trans. on Networking*, vol. 22(1), pp. 285–299, 2014.

[13] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '04, 2004, pp. 142–150.
[14] S. Kamath, D. N. C. Tse, and C. C. Wang, "Two-unicast is hard," in *IEEE Intl. Symposium on Info. Th.*, June 2014, pp. 2147–2151.
[15] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Network coding for computing: Cut-set bounds," *IEEE Trans. on Info. Th.*, vol. 57, no. 2, pp. 1015–1030, Feb 2011.
[16] C. Huang, Z. Tan, and S. Yang, "Upper bound on function computation in directed acyclic networks," in *IEEE Info. Th. Workshop*, April 2015, pp. 1–5.
[17] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Linear codes, target function classes, and network computing capacity," *IEEE Trans. on Info. Th.*, vol. 59, no. 9, pp. 5741–5753, Sept 2013.
[18] A. Ramamoorthy, "Communicating the sum of sources over a network," in *IEEE Intl. Symposium on Info. Th.*, 2008, pp. 1646–1650.
[19] B. K. Rai and B. K. Dey, "On network coding for sum-networks," *IEEE Trans. on Info. Th.*, vol. 58, no. 1, pp. 50 –63, 2012.
[20] A. Ramamoorthy and M. Langberg, "Communicating the sum of sources over a network," *IEEE J. Select. Areas Comm.*, vol. 31(4), pp. 655–665, 2013.
[21] R. Dougherty, C. Freiling, and K. Zeger, "Linear network codes and systems of polynomial equations," *IEEE Trans. Inf. Theor.*, vol. 54, no. 5, pp. 2303–2316, May 2008.
[22] D. R. Stinson, *Combinatorial Designs: Construction and Analysis*. Springer, 2003.
[23] O. Olmez and A. Ramamoorthy, "Fractional repetition codes with flexible repair from combinatorial designs," *IEEE Trans. on Info. Th.*, vol. 62, no. 4, pp. 1565 –1591, 2016.
[24] S. E. Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *48th Annual Allerton Conference on Communication, Control, and Computing*, 2010, pp. 1510 –1517.
[25] L. Tang and A. Ramamoorthy, "Coded Caching with Low Subpacketization Levels," in *Intl. Symp. on Network Coding (NetCod)*, 2016 (to appear).
[26] B. K. Rai and N. Das, "On the capacity of ms/3t and 3s/nt sum-networks," in *IEEE Info. Th. Workshop*. IEEE, 2013, pp. 1–5.
[27] B. Rai and N. Das, "On the capacity of sum-networks," in *Annual Allerton Conference on Comm., Control & Computing*, Oct 2013, pp. 1545–1552.
[28] A. Tripathy and A. Ramamoorthy, "Sum-networks from undirected graphs: Construction and capacity analysis," in *Annual Allerton Conference on Comm., Control & Computing*, Sept 2014, pp. 651–658.
[29] N. Das and B. K. Rai, "On the number of sources and terminals of sum-networks with capacity p/q," in *Communications (NCC), 2015 Twenty First National Conference on*, Feb 2015, pp. 1–6.
[30] A. Tripathy and A. Ramamoorthy, "Capacity of sum-networks for different message alphabets," in *IEEE Intl. Symposium on Info. Th.*, June 2015, pp. 606–610.
[31] R. A. Brualdi, *Combinatorial matrix classes*. Cambridge University Press, 2006, vol. 13.
[32] L. Mirsky, "Combinatorial theorems and integral matrices," *Journal of Combinatorial Theory*, vol. 5, no. 1, pp. 30–44, 1968.
[33] L. Teirlinck, "Non-trivial t-designs without repeated blocks exist for all t," *Discrete Mathematics*, vol. 65, no. 3, pp. 301–311, 1987.
[34] C. J. Colbourn and J. H. Dinitz, *Handbook of combinatorial designs*. CRC press, 2006.

