IEEE/ACM *Transactions on*
Networking

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

Ying Cui, Muriel Médard, Edmund Yeh, Douglas Leith, Ken Duffy

***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 methods for identifying 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 existing solutions. Numerical results illustrate the performance of our approach.**

***Index Terms*—network coding, network mixing, general connection, resource optimization, distributed algorithm.**

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, as shown by [4] using an example from matroid theory.

Different aspects of the connection between a matroidal structure and the network coding problem with general connections have been investigated in the literature [5]–[13]. However, progress in understanding the matroidal structure of the general connection problem has not yet provided simple and useful approaches to generating explicit linear codes. There has been considerable investigation of special cases [15]–[20]. However, the studies of these special cases do not offer satisfactory solutions for the general case.

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 [21]. 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 [22]. 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 [23], a two-step mixing approach is proposed for network resource minimization of general connections, where flow partition and flow

Manuscript received April 26, 2017; revised January 28, 2018; accepted June 5, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. Paschos. Date of publication XX XX, 2018; date of current version XX XX, 2018. The work of Y. Cui was supported by NSFC grant 61401272 and grant 61521062 as well as the Shanghai Key Laboratory Funding under Grant STCSM15DZ2270400. The work of E. Yeh was supported by the National Science Foundation grant CNS-1205562. The paper was presented in part at ICC 2015. *(Corresponding author: Ying Cui.)* Y. Cui is Institute of Wireless Communication Technologies, Shanghai Institute for Advanced Communication and Data Science, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: cuiying@sjtu.edu.cn). M. Médard is with Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: medard@mit.edu). E. M. Yeh is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA. D. Leith is with Trinity College Dublin, Ireland. K. Duffy is with Maynooth University.

Digital Object Identifier: 10.1109/TNET.2018.2865534

rate optimization are considered separately. In [14], [24], we introduce linear network mixing coefficients for general connections that generalize random linear network coding (RLNC) for multicast connections, 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 [14], [24] is a discrete optimization problem that 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 [25] and [26], simple two-flow combinations are proposed for network resource minimization of general connections.

The flow rate optimization in [23], the joint mixing and flow optimization in [14], [24], and the joint two-flow coding and flow optimization in [25], [26] can be solved distributively. However, the separation of flow partition and flow rate optimization in [23] and the pairwise coding in [25], [26] lead in general to feasibility region reduction and network cost increase. The joint mixing and flow rate optimization for general connections of integer flows in [14], [24] allow fairly general coding across flows. However, in [14], [24], we consider integer flow rates and edge capacities, and do not allow flow splitting and coding over time, leading to coded symbols flowing through each edge at an integer rate. The restriction of integer flow rates affects the network 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 [14], [24] to allow flow splitting and coding over time, leading to coded symbols flowing through each edge at a continuous rate, 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, the solution of the two-step mixing in [23], and the integer solution of the discrete joint mixing and flow rate optimization in [14], [24]. 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 its discrete subproblem to a constraint satisfaction problem (CSP) in discrete optimization and applying recent results in the domain [27], and solving its continuous subproblem using a primal-dual method. The distributed algorithm for the continuous mixing formulation is based on penalty methods for nonlinear programming [28]. Note that the methods for solving the continuous problems are new compared to [14], [24]. In addition, note that this paper extends the results in the conference version in [29] which does not present a distributed algorithm for the continuous mixing formulation.

II. Network Model and Definitions

In this section, we first define the network model for general connections of continuous flows. The model is similar to the one we considered in [14], [24] for integer flows, except that here we consider general flow rates and edge capacities, and allow flow splitting and coding over time. Next, we also briefly illustrate the formal relationship between linear network coding and mixing established in [14], [24].

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 $I_i \leq D$ and $O_i \leq D$ for all $i \in \mathcal{V}$, where D is a constant. Consider a finite field $\mathcal{F}$ with size $F = |\mathcal{F}|$. Let $\mathcal{P} = \{1, \cdots, P\}$ denote the set of $P = |\mathcal{P}|$ flows of symbols in finite field $\mathcal{F}$ 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. To be specific, each continuous flow consists of symbols from finite field $\mathcal{F}$, and its source rate (i.e., the number of symbols generated at its source per unit time) can be a real number. 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. We assume different flows do not share a common source node and no source node

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

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., $\cup_{t \in \mathcal{T}} \mathcal{P}_t = \mathcal{P}$. Let $\boldsymbol{\mathcal{P}} \triangleq (\mathcal{P}_t)_{t \in \mathcal{T}}$ denote the demands of all the terminals. We assume no terminal has any outgoing edges.

Let $B_{ij} \in \mathbb{R}^+$ denote the edge capacity for edge (i,j). We assume a cost is incurred on an edge when information is transmitted through the edge and let $U_{ij}(z_{ij})$ denote the cost function for edge (i,j) when the transmission rate through edge (i,j) is $z_{ij} \in [0, B_{ij}]$. Note that we allow flow splitting and coding over time, leading to coded symbols flowing through each edge of the network at a continuous rate.[2] Assume $U_{ij}(z_{ij})$ is convex,[3] non-decreasing, and twice continuously differentiable in z_{ij}. For example, we can choose $U_{ij}(z_{ij}) = a^{z_{ij}}$ with $a \geq 1$ or $U_{ij}(z_{ij}) = {z_{ij}}^a$ with $a \geq 1$. We are interested in the problem of finding linear network coding designs and minimizing the network cost $\sum_{(i,j) \in \mathcal{E}} U_{ij}(z_{ij})$ for general connections of continuous flows under 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.

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 $(i,j) \in \mathcal{E}$, 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 $\mathbf{c}_{ij} \triangleq (c_{ij,p})_{p \in \mathcal{P}} \in \mathcal{F}^P$ 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 $(i,j) \in \mathcal{E}$. Here, $\mathcal{F}^P$ represents the set of global coding vectors, the cardinality of which is F^P. 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 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 based on local and global mixing coefficients established in [14], [24]. Later, we shall see that distributed linear network mixing designs in terms of these mixing coefficients are much easier. 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 $\alpha_{kij} = 0$; if $\beta_{kij} = 1$, we can further determine how symbol σ_{ki} contributes to the linear combination forming symbol σ_{ij} by choosing $\alpha_{kij} \in \mathcal{F}$ (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 $c_{ij,p} = 0$; if $x_{ij,p} = 1$, we can further determine how symbol σ_p contributes to the linear combination forming symbol σ_{ij} (note that $c_{ij,p}$ can be zero when $x_{ij,p} = 1$). Then, we introduce the global mixing vector $\mathbf{x}_{ij} \triangleq (x_{ij,p})_{p \in \mathcal{P}} \in \{0,1\}^P$ for edge $(i,j) \in \mathcal{E}$, which relates to the global coding vector $\mathbf{c}_{ij} = (c_{ij,p})_{p \in \mathcal{P}} \in \mathcal{F}^P$. Here, $\{0,1\}^P$ represents the set of global mixing vectors, the cardinality of which is 2^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 do not change over time.

Global mixing vectors provide a natural way of speaking of flows as possibly coded or not coded without knowledge of the specific values of 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. 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 $\mathbf{e}_p$

[2]A detailed illustration of flow splitting and coding over time can be found in Appendix A.

[3]The convexity assumption precludes the case where transmission rates over some edges are too large compared with others, hence balancing traffic over a network.

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).

Definition 1 (Feasibility of Scalar Linear Network Mixing): [14], [24] 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 $\mathbf{x} = (x_{ij,p})_{(i,j)\in\mathcal{E},p\in\mathcal{P}}$ is called feasible if the following three conditions are satisfied: 1) $\mathbf{x}_{s_p j} = \mathbf{e}_p$ for source edge $(s_p, j) \in \mathcal{E}$, for all $s_p \in \mathcal{S}$ 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, for all $i \notin \mathcal{S}$ 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 $i \in \mathcal{I}_t$, $p \notin \mathcal{P}_t$ and $t \in \mathcal{T}$.

Note that $x_{it,p} = 0$ for all $i \in \mathcal{I}_t$, $p \notin \mathcal{P}_t$ and $t \in \mathcal{T}$ in Condition 3) of 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 $\boldsymbol{\beta} \triangleq (\beta_{kij})_{(k,i),(i,j)\in\mathcal{E}}$), one way to implement mixing when $\mathcal{F}$ is large is to use RLNC [3] (to obtain $\boldsymbol{\alpha} \triangleq (\alpha_{kij})_{(k,i),(i,j)\in\mathcal{E}}$), as discussed in the introduction. Specfcially, when $\beta_{kij} = 1$, α_{kij} can be randomly, uniformly, and independently chosen in $\mathcal{F}$ using RLNC; when $\beta_{kij} = 0$, α_{kij} has to be chosen to be 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 for each edge and allow coded symbols to flow through each edge at a continuous rate.

A. Design Parameter

Now, we generate the mixing design illustrated in Section II-B [14], [24] by considering multiple global mixing vectors for each edge, allowing flows mixed over each edge in different ways. 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). First, we introduce the global and local network mixing vectors, for a given mixing parameter L. Denote $\mathcal{L} \triangleq \{1, \cdots, L\}$. For each $l \in \mathcal{L}$, let $\mathbf{x}_{ij,l} \triangleq (x_{ij,p,l})_{p\in\mathcal{P}} \in \{0,1\}^P$ denote the l-th global network mixing vector over edge $(i,j) \in \mathcal{E}$. Let $\boldsymbol{\beta}_{kij,l,m} \in \{0,1\}$ denote the local mixing coefficient

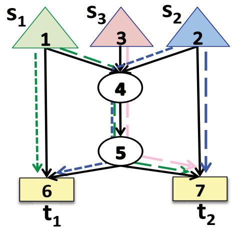


Fig. 1: Illustration of flow partition $\boldsymbol{\mathcal{Y}}$ and mixing parameter L. $\mathcal{P} = \{1,2,3\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$, $R_1 = R_2 = R_3 = 1$, $B_{ij} = 10$ for all $(i,j) \in \mathcal{E}$, $U_{45}(z_{45}) = 10z_{45}$, $U_{ij}(z_{ij}) = z_{ij}$ for all $(i,j) \in \mathcal{E} \setminus \{(4,5)\}$, $\mathcal{T} = \{t_1, t_2\}$, $\mathcal{P}_1 = \{1,2\}$ and $\mathcal{P}_2 = \{1,2,3\}$. Thus, $\boldsymbol{\mathcal{Y}} = \{\{1,2\},\{3\}\}$, $L_{\max} = |\boldsymbol{\mathcal{Y}}| = 2$ and $L \in \{1,2\}$.

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 $l, m \in \mathcal{L}$. Next, we illustrate the maximum number of global network mixing vectors $L_{\max}$. Denote $\boldsymbol{\mathcal{Y}} \triangleq \{\cap_{t\in\mathcal{T}}\mathcal{Y}_t : \mathcal{Y}_t = \mathcal{P}_t \text{ or } \mathcal{Y}_t = \mathcal{P} - \mathcal{P}_t\} - \{\emptyset\}$, which 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 $1 \le L_{\max} \le P$, 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. 1 illustrates an example of flow partition $\boldsymbol{\mathcal{Y}}$ and mixing parameter L for the general case.

Let $f^t_{ij,p,l} \ge 0$ denote the transmission rate of 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 let $z_{ij,l} \ge 0$ denote the transmission rate corresponding to $\mathbf{x}_{ij,l}$ over edge $(i,j) \in \mathcal{E}$, where $l \in \mathcal{L}$. As we allow flow splitting and coding over time, $f^t_{ij,p,l}$ and $z_{ij,l}$ can be real numbers.

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.[4]

[4]Note that (1) with $j = t$, (6) with $i = t$, and (7) with $j = t$ imply $\vee_{i\in\mathcal{I}_t, l\in\mathcal{L}} x_{it,p,l} = 1$ for all $p \in \mathcal{P}_t$ in Condition 3) of Definition 1, where $t \in \mathcal{T}$.

