Network Combination Operations Preserving the Sufficiency of Linear Network Codes

Congduan Li, Steven Weber, John MacLaren Walsh
ECE Department, Drexel University
Philadelphia, PA 19104

Abstract—**Operations that combine smaller networks into a larger network in a manner such that the rate region of the larger network can be directly obtained from the associated rate regions of the smaller networks are defined and presented. The operations are selected to also have the property that the sufficiency of classes of linear network codes and the tightness of the Shannon (polymatroid) outer bound are preserved in the combination. Four such operations are defined, and the classes of linear codes considered include both scalar and vector codes. It is demonstrated that these operations enable one to obtain rate regions for networks of arbitrary size, and to also determine if some classes of linear codes are sufficient and/or the Shannon outer bound bound is tight for these networks.**

I. Introduction

Many important practical problems, including efficient information transfer over networks, the design of efficient distributed information storage systems, and the design of streaming media systems, have been shown to involve determining the capacity region of an abstracted network under network coding. Yan *et al.*'s celebrated paper [1], together with bounds on the entropy region, provides a method, in principle, to calculate inner and outer bounds on the capacity region of networks under network coding. In addition to determining the capacity region when an inner and outer bound is shown to match, other interesting characteristics of the network can be determined through bound comparison, including the sufficiency of classes of linear codes to obtain the entire rate region, and the ability of Shannon-type inequalities to determine the rate region. Although multiple methods can be utilized to directly calculate these bounds through polyhedral projection in this manner [2]–[4], the sheer complexity of the problem, which involves first expressing polyhedra with very large inequality and extremal descriptions, then the subsequent elimination of an exponential number of variables in the projection step, renders it computationally infeasible beyond networks with 10s of edges. Hence, while these computational techniques can render large libraries of difficult to determine and previously unknown capacity regions of small networks [5], other techniques must be developed to tackle networks of larger or arbitrary scale.

Inspired by this observation, in [6], we presented some embedding operations so that the rate region of an embedded network can be obtained directly from the rate region of the larger network. In addition, it was shown that if a smaller network, for which $\mathbb{F}_q$ linear codes do not suffice, is embedded in a larger network (under the embedding operations defined), then $\mathbb{F}_q$ linear codes will not suffice for the larger network, either. In this sense, the embedding operations of [6] provides a method of certifying the insufficiency of linear codes for a network of arbitrary scale without ever having to calculate their rate regions. Furthermore, following an idealogical program inspired by the well-quasi ordering theory of the graph minors project, these embedding operations enables the creation a list of all smallest forbidden embedding networks [5] so that if a larger network has one of these forbidden networks embedded in, this class of codes will not suffice for the larger network.

However, for all networks with arbitrary size N, we can only certify insufficiency of code classes to exhaust the entire rate region with the embedding operations presented in [6]. There is no tool yet to certify sufficiency of classes of codes for networks with arbitrary size, or to determine their associated capacity regions. This paper will define four combination operations so that one can obtain the rate region of the larger network from the associated rate regions of the smaller networks involved in the combination. In addition to the rate region relations, if the smaller networks are known to be $\mathbb{F}_q$ codes sufficient or to be determined by the Shannon outer bound, then the same will be true for the larger network. These combination operations enable one to harness the database of rate regions calculated for small networks through the projection based computational tools to determine the capacity regions of networks of arbitrary scale. Additionally, they enable one to investigate properties of a network with arbitrary size by decomposing (reverse of combination) it into smaller networks and then checking the associated properties of the smaller networks. A journal submission which includes both the embedding and combination operations, together with discussions in network enumeration and rate region computation, is available at [7].

II. Background

This paper studies the multi-source multi-sink network coding problems with hyperedges. A network coding problem in this class, denoted by the symbol A, includes a directed acyclic hypergraph $(\mathcal{V}, \mathcal{E})$ [8] as in Fig. 1, consisting of a set of nodes $\mathcal{V}$ and a set $\mathcal{E}$ of directed hyperedges in the form of ordered pairs $e = (v, \mathcal{A})$ with $v \in \mathcal{V}$ and $\mathcal{A} \subseteq \mathcal{V} \setminus v$. The nodes $\mathcal{V}$ in the graph are partitioned into the set of source nodes $\mathcal{S}$, intermediate nodes $\mathcal{G}$, and sink nodes $\mathcal{T}$, i.e., $\mathcal{V} = \mathcal{S} \cup \mathcal{G} \cup \mathcal{T}$. Each of the source nodes $s \in \mathcal{S}$ will have a single outgoing edge $(s, \mathcal{A}) \in \mathcal{E}$. The source nodes in $\mathcal{S}$ have no incoming edges, the sink nodes $\mathcal{T}$ have no outgoing edges, and the intermediate nodes $\mathcal{G}$ have both incoming and outgoing edges.

The number of sources will be denoted by $|\mathcal{S}| = K$, and each source node $s \in \mathcal{S}$ will be associated with an independent random variable Y_s, $s \in \mathcal{S}$, with entropy $H(Y_s)$, and an associated independent and identically distributed (IID) temporal sequence of random values. For every source $s \in \mathcal{S}$, define $\mathrm{Out}(s)$ to be its single outgoing edge, which is connected to a subset of intermediate nodes and sink nodes. A hyperedge $e \in \mathcal{E}$ connects a source, or an intermediate node to a subset of non-source nodes, i.e., $e = (i, \mathcal{F})$, where $i \in \mathcal{S} \cup \mathcal{G}$ and $\mathcal{F} \subseteq (\mathcal{G} \cup \mathcal{T} \setminus i)$. For brevity, we will refer to hyperedges as edges if there is no confusion. For an intermediate node $g \in \mathcal{G}$, we denote its incoming edges as $\mathrm{In}(g)$ and outgoing edges as $\mathrm{Out}(g)$. For each edge $e = (i, \mathcal{F})$, the associated random variable $U_e = f_e(\mathrm{In}(i))$ is a function of all the inputs of node i, obeying the edge capacity constraint $R_e \geq H(U_e)$. The tail (head) node of edge e is denoted as $\mathrm{Tl}(e)$ ($\mathrm{Hd}(e)$). For notational simplicity, the unique outgoing edge of each source node will be the source random variable, $U_e = Y_s$ if $\mathrm{Tl}(e) = s$, denoting $\mathcal{E}_S = \{e \in \mathcal{E} | \mathrm{Tl}(e) = s, s \in \mathcal{S}\}$ to be the variables associated with outgoing edges of sources, and $\mathcal{E}_U = \mathcal{E} \setminus \mathcal{E}_S$ to be the non-source edge random variables. For each sink $t \in \mathcal{T}$, the collection of sources this sink will demand will be labeled by the non-empty set $\beta(t) \subseteq \mathcal{S}$. Thus, a network can be represented as a tuple $\mathsf{A} = (\mathcal{S}, \mathcal{G}, \mathcal{T}, \mathcal{E}, \beta)$, where $\beta = (\beta(t), t \in \mathcal{T})$. For convenience, networks with K sources and $L = |\mathcal{E}_U|$ edges are referred as (K, L) instances.

As shown in [5], [6], the rate region of a network A, which is usually characterized by inequalities relating source rates and edge capacities, can be expressed in terms of region of entropic vectors Γ_N^* and some network constraints. The expression of the rate region, as an extension of Theorem 1 in [1], is

$$\mathcal{R}_c(\mathsf{A}) = \mathrm{Proj}_{\boldsymbol{r},\boldsymbol{\omega}}(\overline{\mathrm{con}(\Gamma_{\mathrm{N}}^* \cap \mathcal{L}_{13})} \cap \mathcal{L}_{4'5}), \quad (1)$$

where $\mathrm{con}(\mathcal{B})$ is the conic hull of $\mathcal{B}$, and $\mathrm{Proj}_{\boldsymbol{r},\boldsymbol{\omega}}(\mathcal{B})$ is the projection of the set $\mathcal{B}$ on the coordinates $\left[\boldsymbol{r}^T, \boldsymbol{\omega}^T\right]^T$ where $\boldsymbol{r} = [R_e | e \in \mathcal{E}_U]$ and $\boldsymbol{\omega} = [H(Y_s) | s \in \mathcal{S}]$. Further, Γ_N^* and $\mathcal{L}_i, i = 1, 3, 4', 5$ are viewed as subsets of $\mathbb{R}^M$, $M = 2^N - 1 + |\mathcal{E}_U|$, $N = |\mathcal{E}| = |\mathcal{E}_S| + |\mathcal{E}_U|$, with coordinates $[\boldsymbol{h}^T, \boldsymbol{r}^T]^T$, with $\boldsymbol{h} \in \mathbb{R}^{2^N - 1}$ indexed by subsets of $\mathcal{N}$ as is usual in entropic vectors, $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{E}_U|}$ playing the role of the capacities of edges, and any unreferenced dimensions (e.g. $\boldsymbol{r}$ in Γ_N^*) are left unconstrained (e.g. $\boldsymbol{r} \in \mathbb{R}^{|\mathcal{E}_U|}$ in Γ_N^*). The $\mathcal{L}_i, i = 1, 3, 4', 5$ are network constraints representing source independency, intermediate nodes coding, edge capacity constraints, sink nodes decoding constraints, respectively:

