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

Ying Cui, Muriel Médard
Massachusetts Institute of Technology

Edmund Yeh
Northeastern University

Douglas Leith
National University of Ireland Maynooth

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

I. INTRODUCTION

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

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

In general, there are two types of coding approaches for optimizing network use for general connections. The first type of coding is mixing, which consists of coding together flows from sources using the random linear distributed code construction of [3] (originally proposed for multicast connections), as though they were part of a common multicast connection. In this case, no explicit code coefficients are provided and decodability is ensured with high probability by the random coding, given that mixing is properly designed. For example, in [7], a two-step mixing approach is proposed for network resource minimization of general connections, where flow partition and flow rate optimization are considered separately. 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 [8] and [9], simple two-flow combinations are proposed for network resource minimization of general connections. Although the second step flow rate optimization in [7] and the joint two-flow coding and flow optimization in [8], [9] can be solved distributively, the restriction in the separation of flow rate optimization in [7] and the pairwise coding in [8], [9] leads in general to feasibility region reduction and network cost increase. In [10], we propose new methods for constructing linear network codes for general connections of integer flows to minimize the total cost of edge use, based on mixing and explicit construction of linear network codes, respectively. In particular, in [10], we do not allow flow splitting and coding over time, leading to coded symbols flowing through each edge of the network only at an integer rate. The minimum-cost network coding design problems for general connections of integer flows in [10] are discrete optimization problems and 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. Different from [10], we allow flow splitting and coding over time to further reduce network cost. Based on mixing, we formulate
the minimum-cost network coding design problem, which is an instance of mixed discrete-continuous programming, to minimize the total cost of edge use for satisfying general connections of continuous flows. 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 network coding. To solve the mixed discrete-continuous optimization problem, we propose two equivalent alternative formulations with discrete mixing and continuous mixing, respectively, and develop distributed algorithms to solve them. Specifically, the distributed algorithm for the discrete mixing formulation is obtained by relating the optimization problem to a constraint satisfaction problem (CSP) in discrete optimization and applying recent results in the domain. The presented algorithm for the continuous mixing formulation is based on the penalty methods for nonlinear programming in continuous optimization.

II. NETWORK MODEL AND DEFINITIONS

In this section, we first illustrate the network model for general connections of continuous flows, which is similar to that we considered in [10] for integer flows, except that here we consider general flow rates and edge capacities, and allow flow splitting and coding over time. Then, for ease of understanding the formulations proposed later in Sections III, V, and IV, we also briefly illustrate the formal relationship between linear network coding and mixing established in [10].

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 $|\mathcal{V}|$ nodes and $\mathcal{E}$ denotes the set of $|\mathcal{E}|$ 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 $(i, j) \in \mathcal{E}$. 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 $\mathcal{I}_i = |\mathcal{I}_i|$ and $\mathcal{O}_i = |\mathcal{O}_i|$ denote the in degree and out degree of node $i \in \mathcal{V}$, respectively. Assume $\mathcal{I}_i \leq D$ and $\mathcal{O}_i \leq D$ for all $i \in \mathcal{V}$. Let $\mathcal{P} = \{1, \cdots, |\mathcal{P}|\}$ denote the set of $|\mathcal{P}|$ flows to be carried by the network. For each flow $p \in \mathcal{P}$, let $s_p \in \mathcal{V}$ be its source. We consider continuous flows. Let $R_p \in \mathbb{R}^+$ denote the source rate for source $p$, where $\mathbb{R}^+$ denotes the set of non-negative real numbers. Let $\mathcal{S} = \{s_1, \cdots, s_P\}$ denote the set of $P = |\mathcal{P}|$ sources. To simplify notation, we assume different flows do not share a common source node and no source node has any incoming edges. Let $\mathcal{T} = \{t_1, \cdots, t_T\}$ denote the set of $|\mathcal{T}|$ 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., $\cup_{t \in \mathcal{T}} P_t = \mathcal{P}$. To simplify notation, we assume no terminal has any outgoing edges.

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

B. Scalar Time-Invariant Linear Network Coding and Mixing

For ease of exposition, in this part, 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 enables multiple (coded) symbols flow through each edge at a continuous rate. But we shall retain the scalar time-invariant notation.