Problem 1 (Mixing):

$$U^*(L) = \min_{\mathbf{z},\mathbf{f},\mathbf{x},\boldsymbol{\beta}} \sum_{(i,j)\in\mathcal{E}} U_{ij}\left(\sum_{l\in\mathcal{L}} z_{ij,l}\right)$$

$$s.t.\ x_{ij,p,l} \in \{0,1\},\ (i,j)\in\mathcal{E},\ p\in\mathcal{P},\ l\in\mathcal{L} \quad (1)$$

$$\beta_{kij,l,m} \in \{0,1\},\ (k,i),(i,j)\in\mathcal{E},\ l,m\in\mathcal{L} \quad (2)$$

$$f^t_{ij,p,l} \geq 0,\ (i,j)\in\mathcal{E},\ p\in\mathcal{P}_t,\ t\in\mathcal{T},\ l\in\mathcal{L} \quad (3)$$

$$\sum_{p\in\mathcal{P}_t} f^t_{ij,p,l} \leq z_{ij,l},\ (i,j)\in\mathcal{E},\ t\in\mathcal{T},\ l\in\mathcal{L} \quad (4)$$

$$\sum_{l\in\mathcal{L}} z_{ij,l} \leq B_{ij},\ (i,j)\in\mathcal{E} \quad (5)$$

$$\sum_{k\in\mathcal{O}_i, l\in\mathcal{L}} f^t_{ik,p,l} - \sum_{k\in\mathcal{I}_i, l\in\mathcal{L}} f^t_{ki,p,l} = \sigma^t_{i,p},\quad i\in\mathcal{V},\ p\in\mathcal{P}_t,\ t\in\mathcal{T} \quad (6)$$

$$f^t_{ij,p,l} \leq x_{ij,p,l}B_{ij},\ (i,j)\in\mathcal{E},\ p\in\mathcal{P}_t,\ t\in\mathcal{T},\ l\in\mathcal{L} \quad (7)$$

$$\mathbf{x}_{s_p j,l} = \mathbf{e}_p,\ (s_p,j)\in\mathcal{E},\ p\in\mathcal{P},\ l\in\mathcal{L} \quad (8)$$

$$\mathbf{x}_{ij,l} = \vee_{k\in\mathcal{I}_i, m\in\mathcal{L}} \beta_{kij,m,l}\mathbf{x}_{ki,m},\quad i\notin\mathcal{S},\ (i,j)\in\mathcal{E},\ l\in\mathcal{L} \quad (9)$$

$$x_{it,p,l} = 0,\ i\in\mathcal{I}_t,\ p\notin\mathcal{P}_t,\ t\in\mathcal{T},\ l\in\mathcal{L}, \quad (10)$$

where

$$\sigma^t_{i,p} = \begin{cases} R_p, & i = s_p \\ -R_p, & i = t \\ 0, & \text{otherwise} \end{cases} \quad i\in\mathcal{V},\ p\in\mathcal{P}_t,\ t\in\mathcal{T}. \quad (11)$$

Here, $\mathbf{z} \triangleq (z_{ij,l})_{(i,j)\in\mathcal{E},l\in\mathcal{L}}$, $\mathbf{f} \triangleq (f^t_{ij,p,l})_{(i,j)\in\mathcal{E},p\in\mathcal{P}_t,t\in\mathcal{T},l\in\mathcal{L}}$, $\mathbf{x} \triangleq (x_{ij,p,l})_{(i,j)\in\mathcal{E},p\in\mathcal{P},l\in\mathcal{L}}$, and $\boldsymbol{\beta} \triangleq (\beta_{kij,l,m})_{(k,i),(i,j)\in\mathcal{E},l,m\in\mathcal{L}}$.

Problem 1 is a mixed discrete-continuous optimization problem, and does not appear to have a ready solution.

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 (10) is vacuous, and the constraint in (7) is always satisfied by choosing $\beta_{kij,1,1} = 1$ for all $(k,i),(i,j) \in \mathcal{E}$ and choosing $\mathbf{x}$ according to (8) and (9). 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 [22]. The complexity of the optimization for the multicast case is much lower than that for the general case. This is because in the optimization for the multicast case, the variables $\mathbf{x}$ and $\boldsymbol{\beta}$ do not appear, and the constraints in (1), (2), (7), (8), (9) and (10) can be removed.

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 $l \in \mathcal{L}$ 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 cost 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 (2), is equivalent to the minimum-cost flow rate control problem in the second step of the two-step mixing approach in [23]. 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 [23].

Remark 4 (Comparison with Joint Design for Integer Flows): Problem 1 with $L = 1$, $z_{ij,1} \in \{0,1\}$ and $f^t_{ij,p,l} \in \{0,1\}$ instead of (3), is equivalent to the discrete minimum-cost joint mixing and flow rate optimization problem for general connections of integer flows in [14], [24], which does not allow flow splitting and coding over time. Thus, the minimum network cost of Problem 1 is no greater than that of the discrete optimization problem in [14], [24]. If the optimal solution of Problem 1 is a non-integer solution, it has a lower network cost than that of the discrete optimization in [14], [24].[5]

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. 1. 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: $\beta_{145,1,1} = \beta_{245,1,1} = \beta_{345,1,2} = 1$, $\beta_{145,2,1} = \beta_{245,2,1} = \beta_{345,2,2} = 0$, $\beta_{145,m,2} = \beta_{245,m,2} = \beta_{345,m,1} = 0$ 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 ordered edge-mixing index pairs $((i,j),l)$

[5] Due to space limitations, we do not numerically verify the gains of the proposed design in this paper over the ones in [14], [23], [24]. Please note that in [14], we have shown the gain of the proposed solution in [14], [24] over the solution in [23] using numerical experiments, and the gain of the solution of Problem 1 over the solution in [14], [24] is obvious.

such that $f^t_{ij,p,l} = 1$) from the three sources, i.e., $\{((i,j),l) : f^t_{ij,p,l} = 1,\ (i,j) \in \mathcal{E},\ l \in \mathcal{L}\}$ for all $p \in \mathcal{P}_t$ and $t \in \mathcal{T}$, are illustrated using green, blue and pink curves in Fig. 1. Accordingly, choose the transmission rates as follows: $z_{ij,1} = 1$ and $z_{ij,2} = 0$ for all $(i,j) = (1,6),(1,4),(2,7),(2,4),(3,4)$, $z_{45,1} = z_{45,2} = z_{56,1} = z_{57,1} = z_{57,2} = 1$, and $z_{56,2} = 0$.

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. ■

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).[6] 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 $l,m \in \mathcal{L}$ 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 $\mathbf{c}_{ij,l} = \sum_{k\in\mathcal{I}_i,m\in\mathcal{L}} \alpha_{kij,m,l}\mathbf{c}_{ki,m}$ for all $(i,j) = (4,5),(5,6),(5,7)$ and $l \in \mathcal{L}$.

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 (2), we know that for given $(k,i),(i,j) \in \mathcal{E}$, the number of possible choices for $(\beta_{kij,l,m})_{l,m\in\mathcal{L}}$ is L^2. Since $\sum_{(i,j)\in\mathcal{E}} O_j = \sum_{j\in\mathcal{V}} I_j O_j \leq \sum_{j\in\mathcal{V}} D O_j = DE$, the number of possible choices for $\boldsymbol{\beta} = (\beta_{kij,l,m})_{(k,i),(i,j)\in\mathcal{E},l,m\in\mathcal{L}}$ is smaller than or equal to L^2DE. Note that by (8) and (9), $\mathbf{x}$ can be fully determined by $\boldsymbol{\beta}$. Therefore, the number of choices for $\mathbf{x}$ and $\boldsymbol{\beta}$ of Problem 1 is L^2DE, which increases with L.

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 $U^*(L+1) \leq U^*(L)$, where $L \in \{1,\cdots,L_{max}-1\}$.

[6]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), and 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.

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 complete the proof. ■

By Lemma 2, we know that the network cost $U^*(L)$ is non-increasing w.r.t. L. This can also be understood from the example in Fig. 1. 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).

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, i.e., $\mathbf{x}$ and $\boldsymbol{\beta}$ (discrete variables), and the other is the choice of the flow rates, i.e., $\mathbf{z}$ and $\mathbf{f}$ (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 Discrete Mixing for Problem 1):

$$U^*(L) = \min_{\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)} \quad U^*_x(\mathbf{x}),$$

where $U^*_x(\mathbf{x})$ and $\boldsymbol{\mathcal{M}}(L)$ are given by the following two subproblems, respectively.

Subproblem 1 (Flow Optimization for Problem 2): For given $\mathbf{x}$, we have:

$$U^*_x(\mathbf{x}) = \min_{\mathbf{z},\mathbf{f}} \quad \sum_{(i,j)\in\mathcal{E}} U_{ij}\left(\sum_{l\in\mathcal{L}} z_{ij,l}\right)$$
$$s.t. \quad (3),(4),(5),(6),(7).$$

The optimal solution is written as $(\mathbf{z}^*(\mathbf{x}),\mathbf{f}^*(\mathbf{x}))$.

Subproblem 2 (Feasible Discrete Mixing for Problem 2): Find the set $\boldsymbol{\mathcal{M}}(L) \triangleq \{\mathbf{x} :$

$(1),(2),(8),(9),(10),(12)\}$ of feasible $\mathbf{x}$, where (12) is given by:

$$\vee_{i\in\mathcal{I}_t,l\in\mathcal{L}}\, x_{it,p,l}=1,\ p\in\mathcal{P}_t,\ t\in\mathcal{T}. \tag{12}$$

For given $\mathbf{x}$, Subproblem 1 is a convex optimization problem (optimizing $\mathbf{z}$ and $\mathbf{f}$ for given $\mathbf{x}$) and has polynomial-time complexity [30]. On the other hand, Subproblem 2 is a discrete feasibility problem (obtaining the set of feasible $\mathbf{x}$) and is NP-complete in general [31]. Thus, 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 1. Given a feasible $\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)$, Subproblem 1 is convex and can be solved distributively using the primal-dual method [32]. By relaxing the constraints in (4), (5), (6) and (7) of Subproblem 1, we have the Lagrangian function $L(\mathbf{z},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi})$ given in (13), where $\boldsymbol{\lambda}\triangleq(\lambda^t_{ij,l})_{(i,j)\in\mathcal{E},t\in\mathcal{T},l\in\mathcal{L}}\succeq\mathbf{0}$, $\boldsymbol{\eta}\triangleq(\eta_{ij})_{(i,j)\in\mathcal{E}}\succeq\mathbf{0}$, $\boldsymbol{\mu}\triangleq(\mu^t_{i,p})_{i\in\mathcal{V},p\in\mathcal{P}_t,t\in\mathcal{T}}$ and $\boldsymbol{\xi}\triangleq(\xi^t_{ij,p,l})_{(i,j)\in\mathcal{E},p\in\mathcal{P}_t,t\in\mathcal{T},l\in\mathcal{L}}\succeq\mathbf{0}$ denote the Lagrangian multipliers w.r.t. the constraints in (4), (5), (6) and (7) of Subproblem 1, respectively. The partial derivatives of $L(\mathbf{z},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi})$ are given by:

$$\frac{\partial L}{\partial z_{ij,l}}=U'_{ij}\left(\sum_{m\in\mathcal{L}}z_{ij,m}\right)-\sum_{t\in\mathcal{T}}\lambda^t_{ij,l}+\eta_{ij} \tag{14}$$

$$\frac{\partial L}{\partial f^t_{ij,p,l}}=\lambda^t_{ij,l}+\mu^t_{i,p}\mathbf{1}\left[\mathcal{O}_i\neq\emptyset\right]-\mu^t_{j,p}\mathbf{1}\left[\mathcal{I}_j\neq\emptyset\right]+\xi^t_{ij,p,l} \tag{15}$$

$$\frac{\partial L}{\partial \lambda^t_{ij,l}}=\sum_{p\in\mathcal{P}_t}f^t_{ij,p,l}-z_{ij,l} \tag{16}$$

$$\frac{\partial L}{\partial \eta_{ij}}=\sum_{l\in\mathcal{L}}z_{ij,l}-B_{ij} \tag{17}$$

$$\frac{\partial L}{\partial \mu^t_{i,p}}=\sum_{k\in\mathcal{O}_i,l\in\mathcal{L}}f^t_{ik,p,l}-\sum_{k\in\mathcal{I}_i,l\in\mathcal{L}}f^t_{ki,p,l}-\sigma^t_{i,p} \tag{18}$$

$$\frac{\partial L}{\partial \xi^t_{ij,p,l}}=f^t_{ij,p,l}-x_{ij,p,l}B_{ij}, \tag{19}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. The corresponding dual function is given by:

$$g(\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi})=\min_{\mathbf{z},\mathbf{f}}\quad L(\mathbf{z},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}) \quad s.t.\quad (3). \tag{20}$$

The corresponding dual problem is as follows:

$$\max_{\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}}\quad g(\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}) \quad s.t.\quad \boldsymbol{\lambda}\succeq\mathbf{0},\boldsymbol{\eta}\succeq\mathbf{0},\boldsymbol{\xi}\succeq\mathbf{0}. \tag{21}$$

For given $\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)$, the primal optimal $(\mathbf{z}^*(\mathbf{x}),\mathbf{f}^*(\mathbf{x}))$ and the dual optimal $(\boldsymbol{\lambda}^*(\mathbf{x}),\boldsymbol{\eta}^*(\mathbf{x}),\boldsymbol{\mu}^*(\mathbf{x}),\boldsymbol{\xi}^*(\mathbf{x}))$ can be obtained using the primal-dual algoritm summarized in Algorithm 1. The update equations in Algorithm 1 are given below:

$$z_{ij,l}(n+1)=z_{ij,l}(n)-\delta(n)\frac{\partial L}{\partial z_{ij,l}}(n) \tag{22}$$

$$f^t_{ij,p,l}(n+1)=\left(f^t_{ij,p,l}(n)-\delta(n)\frac{\partial L}{\partial f^t_{ij,p,l}}(n)\right)^+ \tag{23}$$

$$\lambda^t_{ij,l}(n+1)=\left(\lambda^t_{ij,l}(n)+\delta(n)\frac{\partial L}{\partial \lambda^t_{ij,l}}(n)\right)^+ \tag{24}$$

$$\eta_{ij}(n+1)=\left(\eta_{ij}(n)+\delta(n)\frac{\partial L}{\partial \eta_{ij}}(n)\right)^+ \tag{25}$$

$$\mu^t_{i,p}(n+1)=\mu^t_{i,p}(n)+\delta(n)\frac{\partial L}{\partial \mu^t_{i,p}}(n) \tag{26}$$

$$\xi^t_{ij,p,l}(n+1)=\left(\xi^t_{ij,p,l}(n)-\delta(n)\frac{\partial L}{\partial \xi^t_{ij,p,l}}(n)\right)^+, \tag{27}$$

where $(x)^+\triangleq\max\{0,x\}$, the partial derivatives of $L(\mathbf{z}(n),\mathbf{f}(n),\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n),\boldsymbol{\xi}(n))$ in (22)-(27) are given by (14)-(19), and $\{\delta(n)\}$ denotes the diminishing stepsize[7] satisfying:

$$\delta(n)\to 0 \text{ as } n\to\infty,\ \sum_{n=1}^{\infty}\delta(n)=\infty,\sum_{n=1}^{\infty}\delta(n)^2<\infty. \tag{28}$$

(22)-(27) can be computed at each edge based on local information. Thus, Algorithm 1 can be implemented locally. In addition, it has been shown [32] that as $n\to\infty$, $(\mathbf{z}(n),\mathbf{f}(n))\to(\mathbf{z}^*(\mathbf{x}),\mathbf{f}^*(\mathbf{x}))$ and $(\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n),\boldsymbol{\xi}(n))\to(\boldsymbol{\lambda}^*(\mathbf{x}),\boldsymbol{\eta}^*(\mathbf{x}),\boldsymbol{\mu}^*(\mathbf{x}),\boldsymbol{\xi}^*(\mathbf{x}))$. In other words, for given $\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)$, Algorithm 1 converges to the primal and dual optimal of Subproblem 1, as $n\to\infty$. Fig. 2 illustrates the convergence of Algorithm 1 of the network in Fig. 1, with $\mathbf{x}$ given in Example 1. From Fig. 2, we can see that $L(\mathbf{z}(n),\mathbf{f}(n),\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n),\boldsymbol{\xi}(n))$ converges to 28, which is the optimal network cost $U^*(\mathbf{x})$ to Subproblem 1, for $\mathbf{x}$ given in Example 1.

