Optimization-Based Linear Network Coding for General Connections of Continuous Flows

Ying Cui
Dept. of EE
Shanghai Jiao Tong University

Muriel Médard
Dept. of EECS
MIT

Edmund Yeh
Dept. of ECE
Northeastern University

Douglas Leith
Dept. of CS
Trinity College Dublin

Ken Duffy
Hamilton Institute
NUI Maynooth

***Abstract*— For general connections, the problem of finding network codes and optimizing resources for those codes is intrinsically difficult and little is known about its complexity. Most of the existing solutions rely on very restricted classes of network codes in terms of the number of flows allowed to be coded together, and are not entirely distributed. In this paper, we consider a new method for constructing linear network codes for general connections of continuous flows to minimize the total cost of edge use based on mixing. We first formulate the minimum-cost network coding design problem. To solve the optimization problem, we propose two equivalent alternative formulations with discrete mixing and continuous mixing, respectively, and develop distributed algorithms to solve them. Our approach allows fairly general coding across flows and guarantees no greater cost than any solution without inter-flow network coding.**

I. Introduction

In the case of general connections (where each destination can request information from any subset of sources), the problem of finding network codes is intrinsically difficult. Little is known about its complexity and its decidability remains unknown. In certain special cases, such as multicast connections (where destinations share all of their demands), it is sufficient to satisfy a Ford-Fulkerson type of min-cut max-flow constraint between all sources to every destination individually. For multicast connections, linear codes are sufficient [1], [2] and a distributed random construction exists [3]. In the literature, linear codes have been the most widely considered. However, in general, linear codes over finite fields may not be sufficient for general connections [4]. In addition, even when we consider simple scalar network codes (with scalar coding coefficients), the problem of code construction for general connections (i.e., neither multicast nor its variations) remains vexing [5]. The main difficulty lies in canceling the effect of flows that are coded together but not destined for a common destination.

The problem of code construction becomes more involved when we seek to limit the use of network links for reasons of network resource management. In the case of multicast connections of continuous flows, it is known that finding a minimum-cost solution for convex cost functions of flows over edges of the network is a convex optimization problem and can be solved distributively using convex decomposition [6]. In the case of general connections of continuous flows, however, network resource minimization, even when considering only restricted code constructions, appears to be difficult.

In general, there are two types of coding approaches for optimizing network use for general connections. The first type of coding is mixing, which consists of coding together flows from sources using the random linear distributed code construction of [3] (originally proposed for multicast connections), as though the flows were parts of a common multicast connection. In this case, no explicit code coefficients are provided and decodability is ensured with high probability by the random coding, given that mixing is properly designed. For example, in [7], a two-step mixing approach is proposed for network resource minimization of general connections, where flow partition and flow rate optimization are considered separately. In [8], we introduce linear network mixing coefficients and present a new method for constructing linear network codes for general connections of integer flows to minimize the total cost of edge use. The minimum-cost network coding design problem in [8] is a discrete optimization problem, which jointly considers mixing and flow optimization. The second type of coding is an explicit linear code construction, where one provides specific linear coefficients, to be applied to flows at different nodes, over some finite field. In this case, the explicit linear code constructions are usually simplified by restricting them to be binary, generally in the context of coding flows together only pairwise. For example, in [9] and [10], simple two-flow combinations are proposed for network resource minimization of general connections.

The flow rate optimization in [7], the joint mixing and flow optimization in [8], and the joint two-flow coding and flow optimization in [9], [10] can be solved distributively. However, the separation of flow partition and flow rate optimization in [7] and the pairwise coding in [9], [10] lead in general to feasibility region reduction and network cost increase. In [8], we do not allow flow splitting and coding over time, leading to coded symbols flowing through each edge of the network at an integer rate. The restriction of integer flow rates affects the network

Y. Cui was supported in part by the National Science Foundation of China grant 61401272. M. Médard was supported by the Air Force Office of Scientific Research (AFOSR) grant FA9550-13-1-0023. E. Yeh was supported by the National Science Foundation grant CNS-1423250.

cost reduction.

In this paper, we consider a new method for constructing linear network codes to minimize the total cost of edge use for satisfying general connections of continuous flows. We generalize the linear network mixing coefficients introduced in [8]. In contrast to [8], we allow flow splitting and coding over time to further reduce network cost. Using mixing with generalized mixing coefficients, we formulate the minimum-cost network coding design problem, which is an instance of mixed discrete-continuous programming. Our mixing-based formulation allows for fairly general coding across flows, offers a tradeoff between performance and computational complexity via tuning a design parameter controlling the mixing effect, and guarantees no greater cost than any solution without inter-flow network coding. To solve the mixed discrete-continuous optimization problem, we propose two equivalent alternative formulations with discrete mixing and continuous mixing, respectively, and develop distributed algorithms to solve them. Specifically, the distributed algorithm for the discrete mixing formulation is obtained by relating the optimization problem to a constraint satisfaction problem (CSP) in discrete optimization and applying recent results in the domain [11]. The distributed algorithm for the continuous mixing formulation is based on penalty methods for nonlinear programming [12] in continuous optimization.

II. Network Model and Definitions

In this section, we first illustrate the network model for general connections of continuous flows. The model is similar to that we considered in [8] for integer flows, except that here we consider general flow rates and edge capacities, and allow flow splitting and coding over time. Next, to facilitate the understanding of the formulations proposed in Sections III, V, and IV, we also briefly illustrate the formal relationship between linear network coding and mixing established in [8].

A. Network Model

We consider a directed acyclic network with general connections. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the directed acyclic graph, where $\mathcal{V}$ denotes the set of $V = |\mathcal{V}|$ nodes and $\mathcal{E}$ denotes the set of $E = |\mathcal{E}|$ edges. To simplify notation, we assume there is only one edge from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$, denoted as edge $(i,j) \in \mathcal{E}$.[1] For each node $i \in \mathcal{V}$, define the set of incoming neighbors to be $\mathcal{I}_i = \{j : (j,i) \in \mathcal{E}\}$ and the set of outgoing neighbors to be $\mathcal{O}_i = \{j : (i,j) \in \mathcal{E}\}$. Let $I_i = |\mathcal{I}_i|$ and $O_i = |\mathcal{O}_i|$ denote the in degree and out degree of node $i \in \mathcal{V}$, respectively. Assume $I_i \le D$ and $O_i \le D$ for all $i \in \mathcal{V}$. Let $\mathcal{P} = \{1, \cdots, P\}$ denote the set of $P = |\mathcal{P}|$ flows to be carried by the network. For each flow $p \in \mathcal{P}$, let $s_p \in \mathcal{V}$ be its source. We consider continuous flows. Let $R_p \in \mathbb{R}^+$ denote the source rate for source p, where $\mathbb{R}^+$ denotes the set of non-negative real numbers. Let $\mathcal{S} = \{s_1, \cdots, s_P\}$ denote the set of $P = |\mathcal{S}|$ sources. To simplify notation, we assume different flows do not share a common source node and no source node has any incoming edges. Let $\mathcal{T} = \{t_1, \cdots, t_T\}$ denote the set of $T = |\mathcal{T}|$ terminals. Each terminal $t \in \mathcal{T}$ demands a subset of $P_t = |\mathcal{P}_t|$ flows $\mathcal{P}_t \subseteq \mathcal{P}$. Assume each flow is requested by at least one terminal, i.e., $\cup_{t\in\mathcal{T}} \mathcal{P}_t = \mathcal{P}$. To simplify notation, we assume no terminal has any outgoing edges.