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

In many cases, we shall see that the specific values of local or global coding coefficients are not required in our design to reduce the network cost. For this purpose, we introduce the mixing concept based on local and global mixing coefficients. Specifically, we introduce 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 $\alpha_{kij} \in \mathcal{F}$. $\beta_{kij} = 1$ indicates that symbol $\sigma_{ki}$ of edge $(k, i) \in \mathcal{E}$ is allowed to contribute to the linear combination over $\mathcal{F}$ forming symbol $\sigma_{ij}$ and $\beta_{kij} = 0$ otherwise. Thus, if $\beta_{kij} = 0$, we have $\alpha_{kij} = 0$ (note that $\alpha_{kij}$ can be zero when $\beta_{kij} = 1$). Similarly, we introduce 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 \( c_{ij,p} \in \mathcal{F} \). \( x_{ij,p} = 1 \)
indicates that flow \( p \) is allowed to be mixed (coded) with other flows, i.e., symbol \( \sigma_p \) is allowed to contribute to the linear combination over \( \mathcal{F} \) forming symbol \( \sigma_{ij} \), and \( x_{ij,p} = 0 \)
otherwise. Thus, if \( x_{ij,p} = 0 \), we have \( c_{ij,p} = 0 \) (note that \( c_{ij,p} \) can be zero when \( x_{ij,p} = 1 \)). Then, we introduce the global mixing vector \( \mathbf{x}_j = (x_{ij,1}, \cdots, x_{ij,p}, \cdots, x_{ij,P}) \in \{0,1\}^P \) for edge \((i,j) \in \mathcal{E}\), which relates to the global coding vector \( \mathbf{c}_j = (c_{ij,1}, \cdots, c_{ij,p}, \cdots, c_{ij,P}) \in \mathcal{F}^P \). Similarly, we consider scalar time-invariant linear network mixing. In other words, \( \beta_{kij} \in \{0,1\} \) and \( x_{ij,p} \in \{0,1\} \) are both scalars, and \( \beta_{kij} \) and \( x_{ij,p} \) do not change over time.

Global mixing vectors provide a natural way of speaking of flows as possibly coded or not without knowledge of the specific values of global coding vectors. Intuitively, global mixing vectors can be treated as a limited representation of global coding vectors. Given network mixing vectors, it may not be sufficient to tell whether a certain symbol can be decoded or not. An example can be found in Fig. 2 of [10]. Therefore, using the network mixing representation, once the extraneous flows are mixed with the desired flows on the paths to each terminal, they are not guaranteed to be cancelled at the terminal. Let \( e_p \) denote the vector with the \( p \)-th element being 1 and all the other elements being 0. Let \( \lor \) denote the “or” operator (logical disjunction). We introduce the definition of feasibility for scalar linear network mixing below.

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

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

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

### III. CONTINUOUS FLOWS WITH MIXING ONLY

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

#### A. Design Parameter

We refer to the number of global network mixing vectors for each edge as the mixing parameter, which is our design parameter and denoted as \( L \in \{1, \cdots, L_{\text{max}}\} \), where \( L_{\text{max}} \) is the maximum number of global network mixing vectors for each edge to guarantee decodability (cf. Definition 1) and is obtained below. Let \( \mathcal{Y} \) denote the set of atoms of the algebra generated by \( \{\mathcal{P}_i : t \in \mathcal{T}\} \), i.e., \( \mathcal{Y} = \bigcap_{i \in \mathcal{T}} \mathcal{Y}_i : \mathcal{Y}_i = \mathcal{P}_i \) or \( \mathcal{Y}_i = \mathcal{P} - \mathcal{P}_i \) - \{0\}. In other words, \( \mathcal{Y} \) gives a set partition of \( \mathcal{P} \) that represents the flows that can be mixed (cf. Definition 1) over an edge in the worst case (i.e., all terminals obtaining flows through the same edge). Choose \( L_{\text{max}} = |\mathcal{Y}| \). Note that \( 1 \leq L_{\text{max}} \leq |\mathcal{P}| \), where \( L_{\text{max}} = 1 \) for the multicast case, i.e., \( \mathcal{P}_t = \mathcal{P} \) for all \( t \in \mathcal{T} \), and \( L_{\text{max}} = |\mathcal{P}| \) 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 the general case.