Next, we consider Subproblem 2. Subproblem 2 can be treated as a CSP and solved distributively

[7]Note that the diminishing stepsize can guarantee convergence, although the associated convergence may be slow. For our problem, it is difficult to determine an appropriate constant stepsize with guarantee of convergence.

$$L\left(\mathbf{z},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}\right)=\sum_{(i,j)\in\mathcal{E}}U_{ij}\left(\sum_{l\in\mathcal{L}}z_{ij,l}\right)+\sum_{\substack{(i,j)\in\mathcal{E},\\ t\in\mathcal{T},l\in\mathcal{L}}}\lambda_{ij,l}^{t}\left(\sum_{p\in\mathcal{P}_t}f_{ij,p,l}^{t}-z_{ij,l}\right)+\sum_{(i,j)\in\mathcal{E}}\eta_{ij}\left(\sum_{l\in\mathcal{L}}z_{ij,l}-B_{ij}\right)$$
$$+\sum_{\substack{i\in\mathcal{V},\\ p\in\mathcal{P}_t,t\in\mathcal{T}}}\mu_{i,p}^{t}\left(\sum_{k\in\mathcal{O}_i,l\in\mathcal{L}}f_{ik,p,l}^{t}-\sum_{k\in\mathcal{I}_i,l\in\mathcal{L}}f_{ki,p,l}^{t}-\sigma_{i,p}^{t}\right)+\sum_{\substack{(i,j)\in\mathcal{E},p\in\mathcal{P}_t,\\ t\in\mathcal{T},l\in\mathcal{L}}}\xi_{ij,p,l}^{t}\left(f_{ij,p,l}^{t}-x_{ij,p,l}B_{ij}\right) \quad (13)$$

Algorithm 1 Primal-dual Method for Subproblem 1 (Flow Optimization)

INPUT: $\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)$

OUTPUT: $\mathbf{z}^*(\mathbf{x}),\mathbf{f}^*(\mathbf{x}),\boldsymbol{\lambda}^*(\mathbf{x}),\boldsymbol{\eta}^*(\mathbf{x}),\boldsymbol{\mu}^*(\mathbf{x}),\boldsymbol{\xi}^*(\mathbf{x})$

1: **initialize** $n=0$, $\mathbf{z}(0),\mathbf{f}(0),\boldsymbol{\lambda}(0),\boldsymbol{\eta}(0),\boldsymbol{\mu}(0),\boldsymbol{\xi}(0)$
2: **loop**
3: For all $(i,j)\in\mathcal{E}$, edge (i,j) updates $z_{ij,l}(n+1),f_{ij,p,l}^{t}(n+1),\lambda_{ij,l}^{t}(n+1),\eta_{ij}(n+1),\mu_{i,p}^{t}(n+1)$ and $\xi_{ij,p,l}^{t}(n+1)$ according to (22), (23), (24), (25), (26) and (27), respectively, under given $\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)$.
4: Set $n=n+1$.
5: **end loop**

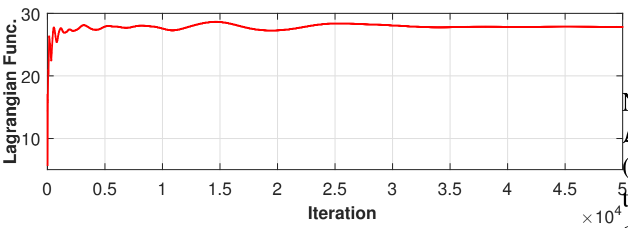


Fig. 2: Convergence of Algorithm 1 (Primal-dual Method for Subproblem 1) for the network in Fig. 1, with $\mathbf{x}$ given in Example 1. The curve represents the Lagrangian function $L\left(\mathbf{z}(n),\mathbf{f}(n),\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n),\boldsymbol{\xi}(n)\right)$ at the n-th iteration, where $L(\cdot)$ is given by (13). Note that in the simulation for this figure, we use $1.1^{\sum_{(i,j)\in\mathcal{E}}U_{ij}\left(\sum_{l\in\mathcal{L}}z_{ij,l}\right)}$ as the objective function, where $U_{ij}(\cdot)$ is given in Fig. 1.

using clause partition and the Communication-Free Learning (CFL) algorithm from [27]. While CSPs are NP-complete in general, CFL provides a probabilistic distributed iterative algorithm with almost sure convergence in finite time. Specifically, the elements of $\mathbf{x}$ can be treated as the variables of the CSP. $\{0,1\}$ can be treated as the finite set of the CSP. From (9), we have an equivalent constraint purely on $\mathbf{x}$, i.e.,

$$\begin{aligned}&\exists\ (\beta_{kij,m,l})_{k\in\mathcal{I}_i,m\in\mathcal{L}},\ \beta_{kij,m,l}\in\{0,1\},\\ &\text{s.t. } \mathbf{x}_{ij,l}=\vee_{k\in\mathcal{I}_i,m\in\mathcal{L}}\beta_{kij,m,l}\mathbf{x}_{ki,m},\\ &\qquad l\in\mathcal{L},\ (i,j)\in\mathcal{E},\ i\notin\mathcal{S}.\end{aligned} \quad (29)$$

In the following, we shall only consider solving for the variables $\mathbf{x}$ of the CSP in a distributed way using clause partition and CFL. Note that we directly choose $\mathbf{x}_{s_pj,l}=\mathbf{e}_p$ for all $l\in\mathcal{L}$, $(s_p,j)\in\mathcal{E}$ and $p\in\mathcal{P}$ according to (8). In addition, $\boldsymbol{\beta}$ can be obtained from feasible $\mathbf{x}$ by (8) and (9).

For notational simplicity, we write the clauses for $\mathbf{x}$ in a more compact form as follows:

$$\begin{aligned}&\phi_{ij,p,l}\Big(\mathbf{x}_{ij,l},\{\mathbf{x}_{ki,m}:m\in\mathcal{L},k\in\mathcal{I}_i\},\\ &\qquad\{\mathbf{x}_{kj,m}:m\in\mathcal{L},k\in\mathcal{I}_j,j\in\mathcal{T}\}\Big)\\ &\triangleq\begin{cases}1, & \text{if } j\notin\mathcal{T},\ (29)\text{ holds}\\ 1, & \text{if } j\in\mathcal{T}\text{ and }p\in\mathcal{P}_j,\ (29)\text{ and (12) hold}\\ 1, & \text{if } j\in\mathcal{T}\text{ and }p\notin\mathcal{P}_j,\ (29)\text{ and (10) hold}\\ 0, & \text{otherwise}\end{cases}\\ &\qquad (i,j)\in\mathcal{E},\ i\notin S,\ p\in\mathcal{P},\ l\in\mathcal{L}.\end{aligned} \quad (30)$$

Note that, when $j\notin\mathcal{T}$, $\{\mathbf{x}_{kj,m}:k\in\mathcal{I}_j,j\in\mathcal{T},m\in\mathcal{L}\}=\emptyset$ and we ignore it in the clause $\phi_{ij,p,l}(\cdot)$. For (12) and (10) in clause $\phi_{ij,p,l}(\cdot)$, we use j as the terminal index instead of t. It can be seen that the constraints in (9) (i.e., (29)), (10) and (12) are considered in clause $\phi_{ij,p,l}(\cdot)$. In addition, the constraint in (8) is considered when choosing $\mathbf{x}_{s_pj,l}=\mathbf{e}_p$ for all $(s_p,j)\in\mathcal{E}$, $p\in\mathcal{P}$ and $l\in\mathcal{L}$. Therefore, all the constraints in Subproblem 2 has been considered in the CSP. We now construct the clause partition of Subproblem 2. Specifically, the set of clauses variable $x_{ij,p,l}$ participates in is as follows:

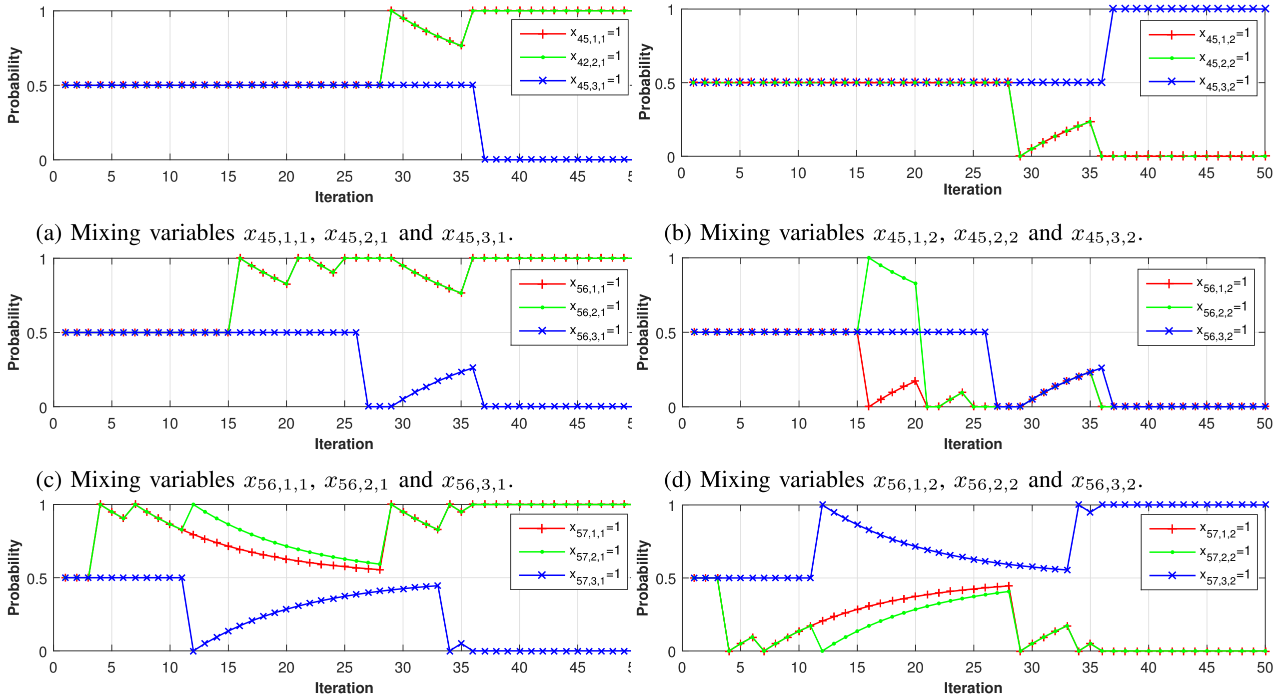
$$\begin{aligned}\Phi_{ij,p,l}\triangleq&\{\phi_{ij,p,l}\}\cup\{\phi_{jk,p,m}:k\in\mathcal{O}_j,m\in\mathcal{L}\}\\ &\cup\{\phi_{kj,p,m}:k\in\mathcal{I}_j,j\in\mathcal{T},m\in\mathcal{L}\}\\ &\quad i\notin S,\ (i,j)\in\mathcal{E},\ p\in\mathcal{P},\ l\in\mathcal{L}.\end{aligned} \quad (31)$$

Note that, when $j\notin\mathcal{T}$, $\{\phi_{kj,p,m}:k\in\mathcal{I}_j,j\in\mathcal{T},m\in\mathcal{L}\}=\emptyset$ and we ignore it in $\Phi_{ij,p,l}$ in (31).

We thus have the following proposition.

Proposition 1 (CSP for Subproblem 2): The CSP with variables $x_{ij,p,l}\in\{0,1\}$, $(i,j)\in\mathcal{E},p\in\mathcal{P},l\in\mathcal{L}$ and clauses (31) has considered all the constraints in Subproblem 2.