Let $B_{ij} \in \mathbb{R}^+$ denote the edge capacity for edge (i,j). Let $z_{ij} \in [0, B_{ij}]$ denote the transmission rate through edge (i,j). We assume a cost is incurred on an edge when information is transmitted through the edge. Let $U_{ij}(z_{ij})$ denote the cost function incurred on edge (i,j) when the transmission rate through edge (i,j) is z_{ij}. Assume $U_{ij}(z_{ij})$ is convex, non-decreasing, and twice continuously differentiable in z_{ij}. We are interested in the problem of finding linear network coding designs and minimizing the network cost $\sum_{(i,j)\in\mathcal{E}} U_{ij}(z_{ij})$ of general connections of continuous flows for those designs.

B. Scalar Time-Invariant Linear Network Coding and Mixing

For ease of exposition, in this section, we illustrate linear network coding and mixing by considering unit flow rate, unit edge capacity and one (coded) symbol transmission for each edge per unit time, and adopt scalar time-invariant notation. Later, in Sections III, V, and IV, we shall consider general flow rates and edge capacities and allow flow splitting and coding over time, which enable multiple (coded) symbols to flow through each edge at a continuous rate.

Consider a finite field $\mathcal{F}$ with size $F = |\mathcal{F}|$. In linear network coding, a linear combination over $\mathcal{F}$ of the symbols in $\{\sigma_{ki} \in \mathcal{F} : k \in \mathcal{I}_i\}$ from the incoming edges $\{(k,i) : k \in \mathcal{I}_i\}$, i.e., $\sigma_{ij} = \sum_{k\in\mathcal{I}_i} \alpha_{kij}\sigma_{ki}$, can be transmitted through the shared edge (i,j), where coefficient $\alpha_{kij} \in \mathcal{F}$ is referred to as the local coding coefficient corresponding to edge $(k,i) \in \mathcal{E}$ and edge $(i,j) \in \mathcal{E}$. On the other hand, the symbol of edge $(i,j) \in \mathcal{E}$ can be expressed as a linear combination over $\mathcal{F}$ of the source symbols $\{\sigma_p \in \mathcal{F} : p \in \mathcal{P}\}$, i.e., $\sigma_{ij} = \sum_{p\in\mathcal{P}} c_{ij,p}\sigma_p$, where coefficient $c_{ij,p} \in \mathcal{F}$ is referred to as the global coding coefficient of flow $p \in \mathcal{P}$ and edge $(i,j) \in \mathcal{E}$. Let $\mathbf{c}_{ij} = (c_{ij,1}, \cdots, c_{ij,p}, \cdots, c_{ij,P}) \in \mathcal{F}^P$ denote the P coefficients corresponding to this linear combination for edge $(i,j) \in \mathcal{E}$, referred to as the global coding vector of edge $(i,j) \in \mathcal{E}$. Note that, we consider scalar time-invariant linear network coding. In other words, $\alpha_{kij} \in \mathcal{F}$ and $c_{ij,p} \in \mathcal{F}$ are both scalars, and α_{kij} and $c_{ij,p}$ do not change over time. When using scalar linear network coding, for each terminal, extraneous flows are allowed to be mixed with the desired flows on the paths to the terminal, as the extraneous flows can be cancelled at intermediate nodes or at the terminal.

In many cases, we shall see that the specific values of the local or global coding coefficients are not required in our design. For this purpose, we introduce the mixing concept

[1] Multiple edges from node i to node j can be modeled by introducing multiple extra nodes, one on each edge.

based on local and global mixing coefficients established in [8]. Specifically, we consider the local mixing coefficient $\beta_{kij} \in \{0,1\}$ corresponding to edge $(k,i) \in \mathcal{E}$ and edge $(i,j) \in \mathcal{E}$, which relates to the local coding coefficient $\alpha_{kij} \in \mathcal{F}$ as follows. $\beta_{kij} = 1$ indicates that symbol σ_{ki} of edge $(k,i) \in \mathcal{E}$ is allowed to contribute to the linear combination over $\mathcal{F}$ forming symbol σ_{ij} and $\beta_{kij} = 0$ otherwise. Thus, if $\beta_{kij} = 0$, we have $\alpha_{kij} = 0$ (note that α_{kij} can be zero when $\beta_{kij} = 1$). Similarly, we consider the global mixing coefficient $x_{ij,p} \in \{0,1\}$ of flow $p \in \mathcal{P}$ and edge $(i,j) \in \mathcal{E}$, which relates to the global coding coefficient $c_{ij,p} \in \mathcal{F}$ as follows. $x_{ij,p} = 1$ indicates that flow p is allowed to be mixed (coded) with other flows, i.e., symbol σ_p is allowed to contribute to the linear combination over $\mathcal{F}$ forming symbol σ_{ij}, and $x_{ij,p} = 0$ otherwise. Thus, if $x_{ij,p} = 0$, we have $c_{ij,p} = 0$ (note that $c_{ij,p}$ can be zero when $x_{ij,p} = 1$). Then, we introduce the global mixing vector $\mathbf{x}_{ij} = (x_{ij,1}, \cdots, x_{ij,p}, \cdots, x_{ij,P}) \in \{0,1\}^P$ for edge $(i,j) \in \mathcal{E}$, which relates to the global coding vector $\mathbf{c}_{ij} = (c_{ij,1}, \cdots, c_{ij,p}, \cdots, c_{ij,P}) \in \mathcal{F}^P$. Similarly, we consider scalar time-invariant linear network mixing. That is, $\beta_{kij} \in \{0,1\}$ and $x_{ij,p} \in \{0,1\}$ are both scalars, and β_{kij} and $x_{ij,p}$ do not change over time.

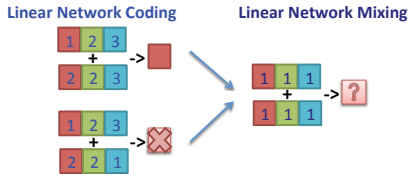

Fig. 1: Comparisons between linear network coding and linear network mixing. Given the two global coding vectors on the left, we can tell that the (red) symbol can be decoded in the first case and cannot be decoded in the second case. However, given the two global mixing vectors on the right (same for the two cases on the left), it is not sufficient to tell whether the (red) symbol can be decoded or not.

Global mixing vectors provide a natural way of speaking of flows as possibly coded or not coded without knowledge of the specific values of the global coding vectors. Intuitively, global mixing vectors can be regarded as a limited representation of global coding vectors. Network mixing vectors may not be sufficient for telling whether a certain symbol can be decoded or not, as illustrated in Fig. 1. Therefore, using the network mixing representation, extraneous flows which are mixed with the desired flows on the paths to each terminal, are not guaranteed to be cancelled at the terminal. Let $\mathbf{e}_p$ denote the vector with the p-th element being 1 and all the other elements being 0. Let $\vee$ denote the "or" operator (logical disjunction). We now define the feasibility for scalar linear network mixing.

Definition 1 (Feasibility of Scalar Linear Network Mixing): For a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of flows $\mathcal{P}$ with sources $\mathcal{S}$ and terminals $\mathcal{T}$, a linear network mixing design $\{\mathbf{x}_{ij} \in \{0,1\}^P : (i,j) \in \mathcal{E}\}$ is called feasible if the following three conditions are satisfied: 1) $\mathbf{x}_{s_p j} = \mathbf{e}_p$ for source edge $(s_p, j) \in \mathcal{E}$, where $s_p \in \mathcal{S}$ and $p \in \mathcal{P}$; 2) $\mathbf{x}_{ij} = \vee_{k \in \mathcal{I}_i} \beta_{kij} \mathbf{x}_{ki}$ for edge $(i,j) \in \mathcal{E}$ not outgoing from a source, where $i \notin \mathcal{S}$ and $\beta_{kij} \in \{0,1\}$; 3) $\vee_{i \in \mathcal{I}_t} x_{it,p} = 1$ for all $p \in \mathcal{P}_t$ and $x_{it,p} = 0$ for all $i \in \mathcal{I}_t$ and $p \notin \mathcal{P}_t$, where $t \in \mathcal{T}$.