[Fig. 1: Illustration of flow partition \( \mathcal{Y} \) and mixing parameter \( L \). \( \mathcal{P} = \{1,2,3\}, \mathcal{S} = \{s_1,s_2,s_3\}, \mathcal{T} = \{t_1,t_2\}, \mathcal{P}_1 = \{1,2\}, \text{ and } \mathcal{P}_2 = \{1,2,3\} \). Thus, \( \mathcal{Y} = \{\{1,2\}, \{3\}\} \), \( L_{\text{max}} = |\mathcal{Y}| = 2 \) and \( L \in \{1,2\} \).

For a given mixing parameter \( L \), we now introduce the global and local network mixing vectors. For each \( l = 1, \cdots, L \), let \( x_{ij,l} = (x_{ij,1,l}, \cdots, x_{ij,p,l}, \cdots, x_{ij,P,l}) \in \{0,1\}^P \) denote the \( l \)-th global network mixing vector over edge \((i,j) \in \mathcal{E} \). Let \( \beta_{kij,l,m} \in \{0,1\} \) denote the local mixing coefficient corresponding to the \( l \)-th global network mixing vector of edge \((k,i) \in \mathcal{E} \) (i.e., \( x_{ki,l}(i) \)) and the \( m \)-th global network mixing vector of edge \((i,j) \in \mathcal{E} \) (i.e., \( x_{ij,l}(m) \)), where \( l = 1, \cdots, L \). In addition, let \( z_{ij,l}(i) \geq 0 \) denote the physical network-coded transmission rate corresponding to \( x_{ij,l}(i) \) over edge \((i,j) \in \mathcal{E} \).

#### B. Problem Formulation

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

**Problem 1 (Continuous Flows with Mixing Only):**

\[
U^*_z(L) = \min_{\alpha_{kij}, \beta_{kij}} \left( \sum_{(i,j) \in \mathcal{E}} \sum_{l=1}^L \left( \left( z_{ij,l} \right) \right) \right) \quad \text{s.t.} \quad 0 \leq z_{ij,l} \leq B_{ij}, \quad (i,j) \in \mathcal{E} \\
\quad \quad x_{ij,p,l} \in \{0,1\}, \quad l = 1, \cdots, L, \quad (i,j) \in \mathcal{E}, \quad p \in \mathcal{P} \\
\quad \quad \beta_{kij,l,m} \in \{0,1\}, \quad l, m = 1, \cdots, L, \quad (k,i),(i,j) \in \mathcal{E}
\]
\[ f_i^t, (l) \geq 0, \quad l = 1, \ldots, L, \quad (i, j) \in \mathcal{E}, \quad p \in \mathcal{P}_t, \quad t \in \mathcal{T} \tag{4} \]

\[ \sum_{p \in \mathcal{P}_t} f_{ij, p} (l) \leq \delta_{ij, l}, \quad t \in \mathcal{T}, \quad l = 1, \ldots, L, \quad (i, j) \in \mathcal{E} \tag{5} \]

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

\[ \sum_{k \in \mathcal{O}} \sum_{l = 1}^{L} f_{ik, p} (l) - \sum_{k \in \mathcal{T}} \sum_{l = 1}^{L} f_{ki, p} (l) = \sigma_i^t, \quad i \in \mathcal{V}, \quad p \in \mathcal{P}_t, \quad t \in \mathcal{T} \tag{7} \]

\[ f_{ij, p} (l) \leq x_{ij, p} (l) B_{ij}, \quad t \in \mathcal{T}, \quad l = 1, \ldots, L, \quad (i, j) \in \mathcal{E}, \quad p \in \mathcal{P}_t \tag{8} \]

\[ x_{ij} (l) = e_p, \quad l = 1, \ldots, L, \quad (s_p, j) \in \mathcal{E}, \quad p \in \mathcal{P} \tag{9} \]