Therefore, a feasible $\mathbf{x}\in\boldsymbol{\mathcal{M}}(L)$ to Subproblem 2 can be found distributively using the probabilistic distributed iterative CFL algorithm [27, Algorithm 1]. Specifically, for all $(i,j)\in\mathcal{E}$, $p\in\mathcal{P}$ and $l\in\mathcal{L}$, in each iteration, each node i realizes a Bernoulli random variable selecting $x_{ij,p,l}$; messages on $\mathbf{x}$ are passed



(a) Mixing variables $x_{45,1,1}$, $x_{45,2,1}$ and $x_{45,3,1}$.

(b) Mixing variables $x_{45,1,2}$, $x_{45,2,2}$ and $x_{45,3,2}$.

(c) Mixing variables $x_{56,1,1}$, $x_{56,2,1}$ and $x_{56,3,1}$.

(d) Mixing variables $x_{56,1,2}$, $x_{56,2,2}$ and $x_{56,3,2}$.

(e) Mixing variables $x_{57,1,1}$, $x_{57,2,1}$ and $x_{57,3,1}$.

(f) Mixing variables $x_{57,1,2}$, $x_{57,2,2}$ and $x_{57,3,2}$.

Fig. 3: Convergence of Algorithm 2 (CFL for Subproblem 2) for the network in Fig. 1. $a = 0.1$ and $b = 0.1$. These convergence curves are for one realization of the random Algorithm 2.

Algorithm 2 CFL for Subproblem 2 (Feasible Discrete Mixing)

Output: $\mathbf{x} \in \boldsymbol{\mathcal{M}}(L)$

For all $(i,j) \in \mathcal{E}$, $p \in \mathcal{P}$ and $l \in \mathcal{L}$, edge (i,j) initializes $q_{ij,p,l}(x) = \frac{1}{2}$, where $x \in \{0,1\}$.
loop
  For all $(i,j) \in \mathcal{E}$, $p \in \mathcal{P}$ and $l \in \mathcal{L}$, edge (i,j) realizes a random variable, selecting $x_{ij,p,l} = x$ with probability $q_{ij,p,l}(x)$, where $x \in \{0,1\}$.
  for $(i,j) \in \mathcal{E}$, $p \in \mathcal{P}$ and $l \in \mathcal{L}$ **do**
    Each edge (i,j) evaluates all the clauses in $\Phi_{ij,p,l}$.
    if all clauses in $\Phi_{ij,p,l}$ are satisfied **then**
      set $q_{ij,p,l}(x) = \begin{cases} 1, & \text{if } x = x_{ij,p,l} \\ 0, & \text{otherwise} \end{cases}$
    else
      set $q_{ij,p,l}(x) = \begin{cases} (1-b)q_{ij,p,l}(x) + \frac{a}{1+a/b}, & \text{if } x = x_{ij,p,l} \\ (1-b)q_{ij,p,l}(x) + \frac{b}{1+a/b}, & \text{otherwise} \end{cases}$
      where $a, b \in (0,1]$ are design parameters.
    end if
  end for
end loop

between adjacent nodes for each node i to evaluate its related clauses in (31); based on the evaluation, each node i updates the distribution of the Bernoulli random variable selecting $x_{ij,p,l}$. The details are summarized in Algorithm 2, which obtains a feasible solution to Subproblem 2 using CFL. Based on the convergence result of CFL [27, Corollary 2], we know that Algorithm 2 can find a feasible solution to Subproblem 2 in almost surely finite time. Fig. 3 illustrates the convergence of Algorithm 2 for the network in Fig. 1. From Fig. 3, we can see that Algorithm 2 converges to a feasible solution (i.e., the feasible solution illustrated in Example 1) to Subproblem 2 quite quickly (within 40 iterations).

Now, we can develop a distributed algorithm to solve Problem 2, relying on the distributed algorithm for Subproblem 1 (i.e., Algorithm 1) and the distributed algorithm for Subproblem 2 (i.e., Algorithm 2), as briefly illustrated in Algorithm 3.[8] Based on the convergence results for Algorithm 1 and Algorithm 2, we can easily see that $U_n \to U^*(L)$ almost surely as $n \to \infty$. Fig. 4 illustrates the convergence of Algorithm 3 at one instance for the network in Fig. 1. From Fig. 4, we can see that Algorithm 3 obtains the optimal network cost 28 to Problem 2 (Problem 1) quite quickly (within 5 iterations for the outer loop).

[8]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 3 CFL-based Optimization for Problem 2 (Discrete Mixing)

initialize $n = 1$ and $U_1 = +\infty$.
loop
  Run the CFL in Algorithm 2.
  if the CFL finds a feasible solution $\mathbf{x}$ to Subproblem 2 **then**
    For the obtained $\mathbf{x}$, run Algorithm 1 to obtain the optimal solution $(\mathbf{z}^*(\mathbf{x}), \mathbf{f}^*(\mathbf{x}))$ to Subproblem 1. Let $\bar{U}_n$ denote the corresponding network cost $U_x^*(\mathbf{x})$.
    Set $U_n = \min\{\bar{U}_n, U_n\}$, $U_{n+1} = U_n$ and $n = n + 1$.
  end if
end loop

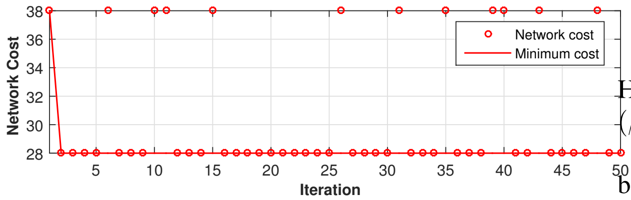


Fig. 4: Convergence of Algorithm 3 (CFL-based Optimization for Problem 2) for the network in Fig. 1. Each dot represents the network cost (obtained by Algorithm 1) of a feasible solution (obtained Algorithm 2). While the curve represents the minimum network cost obtained by Algorithm 3 within a certain number of iterations. The dots and curve are for one realization of the random Algorithm 3.

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 propose a distributed algorithm to solve it.

A. Alternative Formulation

Problem 1 is a mixed discrete-continuous optimization problem. Applying continuous relaxation to (1) and (2) and manipulating (9), we obtain the following continuous optimization problem.

Problem 3 (Equivalent Continuous Mixing for Problem 1):

$$\bar{U}^*(L) = \min_{\mathbf{z},\mathbf{f},\bar{\mathbf{x}},\bar{\boldsymbol{\beta}}} \sum_{(i,j)\in\mathcal{E}} U_{ij}\left(\sum_{L\in\mathcal{L}} z_{ij,l}\right)$$

$$s.t. \ (3), (4), (5), (6), (8), (10)$$

$$\bar{x}_{ij,p,l} \in [0,1], \ (i,j) \in \mathcal{E}, \ p \in \mathcal{P}, \ l \in \mathcal{L} \quad (32)$$

$$\bar{\beta}_{kij,l,m} \in [0,1], \ (k,i),(i,j) \in \mathcal{E}, \ l,m \in \mathcal{L} \quad (33)$$

$$f^t_{ij,p,l} \le \bar{x}_{ij,p,l} B_{ij}, \ (i,j) \in \mathcal{E}, \ p \in \mathcal{P}_t, \ t \in \mathcal{T}, \ l \in \mathcal{L} \quad (34)$$

$$\bar{x}_{ij,p,m} \ge \bar{\beta}_{kij,l,m}\bar{x}_{ki,p,l}, \ k \in \mathcal{I}_i, \ \mathcal{I}_i \ne \emptyset, \ (i,j) \in \mathcal{E}, \ p \in \mathcal{P}, \ l,m \in \mathcal{L} \quad (35)$$

$$\bar{x}_{ij,p,m} \le \sum_{k\in\mathcal{I}_i, l\in\mathcal{L}} \bar{\beta}_{kij,l,m}\bar{x}_{ki,p,l}, \ \mathcal{I}_i \ne \emptyset, \ (i,j) \in \mathcal{E}, \ p \in \mathcal{P}, \ m \in \mathcal{L}. \quad (36)$$

Here, $\bar{\mathbf{x}} \triangleq (\bar{x}_{ij,p,l})_{(i,j)\in\mathcal{E},p\in\mathcal{P},l\in\mathcal{L}}$ and $\bar{\boldsymbol{\beta}} \triangleq (\bar{\beta}_{kij,l,m})_{(k,i),(i,j)\in\mathcal{E},l,m\in\mathcal{L}}$.

Note that Constraints (32) and (33) in Problem 3 can be treated as the continuous relaxation of Constraints (1) and (2) in Problem 1. Constraint (34) in Problem 3 corresponds to Constraint (7) in Problem 1. Constraints (35) and (36) in Problem 3 can be treated as the continuous counterpart of Constraint (9) 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 $(\mathbf{z}, \mathbf{f}, \mathbf{x}, \boldsymbol{\beta})$ is a feasible solution to Problem 1, then $(\mathbf{z}, \mathbf{f}, \bar{\mathbf{x}}, \bar{\boldsymbol{\beta}})$ 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 $(\mathbf{z}, \mathbf{f}, \bar{\mathbf{x}}, \bar{\boldsymbol{\beta}})$ is a feasible solution to Problem 3, then $(\mathbf{z}, \mathbf{f}, \mathbf{x}, \boldsymbol{\beta})$ 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., $U^*(L) = \bar{U}^*(L)$.

Proof: (Sketch) We can easily show that (i) implies (ii) and (iii). Thus, to show Lemma 3, it is sufficient to show (i). To show (i), we first show that Constraint (9) is equivalent to two constraints. Then, we show the first statement of (i) based on the fact that Constraints (32), (33) and (34) in Problem 3 can be treated as the continuous relaxations of Constraints (1), (2) and (7) in Problem 1, respectively; Constraints (35) and (36) in Problem 3 can be treated as the continuous relaxations of the two equivalent constraints of Constraint (9) in Problem 1. Finally, we show the second statement of (i) by showing that a feasible solution of Problem 3 satisfies Constraints (32), (33), (34) and (9). Please refer to Appendix B for the detailed proof. ■

By Lemma 3, solving Problem 1 is equivalent to solving Problem 3. Problem 3 is a (pure) continuous optimization problem. It is not convex due to the constraints in (35) and (36). In general, we can obtain a stationary point to a non-convex (continuous) problem with polynomial-time complexity.

B. Distributed Solution

In this part, we develop a distributed algorithm to obtain a stationary point of Problem 3 with polynomial-time complexity, by using penalty methods [28], the basic idea of which is to eliminate some or all of the constraints and add to the objective function a penalty term that prescribes a high cost to infeasible points.

First, by eliminating the non-convex constraints in (35) and (36) and adding to the objective function of Problem 3 a penalty term reflecting a high cost of violating (35) and (36), we introduce the augmented Lagrangian function $L_c\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}\right)$ given in (37), where $\overline{\boldsymbol{\nu}} \triangleq \left(\overline{\nu}_{kij,p,l,m}\right)_{(k,i),(i,j)\in\mathcal{E},p\in\mathcal{P},m,l\in\mathcal{L}} \succeq \mathbf{0}$ and $\underline{\boldsymbol{\nu}} \triangleq \left(\underline{\nu}_{ij,p,m}\right)_{(i,j)\in\mathcal{E},p\in\mathcal{P},m\in\mathcal{L}} \succeq \mathbf{0}$ denote the Lagrangian multipliers corresponding to the constraints in (35) and (36), respectively, and

$$\overline{g}_{kij,p,l,m} \triangleq \bar{\beta}_{kij,l,m}\bar{x}_{ki,p,l} - \bar{x}_{ij,p,m} \quad (38)$$

$$\underline{g}_{ij,p,m} \triangleq -\sum_{k\in\mathcal{I}_i,l\in\mathcal{L}} \bar{\beta}_{kij,l,m}\bar{x}_{ki,p,l} + \bar{x}_{ij,p,m}. \quad (39)$$

Here, the second and third terms in the augmented Lagrangian function in (37) are the penalty terms that prescribe high costs to infeasible points violating the non-convex constraints in (35) and (36), and c is a positive penalty parameter which determines the severity of the penalty.

We now consider an approximated problem to Problem 3 which minimizes the augmented Lagrangian function in (37) subject to the constraints of Problem 3 except (35) and (36).

Problem 4 (Penalty Approximation for Problem 3): For given $c > 0$, $\overline{\boldsymbol{\nu}} \succeq \mathbf{0}$ and $\underline{\boldsymbol{\nu}} \succeq \mathbf{0}$, we have:

$$\min_{\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}}} \quad L_c\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}\right)$$
$$s.t. \quad \left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}}\right) \in \mathcal{X},$$

where $\mathcal{X} \triangleq \left\{\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}}\right) : (3),(4),(5),(6),(8),(10),(32),(33),(34)\right\}$. The objective function of Problem 4 is differentiable but non-convex. The constraint set $\mathcal{X}$ of Problem 4 is convex. In general, for given $(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})$, we can only obtain a stationary point of Problem 4, denoted as $\left(\mathbf{z}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\mathbf{x}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\boldsymbol{\beta}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})\right)$, e.g., using gradient projection methods, which will be illustrated later.