Note that Condition 3) in Definition 1 ensures that for each terminal, the extraneous flows are not mixed with the desired flows on the paths to the terminal. In other words, using linear network mixing, only mixing is allowed at intermediate nodes. This is not as general as using linear network coding, which allows both mixing and canceling (i.e., removing one or multiple flows from a mixing of flows) at intermediate nodes.

Given a feasible linear network mixing design (specified by $\{\beta_{kij} \in \{0,1\} : (k,i), (i,j) \in \mathcal{E}\}$), one way to accomplish mixing when $\mathcal{F}$ is large is to use random linear network coding [3] (to obtain $\{\alpha_{kij} \in \mathcal{F} : (k,i), (i,j) \in \mathcal{E}\}$), as discussed in the introduction. Note that, in performing random linear network coding based on β_{kij}, α_{kij} can be randomly chosen in $\mathcal{F}$ when $\beta_{kij} = 1$, but α_{kij} must be chosen to be 0 when $\beta_{kij} = 0$.

III. Continuous Flows with Mixing Only

In this section, we consider the minimum-cost scalar time-invariant linear network coding design problem for general connections of continuous flows with mixing only. Starting from this section, we consider multiple global mixing vectors (each may correspond to multiple global coding vectors) for each edge and allow coded symbols to flow through each edge at a continuous rate.

A. Design Parameter

We consider multiple global mixing vectors for each edge. This generalizes the linear network mixing coefficients introduced in [8]. We refer to the number of global network mixing vectors for each edge as the mixing parameter, denoted as $L \in \{1, \cdots, L_{\max}\}$, where $L_{\max}$ is the maximum number of global network mixing vectors necessary for decodability using mixing (cf. Definition 1), and is given as follows. Let $\boldsymbol{\mathcal{Y}}$ denote the set of atoms of the algebra generated by $\{\mathcal{P}_t : t \in \mathcal{T}\}$, i.e., $\boldsymbol{\mathcal{Y}} \triangleq \{\cap_{t \in \mathcal{T}} \mathcal{Y}_t : \mathcal{Y}_t = \mathcal{P}_t \text{ or } \mathcal{Y}_t = \mathcal{P} - \mathcal{P}_t\} - \{\emptyset\}$. In other words, $\boldsymbol{\mathcal{Y}}$ gives a set partition of $\mathcal{P}$ that represents the flows that can be mixed (cf. Definition 1) over an edge in the worst case (i.e., all terminals obtaining flows through the same edge). We choose $L_{\max} = |\boldsymbol{\mathcal{Y}}|$. Note that $1 \leq L_{\max} \leq P$, where $L_{\max} = 1$ for the multicast case, i.e., $\mathcal{P}_t = \mathcal{P}$ for all $t \in \mathcal{T}$, and $L_{\max} = P$ for the unicast case, i.e., $\mathcal{P}_{t'} \cap \mathcal{P}_t = \emptyset$ for all $t \neq t'$ and $t, t' \in \mathcal{T}$. Fig. 2 illustrates an example of flow partition $\boldsymbol{\mathcal{Y}}$ and mixing parameter L for the general case.

For a given mixing parameter L, we now introduce the global and local network mixing vectors. For each $l = 1, \cdots, L$, let $\mathbf{x}_{ij}(l) = (x_{ij,1}(l), \cdots, x_{ij,p}(l), \cdots, x_{ij,P}(l)) \in \{0,1\}^P$ denote the l-th global network mixing vector over edge $(i,j) \in$

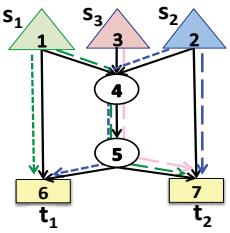

Fig. 2: Illustration of a feasible solution to Problem 1. $\mathcal{P}=\{1,2,3\}$, $\mathcal{S}=\{s_1,s_2,s_3\}$, $R_1=R_2=R_3=1$, $B_{ij}=2$ for all $(i,j)\in\mathcal{E}$, $\mathcal{T}=\{t_1,t_2\}$, $\mathcal{P}_1=\{1,2\}$ and $\mathcal{P}_2=\{1,2,3\}$. Thus, $\boldsymbol{\mathcal{Y}}=\{\{1,2\},\{3\}\}$, $L_{\max}=|\boldsymbol{\mathcal{Y}}|=2$ and $L\in\{1,2\}$.

$\mathcal{E}$. Let $\boldsymbol{\beta}_{kij}(l,m)\in\{0,1\}$ denote the local mixing coefficient corresponding to the l-th global network mixing vector of edge $(k,i)\in\mathcal{E}$ (i.e., $\mathbf{x}_{ki}(l)$) and the m-th global network mixing vector of edge $(i,j)\in\mathcal{E}$ (i.e., $\mathbf{x}_{ij}(m)$), where $l,m=1,\cdots,L$.

B. Problem Formulation

We would like to find the minimum-cost scalar time-invariant linear network coding design with design parameter $L\in\{1,\cdots,L_{\max}\}$ for general connections of continuous flows with mixing only.

Problem 1 (Continuous Flows with Mixing Only):

$$U_x^*(L)=\min_{\substack{\{z_{ij}\},\{z_{ij}(l)\}\{f^t_{ij,p}(l)\}\\ \{x_{ij,p}(l)\},\{\beta_{kij}(l,m)\}}}\sum_{(i,j)\in\mathcal{E}}U_{ij}(z_{ij})$$

$$s.t.\ 0\le z_{ij}\le B_{ij},\ (i,j)\in\mathcal{E} \tag{1}$$

$$x_{ij,p}(l)\in\{0,1\},\ l=1,\cdots,L,\ (i,j)\in\mathcal{E},\ p\in\mathcal{P} \tag{2}$$

$$\beta_{kij}(l,m)\in\{0,1\},\ l,m=1,\cdots,L,\ (k,i),(i,j)\in\mathcal{E} \tag{3}$$

$$f^t_{ij,p}(l)\ge 0,\ l=1,\cdots,L,\ (i,j)\in\mathcal{E},\ p\in\mathcal{P}_t,\ t\in\mathcal{T} \tag{4}$$

$$\sum_{p\in\mathcal{P}_t}f^t_{ij,p}(l)\le z_{ij}(l),\ t\in\mathcal{T},\ l=1,\cdots,L,\ (i,j)\in\mathcal{E} \tag{5}$$

$$\sum_{l=1}^{L}z_{ij}(l)\le z_{ij},\ (i,j)\in\mathcal{E} \tag{6}$$

$$\sum_{k\in\mathcal{O}_i}\sum_{l=1}^{L}f^t_{ik,p}(l)-\sum_{k\in\mathcal{I}_i}\sum_{l=1}^{L}f^t_{ki,p}(l)=\sigma^t_{i,p},\quad i\in\mathcal{V},\ p\in\mathcal{P}_t,\ t\in\mathcal{T} \tag{7}$$

$$f^t_{ij,p}(l)\le x_{ij,p}(l)B_{ij},\ l=1,\cdots,L,\ (i,j)\in\mathcal{E},\quad p\in\mathcal{P}_t,\ t\in\mathcal{T} \tag{8}$$

$$\mathbf{x}_{s_pj}(l)=\mathbf{e}_p,\ l=1,\cdots,L,\ (s_p,j)\in\mathcal{E},\ p\in\mathcal{P} \tag{9}$$

$$\mathbf{x}_{ij}(l)=\vee_{k\in\mathcal{I}_i,m=1,\cdots,L}\beta_{kij}(m,l)\mathbf{x}_{ki}(m),\quad l=1,\cdots,L,\ (i,j)\in\mathcal{E},\ i\notin\mathcal{S} \tag{10}$$

$$x_{it,p}(l)=0,\ l=1,\cdots,L,\ i\in\mathcal{I}_t,\ p\notin\mathcal{P}_t,\ t\in\mathcal{T} \tag{11}$$

where

$$\sigma^t_{i,p}=\begin{cases}R_p, & i=s_p\\ -R_p, & i=t\\ 0, & \text{otherwise}\end{cases}\qquad i\in\mathcal{V},\ p\in\mathcal{P}_t,\ t\in\mathcal{T}. \tag{12}$$

