Joint Scheduling and Multiflow Maximization in Wireless Networks

Yanxiao Liu[†], Shenghao Yang[*], and Cheuk Ting Li[†]
[†]Department of Information Engineering, The Chinese University of Hong Kong
[*]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
Email: yanxiaoliu@link.cuhk.edu.hk, shyang@cuhk.edu.cn, ctli@ie.cuhk.edu.hk

***Abstract*—We revisit a fundamental problem at the heart of network communication theory: the maximum multiflow (MMF) problem in multi-hop networks, with network coding performed at intermediate nodes. To derive the exact-optimal solution to the MMF problem (as opposed to approximations), conventional methods usually involve two steps: first calculate the scheduling rate region, and then find the maximum multiflow that can be supported by the achievable link rates. However, the NP-hardness of the scheduling part makes solving the MMF problem in large networks computationally prohibitive. In this paper, while still focusing on the exact-optimal solution, we provide efficient algorithms that can jointly calculate the scheduling rate region and solve the MMF problem, thereby outputting optimal values *without* requiring the entire scheduling rate region. We prove that our algorithms always output optimal solutions in a finite number of iterations.**

A full version of this paper is accessible at [1]: https://arxiv.org/pdf/2509.14582

I. Introduction

To deploy the next generation mobile system, it is expected that a massive connectivity and emerging applications should be supported. For the design of large-scale, highly-connected wireless mobile systems, it is crucial to have a solid understanding of their theoretic limits, and hence we revisit two multiflow optimization problems: the maximum multiflow (MMF) and maximum concurrent multiflow (MCMF) that can be supported by collision-free link schedules of the networks.

To maximize the total or concurrent throughput between multiple source nodes and sink nodes supported by achievable link rates, traditional methods typically first model wireless interference in networks using a link conflict graph [2]. Next, they compute the scheduling rate region, which is a well-known NP-hard problem that is also difficult to approximate [3]. Finally, they determine the maximum total or concurrent throughput under the constraints defined by the scheduling rate region. Due to the hardness of the scheduling problem, it is unrealistic to directly solve the multiflow maximization problems in large-scale networks in this way. In [4] it has been shown that both and MMF and MCMF problems are NP-hard even in very simple settings, and can be even harder when network coding [5]–[7] is performed. Hence, joint optimization methods usually focus on either approximate solutions of general networks or exact solutions of restricted networks.

In this paper, we are interested in the *exact-optimal* solutions of the multiflow problems for *general* multi-hop networks. Rather than first solving the scheduling problem and then calculating the multiflow values (referred to as *two-step algorithms*), we design efficient algorithms that jointly calculate the maximum total or concurrent multiflow and the scheduling rate region by employing a decomposition method, thereby requiring only a (possibly very small) subset of the scheduling rate region. This joint framework makes our algorithms more practical, while still provably guaranteeing optimal (not approximate or converging-to-optimal) solutions to the multiflow maximization problems in a finite number of iterations. Our algorithms are applicable to the most general setting in multi-hop networks: the *multiple multicast* case, where network coding [5]–[7] at intermediate nodes is allowed.

II. Related Works

A. Maximum Multiflow Problem

The MMF problem studies the maximum throughput between selected source nodes and sink nodes [2], [4], and the maximum concurrent multiflow problem [8] models the case where every sender-receivers session transmits messages concurrently. The NP-hardness of both problems have been proved in [4], even in very simple network settings. In [9], [10], both the MMF and MCMF problems are discussed under the interference model that nodes cannot transmit and receive simultaneously. By enforcing interference constraints on links, [11] guarantees the schedulability and develops constant-approximation algorithms. More linear programming formulation and approximation algorithms can be found in [4], [8], [9]. In [12], the MMF and MCMF problems are discussed by dividing the cases to full-duplex systems and half-duplex systems, both of which are covered by our interference model in this paper. The MMF problem has been extended to unicast networks with network coding in [13].

B. Joint Optimization and Network Coding

Existing works on the MMF (or MCMF) problem [2], [4], [13] only study the *multiple unicast* case, i.e., each source node is paired with one sink node. However, we consider *multiple multicast* in general multi-source multi-sink networks, where each of a number of source nodes transmits a message to a set of sink nodes. In this scenario, network coding [5]–[7] is an effective technique to improve the network performance, and the throughput can increase up to several folds [14], [15]. The joint consideration of throughput, scheduling and network coding has been widely studied in [16]–[20] for various objectives, e.g., maximizing throughput or minimizing the energy consumption under certain constraints. These approaches are either converging-to-optimal with respect to some constraints or only approximate the solutions. The closest work to ours is [16], where the authors decompose the joint optimization of scheduling and network coding into two subproblems. However, we study the multi-source (instead of single source) multi-sink case, where the trade-off between the rates of the sources becomes an important factor of consideration, and the algorithm in [16] is an iterative algorithm that only converges to the optimum, but our algorithms provably output the exact optimum in a finite number of iterations.