As c increases, the approximated problem in Problem 4 becomes increasingly accurate to Problem 3. The penalty method for Problem 3 consists of a sequence of problems obtaining a stationary point of the form in Problem 4 with increasing c. The details of the penalty method for Problem 3 is summarized in Algorithm 4. The update equations in Algorithm 4 are given by:

Algorithm 4 Penalty Method for Problem 3 (Continuous Mixing)

OUTPUT: $\mathbf{z}^\dagger,\overline{\mathbf{x}}^\dagger,\overline{\boldsymbol{\beta}}^\dagger$

1: **initialize** $n = 0,\ c(0) = 1,\ \overline{\boldsymbol{\nu}}(0)$ and $\underline{\boldsymbol{\nu}}(0)$.
2: **loop**
3: Compute a stationary point $\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n)\right)$ of Problem 4, e.g., using Algorithm 5, i.e., $\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n)\right) = \left(\mathbf{z}^\dagger(c(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n)),\overline{\mathbf{x}}^\dagger(c(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n)),\overline{\boldsymbol{\beta}}^\dagger(c(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n))\right)$ obtained by Algorithm 5.
4: For all $(i,j) \in \mathcal{E}$ and $l,m \in \mathcal{L}$, each edge (i,j) updates $c(n+1)$, $\overline{\nu}_{kij,p,l,m}(n+1)$ and $\underline{\nu}_{ij,p,m}(n+1)$ according to (40), (41) and (42), respectively.
5: Set $n = n+1$.
6: **end loop**

$$c(n+1) = \beta c(n) \quad (40)$$

$$\overline{\nu}_{kij,p,l,m}(n+1) = \left(\overline{\nu}_{kij,p,l,m}(n) + c(n)\overline{g}_{kij,p,l,m}(n)\right)^+ \quad (41)$$

$$\underline{\nu}_{ij,p,m}(n+1) = \left(\underline{\nu}_{ij,p,m}(n) + c(n)\underline{g}_{ij,p,m}(n)\right)^+. \quad (42)$$

Here, $\overline{g}_{kij,p,l,m}(n)$ and $\underline{g}_{ij,p,m}(n)$ denote the the values of the functions in (38) and (39) at a stationary point $\left(\mathbf{z}^\dagger(c(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n)),\overline{\mathbf{x}}^\dagger(c(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n)),\overline{\boldsymbol{\beta}}^\dagger(c(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n))\right)$ of Problem 4 at the n-th iteration, which can be obtained in a distributed manner using the gradient projection algorithm in Algorithm 5. We shall illustrate the details of Algorithm 5 later. In addition, the update equations in Step 4 can be computed at each edge based on local information. Therefore, Algorithm 4 can be implemented in a distributed manner. As the number of iterations n goes to infinity, we can obtain a stationary point of Problem 3, as summarized in the following theorem.

Theorem 1 (Convergence of Algorithm 4): As $n \to \infty$, $\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n)\right) \to \left(\mathbf{z}^\dagger,\overline{\mathbf{x}}^\dagger,\overline{\boldsymbol{\beta}}^\dagger\right)$, where $\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n)\right)$ is given by the n-th iteration of Algorithm 4, and $\left(\mathbf{z}^\dagger,\overline{\mathbf{x}}^\dagger,\overline{\boldsymbol{\beta}}^\dagger\right)$ is a stationary point of Problem 3.[9]

Proof: Please refer to Appendix C. ■

Fig. 5 illustrates the convergence of Algorithm 4 for the network in Fig. 1. From Fig. 5, we can see that as n increases, the non-convex constraints in (35) and (36) tend to be satisfied, and the network cost goes to 28, which is the optimal network cost to

[9] The constraint set of Problem 3 can be written as $\left\{\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}}\right) : (35),(36)\right\} \cap \mathcal{X}$, in terms of $\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}}\right)$, where $\mathcal{X}$ is the constraint set of Problem 4.

$$L_c\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}\right)=\sum_{(i,j)\in\mathcal{E}}U_{ij}\left(\sum_{l\in\mathcal{L}}z_{ij,l}\right)+\frac{1}{2c}\sum_{\substack{(k,i),(i,j)\in\mathcal{E},\\ p\in\mathcal{P},l,m\in\mathcal{L}}}\left(\left(\left(\overline{\nu}_{kij,p,l,m}+c\overline{g}_{kij,p,l,m}\right)^+\right)^2-\overline{\nu}^2_{kij,p,l,m}\right)$$
$$+\frac{1}{2c}\sum_{\substack{\mathcal{I}_i\neq\emptyset,\ (i,j)\in\mathcal{E},\\ p\in\mathcal{P},m\in\mathcal{L}}}\left(\left(\left(\underline{\nu}_{ij,p,m}+c\underline{g}_{ij,p,m}\right)^+\right)^2-\underline{\nu}^2_{ij,p,m}\right) \tag{37}$$

Problem 2 (Problem 1). Algorithm 4 converges quite quickly (within 5 iterations for the outer loop).

Now, we focus on obtaining a stationary point $\left(\mathbf{z}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\mathbf{x}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\boldsymbol{\beta}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})\right)$ of Problem 4, using gradient projection methods [28, pp. 228]. We first compute the partial derivatives of $L_c\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}\right)$ in (37) as follows:

$$\frac{\partial L_c}{\partial z_{ij,l}}=U'_{ij}\left(\sum_{m\in\mathcal{L}}z_{ij,m}\right) \tag{43}$$

$$\begin{aligned}\frac{\partial L_c}{\partial \overline{x}_{ij,p,m}}=&-\sum_{k\in\mathcal{I}_i,l\in\mathcal{L}}\left(\overline{\nu}_{kij,p,l,m}+c\overline{g}_{kij,p,l,m}\right)^+\\&+\sum_{k\in\mathcal{O}_j,l\in\mathcal{L}}\overline{\beta}_{kij,l,m}\left(\overline{\nu}_{kij,p,m,l}+c\overline{g}_{kij,p,m,l}\right)^+\\&-\sum_{k\in\mathcal{O}_j,l\in\mathcal{L}}\overline{\beta}_{ijk,m,l}\left(\underline{\nu}_{jk,p,l}+c\underline{g}_{jk,p,l}\right)^+\\&+\left(\underline{\nu}_{ij,p,m}+c\underline{g}_{ij,p,m}\right)^+\end{aligned} \tag{44}$$

$$\begin{aligned}\frac{\partial L_c}{\partial \overline{\beta}_{kij,l,m}}=&\sum_{p\in\mathcal{P}}\overline{x}_{ki,p,l}\Bigg(\left(\overline{\nu}_{kij,p,l,m}+c\overline{g}_{kij,p,l,m}\right)^+\\&-\left(\underline{\nu}_{ij,p,m}+c\underline{g}_{ij,p,m}\right)^+\Bigg).\end{aligned} \tag{45}$$

For given $(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})$, the gradient projection method to compute a stationary point $\left(\mathbf{z}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\mathbf{x}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\boldsymbol{\beta}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})\right)$ of Problem 4 is summarized in Algorithm 5. The update equations in Algorithm 5 are given below:

$$z_{ij,l}(n+1)=\left[z_{ij,l}(n)-\epsilon(n)\frac{\partial L_c}{\partial z_{ij,l}}(n)\right]_* \tag{46}$$

$$\overline{x}_{ij,p,m}(n+1)=\left[\overline{x}_{ij,p,m}(n)-\epsilon(n)\frac{\partial L_c}{\partial \overline{x}_{ij,p,m}}(n)\right]_* \tag{47}$$

$$\overline{\beta}_{kij,l,m}(n+1)=\left[\overline{\beta}_{kij,l,m}(n)-\epsilon(n)\frac{\partial L_c}{\partial \overline{\beta}_{kij,l,m}}(n)\right]_*, \tag{48}$$

where the partial derivatives of $L_c\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}\right)$ in (46), (47) and (48) are given by (43), (44) and (45),

Algorithm 5 Gradient Projection Method for Problem 4 (Penalty Approximation)

INPUT: $c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}$

OUTPUT: $\mathbf{z}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\mathbf{x}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\boldsymbol{\beta}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})$

1: **initialize** $n=0$, $\mathbf{z}(0)$, $\overline{\mathbf{x}}(0)$, $\overline{\boldsymbol{\beta}}(0)$.
2: **loop**
3: For all $(i,j)\in\mathcal{E}$, each edge (i,j) updates $z_{ij,l}(n+1)$, $\overline{x}_{ij,p,m}(n+1)$ and $\overline{\beta}_{kij,l,m}(n+1)$ according to (46), (47) and (48), respectively, where the projection $[\ ]_*$ on the constraint set of Problem 4 is computed using Algorithm 6. In other words, $\left(\mathbf{z}(n+1),\overline{\mathbf{x}}(n+1),\overline{\boldsymbol{\beta}}(n+1)\right)=\left(\mathbf{z}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\mathbf{x}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\boldsymbol{\beta}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$ obtained by Algorithm 6, where $z'_{ij,l}=z_{ij,l}(n)-\epsilon(n)\frac{\partial L_c}{\partial z_{ij,l}}(n)$, $\overline{x}'_{ij,p,m}=\overline{\beta}_{kij,l,m}(n)-\epsilon(n)\frac{\partial L_c}{\partial \overline{\beta}_{kij,l,m}}(n)$, and $\overline{\beta}'_{kij,l,m}=\overline{\beta}_{kij,l,m}(n)-\epsilon(n)\frac{\partial L_c}{\partial \overline{\beta}_{kij,l,m}}(n)$.
4: Set $n=n+1$.
5: **end loop**

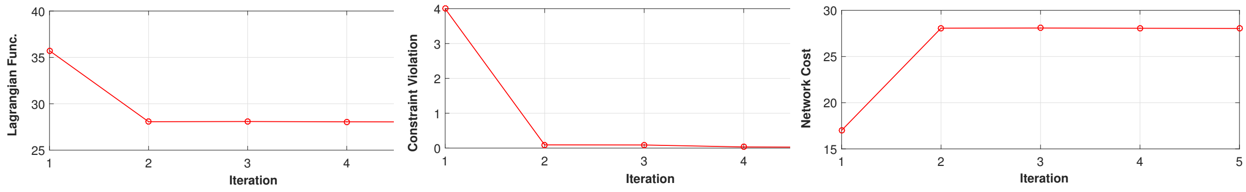
$\{\epsilon(n)\}$ denotes the diminishing stepsize satisfying:

$$\epsilon(n)\to 0 \text{ as } n\to\infty,\ \sum_{n=1}^{\infty}\epsilon(n)=\infty,\sum_{n=1}^{\infty}\epsilon(n)^2<\infty, \tag{49}$$

and $[\cdot]_*$ denotes the projection on the convex constraint set of Problem 4, i.e., the set of solutions satisfying (3)-(6), (8), (10), (32)-(34), which can be obtained in a distributed manner using the primal-dual algorithm in Algorithm 6. We shall illustrate the details of Algorithm 6 later. It has been shown that as $n\to\infty$, $\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n)\right)$ converges to a stationary point $\left(\mathbf{z}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\mathbf{x}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}),\overline{\boldsymbol{\beta}}^\dagger(c,\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}})\right)$ of Problem 4 [28, pp. 232]. Fig. 6 illustrates the convergence of Algorithm 5 for the network in Fig. 1. We can see that Algorithm 5 converges quite quickly (within 50 iterations for the outer loop).

Next, we study the projection of $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$ on the convex constraint set of Problem 4, i.e., $\left[\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right]_*$. First, define the distance between $\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}}\right)$ and $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$ as follows:

$$\begin{aligned}&D\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\\&\triangleq\sum_{(i,j)\in\mathcal{E},l\in\mathcal{L}}\left(z_{ij,l}-{z'}_{ij,l}\right)^2+\sum_{(i,j)\in\mathcal{E},l\in\mathcal{L}}\left(\bar{x}_{ij,p,l}-\bar{x}'_{ij,p,l}\right)^2\end{aligned}$$



(a) Augmented Lagrangian function. (b) Total constraint violation. (c) Network cost.

Fig. 5: Convergence of Algorithm 4 (Penalty Method for Problem 3) for the network in Fig. 1. In (a), the curve represents the augmented Lagrangian function $L_{c(n)}\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n),\overline{\boldsymbol{\nu}}(n),\underline{\boldsymbol{\nu}}(n)\right)$ at the n-th iteration, where $L_c(\cdot)$ is given by (37). In (b), the curve represents $\sum_{(k,i),(i,j)\in\mathcal{E},p\in\mathcal{P},l,m\in\mathcal{L}}\left(\overline{g}_{kij,p,l,m}(n)\right)^{+}+\sum_{\mathcal{I}_i\neq\emptyset,\ (i,j)\in\mathcal{E},p\in\mathcal{P},m\in\mathcal{L}}\left(\underline{g}_{ij,p,m}(n)\right)^{+}$ at the n-th iteration. In (c), the curve represents $\sum_{(i,j)\in\mathcal{E}}U_{ij}\left(\sum_{l\in\mathcal{L}}z_{ij,l}(n)\right)$ at the n-th iteration.

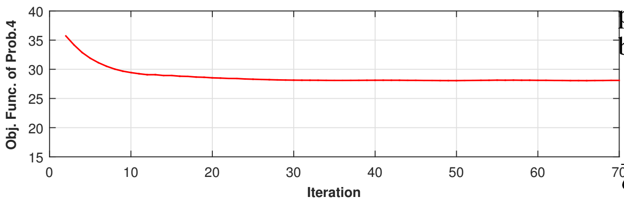


Fig. 6: Convergence of Algorithm 5 (Gradient Projection Method for Problem 4) for the network in Fig. 1. The curve represents $L_c\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n),\overline{\boldsymbol{\nu}},\underline{\boldsymbol{\nu}}\right)$ at the n-th iteration for given $c>0$, $\overline{\boldsymbol{\nu}}\succeq\mathbf{0}$ and $\underline{\boldsymbol{\nu}}\succeq\mathbf{0}$, where $L_c(\cdot)$ is given by (37).

$$+\sum_{k\in\mathcal{I}_i\neq\emptyset,(i,j)\in\mathcal{E},l,m\in\mathcal{L}}\left(\bar{\beta}_{kij,l,m}-\bar{\beta}'_{kij,l,m}\right)^2. \quad (50)$$

The projection of $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$ on the convex constraint set of Problem 4 can be obtained by solving the following problem.