In the above formulation,[2] $f^t_{ij,p}(l)\ge 0$ can be interpreted as the rate of delivering flow $p\in\mathcal{P}_t$ to terminal $t\in\mathcal{T}$ over edge $(i,j)\in\mathcal{E}$ using $\mathbf{x}_{ij}(l)$, and $z_{ij}(l)$ denotes the transmission rate corresponding to $\mathbf{x}_{ij}(l)$ over edge $(i,j)\in\mathcal{E}$, where $l=1,\cdots,L$. Problem 1 is a mixed discrete-continuous optimization problem and is NP-complete in general. For notational simplicity, in this paper, we omit the conditions $\{z_{ij}:(i,j)\in\mathcal{E}\}$, $\{z_{ij}(l):l=1,\cdots,L,(i,j)\in\mathcal{E}\}$, $\{f^t_{ij,p}(l):l=1,\cdots,L,(i,j)\in\mathcal{E},p\in\mathcal{P}_t,t\in\mathcal{T}\}$, $\{x_{ij,p}(l):l=1,\cdots,L,(i,j)\in\mathcal{E},p\in\mathcal{P}\}$, and $\{\beta_{kij}(l,m):l,m=1,\cdots,L,(k,i),(i,j)\in\mathcal{E}\}$ where there is no confusion.

Remark 1 (Problem 1 with $L=1$ for Multicast): For the multicast case (i.e., $\mathcal{P}_t=P$ for all $t\in\mathcal{T}$) and $L=1$, the constraint in (11) does not exist, the constraint in (6) can be satisfied by choosing $z_{ij}(1)=z_{ij}$, and the constraint in (8) is always satisfied by choosing $\beta_{kij}(1,1)=1$ and choosing $x_{ij,p}(1)$ according to (9) and (10). Therefore, in the multicast case, Problem 1 with $L=1$ for general connections reduces to the conventional minimum-cost network coding design problem for the multicast case [6].

Remark 2 (Comparison with Intra-flow Coding): Problem 1 (with any $L\in\{1,\cdots,L_{\max}\}$) with an extra constraint $\sum_{p\in\mathcal{P}}x_{ij,p}(l)\in\{0,1\}$ for all $(i,j)\in\mathcal{E}$ and $l=1,\cdots,L$ is equivalent to a minimum-cost intra-flow coding problem. Thus, the minimum network cost of Problem 1 (with any $L\in\{1,\cdots,L_{\max}\}$) is no greater than the minimum costs for intra-flow coding.

Remark 3 (Comparison with Two-step Mixing): Problem 1 with $L=L_{\max}$ and $\beta_{kij}(l,m)=1$ instead of (3), is equivalent to the minimum-cost flow rate control problem in the second step of the two-step mixing approach in [7]. Thus, the minimum network cost of Problem 1 with $L=L_{\max}$ is no greater than the minimum cost of the two-step mixing approach in [7].

Example 1 (Illustration of Linear Network Mixing): We illustrate a feasible mixing design (corresponding to a feasible solution) to Problem 1 with $L=2$ for the example in Fig. 2. For ease of illustration, in this example, we consider unit source rate and do not consider flow splitting and coding over time. For source edges (1,6), (1,4), (2,7), (2,4) and (3,4), choose the global mixing vectors as follows: $\mathbf{x}_{16}(l)=\mathbf{x}_{14}(l)=(1,0,0)$, $\mathbf{x}_{24}(l)=\mathbf{x}_{27}(l)=(0,1,0)$ and $\mathbf{x}_{34}(l)=(0,0,1)$ for all $l=1,2$. In addition, choose the local coding coefficients as follows: $\beta_{145}(1,1)=\beta_{245}(1,1)=\beta_{345}(1,2)=1$, $\beta_{145}(2,1)=\beta_{245}(2,1)=\beta_{345}(2,2)=0$, $\beta_{145}(m,2)=$

[2]Note that (2) with $j=t$, (7) with $i=t$, and (8) with $j=t$ imply $\vee_{i\in\mathcal{I}_t,l=1,\cdots,L}x_{it,p}(l)=1$ for all $p\in\mathcal{P}_t$, i.e., Condition 3) of Definition 1, where $t\in\mathcal{T}$.

$\beta_{245}(m,2)=\beta_{345}(m,1)=0$ for all $m=1,2$, $\beta_{456}(1,1)=1$, $\beta_{456}(2,1)=\beta_{456}(1,2)=\beta_{456}(2,2)=0$, $\beta_{457}(1,1)=\beta_{457}(2,2)=1$ and $\beta_{457}(1,2)=\beta_{457}(2,1)=0$. Therefore, for edges (4,5), (5,6) and (5,7) not outgoing from a source, the global mixing vectors are given by $\mathbf{x}_{45}(1)=(1,1,0)$, $\mathbf{x}_{45}(2)=(0,0,1)$, $\mathbf{x}_{56}(1)=(1,1,0)$, $\mathbf{x}_{56}(2)=(0,0,0)$, $\mathbf{x}_{57}(1)=(1,1,0)$ and $\mathbf{x}_{57}(2)=(0,0,1)$. On the other hand, flow paths (sets of edge-mixing index pairs $((i,j),l)$ for which the rates of delivering flows are one) from the three sources, i.e., $\{((i,j),l): f^t_{ij,p}(l)=1,\ (i,j)\in\mathcal{E},\ l=1,\cdots,L\}$ for all $p\in\mathcal{P}_t$ and $t\in\mathcal{T}$, are illustrated using green, blue and pink curves in Fig. 2. Accordingly, choose the transmission rates as follows: $z_{ij}(1)=1$ and $z_{ij}(2)=0$ for all $(i,j)=(1,6),(1,4),(2,7),(2,4),(3,4)$, $z_{45}(1)=z_{45}(2)=z_{56}(1)=z_{57}(1)=z_{57}(2)=1$, $z_{56}(2)=0$, and $z_{ij}=z_{ij}(1)+z_{ij}(2)$ for all $(i,j)\in\mathcal{E}$.

The following lemma shows the existence of a feasible linear network code corresponding to Problem 1.

Lemma 1: Suppose Problem 1 is feasible. Then, for each feasible solution, there exists a feasible linear network code with a field size $F>T$ to deliver the desired flows to each terminal.

Proof: Please refer to Appendix A. ■

Note that a feasible linear network code can be obtained from a feasible linear network mixing design (a feasible solution to Problem 1) using random linear network coding [3], as illustrated in Section II-B.

Example 2 (Illustration of Linear Network Coding): We illustrate how to obtain a feasible linear network code using random linear network coding, based on the feasible linear network mixing design illustrated in Example 1. In this example, one local mixing coefficient (global mixing vector) corresponds to one local coding coefficient (global coding vector).[3] For the source edges, choose the global coding vectors as follows: $\mathbf{c}_{ij}(l)=\mathbf{x}_{ij}(l)$ for all $(i,j)=(1,6),(1,4),(2,7),(2,4),(3,4)$ and $l=1,2$. For all $l,m=1,\cdots,L$ and $(k,i),(i,j)\in\mathcal{E}$, if $\beta_{kij}(l,m)=0$, choose $\alpha_{kij}(l,m)=0$; if $\beta_{kij}(l,m)=1$, choose $\alpha_{kij}(l,m)$ uniformly at random from $\mathcal{F}$. Therefore, for the edges not outgoing from a source, the global coding vectors are given by $\mathbf{c}_{ij}(l)=\sum_{k\in\mathcal{I}_i,m=1,\cdots,L}\alpha_{kij}(m,l)\mathbf{c}_{ki}(m)$ for all $(i,j)=(4,5),(5,6),(5,7)$ and $l=1,\cdots,L$.