APPENDIX A
NUMBER OF EDGES ADDED BY ALGORITHM 1

We use the notation defined in equations 3 and 4. Let $\mathcal{I}$ denote an incidence structure on v points and b blocks. The algorithm in Figure 1 take as input the incidence matrix $A_{\mathcal{I}}$ for constructing the normal sum-network. If the input is $A_{\mathcal{I}}^T$ then the sum-network returned is the transpose sum-network.

For the normal sum-network,

- Line 2 adds v edges,
- Line 4 adds $v+b$ source nodes and line 5 adds $v+b$ terminal nodes,
- Lines 9 and 8 together add $\sum_{p\in\mathcal{P}} 2(1+|\langle p\rangle|)$ edges and
- Lines 12 and 15 together add

$$\sum_{p\in\mathcal{P}}(v-1+b-|\langle p\rangle|)+\sum_{B\in\mathcal{B}}(v-|B|+b-|\langle B\rangle|)=(v+b)^2-v-\sum_{p\in\mathcal{P}}|\langle p\rangle|-\sum_{B\in\mathcal{B}}(|B|+|\langle B\rangle|)$$

edges.

For the transpose sum-network,

- Line 2 adds b edges,
- Line 4 adds $v+b$ source nodes and line 5 adds $v+b$ terminal nodes,
- Lines 9 and 8 together add $\sum_{B\in\mathcal{B}} 2(1+|B|)$ edges and
- Lines 12 and 15 together add

$$\begin{aligned}&\sum_{B\in\mathcal{B}}(b-1+v-|\cup_{B'\in\langle B\rangle}B'|)+\sum_{p\in\mathcal{P}}\left(b-|\langle p\rangle|+v-|\cup_{B\in\langle p\rangle}B|\right)\\&\quad=(v+b)^2-b-\sum_{B\in\mathcal{B}}|\cup_{B'\in\langle B\rangle}B'|-\sum_{p\in\mathcal{P}}(|\langle p\rangle|+|\cup_{B\in\langle p\rangle}B|)\end{aligned}$$

edges.

APPENDIX B
PROOF OF LEMMA 2

Proof: Let lowercase x_p, x_B denote realizations of the random variables X_p, X_B respectively and let $\mathcal{X}$ be the set of solutions to the following system of linear equations:

$$(M\,[[r]\cup\mathcal{S}',[r+c]]\otimes I_m)\begin{bmatrix}x_{p_1}^T & \cdots & x_{p_r}^T & x_{B_1}^T & \cdots & x_{B_c}^T\end{bmatrix}^T=\begin{bmatrix}x_1'^T & \cdots & x_r'^T & y_1'^T & \cdots y_t'^T\end{bmatrix}^T, \tag{24}$$

where the RHS $\begin{bmatrix} x_1'^T & \cdots & x_r'^T & y_1'^T & \cdots y_t'^T \end{bmatrix}^T$ is assumed to be fixed and the unknowns are $\begin{bmatrix} x_{p_1}^T & \cdots & x_{p_r}^T & x_{B_1}^T & \cdots & x_{B_c}^T \end{bmatrix}^T$. Because of the condition that $M_A\left[[r] \cup \mathcal{S}', [r+c]\right]$ has full row rank, choosing any value from $\mathcal{F}^m$ for each of the variables in $\{x_{B_j} : B_j \notin \mathcal{B}_{\mathcal{S}'}\}$ fixes the value of $x_{p_i}, x_{B_{\mathcal{S}_j'}}$ for all $i \in [r], j \in [t]$. Since, all random variables in $\{X_{p_i} : i \in [v]\} \cup \{X_{B_j} : j \in [c]\}$ are uniform i.i.d. over $\mathcal{F}^m$ with $|\mathcal{F}| = q$ the probability in the LHS of equation (11) is equal to

$$\sum_{\mathcal{X}} \prod_{i=1}^{r} \Pr(X_{p_i} = x_{p_i}) \prod_{j=1}^{c} \Pr(X_{B_j} = x_{B_j}) = |\mathcal{X}| \left[\frac{1}{q^m}\right]^r \left[\frac{1}{q^m}\right]^c = (q^m)^{c-t} \left[\frac{1}{q^m}\right]^r \left[\frac{1}{q^m}\right]^c = \left[\frac{1}{q^m}\right]^{r+t}.$$