Problem 5 (Projection on Constraint Set of Problem 4): For given $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$, we have:

$$\begin{aligned}\min_{\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{f}}\quad & D\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\\ s.t.\quad & (3),(4),(5),(6),(32),(33),(34).\end{aligned}$$

The optimal solution to Problem 5 is written as $\left(\mathbf{z}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\mathbf{x}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\boldsymbol{\beta}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\mathbf{f}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$. In addition, we have $\left[\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right]_* = \left(\mathbf{z}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\mathbf{x}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\boldsymbol{\beta}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$.

Problem 5 is convex and can be solved using the primal-dual method. By relaxing the constraints in (4), (5), (6) and (34) of Problem 5, for given $(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}')$, we have the following Lagrangian function $L\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}\right)$ given in (51), where $\boldsymbol{\lambda}\triangleq(\lambda^t_{ij,l})_{(i,j)\in\mathcal{E},t\in\mathcal{T},l\in\mathcal{L}}\succeq\mathbf{0}$, $\boldsymbol{\eta}\triangleq(\eta_{ij})_{(i,j)\in\mathcal{E}}\succeq\mathbf{0}$, $\boldsymbol{\mu}\triangleq(\mu^t_{i,p})_{i\in\mathcal{V},p\in\mathcal{P}_t,t\in\mathcal{T}}$ and $\boldsymbol{\xi}\triangleq(\xi^t_{ij,p,l})_{(i,j)\in\mathcal{E},p\in\mathcal{P}_t,t\in\mathcal{T},l\in\mathcal{L}}\succeq\mathbf{0}$ denote the Lagrangian multipliers w.r.t. the constraints in (4), (5), (6) and (34) of Problem 5, respectively, with abuse of notations. The partial derivatives of $L\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}\right)$ are given by:

$$\frac{\partial L}{\partial z_{ij,l}}=2\left(z_{ij,l}-z'_{ij,l}\right)-\sum_{t\in\mathcal{T}}\lambda^t_{ij,l}+\eta_{ij} \quad (52)$$

$$\frac{\partial L}{\partial \bar{x}_{ij,p,m}}=2\left(\bar{x}_{ij,p,m}-\bar{x}'_{ij,p,m}\right)-\sum_{t\in\{t:p\in\mathcal{P}_t\}}\xi^t_{ij,p,m}B_{ij} \quad (53)$$

$$\frac{\partial L}{\partial \bar{\beta}_{kij,l,m}}=2\left(\bar{\beta}_{kij,l,m}-\bar{\beta}'_{kij,l,m}\right) \quad (54)$$

$$\frac{\partial L}{\partial f^t_{ij,p,l}}=\lambda^t_{ij,l}+\mu^t_{i,p}\mathbf{1}\left[\mathcal{O}_i\neq\emptyset\right]-\mu^t_{j,p}\mathbf{1}\left[\mathcal{I}_j\neq\emptyset\right]+\xi^t_{ij,p,l} \quad (55)$$

$$\frac{\partial L}{\partial \lambda^t_{ij,l}}=\sum_{p\in\mathcal{P}_t}f^t_{ij,p,l}-z_{ij,l} \quad (56)$$

$$\frac{\partial L}{\partial \eta_{ij}}=\sum_{l\in\mathcal{L}}z_{ij,l}-B_{ij} \quad (57)$$

$$\frac{\partial L}{\partial \mu^t_{i,p}}=\sum_{k\in\mathcal{O}_i,l\in\mathcal{L}}f^t_{ik,p,l}-\sum_{k\in\mathcal{I}_i,l\in\mathcal{L}}f^t_{ki,p,l}-\sigma^t_{i,p} \quad (58)$$

$$\frac{\partial L}{\partial \xi^t_{ij,p,l}}=f^t_{ij,p,l}-\bar{x}_{ij,p,l}B_{ij}. \quad (59)$$

Similar to Subproblem 2 in Section IV, for given $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$, the primal optimal $\left(\mathbf{z}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\mathbf{x}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\boldsymbol{\beta}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\mathbf{f}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$ and the dual optimal

$$\left(\boldsymbol{\lambda}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\eta}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\mu}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\xi}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$$

of Problem 5 can can be obtained using the primal-dual algoritm summarized in Algorithm 6. The update

$$L\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{f},\boldsymbol{\lambda},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\xi}\right)=D\left(\mathbf{z},\overline{\mathbf{x}},\overline{\boldsymbol{\beta}},\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)+\sum_{\substack{(i,j)\in\mathcal{E},\\ t\in\mathcal{T},l\in\mathcal{L}}}\lambda_{ij,l}^{t}\left(\sum_{p\in\mathcal{P}_t}f_{ij,p,l}^{t}-z_{ij,l}\right)+\sum_{(i,j)\in\mathcal{E}}\eta_{ij}\left(\sum_{l\in\mathcal{L}}z_{ij,l}-B_{ij}\right)$$
$$+\sum_{\substack{i\in\mathcal{V},\\ p\in\mathcal{P}_t,t\in\mathcal{T}}}\mu_{i,p,l}^{t}\left(\sum_{k\in\mathcal{O}_i,l\in\mathcal{L}}f_{ik,p,l}^{t}-\sum_{k\in\mathcal{I}_i,l\in\mathcal{L}}f_{ki,p,l}^{t}-\sigma_{i,p}^{t}\right)+\sum_{\substack{(i,j)\in\mathcal{E},p\in\mathcal{P}_t,\\ t\in\mathcal{T},l\in\mathcal{L}}}\xi_{ij,p,l}^{t}\left(f_{ij,p,l}^{t}-\overline{x}_{ij,p,l}B_{ij}\right) \tag{51}$$

equations in Algorithm 6 are given below:

$$z_{ij,l}(n+1)=z_{ij,l}(n)-\gamma(n)\frac{\partial L}{\partial z_{ij,l}}(n) \tag{60}$$

$$\overline{x}_{ij,p,l}(n+1)=\left[\overline{x}_{ij,p,l}(n)-\gamma(n)\frac{\partial L}{\partial \overline{x}_{ij,p,l}}(n)\right]_{[0,1]} \tag{61}$$

$$\overline{\beta}_{kij,l,m}(n+1)=\left[\overline{\beta}_{kij,l,m}(n)-\gamma(n)\frac{\partial L}{\partial \overline{\beta}_{kij,l,m}}(n)\right]_{[0,1]} \tag{62}$$

$$f_{ij,p,l}^{t}(n+1)=\left(f_{ij,p,l}^{t}(n)-\gamma(n)\frac{\partial L}{\partial f_{ij,p,l}^{t}}(n)\right)^{+} \tag{63}$$

$$\lambda_{ij,l}^{t}(n+1)=\left(\lambda_{ij,l}^{t}(n)+\gamma(n)\frac{\partial L}{\partial \lambda_{ij,l}^{t}}(n)\right)^{+} \tag{64}$$

$$\eta_{ij}(n+1)=\left(\eta_{ij}(n)+\gamma(n)\frac{\partial L}{\partial \eta_{ij}}(n)\right)^{+} \tag{65}$$

$$\mu_{i,p}^{t}(n+1)=\mu_{i,p}^{t}(n)+\gamma(n)\frac{\partial L}{\partial \mu_{i,p}^{t}}(n) \tag{66}$$

$$\xi_{ij,p,l}^{t}(n+1)=\left(\xi_{ij,p,l}^{t}(n)+\gamma(n)\frac{\partial L}{\partial \xi_{ij,p,l}^{t}}(n)\right)^{+}, \tag{67}$$

where the partial derivatives of $L\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n),\mathbf{f}(n),\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n),\boldsymbol{\xi}(n)\right)$ in (60), (61), (62), (63), (64), (65), (66) and (67) are given by (52), (53), (54), (55), (56), (57), (58) and (59), $[x]_{[0,1]}$ denotes the projection of x on [0,1], and $\{\gamma(n)\}$ denotes the diminishing stepsize satisfying

$$\gamma(n)\to 0 \text{ as } n\to\infty,\ \sum_{n=1}^{\infty}\gamma(n)=\infty,\sum_{n=1}^{\infty}\gamma(n)^2<\infty. \tag{68}$$

Note that Algorithm 6 can be implemented in a distributed manner. In addition, it has been shown [32] that as $n\to\infty$, $\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n),\mathbf{f}(n)\right)\to\left(\mathbf{z}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\mathbf{x}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\boldsymbol{\beta}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\mathbf{f}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$ and $\left(\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n),\boldsymbol{\xi}(n)\right)\to\left(\boldsymbol{\lambda}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\eta}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\mu}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\xi}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right)$. In other words, for given $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$, Algorithm 6 converges to the projection of $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$ on the convex constraint set of Problem 4, i.e., $\left[\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)\right]_*$, as $n\to\infty$. Fig. 7 illustrates the convergence of Algorithm 6 for the network in Fig. 1.

Algorithm 6 Primal-dual Method for Problem 5 (Projection)

INPUT: $\overline{\mathbf{z}}'$, $\overline{\mathbf{x}}'$ and $\overline{\boldsymbol{\beta}}'$
OUTPUT: $\mathbf{z}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\mathbf{x}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\overline{\boldsymbol{\beta}}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\mathbf{f}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$, $\boldsymbol{\lambda}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\eta}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right),\boldsymbol{\mu}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$ and $\boldsymbol{\xi}^*\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$

1: **initialize** $n=0$, $\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n),\mathbf{f}(n),\boldsymbol{\lambda}(n),\boldsymbol{\eta}(n),\boldsymbol{\mu}(n)$ and $\boldsymbol{\xi}(n)$.
2: **loop**
3: Update $z_{ij,l}(n+1),\overline{x}_{ij,p,l}(n+1),\overline{\beta}_{kij,l,m}(n+1),f_{ij,p,l}^{t}(n+1),\lambda_{ij,l}^{t}(n+1),\eta_{ij}(n+1),\mu_{i,p}^{t}(n+1)$ and $\xi_{ij,p,l}^{t}(n+1)$ according to (60), (61), (62), (63), (64), (65), (66) and (67), respectively.
4: Set $n=n+1$.
5: **end loop**

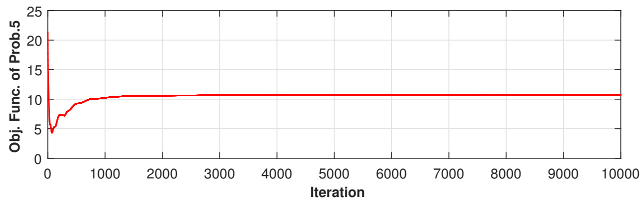


Fig. 7: Convergence of Algorithm 6 (Primal-dual Method for Problem 5) for the network in Fig. 1. The curve represents $D\left(\mathbf{z}(n),\overline{\mathbf{x}}(n),\overline{\boldsymbol{\beta}}(n),\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$ at the n-th iteration for given $\left(\mathbf{z}',\overline{\mathbf{x}}',\overline{\boldsymbol{\beta}}'\right)$, where $D(\cdot)$ is given by (50).

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 existing solutions.

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 $\{f^t_{ij,p}\}$ denote a feasible solution to Problem 1. We shall extend the proof of Lemma 1 in [14], [24] for the integer flows ($f^t_{ij,p} \in \{0, 1\}$) and unit source rates ($R_p = 1$) with one global coding vector over each edge ($z_{ij} \in \{0, 1\}$) to the general continuous flows ($f^t_{ij,p} \in [0, B_{ij}]$) and source rates ($R_p \in \mathbb{R}^+$) with multiple global coding vectors ($z_{ij} \in [0, B_{ij}]$) 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 $\lceil nf^t_{ij,p} \rceil$. 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 $\lceil nf^t_{ij,p} \rceil$. The sum rate of unit rate sub-flows of all the flows over edge (i, j) is less than or equal to $\bar{z}_{ij} = \max_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}_t} \lceil nf^t_{ij,p} \rceil$. 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 [14], [24]. 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 $U^*(1)$ 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 $\{z_{ij,l}\}$ associated with a feasible solution to Problem 1 is also associated with a feasible solution to Problem 3, and vice versa. By noting that $\{z_{ij,l}\}$ fully determines $\sum_{(i,j) \in \mathcal{E}} U_{ij}(\sum_{l \in \mathcal{L}} z_{ij,l})$, the two related feasible solutions for the two problems have the same network cost. 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 (9) is equivalent to the following two constraints in (69) and (70).

$$x_{ij,p,m} \geq \beta_{kij,l,m} x_{ki,p,l}, \; k \in \mathcal{I}_i, \; l \in \mathcal{L} \quad m \in \mathcal{L}, \; \mathcal{I}_i \neq \emptyset, \; (i, j) \in \mathcal{E}, p \in \mathcal{P} \tag{69}$$

$$x_{ij,p,m} \leq \sum_{k \in \mathcal{I}_i, l \in \mathcal{L}} \beta_{kij,l,m} x_{ki,p,l}, \quad m \in \mathcal{L}, \; \mathcal{I}_i \neq \emptyset, \; (i, j) \in \mathcal{E}, p \in \mathcal{P}. \tag{70}$$

Note that Constraints (9), (69) and (70) are for all $m \in \mathcal{L}$, $\mathcal{I}_i \neq \emptyset$, $(i, j) \in \mathcal{E}$ and $p \in \mathcal{P}$. Thus, we prove this equivalence by considering the following two cases for any $m \in \mathcal{L}$, $\mathcal{I}_i \neq \emptyset$, $(i, j) \in \mathcal{E}$ 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 $l \in \mathcal{L}$. Constraint (9) implies that $x_{ij,p,m} = 0$, and Constraints (69) and (70) 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 $l \in \mathcal{L}$, such that $\beta_{kij,l,m} x_{ki,p,l} = 1$. Constraint (9) implies that $x_{ij,p,m} = 1$, and Constraints (69) and (70) 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 (9) is equivalent to Constraints (69) and (70).