$$\mathcal{L}_1 = \{\mathbf{h} \in \mathbb{R}^M : h_{\mathbf{Y}_S} = \Sigma_{s \in \mathcal{S}} h_{Y_s}\} \quad (2)$$

$$\mathcal{L}_3 = \{\mathbf{h} \in \mathbb{R}^M : h_{\mathbf{U}_{\mathrm{Out}(i)} | \mathbf{U}_{\mathrm{In}(i)}} = 0, \forall i \in \mathcal{G}\} \quad (3)$$

$$\mathcal{L}_{4'} = \{(\mathbf{h}^T, \boldsymbol{r}^T)^T \in \mathbb{R}_+^{2^N - 1 + |\mathcal{E}|} : R_e \geq h_{U_e}, e \in \mathcal{E}\} \quad (4)$$

$$\mathcal{L}_5 = \{\mathbf{h} \in \mathbb{R}^M : h_{\mathbf{Y}_{\beta(t)} | \mathbf{U}_{\mathrm{In}(t)}} = 0, \forall t \in \mathcal{T}\}. \quad (5)$$

and we will denote $\mathcal{L}(\mathsf{A}) = \mathcal{L}_1 \cap \mathcal{L}_3 \cap \mathcal{L}_{4'} \cap \mathcal{L}_5$.

While the analytical expression determines, in principle, the rate region of any network under network coding, it is only an implicit characterization. This is because Γ_N^* is unknown and even non-polyhedral for $N \geq 4$. Further, while $\bar{\Gamma}_N^*$ is a convex cone for all N, Γ_N^* is already non-convex by $N = 3$, though it is also known that the closure only adds points at the boundary of $\bar{\Gamma}_N^*$. Thus, the direct calculation of rate regions from (1) for a network with 4 or more variables is infeasible. On a related note, at the time of writing, it appears to be unknown by the community whether or not the closure after the conic hull is actually necessary in (1) (The closure would be unnecessary if $\bar{\Gamma}_N^* = \mathrm{con}(\Gamma_N^*)$, i.e. if every extreme ray in $\bar{\Gamma}_N^*$ had at least one point along it that was entropic (i.e. in Γ_N^*). At present, all that is known is that Γ_N^* has a solid core, i.e. that the closure only adds points on the boundary of $\bar{\Gamma}_N^*$.), and the uncertainty that necessitates its inclusion muddles a number of otherwise simple proofs and ideas. For this reason, some of the discussion in the remainder of the manuscript will study a closely related inner bound to $\mathcal{R}_c(\mathsf{A})$ as follows

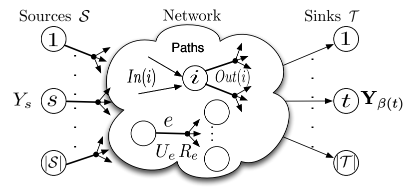


Figure 1: A general network model A

$$\mathcal{R}_*(\mathsf{A}) = \mathrm{Proj}_{\boldsymbol{r},\boldsymbol{\omega}} \mathrm{con}(\Gamma_N^*) \cap \mathcal{L}_{\mathsf{A}}. \quad (6)$$

In all of the cases where the rate region has been computed to date these two regions are equivalent to one another.

Again, both $\mathcal{R}_c(\mathsf{A})$ and its closely related inner bound $\mathcal{R}_*(\mathsf{A})$ are not directly computable because they depend on the unknown region of entropic vectors and its closure. However, replacing Γ_N^* with finitely generated inner and outer bounds, as described in the following corollaries, transforms (1) into a polyhedral computation problem, which involves applying some linear constraints onto a polyhedron and then projecting down onto some coordinates. As described in the introduction and [2]–[6], replacing Γ_N^* with polyhedral inner and outer bounds, typically from $\mathbb{F}_q$ representable matroids and the Shannon outer bound Γ_N, respectively, allows (1) to become a polyhedral computation problem which involves applying some constraints onto a polyhedra and then projecting down onto some coordinates. If the outer and inner bounds on rate region match, we obtain exact rate region.

As shown in [6], there are two types of inner bounds obtained from $\mathbb{F}_q$-representable matroids. One is Γ_N^q, which is obtained directly from the conic hull of $\mathbb{F}_q$-representable matroids on N elements, and is associated with *scalar codes*. The other inner bound, associated with *vector codes*, is $\Gamma_{N,N'}^q$, obtained by N-partitioning the ground sets of $\mathbb{F}_q$-representable matroids on $N' > N$ elements. As $N' \to \infty$, tighter and tighter inner bounds Γ_N^q will be obtained. These bounds are

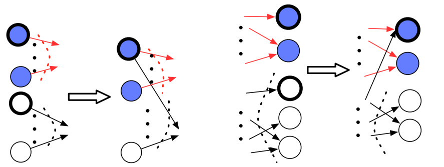
$$\mathcal{R}_o(\mathsf{A}) = \mathrm{Proj}_{\boldsymbol{r},\boldsymbol{\omega}}(\Gamma_N \cap \mathcal{L}(\mathsf{A})) \quad (7)$$

$$\mathcal{R}_{s,q}(\mathsf{A}) = \mathrm{Proj}_{\boldsymbol{r},\boldsymbol{\omega}}(\Gamma_N^q \cap \mathcal{L}(\mathsf{A})) \quad (8)$$

$$\mathcal{R}_q(\mathsf{A}) = \mathrm{Proj}_{\boldsymbol{r},\boldsymbol{\omega}}(\Gamma_{N,\infty}^q \cap \mathcal{L}(\mathsf{A})). \quad (9)$$

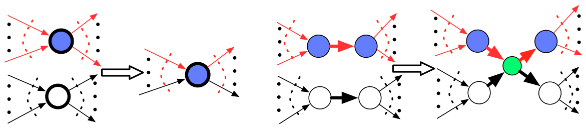
III. Combination Operation Definitions

In this section we propose a series of combination operations relating smaller networks with larger networks in a manner such that the rate region of the larger network can



(a) Sources merge: the merged source will serve for the new larger network.

(b) Sinks merge: input and output of the sinks are unioned.



(c) Intermediate nodes merge: input and output of the nodes are unioned.

(d) Edges merge: one extra node and four associated edges are added to replace the two edges.

Figure 2: Combination operations on two smaller networks to form a larger network. Thickly lined nodes (edges) are merged.

be easily derived from those of the smaller ones. In addition, the sufficiency of a class of linear network codes is inherited in the larger network from the smaller one. Throughout the following, the network $\mathsf{A} = (\mathcal{S}, \mathcal{G}, \mathcal{T}, \mathcal{E}, \beta)$ is a combination of two *disjoint* networks $\mathsf{A}_i = (\mathcal{S}_i, \mathcal{G}_i, \mathcal{T}_i, \mathcal{E}_i, \beta_i)$, $i \in \{1, 2\}$, meaning $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, $\mathcal{G}_1 \cap \mathcal{G}_2 = \emptyset$, $\mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset$, $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$, and $\beta_1(t_1) \cap \beta_2(t_2) = \emptyset, \forall t_1 \in \mathcal{T}_1, t_2 \in \mathcal{T}_2$.

The operations we will define will merge network elements, i.e., sources, intermediate nodes, sink nodes, edges, etc, and are depicted in Fig. 2. Since each merge will combine one or several pairs of elements, with each pair containing one element from A_1 and the other from A_2, each merge definition will involve a bijection π indicating which element from the appropriate set of A_2 is paired with its argument in A_1.

We first consider the sources merge operation, in which the merged sources will function as identical sources for both sub-networks, as shown in Fig. 2a. A sink requiring sources involved in the merge will require the merged source instead.