C. Network Cost and Complexity Tradeoff

The design parameter L in Problem 1 determines the complexity and network cost tradeoff. First, we illustrate the impact of L on the complexity of Problem 1. By (3), we know that for given $(k,i),(i,j)\in\mathcal{E}$, the cardinality of $\{\beta_{kij}(l,m): l,m=1,\cdots,L\}$ is L^2. Since $\sum_{(i,j)\in\mathcal{E}}O_j=\sum_{j\in\mathcal{V}}I_jO_j\leq\sum_{j\in\mathcal{V}}DO_j=DE$, the cardinality of $\{\beta_{kij}(l,m): l,m=1,\cdots,L,\ (k,i),(i,j)\in\mathcal{E}\}$ is smaller than or equal to L^2DE. Note that by (9) and (10), $\{x_{ij,p}(l)\}$ can be fully determined by $\{\beta_{kij}(l,m)\}$. Therefore, the cardinality of the discrete variables $\{x_{ij,p}(l)\}$ and $\{\beta_{kij}(l,m)\}$ of Problem 1 is L^2DE, which increases as L increases.

Next, we discuss the impact of L on the network cost.

Lemma 2: If Problem 1 is feasible for design parameter L, then Problem 1 is feasible for design parameter $L+1$ and $U_x^*(L+1)\leq U_x^*(L)$, where $L=1,\cdots,L_{\max}-1$.

Proof: Given a feasible solution to Problem 1 with design parameter L, by setting variables w.r.t. index $l=L+1$ or $m=L+1$ to be zero, we can easily construct a feasible solution to Problem 1 with design parameter $L+1$. This feasible solution corresponds to the same network cost as the one with design parameter L. But the network cost with design parameter $L+1$ can be further optimized by solving Problem 1 with design parameter $L+1$. Therefore, we can show $U_x^*(L+1)\leq U_x^*(L)$ for all $L=1,\cdots,L_{\max}-1$. ■

By Lemma 2, we know that the network cost $U_x^*(L)$ is non-increasing w.r.t. L. This can also be understood from the example in Fig. 2. Note that by Condition 3) in Definition 1, flow 3 is not allowed to be mixed with flow 1 and flow 2 on their paths to terminal t_1. When $L=1<L_{\max}$, flow 3 cannot be delivered over edge $(4,5)$ to terminal t_2 using feasible mixing. In other words, Problem 1 with $L=1$ is not feasible (i.e., of infinite network cost). However, when $L=2=L_{\max}$, flow 3 can be delivered to terminal t_2 without mixing with flow 1 and flow 2 over edge $(4,5)$, e.g., using global mixing vectors $\mathbf{x}_{45}(1)=(1,1,0)$ and $\mathbf{x}_{45}(2)=(0,0,1)$ over edge $(4,5)$. In other words, Problem 1 with $L=2$ is feasible (i.e., of finite network cost). Thus, we can see the impact of L on the network cost shown in Lemma 2.

IV. Alternative Formulation with Discrete Mixing

Problem 1 is a mixed discrete-continuous optimization problem with two main challenges. One is the choice of the network mixing coefficients (discrete variables), and the other is the choice of the flow rates (continuous variables). In this section, we first propose an equivalent alternative formulation of Problem 1 which naturally subdivides Problem 1 according to these two aspects. Then, we propose a distributed algorithm to solve it.

A. Alternative Formulation

Problem 1 is equivalent to the following problem.

Problem 2 (Equivalent Problem of Problem 1):

$$U_x^*(L)=\min_{\{x_{ij,p}(l)\}\in\boldsymbol{\mathcal{M}}(L)}\ U_x^*(\{x_{ij,p}(l)\})$$

where $U_x^*(\{x_{ij,p}(l)\})$ and $\boldsymbol{\mathcal{M}}(L)$ are given by the following two subproblems.

[3]Note that when flow splitting or coding over time happens, one local mixing coefficient (global mixing vector) may correspond to multiple local coding coefficients (global coding vectors). In this case, a linear network code can be designed in a similar way based on the sub-flows and sub-edges established in the proof of Lemma 1.

Subproblem 1 (Subproblem of Problem 2: Flow Optimization): For given $\{x_{ij,p}(l)\}$, we have:

$$U_x^*(\{x_{ij,p}(l)\}) = \min_{\{z_{ij}\},\{z_{ij}(l)\},\{f_{ij,p}^t(l)\}} \sum_{(i,j)\in\mathcal{E}} U_{ij}(z_{ij})$$
$$s.t. \quad (1),(4),(5),(6),(7),(8)$$

Subproblem 2 (Subproblem of Problem 2: Feasible Mixing): Find the set $\boldsymbol{\mathcal{M}}(L) \triangleq \{\{x_{ij,p}(l)\} : (2),(3),(9),(10),(11),(13)\}$ of feasible $\{x_{ij,p}(l)\}$, where (13) is given by:

$$\vee_{i\in\mathcal{I}_t, l=1,\cdots,L} x_{it,p}(l) = 1,\ p\in\mathcal{P}_t,\ t\in\mathcal{T}. \tag{13}$$

Note that for given $\{x_{ij,p}(l)\}$, Subproblem 1 is a convex optimization problem (involving continuous flow rates) and hence has polynomial-time complexity. On the other hand, Subproblem 2 is a discrete feasibility problem (involving discrete mixing coefficients) and is NP-complete in general. Therefore, Problem 2 is still a mixed discrete-continuous optimization problem and is NP-complete in general.

B. Distributed Solution

In this part, we develop a distributed algorithm to solve Problem 2 by solving Subproblem 1 and Subproblem 2, respectively, in a distributed manner. First, we consider Subproblem 2. Subproblem 2 can be treated as a CSP and solved distributively using clause partition and the Communication-Free Learning (CFL) algorithm from [11]. While CSPs are NP-complete in general, CFL provides a probabilistic distributed iterative algorithm with almost sure convergence in finite time. Specifically, $\{x_{ij,p}(l)\} \cup \{\beta_{kij}(l,m)\}$ can be treated as the variables of the CSP. $\{0,1\}$ can be treated as the finite set of the CSP. From (10), we have an equivalent constraint purely on $\{x_{ij,p}(l)\}$, i.e.,

$$\begin{aligned}&\exists\ \beta_{kij}(m,l)\in\{0,1\}\ \forall k\in\mathcal{I}_i, m=1,\cdots,L,\\ &\text{s.t. } \mathbf{x}_{ij}(l) = \vee_{k\in\mathcal{I}_i, m=1,\cdots,L}\beta_{kij}(m,l)\mathbf{x}_{ki}(m),\\ &\qquad l=1,\cdots,L,\ (i,j)\in\mathcal{E},\ i\notin\mathcal{S}.\end{aligned} \tag{14}$$

In the following, we shall only consider solving for the variables $\{x_{ij,p}(l)\}$ of the CSP in a distributed way using clause partition and CFL, as $\{\beta_{kij}(l,m)\}$ can be obtained from feasible $\{x_{ij,p}(l)\}$ by (9) and (10). In addition, we directly choose $\mathbf{x}_{s_pj}(l) = \mathbf{e}_p$ for all $l = 1,\cdots,L$, $(s_p,j)\in\mathcal{E}$ and $p\in\mathcal{P}$ according to (9).