\[ x_{ij} (l) = \vee_{k \in \mathcal{O}, m = 1, \ldots, l} \beta_{kij} (m, l) x_{kij} (m), \]

\[ l = 1, \ldots, L, \quad (i, j) \in \mathcal{E}, \quad i \notin S \tag{10} \]

\[ x_{it, p} (l) = 0, \quad l = 1, \ldots, L, \quad i \notin \mathcal{T}, \quad p \notin \mathcal{P}_t, \quad t \in \mathcal{T} \tag{11} \]

where

\[ \sigma_i^t = \begin{cases} R_{ip}, & i = s_p \\ -R_{ip}, & i = t \\ 0, & \text{otherwise} \end{cases}, \quad i \in \mathcal{V}, \quad p \in \mathcal{P}_t, \quad t \in \mathcal{T} \tag{12} \]

In the above formulation, \( f_{ij, p} (l) \geq 0 \) can be interpreted as the transmission rate of flow \( p \) in \( \mathcal{P}_t \) to terminal \( t \in \mathcal{T} \) corresponding to \( x_{ij} (l) \) over edge \((i, j) \in \mathcal{E}\), where \( l = 1, \ldots, L \). For notational simplicity, in this paper, we omit the conditions in (7) does not exist, the constraint in (6) can be satisfied by choosing \( z_{ij} (l) = 1 \), and the constraint in (8) is always satisfied by choosing \( \beta_{kij} (1, 1) = 1 \) and choosing \( x_{ij, p} (1) \) accordingly by (9) and (10). Therefore, in the multicast case, Problem 1 w.r.t. \( L = 1 \) for general connections reduces to the conventional minimum-cost network coding design problem for the multicast case [6].

Remark 2 (Comparisons with Routing and Two-step Mixing): Problem 1 w.r.t. \( L = 1 \) with an extra constraint \( \sum_{p \in \mathcal{P}} x_{ij, p} (1) \in \{0, 1\} \) for all \((i, j) \in \mathcal{E}\) is equivalent to the minimum-cost routing problem. Thus, the minimum network cost of Problem 1 w.r.t. \( L = 1 \) is no greater than the minimum costs of routing. On the other hand, Problem 1 w.r.t. \( L = L_{\text{max}} \) with \( \beta_{kij} (l, m) = 1 \) instead of (3), is equivalent to the minimum-cost flow rate control problem in the second step of the two-step mixing approach in [7]. Thus, the minimum network cost of Problem 1 w.r.t. \( L = L_{\text{max}} \) is no greater than the minimum cost of the two-step mixing approach for general connections in [7].

The following lemma shows the existence of a feasible linear network code corresponding to Problem 1.²

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.

C. Network Cost and Complexity Tradeoff

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

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

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

By Lemma 2, we know that the network cost \( U^* (L) \) decreases as the design parameter \( L \) increases. 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 \( L = 1 \), flow 3 cannot be delivered over edge \((4, 5)\) to terminal \( t_2 \) using feasible mixing. In other words, Problem 1 w.r.t. \( L = 1 \) is not feasible (i.e., of infinite network cost). However, when \( L = 2 \), 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 \( x_{15} (1) = (1, 1, 0) \) and \( x_{15} (2) = (0, 0, 1) \) over edge \((4, 5)\). In other words, Problem 1 w.r.t. \( L = 2 \) is feasible (i.e., of finite network cost). Thus, we can see the impact of \( L \) on the network cost shown in Lemma 2.

IV. ALTERNATIVE FORMULATION WITH DISCRETE MIXING

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

Note that (2) with \( j = t \), (7) with \( i = t \), and (8) with \( j = t \) imply \( \forall_{t \in \mathcal{T}} x_{it, p} = 1 \) for all \( p \in \mathcal{P}_t \) in Condition 3) of Definition 1.

²We omit all the proofs due to page limitation. Please refer to [13] for the details.
A. Alternative Formulation

Problem 1 is equivalent to the following problem.