Definition 1 (Source Merge $(\mathsf{A}_1.\hat{\mathcal{S}} = \mathsf{A}_2.\pi(\hat{\mathcal{S}}))$ – Fig. 2a)**:** Merging the sources $\hat{\mathcal{S}} \subseteq \mathcal{S}_1$ from network A_1 with the sources $\pi(\hat{\mathcal{S}}) \subseteq \mathcal{S}_2$ from a disjoint network A_2, will produce a network A with $i)$ merged sources $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \setminus \pi(\hat{\mathcal{S}})$, $ii)$ $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, $iii)$ $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, $iv)$ $\mathcal{E} = (\mathcal{E}_1 \cup \mathcal{E}_2 \setminus \mathcal{A}) \cup \mathcal{B}$, where $\mathcal{A} = \{e \in \mathcal{E}_1 \cup \mathcal{E}_2 | \mathsf{Tl}(e) \in \hat{\mathcal{S}} \cup \pi(\hat{\mathcal{S}})\}$ includes the edges connected with the sources involved in the merge, $\mathcal{B} = \{(s, \mathcal{F}_1 \cup \mathcal{F}_2) | s \in \hat{\mathcal{S}}, (s, \mathcal{F}_1) \in \mathcal{E}_1, (\pi(s), \mathcal{F}_2) \in \mathcal{E}_2\}$ includes the new edges connected with the merged sources, and $v)$ updated sink demands

$$\beta(t) = \begin{cases} \beta_1(t) & t \in \mathcal{T}_1 \\ \left(\beta_2(t) \setminus \pi(\hat{\mathcal{S}})\right) \cup \pi^{-1}\left(\pi(\hat{\mathcal{S}}) \cap \beta_2(t)\right) & t \in \mathcal{T}_2 \end{cases}.$$

Similar to source merge, we can merge sink nodes of two networks, as demonstrated in Fig. 2b. When two sinks are merged into one sink, we simply union their input and demands as the input and demands of the merged sink.

Definition 2 (Sink Merge $(\mathsf{A}_1.\hat{\mathcal{T}} + \mathsf{A}_2.\pi(\hat{\mathcal{T}}))$ – Fig. 2b.)**:** Merging the sinks $\hat{\mathcal{T}} \subseteq \mathcal{T}_1$ from network A_1 with the sinks $\pi(\hat{\mathcal{T}}) \subseteq \mathcal{T}_2$ from the disjoint network A_2 will produce a network A with $i)$ $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$; $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, $ii)$ $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 \setminus \pi(\hat{\mathcal{T}})$, $iii)$ $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{A} \setminus \mathcal{B}$, where $\mathcal{A} = \{(g_2, \mathcal{F}_1 \cup \mathcal{F}_2) | g_2 \in \mathcal{G}_2, \mathcal{F}_1 \subseteq \hat{\mathcal{T}}, \mathcal{F}_2 \subseteq \mathcal{T}_2, (g_2, \pi(\mathcal{F}_1) \cup \mathcal{F}_2) \in \mathcal{E}_2\}$ updates the head nodes of edges in A_2 with new merged sinks, $\mathcal{B} = \{(g_2, \mathcal{F}_2) \in \mathcal{E}_2 | \mathcal{F}_2 \cap \pi(\hat{\mathcal{T}}) \neq \emptyset\}$ includes the edges connected to sinks in $\pi(\hat{(\mathcal{T})})$, and $v)$ updated sink demands

$$\beta(t) = \begin{cases} \beta_i(t) & t \in \mathcal{T}_i \setminus \hat{\mathcal{T}}, i \in \{1, 2\} \\ \beta_1(t) \cup \beta_2(\pi(t)) & t \in \hat{\mathcal{T}} \end{cases}. \quad (10)$$

Next, we define intermediate nodes merge. When two intermediate nodes are merged, we union their incoming and outgoing edges as the incoming and outgoing edges of the merged node, respectively, as illustrated in Fig. 2c.

Definition 3 (Intermediate Node Merge $(\mathsf{A}_1.g + \mathsf{A}_2.\pi(g))$ – Fig. 2c)**:** Merging the intermediate node $g \in \mathcal{G}_1$ from network A_1 with the intermediate node $\pi(g) \in \mathcal{G}_2$ from the disjoint network A_2 will produce a network A with $i)$ $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, $ii)$ $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 \setminus \pi(g)$, $iii)$ $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, $iv)$ $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{A} \cup \mathcal{B} \setminus \mathcal{C} \setminus \mathcal{D}$, where $\mathcal{A} = \{(g_2, \mathcal{F}_2 \setminus \pi(g) \cup g) | g_2 \in \mathcal{G}_2, (g_2, \mathcal{F}_2 \cup \pi(g)) \in \mathcal{E}_2\}$ updates the head nodes of edges in A_2 that have $\pi(g)$ as head node, $\mathcal{B} = \{(g, \mathcal{F}_2) | (\pi(g), \mathcal{F}_2) \in \mathcal{E}_2\}$ updates the tail node of edges in A_2 that have $\pi(g)$ as tail node, $\mathcal{C} = \{e \in \mathcal{E}_2 | \mathsf{Tl}(e) = \pi(g)\}$ includes the edges in A_2 that have $\pi(g)$ as tail node, $\mathcal{D} = \{e \in \mathcal{E}_2 | \pi(g) \in \mathsf{Hd}(e)\}$ includes the edges in A_2 that have $\pi(g)$ as head node; and $v)$ updated sink demands

$$\beta(t) = \begin{cases} \beta_1(t) & t \in \mathcal{T}_1 \\ \beta_2(t) & t \in \mathcal{T}_2 \end{cases} \quad (11)$$

Finally, we define edge merge. As demonstrated in Fig. 2d, when two edges are merged, one new node and four new edges will be added to create a "cross" component so that the transmission will be in the new component instead of the two edges being merged.

Definition 4 (Edge Merge $(\mathsf{A}_1.e + \mathsf{A}_2.\pi(e))$ – Fig. 2d)**:** Merging edge $e \in \mathcal{E}_1$ from network A_1 with edge $\pi(e) \in \mathcal{E}_2$ from disjoint network A_2 will produce a network A with $i)$ $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, $ii)$ $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 \cup g_0$, where $g_0 \notin \mathcal{G}_1, g_0 \notin \mathcal{G}_2$, $iii)$ $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, $iv)$ $\mathcal{E} = (\mathcal{E}_1 \setminus e) \cup (\mathcal{E}_2 \setminus \pi(e)) \cup \{(\mathsf{Tl}(e), g_0), (\mathsf{Tl}(\pi(e)), g_0), (g_0, \mathsf{Hd}(e)), (g_0, \mathsf{Hd}(\pi(e)))\}$; and $v)$ updated sink demands given by (11).

It is not difficult to see that this edge merge operation can be thought of as a special node merge operation. Suppose the edges being merged are $\mathsf{A}_1.e$, $\mathsf{A}_2.\pi(e)$. If two virtual nodes g_1, g_2 are added on $e, \pi(e)$, respectively, splitting them each into two edges, so that $e, \pi(e)$ go into and flow out g_1, g_2, respectively, then, the merge of g_1, g_2 gives the same network as merging $e, \pi(e)$.

IV. Rate Regions Resulting from the Operations

Here we prove that the combination operations enable the rate regions of the small networks to be combined to produce the rate region of the resulting large network, and also preserve sufficiency of classes of codes and tightness of other bounds.

Theorem 1: Suppose a network A is obtained by merging $\hat{\mathcal{S}}$ with $\pi(\hat{\mathcal{S}})$, i.e., $\mathsf{A}_1.\hat{\mathcal{S}} = \mathsf{A}_2.\pi(\hat{\mathcal{S}})$. Then, for $l \in \{*, q, (s,q), o\}$,

$$\mathcal{R}_l(\mathsf{A}) = \text{Proj}((\mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)) \cap \mathcal{L}_0), \tag{12}$$

with $\mathcal{L}_0 = \left\{H(Y_s) = H(Y_{\pi(s)}), \forall s \in \hat{\mathcal{S}}\right\}$, and the dimensions kept in the projection are $(H(Y_s), s \in \mathcal{S})$ and $(R_e, e \in \mathcal{E})$, where $\mathcal{S}, \mathcal{E}$ represent the source and edge sets of the merged network A, respectively.

Remark 1: The inequality description of the polyhedral cone $\text{Proj}((\mathcal{P}_1 \times \mathcal{P}_2) \cap \mathcal{L}_0)$ for two polyhedral cones $\mathcal{P}_j, j \in \{1,2\}$ can be created by concatenating the inequality descriptions for $\mathcal{P}_1$ and $\mathcal{P}_2$, then replacing the variable $H(Y_{\pi(s)})$ with the variable $H(Y_s)$ for each $s \in \hat{\mathcal{S}}$.