Next, we show that the first statement of (i) holds. Suppose $\{z_{ij,l}\}, \{f^t_{ij,p,l}\}, \{x_{ij,p,l}\}, \{\beta_{kij,l,m}\}$ is a feasible solution to Problem 1. Let $\bar{x}_{ij,p,l} = x_{ij,p,l} \in \{0, 1\}$ for all $l \in \mathcal{L}$, $(i, j) \in \mathcal{E}$ 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 $l, m \in \mathcal{L}$. Since Constraints (32), (33) and (34) in Problem 3 can be treated as the continuous relaxation of Constraints (1), (2) and (7) in Problem 1, $\{f^t_{ij,p,l}\}, \{\bar{x}_{ij,p,l}\}, \{\bar{\beta}_{kij,l,m}\}$ satisfies Constraints (32), (33) and (34). In addition, since Constraint (9) is equivalent to Constraints (69) and (70), and Constraints (35) and (36) can be treated as the continuous relaxation of Constraints (69) and (70), $\{\bar{x}_{ij,p,l}\}, \{\bar{\beta}_{kij,l,m}\}$ satisfies Constraints (35) and (36). Therefore, we can show $\{z_{ij,l}\}, \{f^t_{ij,p,l}\}, \{\bar{x}_{ij,p,l}\}, \{\bar{\beta}_{kij,l,m}\}$ is a feasible solution to Problem 3.

Finally, we show that the second statement of (i) holds. Suppose $\{z_{ij,l}\}, \{f^t_{ij,p,l}\}, \{\bar{x}_{ij,p,l}\}, \{\bar{\beta}_{kij,l,m}\}$ is a feasible solution to Problem 3. Let $x_{ij,p,l} = \lceil \bar{x}_{ij,p,l} \rceil$ for all $l \in \mathcal{L}$, $(i, j) \in \mathcal{E}$ 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 $l, m \in \mathcal{L}$. 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 $\bar{x}_{ij,p,l} \in (0, 1]$ ($\bar{\beta}_{kij,l,m} \in (0, 1]$), then $x_{ij,p,l} = 1$ ($\beta_{kij,l,m} = 1$). It is obvious that $\{f^t_{ij,p,l}\}, \{x_{ij,p,l}\}, \{\beta_{kij,l,m}\}$ satisfies Constraints (1), (2) and (7). It remains to show $\{x_{ij,p,l}\}, \{\beta_{kij,l,m}\}$ satisfies Constraint (9). Note that Constraint (9) is for all $m \in \mathcal{L}$, $\mathcal{I}_i \neq \emptyset$, $(i, j) \in \mathcal{E}$ and $p \in \mathcal{P}$. Thus, similarly, we prove this result by considering the following two cases for any $m \in$

$\mathcal{L}$, $\mathcal{I}_i \neq \emptyset$, $(i,j) \in \mathcal{E}$ 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 $l \in \mathcal{L}$. Constraints (35) and (36) 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 $l \in \mathcal{L}$ 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 $l \in \mathcal{L}$. Thus, in this case, we can show $\{x_{ij,p,l}\}, \{\beta_{kij,l,m}\}$ satisfies Constraint (9). Second, consider the case where there exists at least one pair (k,l), where $k \in \mathcal{I}_i$ and $l \in \mathcal{L}$, such that $\bar{\beta}_{kij,l,m}\bar{x}_{ki,p,l} \in (0,1]$. Constraints (35) and (36) together with Constraints (32) and (33) 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 (32) and (33) 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 (9). 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 $\{z_{ij,l}\}, \{f^t_{ij,p,l}\}, \{x_{ij,p,l}\}, \{\beta_{kij,l,m}\}$ is a feasible solution to Problem 1.

Therefore, we complete the proof of Lemma 3.

APPENDIX C: PROOF OF THEOREM 1

In the following, we prove a theorem, i.e., Theorem 2, which is more general than Theorem 1. For ease of illustration, we first introduce some notations. Denote $\mathbf{x} \triangleq (x_1, \cdots, x_n)$, $\mathbf{z} \triangleq (z_1, \cdots, z_r)$ and $\nabla_{\mathbf{x}} f \triangleq \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n}\right)^T$, where $r \leq n$. Consider the following optimization problem.

Problem 6 (Equality and Inequality Constrained Problem):

$$\begin{aligned} &\min_{\mathbf{x}} \ f(\mathbf{x}) \\ &s.t.\ h_i(\mathbf{x}) = 0, \cdots, h_m(\mathbf{x}) = 0, \\ &\quad\ g_j(\mathbf{x}) \leq 0, \cdots, g_r(\mathbf{x}) \leq 0. \end{aligned}$$

Its augmented Lagrangian function is given by [28, pp. 406]:

$$\begin{aligned} L_c(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) =& f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \frac{c}{2}\|\mathbf{h}(\mathbf{x})\|^2 \\ &+ \frac{1}{2c}\sum_{j=1}^{r}\left((\max(0, \mu_j + c g_j(\mathbf{x})))^2 - \mu_j^2\right), \end{aligned} \tag{71}$$

where $\mathbf{h} \triangleq (h_1, \cdots, h_m)$ and $\boldsymbol{\lambda} \triangleq (\lambda_1, \cdots, \lambda_m)^T$. Convert Problem 6 to the following problem [28, pp. 406]:

Problem 7 (Equality Constrained Problem):

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} \ f(\mathbf{x}) \\ &s.t.\ h_i(\mathbf{x}) = 0, \cdots, h_m(\mathbf{x}) = 0, \\ &\quad\ g_j(\mathbf{x}) + z_j^2 = 0, \cdots, g_r(\mathbf{x}) + z_r^2 = 0. \end{aligned}$$

Its augmented Lagrangian function is given by [28, 398]:

$$\begin{aligned} \bar{L}_c(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \frac{c}{2}\|\mathbf{h}(\mathbf{x})\|^2 \\ + \sum_{j=1}^{r}\left(\mu_j\left(g_j(\mathbf{x}) + z_j^2\right) + \frac{c}{2}|g_j(\mathbf{x}) + z_j^2|^2\right), \end{aligned} \tag{72}$$

where $\mathbf{h} \triangleq (h_1, \cdots, h_m)$, $\boldsymbol{\lambda} \triangleq (\lambda_1, \cdots, \lambda_m)^T$ and $\boldsymbol{\mu} \triangleq (\mu_1, \cdots, \mu_r)^T$.

Assume f, h_i, $i = 1, \cdots, m$ and g_j, $j = 1, \cdots, r$ are continously differentiable. Assume the constraint set $\mathcal{X} \triangleq \{\mathbf{x} \in \mathbb{R}^n | h_i(\mathbf{x}) = 0, g_j(\mathbf{x}) \leq 0, i = 1, \cdots m, j = 1, \cdots r\}$ of Problem 6 is nonempty. The following theorem shows that a stationary point of Problem 6 can be obtained using the penalty method considered in this paper. Note that Theorem 2 extends Proposition 4.2.1 in [28]. In addition, Theorem 2 implies Theorem 1.

Theorem 2: For $n = 0, 1, \cdots$, let $\mathbf{x}(n) \in \mathcal{X}$ be a stationary point of $L_{c(n)}(\mathbf{x}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))$, i.e., $\nabla_{\mathbf{x}} L_{c(n)}(\mathbf{x}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))^T(\mathbf{x} - \mathbf{x}(n)) \geq \mathbf{0}$ for all $\mathbf{x} \in \mathcal{X}$, where $\{\boldsymbol{\lambda}(n)\}$ and $\{\boldsymbol{\mu}(n)\}$ are bounded and $\{c(n)\}$ satisfies $0 < c(n) < c(n+1)$ for all n and $c(n) \to \infty$ as $n \to \infty$. Assume $\mathbf{x}(n) \to \mathbf{x}^* \in \mathcal{X}$, $\lambda_i(n) \to \lambda_i^*$ and $\mu_j(n) \to \mu_j^*$, where $\lambda_i(n+1) = \lambda_i(n) + c(n)h_i(\mathbf{x}(n))$, $i = 1, \cdots, m$ and $\mu_j(n+1) = \mu_j(n) + c(n)(g_j(\mathbf{x}(n)) + z_j(n)^2)$, $j = 1, \cdots, r$. Then, $\mathbf{x}^*$ is a stationary point of the problem 6, *i.e.*, $\nabla_{\mathbf{x}} f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$.

Proof: By the proof in [28, 405], we know that for given $c(n)$, $\boldsymbol{\lambda}(n)$ and $\boldsymbol{\mu}(n)$, we have:

$$\begin{aligned} &L_{c(n)}(\mathbf{x}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)) = \min_{\mathbf{z}} \bar{L}_{c(n)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)) \\ =& \bar{L}_{c(n)}(\mathbf{x}, \mathbf{z}(\mathbf{x}, c(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)), \forall \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{73}$$

where

$$\mathbf{z}(\mathbf{x}, c(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)) \triangleq \arg\min_{\mathbf{z}} \bar{L}_{c(n)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)).$$

First, we show that $(\mathbf{x}(n), \mathbf{z}(n))$ is a stationary point of $\bar{L}_{c(n)}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))$. By (73), we have:

$$\begin{aligned} &\nabla_{\mathbf{z}} \bar{L}_{c(n)}(\mathbf{x}, \mathbf{z}(\mathbf{x}, c(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))^T \\ &(\mathbf{z} - \mathbf{z}(\mathbf{x}, c(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))) = 0, \ \forall \mathbf{x} \in \mathcal{X} \\ \Rightarrow& \nabla_{\mathbf{z}} \bar{L}_{c(n)}(\mathbf{x}(n), \mathbf{z}(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))^T(\mathbf{z} - \mathbf{z}(n)) = 0, \ \forall \mathbf{z} \in \mathbb{R}^r, \end{aligned} \tag{74}$$

where $\mathbf{z}(n) \triangleq \mathbf{z}(\mathbf{x}(n), c(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))$. Since $\mathbf{x}(n)$ is a stationary point of $L_{c(n)}(\mathbf{x}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))$, we have:

$$\nabla_{\mathbf{x}} L_{c(n)}(\mathbf{x}(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n))^T(\mathbf{x} - \mathbf{x}(n)) \geq 0, \ \forall \mathbf{x} \in \mathcal{X}. \tag{75}$$

By (73) and (75), we can get:

$$\nabla_{\mathbf{x}} \bar{L}_{c(n)} \left(\mathbf{x}(n), \mathbf{z}(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)\right)^T \left(\mathbf{x} - \mathbf{x}(n)\right) \geq 0, \ \forall \mathbf{x} \in \mathcal{X}. \tag{76}$$

By (74) and (76), we can get:

$$\begin{aligned}&\nabla_{\mathbf{x}} \bar{L}_{c(n)} \left(\mathbf{x}(n), \mathbf{z}(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)\right)^T \left(\mathbf{x} - \mathbf{x}(n)\right) \\ &+ \nabla_{\mathbf{z}} \bar{L}_{c(n)} \left(\mathbf{x}(n), \mathbf{z}(n), \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)\right)^T \left(\mathbf{z} - \mathbf{z}(n)\right) \geq 0, \\ &\forall \mathbf{x} \in \mathcal{X}, \ \forall \mathbf{z} \in \mathbb{R}^r. \end{aligned} \tag{77}$$

Thus, we can show that $(\mathbf{x}(n), \mathbf{z}(n))$ is a stationary point of $\bar{L}_{c(n)} \left(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}(n), \boldsymbol{\mu}(n)\right)$.