For notational simplicity, we write the clauses for $\{x_{ij,p}(l)\}$ in a more compact form as follows:

$$\begin{aligned}&\phi_{ij,p}^{x,l}\Big(\mathbf{x}_{ij}(l), \{\mathbf{x}_{ki}(m) : m=1,\cdots,L, k\in\mathcal{I}_i\},\\ &\qquad \{\mathbf{x}_{kj}(m) : m=1,\cdots,L, k\in\mathcal{I}_j, j\in\mathcal{T}\}\Big)\\ &\triangleq \begin{cases}1, & \text{if } j\notin\mathcal{T}, \text{ (14) holds}\\ 1, & \text{if } j\in\mathcal{T} \text{ and } p\in\mathcal{P}_j, \text{ (14) and (13) hold}\\ 1, & \text{if } j\in\mathcal{T} \text{ and } p\notin\mathcal{P}_j, \text{ (14) and (11) hold}\\ 0, & \text{otherwise}\end{cases}\\ &\quad l=1,\cdots,L,\ (i,j)\in\mathcal{E},\ p\in\mathcal{P},\ i\notin S.\end{aligned} \tag{15}$$

Note that, when $j\notin\mathcal{T}$, $\{\mathbf{x}_{kj}(m) : m=1,\cdots,L, k\in\mathcal{I}_j, j\in\mathcal{T}\} = \emptyset$ and we ignore it in the clause $\phi_{ij,p}^{x,l}(\cdot)$. For (13) and (11) in clause $\phi_{ij,p}^{x,l}(\cdot)$, we use j as the terminal index instead of t. It can be seen that the constraints in (10) (i.e., (14)), (11) and (13) are considered in clause $\phi_{ij,p}^{x,l}(\cdot)$. In addition, the constraint in (9) is considered when choosing $\mathbf{x}_{s_pj}(l) = \mathbf{e}_p$ for all $l = 1,\cdots,L$, $(s_p,j)\in\mathcal{E}$ and $p\in\mathcal{P}$. Therefore, the CSP has considered all the constraints in Subproblem 2.

We now construct the clause partition of Subproblem 2. Specifically, the set of clauses variable $x_{ij,p}(l)$ participates in is as follows:

$$\begin{aligned}\Phi_{ij,p}^{x,l} \triangleq &\left\{\phi_{ij,p}^{x,l}\right\} \cup \left\{\phi_{jk,p}^{x,m} : m=1,\cdots,L, k\in\mathcal{O}_j\right\}\\ &\cup \left\{\phi_{kj,p}^{x,m} : m=1,\cdots,L, k\in\mathcal{I}_j, j\in\mathcal{T}\right\}\\ &l=1,\cdots,L,\ (i,j)\in\mathcal{E},\ p\in\mathcal{P},\ i\notin S.\end{aligned} \tag{16}$$

Note that, when $j\notin\mathcal{T}$, $\left\{\phi_{kj,p}^{x,m} : m=1,\cdots,L, k\in\mathcal{I}_j, j\in\mathcal{T}\right\} = \emptyset$ and we ignore it in $\Phi_{ij,p}^{x,l}$ in (16).

Based on the clause partition, a feasible $\{x_{ij,p}(l)\}\in\boldsymbol{\mathcal{M}}(L)$ to Subproblem 2 can be found distributively using the probabilistic distributed iterative CFL algorithm [11, Algorithm 1]. Specifically, for all $(i,j)\in\mathcal{E}$, $p\in\mathcal{P}$ and $l=1,\cdots,L$, in each iteration, each node i realizes a Bernoulli random variable selecting $x_{ij,p}(l)$; messages on $\{x_{ij,p}(l)\}$ are passed between adjacent nodes for each node i to evaluate its related clauses in (16); based on the evaluation, each node i updates the distribution of the Bernoulli random variable selecting $x_{ij,p}(l)$. Given a feasible $\{x_{ij,p}(l)\}\in\boldsymbol{\mathcal{M}}(L)$ obtained by CFL, Subproblem 1 is convex and can be solved distributively using standard convex decomposition. We omit the details here due to the page limitation.

Now, we can develop a distributed algorithm to solve Problem 2 based on CFL and convex decomposition, as briefly illustrated in Algorithm 1.[4]

[4]In Step 3, CFL is run for a sufficiently long time. Step 4 (Step 6) can be implemented with a master node obtaining the network convergence information of CFL (network cost) from all nodes or with all nodes computing the average convergence indicator of CFL (average network cost) locally via a gossip algorithm.

Algorithm 1 Algorithm for Problem 2

initialize $n = 1$ and $U_1 = +\infty$.
loop
  Run CFL to the CSP corresponding to Subproblem 2.
  if the CFL finds a feasible solution **then**
    For the obtained feasible solution to Subproblem 2, solve Subproblem 1 distributively using convex decomposition. Let $\bar{U}_n$ denote the corresponding network cost.
    if $\bar{U}_n < U_n$ **then**
      set $U_{n+1} = \bar{U}_n$ and $n = n + 1$.
    end if
  end if
end loop

Based on the convergence result of CFL [11, Corollary 2], we can easily see that $U_n \to U_x^*(L)$ almost surely as $n \to \infty$, if Problem 2 is feasible.

V. Alternative Formulation with Continuous Mixing

The complexity of solving Problem 2 mainly lies in solving for the network mixing coefficients (discrete variables) in Subproblem 2. In this section, we first propose an equivalent alternative formulation of Problem 1 (Problem 2) with continuous mixing. Then, we elaborate on some distributed algorithms to solve it.

A. Alternative Formulation

Problem 1 (Problem 2) is a mixed discrete-continuous optimization problem. Applying continuous relaxation to (2) and (3) and manipulating (10), we obtain the following continuous optimization problem.

Problem 3 (Continuous Formulation of Problem 1):

$$\bar{U}_x^*(L) = \min_{\substack{\{z_{ij}\},\{z_{ij}(l)\}\{f_{ij,p}^t(l)\} \\ \{\bar{x}_{ij,p}(l)\},\{\bar{\beta}_{kij}(l,m)\}}} \sum_{(i,j)\in\mathcal{E}} U_{ij}(z_{ij})$$

$$s.t.\ (1),(4),(5),(6),(7),(9),(11)$$

$$\bar{x}_{ij,p}(l) \in [0,1],\ l = 1,\cdots,L,\ (i,j) \in \mathcal{E},\ p \in \mathcal{P} \quad (17)$$

$$\bar{\beta}_{kij}(l,m) \in [0,1],\ l,m = 1,\cdots,L,\ (k,i),(i,j) \in \mathcal{E} \quad (18)$$

$$f_{ij,p}^t(l) \le \bar{x}_{ij,p}(l) B_{ij},\ l = 1,\cdots,L,\ (i,j) \in \mathcal{E},\ p \in \mathcal{P}_t,\ t \in \mathcal{T} \quad (19)$$

$$\bar{x}_{ij,p}(m) \ge \bar{\beta}_{kij}(l,m)\bar{x}_{ki,p}(l),\ k \in \mathcal{I}_i,\ l = 1,\cdots,L,\ m = 1,\cdots,L,\ \mathcal{I}_i \ne \emptyset,\ (i,j) \in \mathcal{E}, p \in \mathcal{P} \quad (20)$$

$$\bar{x}_{ij,p}(m) \le \sum_{k\in\mathcal{I}_i, l=1,\cdots,L} \bar{\beta}_{kij}(l,m)\bar{x}_{ki,p}(l),\ m = 1,\cdots,L,\ \mathcal{I}_i \ne \emptyset,\ (i,j) \in \mathcal{E}, p \in \mathcal{P} \quad (21)$$

Note that Constraints (17) and (18) in Problem 3 can be treated as the continuous relaxation of Constraints (2) and (3) in Problem 1. Constraint (19) in Problem 3 corresponds to Constraint (8) in Problem 1. Constraints (20) and (21) in Problem 3 can be treated as the continuous counterpart of Constraint (10) in Problem 1. The following lemma shows the relationship between Problem 1 (mixed discrete-continuous optimization problem) and Problem 3 (continuous optimization problem).