Proof: Select any point $\mathbf{R} \in \mathcal{R}_*(\mathsf{A})$. Then there exists a conic combination of some points in $\mathcal{R}_*(\mathsf{A})$ that are associated with entropic vectors in Γ_N^* such that $\mathbf{R} = \sum_{\mathbf{r}_j \in \mathcal{R}_*(\mathsf{A})} \alpha_j \mathbf{r}_j$, where $\alpha_j \geq 0, \forall j$. For each $\mathbf{r}_j$, there exist random variables $\mathbf{Y}_{\mathcal{S}}^{(j)}, U_i^{(j)}, i \in \mathcal{E} \setminus e$, such that the entropy vector $\mathbf{h}^{(j)} = \left[H(\mathcal{A}) \middle| \mathcal{A} \subseteq \left\{Y_s^{(j)}, U_i^{(j)} | s \in \mathcal{S}, i \in \mathcal{E}\right\}\right]$ is in Γ_N^*, where N is the number of variables in A. Furthermore, their entropies satisfy all the constraints determined by A. When decomposing A into $\mathsf{A}_1, \mathsf{A}_2$, let i.i.d. copies of variables $Y_s^{(j)}, s \in \hat{\mathcal{S}}$ work as sources $\pi(\hat{\mathcal{S}}) \subseteq \mathcal{S}_2$. The associated edges connecting $\hat{\mathcal{S}}$ and nodes in $\mathcal{G}_2$ will then connect $\pi(\hat{\mathcal{S}})$ and nodes in $\mathcal{G}_2$. Then the random variables $\{Y_s^{(j)}, U_i^{(j)} | s \in \mathcal{S}_1, i \in \mathcal{E}_1\}, \{Y_s^{(j)}, U_i^{(j)} | s \in \mathcal{S}_2, i \in \mathcal{E}_2\}$ will satisfy the network constraints determined by $\mathsf{A}_1, \mathsf{A}_2$, and also $\mathcal{L}_0$. Thus, $\mathbf{R} \in \text{Proj}((\mathcal{R}_*(\mathsf{A}_1) \times \mathcal{R}_*(\mathsf{A}_2)) \cap \mathcal{L}_0)$. Similarly, if $\mathbf{R}$ is achievable by $\mathbb{F}_q$ codes, vector or scalar, the same code applied to the part of A that is $\mathsf{A}_1, \mathsf{A}_2$ will achieve $\mathbf{R}_1, \mathbf{R}_2$, respectively. Putting these together, we have $\mathcal{R}_l(\mathsf{A}) \subseteq \text{Proj}((\mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)) \cap \mathcal{L}_0), l \in \{*, q, (s,q)\}$.

Next, if we select two points $\mathbf{R}_1 \in \mathcal{R}_*(\mathsf{A}_1)$, $\mathbf{R}_2 \in \mathcal{R}_*(\mathsf{A}_2)$ such that $H(Y_s) = H(Y_{\pi(s)}), \forall s \in \hat{\mathcal{S}}$, then there exist conic combinations $\mathbf{R}_1 = \sum_{\mathbf{r}_{i,j} \in \mathcal{R}_*(\mathsf{A}_i)} \alpha_{i,j} \mathbf{r}_{i,j}$ for $i = 1, 2$, and for each $\mathbf{r}_{i,j}$ there exist a set of variables associated with sources and edges. Since $H(Y_s) = H(Y_{\pi(s)})$ and sources are independent and uniformly distributed, we can let the associated variables $Y_s^{(j)}$ and $Y_{\pi(s)}^{(j)}$ be the same variables. Then, after combination, the entropy vector of all variables $\{Y_s^{(j)}, U_e^{(j)} | s \in \mathcal{S}, e \in \mathcal{E}\}$ will be in Γ_N^*. Furthermore, their entropies, together with the rate vectors from $\mathbf{R}_1, \mathbf{R}_2$, will satisfy all network constraints of A, and there will be an associated point $\mathbf{r} = \mathbf{r}_1 \times \mathbf{r}_2$ with $\mathcal{L}_0$. Using the same conic combination, we will find the associated point $\mathbf{R} = \mathbf{R}_1 \times \mathbf{R}_2 \cap \mathcal{L}_0$. Hence, $\text{Proj}((\mathcal{R}_*(\mathsf{A}_1) \times \mathcal{R}_*(\mathsf{A}_2)) \cap \mathcal{L}_0) \subseteq \mathcal{R}_*(\mathsf{A})$. Now suppose there exists a sequence of network codes for A_1 and A_2 achieving $\mathbf{R}_1, \mathbf{R}_2$. By using the same source bits as the source inputs for s in A_1 and $\pi(s)$ in A_2 for each $s \in \hat{S}$, we have the same effect as using these source bits as the inputs for s in the source merged A and achieving the associated rate vector $\mathbf{R}$, implying $\mathbf{R} \in \mathcal{R}_l(\mathsf{A})$, $l \in \{q, (s,q)\}$, and hence $\mathcal{R}_{s,q}(\mathsf{A}) \supseteq \text{Proj}((\mathcal{R}_{s,q}(\mathsf{A}_1) \times \mathcal{R}_{s,q}(\mathsf{A}_2)) \cap \mathcal{L}_0)$ and $\mathcal{R}_q(\mathsf{A}) \supseteq \text{Proj}((\mathcal{R}_q(\mathsf{A}_1) \times \mathcal{R}_q(\mathsf{A}_2)) \cap \mathcal{L}_0)$. Together with the statements above, this proves (12) for $l \in \{*, q, (s,q)\}$.

Furthermore, any point $\mathbf{R} \in \mathcal{R}_o(\mathsf{A})$, is the projection of some point $[\mathbf{h}, \mathbf{r}] \in \Gamma_N \cap \mathcal{L}_\mathsf{A}$, where $\mathbf{h} \in \Gamma_N$ and $\mathbf{r} = [R_e | e \in \mathcal{E}]$. Because the Shannon inequalities and network constraints in $\Gamma_N \cap \mathcal{L}_\mathsf{A}$ form a superset (i.e., include all of) of the network constraints in $\Gamma_N \cap \mathcal{L}(\mathsf{A}_i)$, the subvectors $[\mathbf{h}^i, \mathbf{r}^i]$ of $[\mathbf{h}, \mathbf{r}]$ associated only with the variables in A_i (with $Y_{\pi(s)}$ being recognized as Y_s for all $s \in \hat{S}$) are in $\Gamma_{N_i} \cap \mathcal{L}(\mathsf{A}_i)$ and obey $\mathcal{L}_0$, implying $\mathbf{R} \in \text{Proj}((\mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)) \cap \mathcal{L}_0)$, and hence $\mathcal{R}_o(\mathsf{A}) \subseteq \text{Proj}((\mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)) \cap \mathcal{L}_0)$.