Next, we show that $(\mathbf{x}^*, \mathbf{z}^*)$ is a stationary point of $f(\mathbf{x}) + {\boldsymbol{\lambda}^*}^T \mathbf{h}(\mathbf{x}) + \sum_{j=1}^{r} \mu_j^*(g_j(\mathbf{x}) + z_j^2)$. By (72), we know:

$$\begin{aligned}&\nabla_{\mathbf{x}} \bar{L}_c \left(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}\right) = \nabla f(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})\boldsymbol{\lambda} + c\nabla \mathbf{h}(\mathbf{x})\mathbf{h}(\mathbf{x}) \\ &+ \sum_{j=1}^{r} \left(\nabla g_j(\mathbf{x})\mu_j + c(g_j(\mathbf{x}) + z_j^2)\nabla g_j(\mathbf{x})\right) \end{aligned} \tag{78}$$

$$\nabla_{z_j} \bar{L}_c \left(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}\right) = 2\mu_j z_j + 2cz_j \left(g_j(\mathbf{x}) + z_j^2\right), \ j = 1, \cdots, r. \tag{79}$$

Substituting (78) and (79) into (77), we have:

$$\begin{aligned}&\Bigg(\nabla f(\mathbf{x}(n)) + \nabla \mathbf{h}(\mathbf{x}(n)) \left(\boldsymbol{\lambda}(n) + c(n)\mathbf{h}(\mathbf{x}(n))\right) \\ &+ \sum_{j=1}^{r} \nabla g_j(\mathbf{x}(n)) \left(\mu_j(n) + c(n) \left(g_j(\mathbf{x}(n)) + z_j(n)^2\right)\right) \Bigg)^T \\ &(\mathbf{x} - \mathbf{x}(n)) + \sum_{j=1}^{r} \left(2z_j(n) \left(\mu_j(n) + c(n) \left(g_j(\mathbf{x}(n)) + z_j(n)^2\right)\right)\right) \\ &(z_j - z_j(n)) \geq 0, \ \forall \mathbf{x} \in \mathcal{X}, \ \forall z_j \in \mathbb{R}. \end{aligned} \tag{80}$$

Since $\mathbf{x}(n) \to \mathbf{x}^*$, $\lambda_i(n) \to \lambda_i^*$ for all $i = 1, \cdots, m$, $\mu_j(n) \to \mu_j^*$ and $z_j(n) \to z_j^*$ for all $j = 1, \cdots, r$, we have:

$$\begin{aligned}&\left(\nabla f(\mathbf{x}^*) + \nabla \mathbf{h}(\mathbf{x}^*)\boldsymbol{\lambda}^* + \sum_{j=1}^{r} \left(\nabla g_j(\mathbf{x}^*)\mu_j^*\right)\right)^T (\mathbf{x} - \mathbf{x}^*) \\ &+ \sum_{j=1}^{r} 2z_j^*\mu_j^* \left(z_j - z_j^*\right) \geq 0, \ \forall \mathbf{x} \in \mathcal{X}, \ z_j \in \mathbb{R}. \end{aligned} \tag{81}$$

Since the L.H.S of (81) is the gradient of $f(\mathbf{x}) + {\boldsymbol{\lambda}^*}^T \mathbf{h}(\mathbf{x}) + \sum_{j=1}^{r} \mu_j^* \left(g_j(\mathbf{x}) + z_j^2\right)$, we can show that $(\mathbf{x}^*, \mathbf{z}^*)$ is a stationary point of $f(\mathbf{x}) + {\boldsymbol{\lambda}^*}^T \mathbf{h}(\mathbf{x}) + \sum_{j=1}^{r} \mu_j^*(g_j(\mathbf{x}) + z_j^2)$.

Finally, we show that $\mathbf{x}^*$ is the stationary point of $f(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{X}$. We denote $\mathcal{Y} \triangleq \big\{(\mathbf{x}, \mathbf{z}) \big| h_i(\mathbf{x}) = 0, i = 1, \cdots, m, g_j(\mathbf{x}) + z_j^2 = 0, z_j \in \mathbb{R}, j = 1, \cdots, r\big\}$. Note that $(\mathbf{x}, \mathbf{z}) \in \mathcal{Y}$ implies $\mathbf{x} \in \mathcal{X}$. For all $(\mathbf{x}, \mathbf{z}) \in \mathcal{Y}$, we have ${\boldsymbol{\lambda}^*}^T \mathbf{h}(\mathbf{x}) + \sum_{j=1}^{r} \mu_j^*(g_j(\mathbf{x}) + z_j^2) = 0$. Note that, we have shown that $(\mathbf{x}^*, \mathbf{z}^*)$ is a stationary point of $f(\mathbf{x}) + {\boldsymbol{\lambda}^*}^T \mathbf{h}(\mathbf{x}) + \sum_{j=1}^{r} \mu_j^*(g_j(\mathbf{x}) + z_j^2)$. Thus, $(\mathbf{x}^*, \mathbf{z}^*)$ is the stationary point of $f(\mathbf{x})$. So, we have:

$$\nabla_{\mathbf{x}} f(\mathbf{x})^T (\mathbf{x} - \mathbf{x}^*) + \nabla_{\mathbf{z}} f(\mathbf{x})^T (\mathbf{z} - \mathbf{z}^*) \geq 0, \ \forall (\mathbf{x}, \mathbf{z}) \in \mathcal{Y}. \tag{82}$$

Since $\nabla_{\mathbf{z}} f(\mathbf{x}) = \mathbf{0}$, we have $\nabla_{\mathbf{x}} f(\mathbf{x})^T (\mathbf{x} - \mathbf{x}^*) \geq 0$, for all $\mathbf{x} \in \mathcal{X}$. Thus, we can show that $\mathbf{x}^*$ is the stationary point of $f(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{X}$. ∎

References

[1] R. Koetter and M. Médard, "Beyond routing: an algebraic approach to network coding," in *21st Annual Joint Conf. of the IEEE Computer and Commun. Societies*, vol. 1, Oct 2002, pp. 122–130 vol.1.

[2] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, 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," *IEEE Trans. Inform. Theory*, 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 *2005 IEEE ISIT*, Sep 2005, pp. 264–267.

[5] S. El Rouayheb, A. Sprintson, and C. Georghiades, "A new construction method for networks from matroids," in *2009 IEEE ISIT*, Jun 2009, pp. 2872–2876.

[6] Q. Sun, S. T. Ho, and S.-Y. Li, "On network matroids and linear network codes," in *2008 IEEE ISIT*, Jul 2008, pp. 1833–1837.

[7] R. Dougherty, C. Freiling, and K. Zeger, "Matroidal networks," in *2007 45th Annual Allerton Conf. on Commun., Control, and Computing (Allerton)*, Sep 2007.

[8] A. Kim and M. Médard, "Scalar-linear solvability of matroidal networks associated with representable matroids," in *2010 6th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, Sep 2010, pp. 452–456.

[9] X. Yan, R. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," in *2007 IEEE ISIT*, Jun 2007, pp. 116–120.

[10] T. Chan, A. Grant, and D. Pfluger, "Truncation technique for characterizing linear polymatroids," *IEEE Trans. Inform. Theory*, vol. 57, no. 10, pp. 6364–6378, 2011.

[11] R. Dougherty, C. Freiling, and K. Zeger, "Networks, matroids, and non-shannon information inequalities," *IEEE Trans. Inform. Theory*, vol. 53, no. 6, pp. 1949–1969, Jun 2007.

[12] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and matroid theory," *IEEE Trans. Inform. Theory*, vol. 56, no. 7, pp. 3187–3195, 2010.

[13] A. Salimi, M. Médard, and S. Cui, "On the representability of integer polymatroids: Applications in linear code construction," in *2015 53rd Annual Allerton Conf. on Commun., Control, and Computing (Allerton)*, Monticello, IL, USA, Sep 2015.

[14] Y. Cui, M. Médard, E. Yeh, D. Leith, K. Duffy, F. Lai, and D. Pandya, "A linear network code construction for general integer connections based on the constraint satisfaction problem," *IEEE/ACM Trans. Networking*, vol. 25, no. 6, pp. 3441–3454, Dec 2017.

[15] W. Zeng, V. R. Cadambe, and M. Médard, "An edge reduction lemma for linear network coding and an application to two-unicast networks," in *2012 50th Annual Allerton Conf. on Commun., Control, and Computing (Allerton)*, 2012, pp. 509–516.

[16] C.-C. Wang and N. Shroff, "Pairwise intersession network coding on directed networks," *IEEE Trans. Inform. Theory*, vol. 56, no. 8, pp. 3879–3900, Aug 2010.

[17] S. Kamath, D. Tse, and V. Anantharam, "Generalized network sharing outer bound and the two-unicast problem," in *2011 Int. Symp. on Network Coding (NetCod)*, Jul 2011, pp. 1–6.

[18] S. Kamath, V. Anantharam, D. Tse, and C. C. Wang, "The two-unicast problem," *IEEE Trans. Inform. Theory*, vol. PP, no. 99, pp. 1–1, 2016.

[19] H. Maleki, V. R. Cadambe, and S. A. Jafar, “Index coding-an interference alignment perspective,” *arXiv preprint arXiv:1205.1483*, 2012.
[20] C. Li, S. Weber, and J. M. Walsh, “On multi-source networks: Enumeration, rate region computation, and hierarchy,” *CoRR*, vol. abs/1507.05728, 2015. [Online]. Available: http://arxiv.org/abs/1507.05728
[21] S. Kamath, D. N. Tse, and C.-C. Wang, “Two-unicast is hard,” in *2014 IEEE ISIT*, Jun 2014.
[22] 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,” *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2608–2623, 2006.
[23] D. S. Lun, M. Médard, T. Ho, and R. Koetter, “Network coding with a cost criterion,” in *2004 Int. Symp. Inform. Theory and its Applications (ISITA)*, 2004, pp. 1232–1237.
[24] Y. Cui, M. Medard, D. Pandya, E. Yeh, D. Leith, and K. Duffy, “A linear network code construction for general integer connections based on the constraint satisfaction problem,” in *2015 IEEE GLOBECOM*, Dec 2015, pp. 1–7.
[25] A. ParandehGheibi, A. Ozdaglar, M. Effros, and M. Médard, “Optimal reverse carpooling over wireless networks - a distributed optimization approach,” in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, Mar 2010, pp. 1–6.
[26] 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 *2007 IEEE ISIT*, Jun 2007, pp. 121–125.
[27] K. Duffy, C. Bordenave, and D. Leith, “Decentralized constraint satisfaction,” *IEEE/ACM Trans. Networking*, vol. 21, no. 4, pp. 1298–1308, Aug 2013.
[28] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.
[29] Y. Cui, M. Médard, E. Yeh, D. Leith, and K. Duffy, “Optimization-based linear network coding for general connections of continuous flows,” in *IEEE ICC*, London, UK, Jun 2015, pp. 4492 –4498.
[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press, 2004.
[31] D. Bertsimas and R. Weismantel, *Optimization over integers*. Belmont, Massachusetts, USA: Dynamic ideas, 2005.
[32] S. Boyd, *Lecture notes of EE364b: Primal-Dual Subgradient Method*. Standford Univ.

Ying Cui (S’08-M’12) received her B.E. degree in Electronic and Information Engineering from Xi’an Jiao Tong University, China, in 2007 and her Ph.D. degree in Electronic and Computer Engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2011. From June 2012 to June 2013, she was a Postdoctoral Research Associate in the Department of Electrical and Computer Engineering at Northeastern University, US. From July 2013 to December 2014, she was a Postdoctoral Research Associate in the Department of Electrical Engineering and Computer Science at Massachusetts Institute of Technology (MIT), US. Since January 2015, she has been an Associate Professor in the Department of Electronic Engineering at Shanghai Jiao Tong University, China. Her current research interests include cache-enabled wireless networks, future Internet architecture and delay-sensitive cross-layer control. She serves as an editor for the IEEE Transactions on Wireless Communications. She was selected to the Thousand Talents Plan for Young Professionals of China in 2013. She received the Best Paper Award at IEEE ICC, London, UK, June 2015.

Muriel Mdard (S’91M’95SM’02F’08) is the Cecil H. Green Professor of Electrical Engineering and Computer Science at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. Her research interests are in the areas of network coding and reliable communications. She has served as an Editor of many IEEE publications, and is currently the Editor-in-Chief of the IEEE J. Sel. Areas Commun. She served on the Board of Governors of the IEEE Information Theory Society, for which she was President in 2012 and received that Societys Aaron D. Wyner Distinguished Service Award in 2017. She has served as a TPC Chair or general Chair for several IEEE conferences. She was the recipient of the 2013 MIT Graduate Student Council EECS Mentor Award, the 2009 Communication Society and Information Theory Society Joint Paper Award, the 2009 William R. Bennett Prize in the Field of Communications Networking, the 2002 IEEE Leon K. Kirchmayer Prize Paper Award, and several conference paper awards. She was also a co-recipient of the MIT 2004 Harold E. Edgerton Faculty Achievement Award. In 2007, she was named a Gilbreth Lecturer by the U.S. National Academy of Engineering.

Edmund Yeh (SM’12) received his B.S. in Electrical Engineering with Distinction and Phi Beta Kappa from Stanford University in 1994. He then studied at Cambridge University on the Winston Churchill Scholarship, obtaining his M.Phil in Engineering in 1995. He received his Ph.D. in Electrical Engineering and Computer Science from MIT in 2001. He is currently Professor of Electrical and Computer Engineering at Northeastern University. He was previously Assistant and Associate Professor of Electrical Engineering, Computer Science, and Statistics at Yale University. Professor Yeh is the recipient of the Alexander von Humboldt Research Fellowship, the Army Research Office Young Investigator Award, the National Science Foundation and Office of Naval Research Graduate Fellowships, the Barry M. Goldwater Scholarship, the Frederick Emmons Terman Engineering Scholastic Award, and the President’s Award for Academic Excellence (Stanford University). He received Best Paper Awards at ACM ICN 2017, IEEE ICC 2015, and at IEEE ICUFN 2012.

Douglas Leith (M’02SM’09) graduated from the University of Glasgow in 1986 and was awarded his PhD, also from the University of Glasgow, in 1989. In 2001, Prof. Leith moved to the National University of Ireland, Maynooth and then in Dec 2014 to Trinity College Dublin to take up the Chair of Computer Systems in the School of Computer Science and Statistics. His current research interests include wireless networks, network congestion control, distributed optimization and data privacy.

Ken R. Duffy is a Professor at the Hamilton Institute, National University of Ireland Maynooth. He received the B.A.(mod) and Ph.D. degrees in mathematics from the Trinity College Dublin. His primary research interests are in probability and statistics, and their applications in science and engineering.