Problem 2 (Equivalent Problem of Problem 1):

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

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

Subproblem 1 (Subproblem of Problem 2: Flow Optimization):

For given \(\{x_{ij,p}(l)\}\), we have:

\[
U_x^*(\{x_{ij,p}(l)\}) = \min_{\{z_{ij}\} \in \mathcal{E}} \sum_{(i,j) \in \mathcal{E}} U_{ij}(z_{ij})
\]

s.t. \((1), (4), (5), (6), (7), (8)\)

Subproblem 2 (Subproblem of Problem 2: Feasible Mixing):

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

\[
\forall i \in \mathcal{I}, l = 1, \ldots, \Omega, x_{ij,p}(l) = 1, \quad p \in \mathcal{P}, \quad t \in \mathcal{T}. \tag{13}
\]

B. Distributed Solution

In this part, we develop a distributed algorithm to solve Problem 2 via solving Subproblem 1 and Subproblem 2, respectively, in a distributed manner. First, we consider Subproblem 2. Subproblem 2 can be treated as a CSP and solved distributively using clause partition and Communication-Free Learning (CFL) algorithm from [11]. Specifically, \(\{x_{ij,p}(l)\} \cup \{\beta_{kj}(l, m)\}\) can be treated as the variables of the CSP. \(0, 1\) can be treated as the finite set of the CSP. From (10), we have an equivalent constraint purely on \(\{x_{ij,p}(l)\}\), i.e.,

\[
\exists \beta_{kj}(l, m) \in \{0, 1\} \quad \forall k \in \mathcal{I}, m = 1, \ldots, \Omega,
\]

s.t. \(x_{ij}(l) = \forall k \in \mathcal{I}, m = 1, \ldots, \Omega, \beta_{kj}(l, m)x_{ki}(m),\)

\(l = 1, \ldots, \Omega, (i, j) \in \mathcal{E}, \quad i \notin \mathcal{S}. \tag{14}\)

In the following, we shall only consider solving variables \(\{x_{ij,p}(l)\}\) of the CSP in a distributed way using clause partition and CFL, as \(\{\beta_{kj}(l, m)\}\) can be obtained from feasible \(\{x_{ij,p}(l)\}\) by (9) and (10). In addition, we directly choose \(x_{ij,p}(l) = e_p\) for all \(l = 1, \ldots, \Omega, (s_p, j) \in \mathcal{E}\) and \(p \in \mathcal{P}\) according to (9) and shall not solve it from the CSP.

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

\[
\phi^x_{ij,p}(x_{ij}(l), \{x_{ki}(m) : m = 1, \ldots, \Omega, l, k \in \mathcal{I}\})
\]

\[
\phi^x_{ij,p}(x_{ij}(l), \{x_{ki}(m) : m = 1, \ldots, \Omega, l, k \in \mathcal{I}, j \in \mathcal{T}\})
\]

\[
\Delta \triangleq \begin{cases} 1, & \text{if } j \notin \mathcal{T}, \text{ (14) holds} \\ 1, & \text{if } j \in \mathcal{T} \text{ and } p \in \mathcal{P}, \text{ (14) and (13) hold} \\ 1, & \text{if } j \in \mathcal{T} \text{ and } p \notin \mathcal{P}, \text{ (14) and (11) hold} \\ 0, & \text{otherwise} \end{cases}
\]

\(l = 1, \ldots, \Omega, (i, j) \in \mathcal{E}, \quad p \in \mathcal{P}, \quad i \notin \mathcal{S}. \tag{15}\)

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

We now construct the clause partition of Subproblem 2.

Specifically, the set of clauses variable \(x_{ij,p}(l)\) participates in is as follows:

\[
\Phi^x_{ij,p} \triangleq \left\{ \phi^x_{ij,p} : m = 1, \ldots, \Omega, l, k \in \mathcal{I}, j \in \mathcal{T} \right\}
\]

\[
\cup \left\{ \phi^x_{kj,p} : m = 1, \ldots, \Omega, l, k \in \mathcal{I}, j \in \mathcal{T} \right\}
\]

\((i, j) \in \mathcal{E}, \quad p \in \mathcal{P}, \quad i \notin \mathcal{S}. \tag{16}\)