Next, if we select two points $\mathbf{R}_1 \in \mathcal{R}_o(\mathsf{A}_1)$, $\mathbf{R}_2 \in \mathcal{R}_o(\mathsf{A}_2)$ such that $H(Y_s) = H(Y_{\pi(s)}), \forall s \in \hat{\mathcal{S}}$, then there exists $[\mathbf{h}^i, \mathbf{r}^i] \in \Gamma_{N_i} \cap \mathcal{L}(\mathsf{A}_i)$, where $\mathbf{h}^i \in \Gamma_{N_i}$ and $\mathbf{r}^i = [R_e | e \in \mathcal{E}_i]$, such that $\mathbf{R}_i = \text{Proj}_{\boldsymbol{r}_i, \boldsymbol{\omega}_i}[\mathbf{h}^i, \mathbf{r}^i]$, $i \in \{1,2\}$ with $h^1_{X_s} = h^2_{X_{\pi(s)}}$ for all $s \in \hat{S}$. Define $\mathbf{h}$ whose element associated with the subset $\mathcal{A}$ of $\mathcal{N} = \mathcal{S} \cup \mathcal{E}$ is $h_\mathcal{A} = h^1_{\mathcal{A} \cap \mathcal{N}_1} + h^2_{\mathcal{A} \cap \mathcal{N}_2} - h^2_{\mathcal{A} \cap \pi(\hat{\mathcal{S}})}$ where $\mathcal{N}_i = \mathcal{S}_i \cup \mathcal{E}_i$, $i \in \{1,2\}$. By virtue of its creation this way, this function is submodular and $\mathbf{h} \in \Gamma_N$. Since the two networks are disjoint, the list of equalities in $\mathcal{L}_3(\mathsf{A})$ is simply the concatenation of the lists in $\mathcal{L}_3(\mathsf{A}_1)$ and $\mathcal{L}_3(\mathsf{A}_2)$, each of which involved inequalities in disjoint variables $\mathcal{N}_1$ and $\mathcal{N}_2$, and the same thing holds for $\mathcal{L}_{4'}$ with consideration of $\mathbf{r}^i$. Furthermore, since $\mathbf{h}^i \in \mathcal{L}_2(\mathsf{A}_i)$ and $h^1_{Y_s} = h^2_{Y_{\pi(s)}}, s \in \hat{\mathcal{S}}$, $\mathbf{h}$ obeys $\mathcal{L}_2(\mathsf{A})$. The definition of $\mathbf{h}$, together with $\mathbf{h}^i \in \mathcal{L}_1(\mathsf{A}_i)$, $i \in \{1,2\}$ and $h^1_{Y_s} = h^2_{Y_{\pi(s)}}$, $s \in \hat{\mathcal{S}}$, implies that $\mathbf{h} \in \mathcal{L}_1(\mathsf{A})$. Finally $\mathbf{h}^1 \in \mathcal{L}(\mathsf{A}_1)$ and $\mathbf{h}^2 \in \mathcal{L}(\mathsf{A}_2)$ imply $\mathbf{h} \in \mathcal{L}_5(\mathsf{A})$. Putting these facts together we observe that $[\mathbf{h}, \mathbf{r}] \in \Gamma_N \cap \mathcal{L}_\mathsf{A}$, so $\mathbf{R} \in \mathcal{R}_o(\mathsf{A})$, implying $\mathcal{R}_o(\mathsf{A}) \supseteq \text{Proj}((\mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)) \cap \mathcal{L}_0)$. ■

Theorem 2: Suppose a network A is obtained by merging sink nodes $\hat{\mathcal{T}}$ with $\pi(\hat{\mathcal{T}})$, i.e., $(\mathsf{A}_1.\hat{\mathcal{T}} + \mathsf{A}_2.\pi(\hat{\mathcal{T}}))$. Then

$$\mathcal{R}_l(\mathsf{A}) = \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2),\ l \in \{*, q, (s,q), o\} \tag{13}$$

with the index on the dimensions mapping from $\{e \in \mathcal{E}_2 | \mathsf{Hd}(e) \in \pi(\hat{\mathcal{T}})\}$ to $\{e \in \mathcal{E} | \mathsf{Hd}(e) \in \hat{\mathcal{T}}, \mathsf{Tl}(e) \in \mathcal{G}_2\}$.

Proof: Consider a point $\mathbf{R} \in \mathcal{R}_l(\mathsf{A})$ with conic combination of $\mathbf{R} = \sum_{\mathbf{r}_j \in \mathcal{R}_l(\mathsf{A})} \alpha_{l,j} \mathbf{r}_{l,j}$, where $\alpha_{l,j} \geq 0$ for any j and $l \in \{*, q, (s,q), o\}$. Each $\mathbf{r}_{l,j}$ has associated random variables or the associated codes. Due to the independence of sources in networks $\mathsf{A}_1, \mathsf{A}_2$, and the fact that their sources and intermediate nodes are disjoint, the variables arriving at a merged sink node from A_1 will be independent of the sources in A_2 and the variables arriving at a merged sink node from A_2 will be independent of the sources in A_1. In particular, Shannon type inequalities imply the Markov chains $H(\mathbf{Y}_{\mathcal{S}_1} | \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_1}, \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_2}) = H(\mathbf{Y}_{\mathcal{S}_1} | \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_1})$ and $H(\mathbf{Y}_{\mathcal{S}_2} | \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_1}, \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_2}) = H(\mathbf{Y}_{\mathcal{S}_2} | \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_2})$ for all $t \in \mathcal{T}$ (even if the associated "entropies" are only in Γ_N and not necessarily $\bar{\Gamma}_N^*$). This then implies, together with the independence of the sources, that $H(\mathbf{Y}_{\beta(t)} | \mathbf{U}_{\text{In}(t)}) = H(\mathbf{Y}_{\beta(t) \cap \mathcal{S}_1} | \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_1}) + H(\mathbf{Y}_{\beta(t) \cap \mathcal{S}_2} | \mathbf{U}_{\text{In}(t) \cap \mathcal{E}_2})$, showing that the constraints in $\mathcal{L}_5(\mathsf{A})$ imply the constraints in $\mathcal{L}_5(\mathsf{A}_1)$ and $\mathcal{L}_5(\mathsf{A}_2)$. Furthermore, given the disjoint nature of A_1 and A_2, the constraints in $\mathcal{L}_i(\mathsf{A})$, are simply the concatenation of the constraints in $\mathcal{L}_i(\mathsf{A}_1)$ and $\mathcal{L}_i(\mathsf{A}_2)$, for $i \in \{2, 3, 4'\}$. Furthermore, the joint independence of all of $\mathbf{Y}_{\mathcal{S}_1}, \mathbf{Y}_{\mathcal{S}_2}$ imply the marginal independence of the collections of variables $\mathbf{Y}_{\mathcal{S}_1}$ and $\mathbf{Y}_{\mathcal{S}_2}$, so that $\mathcal{L}_1(\mathsf{A})$ implies $\mathcal{L}_1(\mathsf{A}_i), i \in \{1,2\}$. This shows that $\mathbf{r}_{l,j} \in \mathbf{r}^1_{l,j} \times \mathbf{r}^2_{l,j}$ and further $\mathbf{R} \in \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)$, and hence $\mathcal{R}_l(\mathsf{A}) \subseteq \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2), l \in \{*, q, (s,q), o\}$.

Next, consider two points $\mathbf{R}_i \in \mathcal{R}_l(\mathsf{A}_i),\ i \in \{1,2\}$ for any $l \in \{q, (s,q), o\}$. By definition these are projections of $[\mathbf{h}^i, \mathbf{r}^i] \in \Gamma^q_{N_i,\infty} \cap \mathcal{L}(\mathsf{A}_i)$, $[\mathbf{h}^i, \mathbf{r}^i] \in \Gamma^q_{N_i} \cap \mathcal{L}(\mathsf{A}_i)$, $[\mathbf{h}^i, \mathbf{r}^i] \in \Gamma_{N_i} \cap \mathcal{L}(\mathsf{A}_i)$, respectively, for $i \in \{1,2\}$, where $\mathbf{h}^i \in \Gamma_{N_i}$ and $\mathbf{r}^i = [R_e | e \in \mathcal{E}_i]$. Define $\mathbf{h}$ with value associated with subset $\mathcal{A} \subseteq \mathcal{N}$ of $h_{\mathcal{A}} = h^1_{\mathcal{A}\cap\mathcal{N}_1} + h^2_{\mathcal{A}\cap\mathcal{N}_2}$, then it is easily verified that the resulting $[\mathbf{h}, \mathbf{r}^1, \mathbf{r}^2] \in \Gamma^q_{N,\infty} \cap \mathcal{L}_{\mathsf{A}}$, $[\mathbf{h}, \mathbf{r}^1, \mathbf{r}^2] \in \Gamma^q_N \cap \mathcal{L}_{\mathsf{A}}$, $[\mathbf{h}, \mathbf{r}^1, \mathbf{r}^2] \in \Gamma_N \cap \mathcal{L}_{\mathsf{A}}$, respectively, (simply use the same codes from A_1 and A_2 on the corresponding parts of A). Since $\mathbf{R} = \mathrm{Proj}_{\boldsymbol{\omega},\boldsymbol{r}}[\mathbf{h}, \mathbf{r}^1, \mathbf{r}^2]$, we have proven $\mathbf{R} \in \mathcal{R}_l(\mathsf{A})$, and hence that $\mathcal{R}_l(\mathsf{A}) \supseteq \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)$. Further, for two points $\mathbf{R}_i \in \mathcal{R}_*(\mathsf{A}_i),\ i \in \{1,2\}$, there exist a conic combination of $\mathbf{r}^i_j$, $\mathbf{R}_i = \sum_{\mathbf{r}^i_j \in \mathcal{R}_*(\mathsf{A}_i)} \alpha^i_j \mathbf{r}^i_j$, with associated random variables $\left\{Y^{(j)}_s, U^{(j)}_i | s \in \mathcal{S}_1, i \in \mathcal{E}_1\right\}, \left\{Y^{(j)}_s, U^{(j)}_i | s \in \mathcal{S}_2, i \in \mathcal{E}_2\right\}$ satisfying the network constraints determined by $\mathsf{A}_1, \mathsf{A}_2$. Due to the independence of sources and disjoint of edge variables, the union of variables in $\mathsf{A}_1, \mathsf{A}_2$ will satisfy the network constraints in the merged A. With the same conic combinations, we have $\mathbf{R} = \sum_{\mathbf{r}^i_j \in \mathcal{R}_*(\mathsf{A}_i)} [\alpha^1_j \mathbf{r}^1_j, \alpha^2_j \mathbf{r}^2_j] \in \mathcal{R}_*(\mathsf{A})$. Thus, $\mathcal{R}_*(\mathsf{A}) \supseteq \mathcal{R}_*(\mathsf{A}_1) \times \mathcal{R}_*(\mathsf{A}_2)$. ■