Let $\mathcal{X}'$ be the set of solutions to the variables $\{x_{p_i}, \{x_{B_j} : A(i,j) = 1\}\}$ such that $x_i' = x_{p_i} + \sum_{j:A(i,j)=1} x_{B_j}$. Then

$$\Pr(X_{p_i}' = x_i') = \sum_{\mathcal{X}'} \Pr(X_{p_i} = x_{p_i}) \prod_{j:A(i,j)=1} \Pr(X_{B_j} = x_{B_j}) = (q^m)^{|\langle p_i \rangle|} \frac{1}{q^m} \left[\frac{1}{q^m}\right]^{|\langle p_i \rangle|} = \frac{1}{q^m}. \tag{25}$$

Let $\mathcal{X}''$ be the set of solutions to the variables $\{x_{p_i} : p_i \in B_j\} \cup \{x_B : B \in \langle B_j \rangle\}$ such that $y_j' = \sum_{i:A(i,j)=1} x_{p_i} + \sum_{B \in \langle B_j \rangle} x_B$. Then we have that

$$\begin{aligned} \Pr(X_{B_j}' = y_j') &= \sum_{\mathcal{X}''} \prod_{i:A(i,j)=1} \Pr(X_{p_i} = x_{p_i}) \prod_{B \in \langle B_j \rangle} \Pr(X_B = x_B) \\ &= (q^m)^{|B_j| + |\langle B_j \rangle| - 1} \prod_{i:A(i,j)=1} \frac{1}{q^m} \prod_{B \in \langle B_j \rangle} \frac{1}{q^m} = \frac{1}{q^m}, \end{aligned}$$

and we get that the RHS of equation (11) is

$$\prod_{i=1}^{r} \Pr\left(X_{p_i}' = x_i'\right) \prod_{j=1}^{t} \Pr\left(X_{B_{\mathcal{S}_j'}} = y_j'\right) = \left[\frac{1}{q^m}\right]^{r+t}.$$

■

APPENDIX C
REMARK ABOUT NON-APPLICABILITY OF THEOREM VI.5 IN [8] FOR SUM-NETWORKS

The capacity of multiple-unicast networks is known to be independent of the alphabet chosen for communication [8, Theorem VI.5]. The core idea there was this. Consider alphabets $\mathcal{F}_1, \mathcal{F}_2$ of different cardinality. Suppose there exists a (m_1, n_1) network code over $\mathcal{F}_1$ that satisfies the demands of every terminal in the network. Then for any $\epsilon > 0$, [8] described a procedure to simulate a (m_2, n_2) network code over $\mathcal{F}_2$ using the (m_1, n_1) network code over $\mathcal{F}_1$ such that $m_2/n_2 \geq m_1/n_1 - \epsilon$. The values of the parameters m_1, n_1, m_2, n_2 are determined by the value of ϵ and $|\mathcal{F}_1|, |\mathcal{F}_2|$. The simulation procedure uses two one-to-one functions $\boldsymbol{h_0} : \mathcal{F}_2^{m_2} \to \mathcal{F}_1^{m_1}$ and $\boldsymbol{h} : \mathcal{F}_1^{n_1} \to \mathcal{F}_2^{n_2}$. We informally describe the simulation procedure as applied to every component in a multiple-unicast network below.