Note that, when \(j \notin \mathcal{T}\), \(\phi^x_{ij,p}(\cdot) = \emptyset\) and we ignore it in \(\Phi^x_{ij,p}\) in (16). A feasible \(\{x_{ij,p}(l)\} \in \mathcal{M}(L)\) to Subproblem 2 can be found distributively using the CFL algorithm [11, Algorithm 1] based on the clause partition. Specifically, for all \((i, j) \in \mathcal{E}, p \in \mathcal{P}\) and \(l = 1, \ldots, \Omega\), node \(i\) realizes a Bernoulli random variable selecting \(x_{ij,p}(l)\). Pass message on \(x_{ij,p}(l)\) between adjacent nodes to evaluate its related clauses in (16) and update the distribution of its related Bernoulli random.

Given a feasible \(\{x_{ij,p}(l)\} \in \mathcal{M}(L)\) obtained by CFL, Subproblem 1 is convex and can be solved distributively using standard convex decomposition. We omit the details here due to the page limitation. Now, we can develop a distributed algorithm to solve Problem 2 based on CFL and convex decomposition, as briefly illustrated in Algorithm 1.

---

Algorithm 1  Algorithm for Problem 2

1: initialize \(n = 1\) and \(U_1 = +\infty\).

2: loop

3: Run CFL to the CSP corresponding to Subproblem 2.

4: if the CFL finds a feasible solution then

5: For the obtained feasible solution to Subproblem 2, solve Subproblem 1 distributively using convex decomposition. Let \(U_n\) denote the corresponding network cost.

6: if \(U_n < U_{n-1}\) then

7: set \(U_{n+1} = U_n\) and \(n = n + 1\).

8: end if

9: end if

10: end loop

---

3In 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.
Based on the convergence result of CFL [11, Corollary 2],
we can easily see that \( U_n \to U_x^*(L) \) almost surely as \( n \to \infty \),
if Problem 2 is feasible.

V. ALTERNATIVE FORMULATION WITH CONTINUOUS MIXING

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

A. Alternative Formulation

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

Problem 3 (Continuous Formulation of Problem 1):

\[
U_x^*(L) = \min_{\{z_{ij}\}, \{\bar{z}_{ij}(l)\}, \{f_{ij,p}(l)\}, \{\bar{f}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\}} \sum_{(i,j) \in E} U_{ij}(z_{ij})
\]

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

\[
\bar{z}_{ij,p}(l) \in [0,1], \quad l = 1, \ldots, L, \quad (i,j) \in E, \quad p \in P
\]

\[
\bar{f}_{ij,p}(l) \leq \bar{z}_{ij,p}(l) B_{ij}, \quad l = 1, \ldots, L, \quad (i,j) \in E,
\]

\[
\bar{x}_{ij,p}(l) \geq \bar{\beta}_{kij}(l,m) \bar{x}_{ki,p}(l), \quad k \in I_i, \quad l = 1, \ldots, L
\]

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

Lemma 3 (Relationship between Problem 1 and Problem 3):

(i) If \( \{z_{ij}\}, \{\bar{z}_{ij}(l)\}, \{f_{ij,p}(l)\}, \{\bar{f}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\} \) is a feasible solution to Problem 1, then \( \{z_{ij}\}, \{\bar{z}_{ij}(l)\}, \{f_{ij,p}(l)\}, \{\bar{f}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\} \) is a feasible solution to Problem 3, where \( \bar{x}_{ij,p}(l) = x_{ij,p}(l) \) and \( \bar{\beta}_{kij}(l,m) = \beta_{kij}(l,m) \).

(ii) If \( \{z_{ij}\}, \{\bar{z}_{ij}(l)\}, \{f_{ij,p}(l)\}, \{\bar{f}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\} \) is a feasible solution to Problem 3, then \( \{z_{ij}\}, \{\bar{z}_{ij}(l)\}, \{f_{ij,p}(l)\}, \{\bar{f}_{ij,p}(l)\}, \{\bar{\beta}_{kij}(l,m)\} \) is a feasible solution to Problem 1, where \( x_{ij,p}(l) = [\bar{x}_{ij,p}(l)] \) and \( \beta_{kij}(l,m) = [\bar{\beta}_{kij}(l,m)] \).