Theorem 3: Suppose a network A is obtained by merging g and $\pi(g)$, i.e., $\mathsf{A}_1.g + \mathsf{A}_2.\pi(g)$. Then

$$\mathcal{R}_l(\mathsf{A}) = \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2),\ l \in \{*, q, (s,q), o\} \qquad (14)$$

with dimensions/ indices mapping from $\{e \in \mathcal{E}_2 | \mathsf{Hd}(e) = \pi(g)\}$ to $\{e \in \mathcal{E} | \mathsf{Hd}(e) = g, \mathsf{Tl}(e) \in \mathcal{G}_2\}$ and from $\{e \in \mathcal{E}_2 | \mathsf{Tl}(e) = \pi(g)\}$ to $\{e \in \mathcal{E} | \mathsf{Tl}(e) = g, \mathsf{Hd}(e) \in \mathcal{G}_2\}$.

Proof: Consider a point $\mathbf{R} \in \mathcal{R}_l(\mathsf{A})$ for any $l \in \{*, q, (s,q)\}$ and all random variables associated with each component $\mathbf{r}_{l,j}$ in the conic combinations $\mathbf{R} = \sum_{\mathbf{r}_j \in \mathcal{R}_l(\mathsf{A})} \alpha_{l,j} \mathbf{r}_{l,j}$, where $\alpha_{l,j} \geq 0$ for any j and $l \in \{*, q, (s,q), o\}$. The associated variables satisfy $\mathcal{L}_i(\mathsf{A}), i = 1, 3, 4', 5$. Partition the incoming edges of the merged node g in A, $\mathrm{In}(g)$, up into $\mathrm{In}_1(g) = \mathrm{In}(g) \cap \mathcal{E}_1$ the edges from A_1, and $\mathrm{In}_2(g) = \mathrm{In}(g) \setminus \mathrm{In}_1(g)$, the new incoming edges resulting from the merge. Similarly, partition the outgoing edges $\mathrm{Out}(g)$ up into $\mathrm{Out}_1(g) = \mathrm{Out}(g) \cap \mathcal{E}_1$ and $\mathrm{Out}_2(g) = \mathrm{Out}(g) \setminus \mathrm{Out}_1(g)$. The $\mathcal{L}_3$ constraints dictate that there exist functions f_e such that for each $e \in \mathrm{Out}(g)$, $U_e = f_e(U_{\mathrm{In}_1(g)}, U_{\mathrm{In}_2(g)})$. Define the new functions f'_e via

$$f'_e(U_{\mathrm{In}_1(g)}, U_{\mathrm{In}_2(g)}) = \begin{cases} f_e(U_{\mathrm{In}_1(g)}, \mathbf{0}) & e \in \mathrm{Out}_1(g) \\ f_e(\mathbf{0}, U_{\mathrm{In}_2(g)}) & e \in \mathrm{Out}_2(g) \end{cases} \qquad (15)$$

i.e., set the possible value for the incoming edges from the other part of the network (possibly erroneously) to a particular constant value among their possible values – let's label it $\mathbf{0}$. The network code using these new functions f'_e will utilize the same rates as before. The constraints and the topology of the merged network further dictated that $U_{\mathrm{In}_i(g)}$ were expressible as a function of $\mathcal{S}_i, i \in \{1,2\}$. In the remainder of the network (moving toward the sink nodes) after the merged nodes, at no other point is any information from the sources in the other part of the network encountered, and the decoders at the sink nodes in $\mathcal{T}_2$ need to work equally well decoding subsets of $\mathcal{S}_2$, regardless of the value of $\mathcal{S}_1$. Since the erroneous value for the $U_{\mathrm{In}_1(g)}$ used for $f'_e, e \in \mathrm{Out}_2(g)$ was still a valid possibility for some (possible other) value(s) of the sources in $\mathcal{S}_1$, the sinks must still produce the correct values for their subsets of $\mathcal{S}_2$. A parallel argument for $\mathcal{T}_1$ shows that they still correctly decode their sources, which were subsets of $\mathcal{S}_1$, even though the f_es were changed to f'_es. Note further that (15) will still be scalar/vector linear if the original f_es were as well.

However, since the $f_{e'}$s no longer depend on the other half of the network, the resulting code can be used as separate codes for A_1 and A_2, given the associated rate points $\mathbf{R}_i$ by keeping the elements in $\mathbf{R}$ associated with $\mathsf{A}_i, i \in \{1,2\}$ (or the associated rate points $\mathbf{r}^i_{l,j}$ by keeping elements in $\mathbf{r}_{l,j}$) in the natural way, implying that $\mathbf{R} \in \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)$. This then implies that $\mathcal{R}_l(\mathsf{A}) \subseteq \mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)$ for all $l \in \{*, q, (s,q)\}$. The opposite containment is obvious, since any rate points or codes for the two networks can be utilized in the trivial manner for the merged network. This proves (14) for $l \in \{*, q, (s,q)\}$.

Next, consider any pair $\mathbf{R}_i \in \mathcal{R}_o(\mathsf{A}_i)\ i \in \{1,2\}$, which are, by definition, projections of some $[\mathbf{h}^i, \mathbf{r}^i] \in \Gamma_{N_i} \cap \mathcal{L}(\mathsf{A}_i)$, where $\mathbf{h}^i \in \Gamma_{N_i}$ and $\mathbf{r}^i = [R_e | e \in \mathcal{E}_i],\ i \in \{1,2\}$. Defining $\mathbf{h}$ whose element associated with the subset $\mathcal{A} \subset \mathcal{N}$ is $h_{\mathcal{A}} = h^1_{\mathcal{A}\cap\mathcal{N}_1} + h^2_{\mathcal{A}\cap\mathcal{N}_2}$, where the intersection respects the remapping of edges under the intermediate node merge, we observe that $[\mathbf{h}, \mathbf{r}^1, \mathbf{r}^2] \in \Gamma_N \cap \mathcal{L}_{\mathsf{A}}$, and hence its projection $\mathbf{R} \in \mathcal{R}_o(\mathsf{A})$, proving $\mathcal{R}_o(\mathsf{A}) \supseteq \mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)$.

Finally, consider a point $\mathbf{R} \in \mathcal{R}_o(\mathsf{A})$, which is a projection of some $[\mathbf{h}, \mathbf{r}] \in \Gamma_N \cap \mathcal{L}_{\mathsf{A}}$, where $\mathbf{h} \in \Gamma_N$ and $\mathbf{r} = [R_e | e \in \mathcal{E}]$. For every $\mathcal{A} \subseteq \mathcal{N}_i$, define $h^i_{\mathcal{A}} = h_{\mathcal{A}\cup\mathcal{S}_{3-i}} - h_{\mathcal{S}_{3-i}}$, and define $\mathbf{h}'$ with $h'_{\mathcal{A}} = h^1_{\mathcal{A}\cap\mathcal{N}_1} + h^2_{\mathcal{A}\cap\mathcal{N}_2}$ and $\mathbf{R}' = \mathrm{proj}_{\boldsymbol{\omega},\boldsymbol{r}}\mathbf{h}'$. We see that $\mathbf{h}^i \in \mathcal{L}(\mathsf{A}_i), i \in \{1,2\}$, because conditioning reduces entropy and entropy is non-negative, but all of the conditional entropies at nodes other than g were already zero, while at g, the conditioning on the sources from the other network will enable the same conditional entropy of zero since the incoming edges from the other network were functions of them. This shows that $\mathbf{R}' \in \mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)$. Owing to the independence of the sources $\mathrm{proj}_{\boldsymbol{\omega}}\mathbf{h} = \mathrm{proj}_{\boldsymbol{\omega}}\mathbf{h}'$, while $\mathrm{proj}_{\boldsymbol{r}}\mathbf{h} \geq \mathrm{proj}_{\boldsymbol{r}}\mathbf{h}'$ due to the fact that conditioning reduces entropy. The coordinate convex nature then implies that $\mathbf{R} \in \mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)$ showing that $\mathcal{R}_o(\mathsf{A}) \subseteq \mathcal{R}_o(\mathsf{A}_1) \times \mathcal{R}_o(\mathsf{A}_2)$ and completing the proof. ■