- At each source node with no incoming edges: A message in $\mathcal{F}_2^{m_2}$ is observed at a source node. This message is mapped to a symbol in $\mathcal{F}_1^{m_1}$ using the function $\boldsymbol{h_0}$. This symbol is used as an argument for the encoding function of this node in the (m_1, n_1) network code; the value returned belongs to the set $\mathcal{F}_1^{n_1}$. The value returned by the encoding function is then mapped by $\boldsymbol{h}$ to an element in $\mathcal{F}_2^{n_2}$, which is transmitted along the outgoing edge.
- At each intermediate node in the network: Each intermediate node observes as many values from $\mathcal{F}_2^{n_2}$ as the number of its incoming edges. Since $\boldsymbol{h}$ is a one-to-one function, for each received symbol in $\mathcal{F}_2^{n_2}$ the node can obtain its pre-image under $\boldsymbol{h}$ in $\mathcal{F}_1^{n_1}$. After obtaining the pre-images for each received value, the node can use them as arguments for its encoding function in the (m_1, n_1) network code and obtain the values that must be transmitted along its outgoing edges. These returned values are in $\mathcal{F}_1^{n_1}$ and they are mapped to symbols in $\mathcal{F}_2^{n_2}$ by $\boldsymbol{h}$ before transmission.
- Decoding at each terminal node in the network: At each terminal node, the received values in $\mathcal{F}_2^{n_2}$ are mapped to their pre-images in $\mathcal{F}_1^{n_1}$ under $\boldsymbol{h}$. These pre-images are used as arguments for the decoding function of this terminal in the (m_1, n_1) network code. The value returned by the decoding function is an element of $\mathcal{F}_1^{m_1}$ that is the image under $\boldsymbol{h_0}$ of the demanded message at this terminal. Since $\boldsymbol{h_0}$ is also a one-to-one function, each terminal can recover its required message.

This simulation procedure however cannot be applied to sum-networks as is illustrated by the example below.

Example 11: Consider a simple sum-network shown in Figure 7, terminal t wants to evaluate $X_1 + X_2$ where $X_1, X_2 \in \mathcal{F}_1$ are random variables observed at source nodes s_1, s_2 respectively. We have a scalar network code (rate $= 1$) that satisfies the problem, described as follows.

1) Edge functions:

$$\phi_{e_1}(X_1) = X_1, \quad \phi_{e_2}(X_2) = X_2.$$

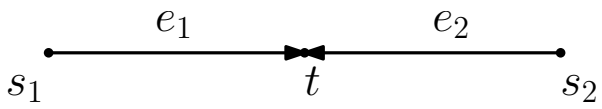


Fig. 7. A simple sum-network. Both edges can transmit one symbol in $\mathcal{F}_1$ from tail to head in one channel use.

2) Decoding function:

$$\psi(\phi_{e_1}(X_1), \phi_{e_2}(X_2)) = \phi_{e_1}(X_1) + \phi_{e_2}(X_2) = X_1 + X_2$$

and $X_1 + X_2$ is the only value terminal t is interested in decoding.

We use the procedure outlined in [8] to extend the network code for another alphabet $\mathcal{F}_2$. Let $\mathcal{F}_1 = GF(3), \mathcal{F}_2 = GF(2)$. Setting $\epsilon = 2^{1-\gamma}/\log_2 3$ where $\gamma > 1$, we obtain the following values

$$m_1 = n_1 = \lceil 2^\gamma \rceil, n_2 = \left\lceil \frac{2^\gamma}{\log_2 3} \right\rceil \text{ and } m_2 = n_2 - 1.$$

Let $h_0 : \mathcal{F}_2 \to \mathcal{F}_1$ be such that

$$h_0(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x = 1. \end{cases}$$

and let $\hat{h}_0 : \mathcal{F}_1 \to \mathcal{F}_2$ be such that $\hat{h}_0(h_0(x)) = x$ for all $x \in \mathcal{F}_2$ and arbitrary otherwise. Then we can define an injection $\boldsymbol{h_0} : \mathcal{F}_2^{m_2} \to \mathcal{F}_1^{m_1}$ as the component-wise application of h_0 to each of the elements in the argument.