Lemma 3 (Relationship between Problem 1 and Problem 3): (i) If $\{z_{ij}\},\{z_{ij}(l)\},\{f_{ij,p}^t(l)\},\{x_{ij,p}(l)\},\{\beta_{kij}(l,m)\}$ is a feasible solution to Problem 1, then $\{z_{ij}\},\{z_{ij}(l)\},\{f_{ij,p}^t(l)\},\{\bar{x}_{ij,p}(l)\},\{\bar{\beta}_{kij}(l,m)\}$ is a feasible solution to Problem 3, where $\bar{x}_{ij,p}(l) = x_{ij,p}(l)$ and $\bar{\beta}_{kij}(l,m) = \beta_{kij}(l,m)$; if $\{z_{ij}\},\{z_{ij}(l)\},\{f_{ij,p}^t(l)\},\{\bar{x}_{ij,p}(l)\},\{\bar{\beta}_{kij}(l,m)\}$ is a feasible solution to Problem 3, then $\{z_{ij}\},\{z_{ij}(l)\},\{f_{ij,p}^t(l)\},\{x_{ij,p}(l)\},\{\beta_{kij}(l,m)\}$ is a feasible solution to Problem 1, where $x_{ij,p}(l) = \lceil \bar{x}_{ij,p}(l) \rceil$ and $\beta_{kij}(l,m) = \lceil \bar{\beta}_{kij}(l,m) \rceil$. (ii) The feasibilities of Problem 1 and Problem 3 imply each other. (iii) The optimal values of Problem 1 and Problem 3 are the same, i.e., $U_x^*(L) = \bar{U}_x^*(L)$.

Proof: Please refer to Appendix B. ■

By Lemma 3, solving Problem 1 is equivalent to solving Problem 3.

B. Distributed Solution

Problem 3 is a (pure) continuous optimization problem. It is not convex due to Constraints (20) and (21). Several penalty methods [12] can be applied to find a local minimum of Problem 3 with polynomial-time complexity. Those methods can also be implemented in a distributed manner using standard decomposition. On the other hand, by weak duality [12], dual method can be applied to find a lower bound of the global minimum value of Problem 3. The difference between an obtained local minimum and this lower bound can serve as an upper bound on the performance gap between the local minimum and the global minimum of Problem 3.

VI. Conclusion

In this paper, we considered linear network code constructions for general connections of continuous flows to minimize the total cost of edge use based on mixing. To solve the minimum-cost network coding design problem, we proposed two equivalent alternative formulations with discrete mixing and continuous mixing, respectively, and developed distributed algorithms to solve them. Our approach allows fairly general coding across flows and guarantees no greater cost than any solution without inter-flow network coding.

Appendix A: Proof of Lemma 1

First, we consider $L = 1$. We omit the index terms (1) and $(1,1)$ behind the variables for notational simplicity. Let $\{z_{ij}\}$, $\{x_{ij,p}\}$, $\{\beta_{kij}\}$ and $\{f_{ij,p}^t\}$ denote a feasible solution

to Problem 1. We shall extend the proof of Lemma 1 in [8] for the integer flows ($f_{ij,p}^t \in \{0,1\}$) and unit source rates ($R_p = 1$) with one global coding vector over each edge ($z_{i,j} \in \{0,1\}$) to the general continuous flows ($f_{ij,p}^t \in [0, B_{ij}]$) and source rates ($R_p \in \mathbb{R}^+$) with multiple global coding vectors ($z_{i,j} \in [0, B_{ij}]$) over each edge. In the general case, we code over time $n \geq 1$. For all $p \in \mathcal{P}$, convert source p with source rate R_p over time n to $\lfloor nR_p \rfloor$ unit rate sub-sources $p_1, \cdots, p_{\lfloor nR_p \rfloor}$. For each edge $(i,j) \in \mathcal{E}$, allow the total number of the sub-flows of flow $p \in \mathcal{P}_t$ to terminal $t \in \mathcal{T}$ to be fewer than or equal to $\lceil nf_{ij,p}^t \rceil$. Therefore, the flow path of flow p can be decomposed into $\lfloor nR_p \rfloor$ unit rate sub-flow paths $p_1, \cdots, p_{\lfloor nR_p \rfloor}$ from source $p \in \mathcal{P}_t$ to terminal $t \in \mathcal{T}$. The sum rate of unit rate sub-flows of flow p over edge $(i,j) \in \mathcal{E}$ is less than or equal to $\lceil nf_{ij,p}^t \rceil$. The sum rate of unit rate sub-flows of all the flows over edge (i,j) is less than or equal to $\bar{z}_{ij} = \max_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_t} \lceil nf_{ij,p}^t \rceil$. Decompose edge (i,j) into $\bar{z}_{ij}$ sub-edges. Let sub-flows to terminal t pass different sub-edges, i.e., each sub-edge transmit at most one sub-flow to terminal t. We have now reduced the general case to the special case considered in Lemma 1 in [8]. Therefore, we can show that there exists a feasible linear network code over time n. The associated average sum transmission rate over edge (i,j) is $\bar{z}_{ij}/n$. Note that $\bar{z}_{ij}/n - z_{ij}/n \leq P/n$. Therefore, this code design can achieve the minimum cost $U_x^*(1)$ by taking n arbitrarily large.

When $L > 1$, we can convert each edge $(i,j) \in \mathcal{E}$ into L edges. Then, we can apply the above proof for $L = 1$ to the equivalent constructed network.

Appendix B: Proof of Lemma 3

It is obvious that (i) implies (ii). Next, we show that (i) implies (iii). Suppose (i) holds, which indicates that each $\{z_{ij}\}$ associated with a feasible solution to Problem 1 is also associated with a feasible solution to Problem 3, and vice versa. In addition, $\{z_{ij}\}$ fully determines $\sum_{(i,j) \in \mathcal{E}} U_{ij}(z_{ij})$. Thus, the set of feasible network costs to Problem 1 is the same as that to Problem 3, implying the optimal values of the two problems are the same. Therefore, we can show that (i) implies (iii). Thus, to show Lemma 3, it is sufficient to show (i). Note that in the proof, we only need to consider the different constrains between Problem 1 and Problem 3.

To show (i), we first show that when $x_{ij,p}(l) \in \{0,1\}$ and $\beta_{kij}(l,m) \in \{0,1\}$, Constraint (10) is equivalent to the following two constraints in (22) and (23).

$$x_{ij,p}(m) \geq \beta_{kij}(l,m) x_{ki,p}(l), \ k \in \mathcal{I}_i, \ l = 1, \cdots, L \\ m = 1, \cdots, L, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}, p \in \mathcal{P} \quad (22)$$

$$x_{ij,p}(m) \leq \sum_{k \in \mathcal{I}_i, l = 1, \cdots, L} \beta_{kij}(l,m) x_{ki,p}(l), \\ m = 1, \cdots, L, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}, p \in \mathcal{P} \quad (23)$$

Note that Constraints (10), (22) and (23) are for all $m = 1, \cdots, L, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}$ and $p \in \mathcal{P}$. Thus, we prove this equivalence by considering the following two cases for any $m = 1, \cdots, L, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}$ and $p \in \mathcal{P}$. First, consider the case where $\beta_{kij}(l,m) x_{ki,p}(l) = 0$ for all $k \in \mathcal{I}_i$ and $l = 1, \cdots, L$. Constraint (10) implies that $x_{ij,p}(m) = 0$, and Constraints (22) and (23) also imply that $x_{ij,p}(m) = 0$. Second, consider the case where there exists at least one pair (k,l), where $k \in \mathcal{I}_i$ and $l = 1, \cdots, L$, such that $\beta_{kij}(l,m) x_{ki,p}(l) = 1$. Constraint (10) implies that $x_{ij,p}(m) = 1$, and Constraints (22) and (23) also imply that $x_{ij,p}(m) = 1$. Note that under the integer constraints $x_{ij,p}(l) \in \{0,1\}$ and $\beta_{kij}(l,m) \in \{0,1\}$, the above two cases are the only two possible cases. Therefore, we can show Constraint (10) is equivalent to Constraints (22) and (23).