Theorem 4: Suppose a network A is obtained by merging e and $\pi(e)$, i.e., $\mathsf{A}_1.e + \mathsf{A}_2.\pi(e)$. Then, for $l \in \{*, q, (s,q), o\}$,

$$\mathcal{R}_l(\mathsf{A}) = \mathrm{Proj}_{\setminus\{e,\pi(e)\}}((\mathcal{R}_l(\mathsf{A}_1) \times \mathcal{R}_l(\mathsf{A}_2)) \cap \mathcal{L}'_0), \qquad (16)$$

with $\mathcal{L}'_0 = \left\{R_{(\mathrm{Tl}(j),g_0)} \geq R_j, R_{(g_0,\mathrm{Hd}(j))} \geq R_j, j \in \{e, \pi(e)\}\right\}$, and projection dimension $\setminus\{e, \pi(e)\}$ means projecting out dimensions associated with $e, \pi(e)$. Furthermore, $\mathcal{R}(\mathsf{A}_q) \times \mathcal{R}(\mathsf{A}_q)$ and $\mathcal{L}'_0$ are viewed in the dimension of $|\mathcal{N}_1| + |\mathcal{N}_2| + 4$ with assumption that all dimensions not shown are unconstrained.

Proof: As observed after the definition of edge merge, one can think of edge merge as the concatenation of two operations: $i)$ split e in A_1 and $\pi(e)$ in A_2 each up into two edges with a new intermediate node (g and $\pi(g)$, respectively) in between them, forming A'_1 and A'_2, respectively, followed by $ii)$ intermediate node merge of $\mathsf{A}'_1.g+\mathsf{A}'_2.\pi(g)$. It is clear that $\mathcal{L}'_0$ describes the operation that must happen to the rate region of A_i, $i \in \{1,2\}$ to get the rate region of A'_i, because the contents of the old edge e or $\pi(e)$ must now be carried by both new edges after the introduction of the new intermediate node. Applying Thm. 3 to A'_1 and A'_2 yields (16). ■

With Theorems 1 – 4, one can easily derive the following corollary regarding the preservation of sufficiency of linear network codes and tightness of Shannon outer bound.

Corollary 1: Let network A be a combination of networks $\mathsf{A}_1, \mathsf{A}_2$ via one of the combination operations. If $\mathbb{F}_q$ vector (scalar) linear codes suffice or the Shannon outer bound is tight for both $\mathsf{A}_1, \mathsf{A}_2$, then the same will be true for A. Equivalently, if $\mathcal{R}_l(\mathsf{A}_i) = \mathcal{R}_*(\mathcal{A}_i), i \in \{1,2\}$ for some $l \in \{o, q, (s,q)\}$ then also $\mathcal{R}_l(\mathsf{A}) = \mathcal{R}_*(\mathsf{A})$.

V. Experimental Results

In this section, we first present some experimental results on numbers of non-isomorphic networks. Then we utilize an example to show the rate region relations and the preservation properties resulting from the operations presented in §III.

It is not difficult to see a *minimal* network instance with simplest connected structure, should obey the following constraints:

Source minimality:

(C1) all sources cannot be only directly connected with sinks: $\forall s \in \mathcal{S}$, $\mathrm{Hd}(\mathrm{Out}(s)) \cap \mathcal{G} \neq \emptyset$;

(C2) sinks do not demand sources to which they are directly connected: $\forall s \in \mathcal{S},\ t \in \mathcal{T}$, if $t \in \mathrm{Hd}(\mathrm{Out}(s))$ then $s \notin \beta(t)$;

(C3) every source is demanded by at least one sink: $\forall s \in \mathcal{S}$, $\exists\, t \in \mathcal{T}$ such that $s \in \beta(t)$;

(C4) sources connected to the same intermediate node and demanded by the same set of sinks should be merged: $\nexists s, s' \in \mathcal{S}$ such that $\mathrm{Hd}(\mathrm{Out}(s)) = \mathrm{Hd}(\mathrm{Out}(s'))$ and $\gamma(s) = \gamma(s')$, where $\gamma(s) = \{t \in \mathcal{T} | s \in \beta(t)\}$;

Node minimality:

(C5) intermediate nodes with identical inputs should be merged: $\nexists\, k, l \in \mathcal{G}$ such that $\mathrm{In}(k) = \mathrm{In}(l)$;

(C6) intermediate nodes should have nonempty inputs and outputs, and sink nodes should have nonempty inputs: $\forall g \in \mathcal{G}, t \in \mathcal{T}, \mathrm{In}(g) \neq \emptyset, \mathrm{Out}(g) \neq \emptyset, \mathrm{In}(t) \neq \emptyset$;

Edge minimality:

(C7) all hyperedges must have at least one head: $\nexists e \in \mathcal{E}$ such that $\mathrm{Hd}(e) = \emptyset$;

(C8) identical edges should be merged: $\nexists e, e' \in \mathcal{E}$ with $\mathrm{Tl}(e) = \mathrm{Tl}(e')$, $\mathrm{Hd}(e) = \mathrm{Hd}(e')$;

(C9) intermediate nodes with unit in and out degree, and whose in edge is not a hyperedge, should be removed: $\nexists e, e' \in \mathcal{E}, g \in \mathcal{G}$ such that $\mathrm{In}(g) = e$, $\mathrm{Hd}(e) = g$, $\mathrm{Out}(g) = e'$;

Sink minimality:

(C10) there must exist a path to a sink from every source wanted by that sink: $\forall t \in \mathcal{T}, \beta(t) \subseteq \sigma(t)$, where $\sigma(t) = \{k \in \mathcal{S} | \exists$ a path from k to $t\}$;

(C11) every pair of sinks must have a distinct set of incoming edges: $\forall t, t' \in \mathcal{T}, i \neq j$, $\mathrm{In}(t) \neq \mathrm{In}(t')$;

(C12) if one sink receives a superset of inputs of a second sink, then the two sinks should have no common sources in demand: If $\mathrm{In}(t) \subseteq \mathrm{In}(t')$, then $\beta(t) \cap \beta(t') = \emptyset$;

(C13) if one sink receives a superset of inputs of a second sink, then the sink with superset input should not have direct access to the sources that demanded by the sink with subset input: If $\mathrm{In}(t) \subseteq \mathrm{In}(t')$ then $t' \notin \mathrm{Hd}(\mathrm{Out}(s))$ for all $s \in \beta(t)$.

Connectivity:

(C14) the direct graph associated with the network A is weakly connected.

To better highlight this definition of network minimality, we explain the conditions involved in greater detail. The first condition (**C1**) requires that a source cannot be only directly connected with some sinks, for otherwise no sink needs to demand it, according to (**C2**) and (**C10**). Therefore, this source is extraneous. The condition (**C2**) holds because otherwise the demand of this sink will always be trivially satisfied, hence removing this reconstruction constraint will not alter the rate region. Note that other sources not demanded by a given sink can be directly available to that sink as side information (e.g., as in index coding problems), as long as condition (**C13**) is satisfied. The condition (**C3**) indicates that each source must be demanded by some sink nodes, for otherwise it is extraneous and can be removed. The condition (**C4**) says that no two sources have exactly the same paths and set of demanders (sinks requesting the source), because in that case the network can be simplified by combining the two sources as a super-source. The condition (**C5**) requires that no two intermediate nodes have exactly the same input, for otherwise the two nodes can be combined. The condition (**C6**) requires that no nodes have empty input except the source nodes, for otherwise these nodes are useless and extraneous from the standpoint of satisfying the network coding problem. The condition (**C7**) requires that every edge variable must be in use in the network, for otherwise it is also extraneous and can be removed. The condition (**C8**) guarantees that there is no duplication of hyperedges, for otherwise they can be merged with one another. The condition (**C9**) says that there is no trivial relay node with only one non-hyperedge input and output, for otherwise the head of the input edge can be replaced with the head of the output edge. The condition (**C10**) reflects the fact that a sink can only decode the sources to which it has at least one path of access, and any demanded source not meeting this constraint will be forced to have entropy rate of zero. The condition (**C11**) indicates the trivial requirement that no two decoders should have the same input, for otherwise these two decoders can be combined. The condition (**C12**) simply stipulates that implied capabilities of sink nodes are not to be stated, but rather inferred from the implications. In particular, if $\mathrm{In}(t) \subseteq \mathrm{In}(t')$, and $\beta(t) \cap \beta(t') \neq \emptyset$, the decoding