$$\boldsymbol{h_0}(b_1, b_2, \ldots, b_{m_2}) = \begin{bmatrix} h_0(b_1) & h_0(b_2) & \ldots & h_0(b_{m_2}) & \mathbf{0}_{m_1 - m_2} \end{bmatrix}$$

where $b_1, b_2, \ldots b_{m_2} \in \mathcal{F}_2$ and $\mathbf{0}_{m_1 - m_2}$ is a zero vector with $m_1 - m_2$ components. We define $\hat{\boldsymbol{h}}_\mathbf{0} : \mathcal{F}_1^{m_1} \to \mathcal{F}_2^{m_2}$ as

$$\hat{\boldsymbol{h}}_\mathbf{0}(a_1, a_2, \ldots, a_{m_1}) = \begin{bmatrix} \hat{h}_0(a_1) & \hat{h}_0(a_2) & \ldots & \hat{h}_0(a_{m_2}) \end{bmatrix}$$

where $a_1, a_2, \ldots, a_{m_1} \in \mathcal{F}_1$.

Also we let $\boldsymbol{h} : \mathcal{F}_1^{n_1} \to \mathcal{F}_2^{n_2}$ be an arbitrary injection and $\hat{\boldsymbol{h}} : \mathcal{F}_2^{n_2} \to \mathcal{F}_1^{n_1}$ is such that $\hat{\boldsymbol{h}}(\boldsymbol{h}(\boldsymbol{x})) = \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathcal{F}_1^{n_1}$ and arbitrary otherwise. This is possible because $3^{\lceil 2^\gamma \rceil} \geq 2^{\lceil 2^\gamma / \log_2 3 \rceil}$ for any $\gamma > 1$. We now use the extended network code to satisfy the sum network for when the source random variables take values in the alphabet $\mathcal{F}_2^{m_2}$. Suppose a particular realization of $X_1 \in \mathcal{F}_2^{m_2}$ and $X_2 \in \mathcal{F}_2^{m_2}$ is such that

$$\boldsymbol{x_1} = (1, 1, \ldots, 1) = \mathbf{1}_{m_2} \text{ and } \boldsymbol{x_2} = (1, 1, \ldots, 1) = \mathbf{1}_{m_2}.$$

Following steps in [8] for the decoding function we get that terminal t carries out the following operation to obtain the value of $\boldsymbol{x_1} + \boldsymbol{x_2}$

$$\begin{aligned} \hat{\boldsymbol{h}}_\mathbf{0}(\psi_t(\phi_{e_1}(\boldsymbol{h_0}(\boldsymbol{x_1})), \phi_{e_2}(\boldsymbol{h_0}(\boldsymbol{x_2})))) &= \hat{\boldsymbol{h}}_\mathbf{0}(\boldsymbol{h_0}(\boldsymbol{x_1}) + \boldsymbol{h_0}(\boldsymbol{x_2})) \\ &= \hat{\boldsymbol{h}}_\mathbf{0}([\mathbf{1}_{m_2}\ \mathbf{0}_{m_1 - m_2}] + [\mathbf{1}_{m_2}\ \mathbf{0}_{m_1 - m_2}]) \\ &= \hat{\boldsymbol{h}}_\mathbf{0}([\mathbf{2}_{m_2}\ \mathbf{0}_{m_1 - m_2}]) \end{aligned}$$

where $\mathbf{2}_{m_2}$ is a vector of m_2 2's.

Since $\hat{h}_0(2)$ is arbitrarily assigned, $\hat{\boldsymbol{h}}_\mathbf{0}([\mathbf{2}_{m_2}\ \mathbf{0}_{m_1 - m_2}])$ need not equal $\mathbf{0}_{m_2}$ which is the right value of $\boldsymbol{x_1} + \boldsymbol{x_2}$. Thus the simulated (m_2, n_2) network code over $\mathcal{F}_2$ does not correctly evaluate the required sum.

In fact, the computation capacity of sum-networks explicitly depends on the alphabet used in message generation and communication, as described in Sections V, VI. Moreover, the choice of alphabet can significantly reduce the computation capacity of the same sum-network as discussed in Section VII.