Next, we show that the first statement of (i) holds. Suppose $\{z_{ij}\}, \{z_{ij}(l)\}, \{f_{ij,p}^t(l)\}, \{x_{ij,p}(l)\}, \{\beta_{kij}(l,m)\}$ is a feasible solution to Problem 1. Let $\bar{x}_{ij,p}(l) = x_{ij,p}(l) \in \{0,1\}$ for all $l = 1, \cdots, L, \ (i,j) \in \mathcal{E}$ and $p \in \mathcal{P}$, and $\bar{\beta}_{kij}(l,m) = \beta_{kij}(l,m) \in \{0,1\}$ for all $k \in \mathcal{I}_i, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}$ and $l, m = 1, \cdots, L$. Since Constraints (17), (18) and (19) in Problem 3 can be treated as the continuous relaxation of Constraints (2), (3) and (8) in Problem 1, $\{f_{ij,p}^t(l)\}, \{\bar{x}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\}$ satisfies Constraints (17), (18) and (19). In addition, since Constraint (10) is equivalent to Constraints (22) and (23), and Constraints (20) and (21) can be treated as the continuous relaxation of Constraints (22) and (23), $\{\bar{x}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\}$ satisfies Constraints (20) and (21). Therefore, we can show $\{z_{ij}\}, \{z_{ij}(l)\}, \{f_{ij,p}^t(l)\}, \{\bar{x}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\}$ is a feasible solution to Problem 3.

Finally, we show that the second statement of (i) holds. Suppose $\{z_{ij}\}, \{z_{ij}(l)\}, \{f_{ij,p}^t(l)\}, \{\bar{x}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\}$ is a feasible solution to Problem 3. Let $x_{ij,p}(l) = \lceil \bar{x}_{ij,p}(l) \rceil$ for all $l = 1, \cdots, L, \ (i,j) \in \mathcal{E}$ and $p \in \mathcal{P}$, and $\beta_{kij}(l,m) = \lceil \bar{\beta}_{kij}(l,m) \rceil$ for all $k \in \mathcal{I}_i, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}$ and $l, m = 1, \cdots, L$. In other words, if $\bar{x}_{ij,p}(l) = 0$ ($\bar{\beta}_{kij}(l,m) = 0$), then $x_{ij,p}(l) = 0$ ($\beta_{kij}(l,m) = 0$); if $\bar{x}_{ij,p}(l,m) \in (0,1]$ ($\bar{\beta}_{kij}(l,m) \in (0,1]$), then $x_{ij,p}(l) = 1$ ($\beta_{kij}(l,m) = 1$). It is obvious that $\{f_{ij,p}^t(l)\}, \{x_{ij,p}(l)\}, \{\beta_{kij}(l,m)\}$ satisfies Constraints (2), (3) and (8). It remains to show $\{x_{ij,p}(l)\}, \{\beta_{kij}(l,m)\}$ satisfies Constraint (10). Note that Constraint (10) is for all $m = 1, \cdots, L, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}$ and $p \in \mathcal{P}$. Thus, similarly, we prove this result by considering the following two cases for any $m = 1, \cdots, L, \ \mathcal{I}_i \neq \emptyset, \ (i,j) \in \mathcal{E}$ and $p \in \mathcal{P}$. First, consider the case where $\bar{\beta}_{kij}(l,m) \bar{x}_{ki,p}(l) = 0$ for all $k \in \mathcal{I}_i$ and $l = 1, \cdots, L$. Constraints (20) and (21) imply that $\bar{x}_{ij,p}(m) = 0$, and hence, we have $x_{ij,p}(m) = \lceil \bar{x}_{ij,p}(m) \rceil = 0$. In addition, $\bar{\beta}_{kij}(l,m) \bar{x}_{ki,p}(l) = 0$ for all $k \in \mathcal{I}_i$ and $l = 1, \cdots, L$ also implies $\beta_{kij}(l,m) x_{ki,p}(l) = \lceil \bar{\beta}_{kij}(l,m) \rceil \lceil \bar{x}_{ki,p}(l) \rceil = 0$ for all $k \in \mathcal{I}_i$ and $l = 1, \cdots, L$. Thus, in this case, we can show $\{x_{ij,p}(l)\}, \{\beta_{kij}(l,m)\}$ satisfies Constraint (10). Second, consider the case where there exists at least one pair (k,l), where $k \in \mathcal{I}_i$ and $l = 1, \cdots, L$, such that

$\bar{\beta}_{kij}(l,m)\bar{x}_{ki,p}(l) \in (0,1]$. Constraints (20) and (21) together with Constraints (17) and (18) imply that $\bar{x}_{ij,p}(m) \in (0,1]$, and hence, we have $x_{ij,p}(m) = \lceil \bar{x}_{ki,p}(l) \rceil = 1$. In addition, $\bar{\beta}_{kij}(l,m)\bar{x}_{ki,p}(l) \in (0,1]$ together with Constraints (17) and (18) also imply $\beta_{kij}(l,m)x_{ki,p}(l) = \lceil \bar{\beta}_{kij}(l,m) \rceil \lceil \bar{x}_{ki,p}(l) \rceil = 1$. Thus, in this case, we can show $\{x_{ij,p}(l)\}, \{\beta_{kij}(l,m)\}$ satisfies Constraint (10). Note that under the continuous constraints $x_{ij,p}(l) \in [0,1]$ and $\beta_{kij}(l,m) \in [0,1]$, the above two cases are the only two possible cases. Therefore, we can show $\{z_{ij}\}, \{z_{ij}(l)\}, \{f^t_{ij,p}(l)\}, \{x_{ij,p}(l)\}, \{\beta_{kij}(l,m)\}$ is a feasible solution to Problem 1.

Therefore, we complete the proof of Lemma 3.

REFERENCES

[1] R. Koetter and M. Médard, "Beyond routing: an algebraic approach to network coding," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, Oct 2002, pp. 122–130 vol.1.

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

[3] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.

[4] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, Sept 2005, pp. 264–267.

[5] S. Kamath, D. N. Tse, and C.-C. Wang, "Two-unicast is hard," in *Information Theory, 2014. ISIT 2014. IEEE International Symposium on*, June 2014.

[6] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2608–2623, 2006.

[7] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *in Proc. 2004 International Symposium on Information Theory and its Applications (ISITA 2004)*, 2004, pp. 1232–1237.

[8] Y. Cui, M. Médard, E. Yeh, D. Leith, and K. Duffy. (2015, July) A linear network code construction for general integer connections based on the constraint satisfaction problem. ISIT Technical Report. [Online]. Available: http://arxiv.org/abs/1502.06321

[9] A. ParandehGheibi, A. Ozdaglar, M. Effros, and M. Médard, "Optimal reverse carpooling over wireless networks - a distributed optimization approach," in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, March 2010, pp. 1–6.

[10] C.-C. Wang and N. Shroff, "Beyond the butterfly - a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, June 2007, pp. 121–125.

[11] K. Duffy, C. Bordenave, and D. Leith, "Decentralized constraint satisfaction," *Networking, IEEE/ACM Transactions on*, vol. 21, no. 4, pp. 1298–1308, Aug 2013.

[12] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.