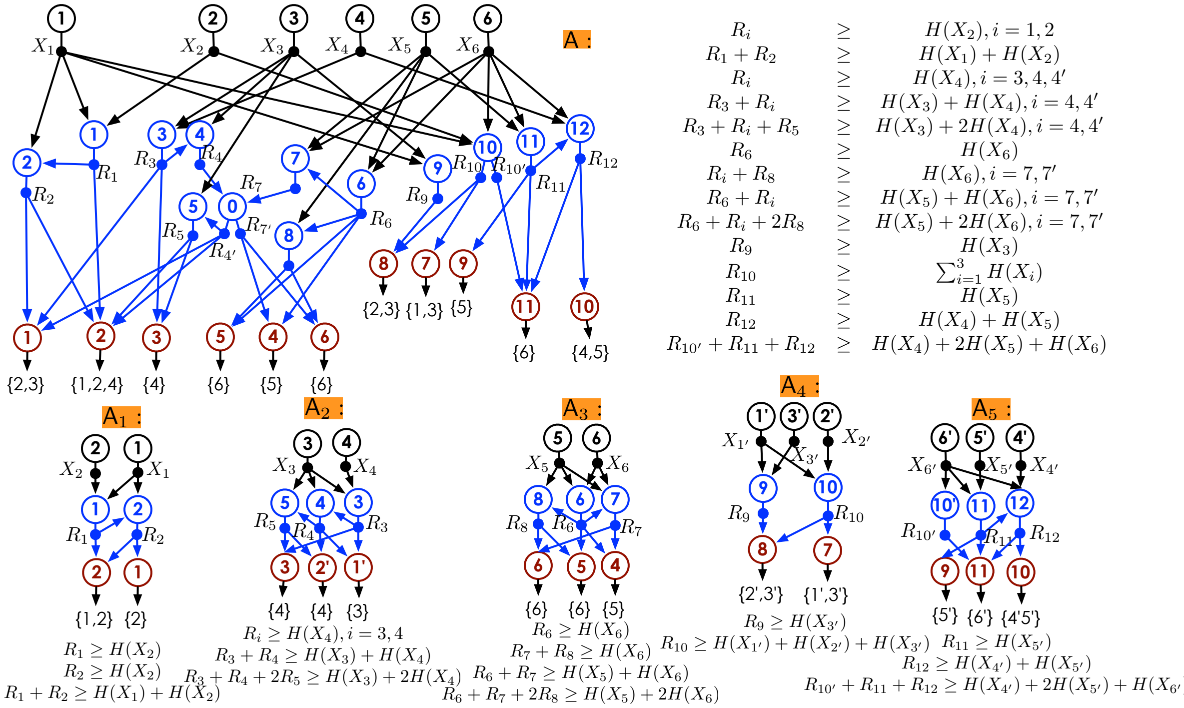


Figure 3: A large network and its capacity region created with the operations in this paper from the 5 networks below it.

ability of $\beta(t)$ is implied at t': pursuing minimality, we only let t' demand extra sources, if any. The condition (**C13**) is also necessary because the availability of s is already implied by having access to $\text{In}(t)$, hence, there is no need to have direct access to s.

By using our enumeration tool [7] with consideration of the conditions above, we obtained the list of non-isomorphic (symmetry removed) network instances for different network sizes with $K + L \leq 5$. We give the numbers of network instances as following

(K, L)	$\lvert\mathcal{Z}\rvert$	$\lvert\hat{\mathcal{Z}}\rvert$
(1,2)	4	7
(1,3)	132	749
(1,4)	18027	420948
(2,1)	1	1
(2,2)	333	1 270
(2,3)	485 890	5 787 074
(3,1)	9	31
(3,2)	239 187	2 829 932
(4,1)	536	10478
Total	744 119	9 050 490

where $\lvert\hat{\mathcal{Z}}\rvert, \lvert\mathcal{Z}\rvert$ represents the number of isomorphic networks and non-isomorphic networks, respectively. As [7] shows, for all networks with $K + L \leq 5$, we proved that Shannon outer bound is tight and linear codes suffice.

Next, we give an example to demonstrate the rate region relations and achieving codes from combination operations.

Example 1: A $(6, 15)$ network instance A can be obtained by combining five smaller networks $\mathsf{A}_1, \ldots, \mathsf{A}_5$, of which the representations are shown in Fig. 3. The combination process is I) $\mathsf{A}_{12} = \mathsf{A}_1.\{t_1, t_2\} + \mathsf{A}_2.\{t_{1'}, t_{2'}\}$; II) $\mathsf{A}_{123} = \mathsf{A}_{12}.e_4 + \mathsf{A}_3.e_7$ with extra node g_0 and edges $e_{4'}, e_{7'}$; III) $\mathsf{A}_{45} = \mathsf{A}_4.g_{10} + \mathsf{A}_5.g_{10'}$; IV) $\mathsf{A} = \mathsf{A}_{123}.\{X_1, \ldots, X_6\} = \mathsf{A}_{45}.\{X_{1'}, \ldots, X_{6'}\}$. From the software calculations and analysis [9], one obtains the rate regions below the 5 small networks. According to the theorems in §IV, the rate region $\mathcal{R}_*(\mathsf{A})$ for A obtained from $\mathcal{R}_*(\mathsf{A}_1), \ldots, \mathcal{R}_*(\mathsf{A}_5)$, is depicted next to it. Additionally, since calculations showed binary codes and the Shannon outer bound suffice for $\mathsf{A}_i,\ i \in \{1, \ldots, 5\}$, Corollary 1 dictates the same for A.

Note that, by repeating one or more combination operations, we can calculate rate regions and tell sufficiency property for many instances with large network sizes. If we integrate embedding operations into the process, as shown in [7], more and more solvable networks can be obtained.

VI. Conclusion

This paper introduced several operations for combining smaller networks into bigger networks with the property that the rate region of the larger network could be determined from the rate regions of the smaller networks. Additionally, if certain classes of linear codes suffice (or Shannon outer bound is tight) for the smaller networks, then the same class of codes will suffice (or Shannon outer bound will be tight) for the combined larger network. The operators create a way to put together small computationally derived rate regions to get rate regions of networks of arbitrary scale. Additionally, they enable one to determine rate regions and the sufficiency of classes of codes to exhaust it for a larger network by decomposing it (reversing the operations) into smaller networks, then investigating the same properties of the smaller component networks.

Acknowledgment

Support from NSF under CCF 1016588 & 1421828 is gratefully acknowledged.

References

[1] X. Yan, R. Yeung, and Z. Zhang, "An implicit characterization of the achievable rate region for acyclic multisource multisink network coding," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5625–5639, Sept 2012.

[2] C. Li, J. M. Walsh, and S. Weber, "Computational approaches for determining rate regions and codes using entropic vector bounds," in *50th Annual Allerton Conference on Communication, Control and Computing*, Oct 2012, pp. 1 –9.

[3] C. Li, J. Apte, J. M. Walsh, and S. Weber, "A new computational approach for determining rate regions and optimal codes for coded networks," in *IEEE International Symposium on Network Coding*, Jun 2013, pp. 1 –6.

[4] J. Apte, C. Li, and J. Walsh, "Algorithms for computing network coding rate regions via single element extensions of matroids," in *2014 IEEE International Symposium on Information Theory (ISIT)*, June 2014.

[5] C. Li, S. Weber, and J. M. Walsh, "Multilevel diversity coding systems: Rate regions, codes, computation, & forbidden minors," *CoRR*, vol. abs/1407.5659, 2014.

[6] C. Li, S. Weber, and J. Walsh, "Network embedding operations preserving the insufficiency of linear network codes," in *52nd Annual Allerton Conference on Communication, Control, and Computing, 2014*, Oct 2014.

[7] C. Li, S. Weber, and J. MacLaren Walsh, "On Multi-source Networks: Enumeration, Rate Region Computation, and Hierarchy," *ArXiv e-prints*, Jul. 2015, available at http://arxiv.org/abs/1507.05728.

[8] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen, "Directed hypergraphs and applications," *Discrete applied mathematics*, vol. 42, no. 2, pp. 177–201, 1993.

[9] C. Li, J. M. Walsh, and S. Weber, "Software for computing bounds on entropic vectors region and network rate region," available at http://www.ece.drexel.edu/walsh/aspitrg/software.html.