III. Network Model

A. Network Model

We use a link-wise network model [2], [4], [21], [22], where each link is associated with a *collision set*, including all the links that can be interfered by it. It is called the binary interference model [2], [21], and it is not difficult to cover the physical interference model by signal-to-interference-and-noise ratio [19], [22]. We assume the network is acyclic and *discrete* [21]–[23] in the sense that time is slotted and the link delays are multiples of a length of a time slot, which is justified in [23]. The intermediate nodes can wait until enough packets are collected before performing coding on the packet.

The network can be modeled by a tuple $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I}, D)$, where $\mathcal{V}$ is the *node set*, $\mathcal{L} \subseteq \mathcal{V}^2$ is the *link set*, $\mathcal{I} = (\mathcal{I}(l), l \in \mathcal{L})$ is the set of *collision sets* where $l' \in \mathcal{I}(l)$ if l' is in the interference range of l and $D : \mathcal{L}^2 \to \mathbb{Z}$ is the link-wise delay matrix specifies the delays between links. We assume each link has a unit bandwidth and allow parallel links between nodes. $(\mathcal{L}, \mathcal{I}, D)$ can form a weighted, directed graph $\mathcal{N}$ where $\mathcal{L}$ is the finite vertex set, (l, l') is an edge if $l' \in \mathcal{I}(l)$, and $D(l, l')$ is the weight on the directed edge (l, l'), which degrades to an unweighted graph $(\mathcal{L}, \mathcal{I})$ if delays are ignored [4], [13]. This graphical approach helps the discussion on our algorithms.

We now describe the communication task over the network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I}, D)$. For $k \in \mathbb{N}_+$, let $\mathcal{S} = \{s_1, \ldots, s_k\} \subseteq \mathcal{V}$ be the set of source nodes. We assume the information sources at different source nodes are mutually independent. Each source node s_i is associated with a set of sink nodes $\mathcal{D}_{s_i} \subseteq \mathcal{V}$ that have to decode the information at s_i. We allow a node to be both a source node and a sink node corresponding to another source node. For $i \neq j$, we may have $\mathcal{D}_{s_i} \cap \mathcal{D}_{s_j} \neq \emptyset$, i.e., different source nodes can share same sink nodes. Each link $l \in \mathcal{L}$ represents a point-to-point channel with unit capacity. The sets of input channels and output channels of a node $v \in \mathcal{V}$ are denoted by $\mathrm{In}(v) \subseteq \mathcal{L}$ and $\mathrm{Out}(v) \subseteq \mathcal{L}$, respectively.

To explain the model, we use a line network [21], [22], [24], [25] as an example.

Example 1. Consider an L-hop unicast line network: there are $L+1$ nodes $\mathcal{V} = \{1, 2, \ldots, L+1\}$, with link set

$$\mathcal{L} = \{l_i \triangleq (i, i+1), i = 1, \ldots, L\}.$$

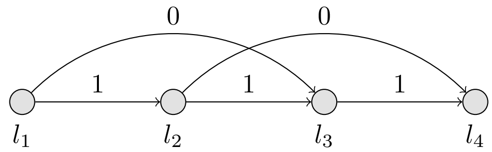

Fig. 1. $\mathcal{N}_{4,1}^{D=1}$: nodes represents network link, edges represent the collision relations between nodes, and edge weights represent propagation delays.

Each link has a unit delay. We consider a K-hop interference model: the reception of a node has possible collisions from nodes in K hops. The nodes are half-duplex. The collision set of l_i is

$$\mathcal{I}_K(l_i) = \{l_j : j \neq i, |i+1-j| \leq K\}. \tag{1}$$

The link l_i is active in time slot t if node i sends a signal in time slot t to node $i+1$. Hence the link-wise delay matrix is

$$D(l_i, l_j) = 1 - |i+1-j|. \tag{2}$$

We denote it by $\mathcal{N}_{L,K}^{D=1}$, and it can be represented by a graph, where $\mathcal{N}_{4,1}^{D=1}$ is shown in Figure 1 as an example.

B. Collision-free Schedules and Rate Region

We define collision-free schedules, similar to [21], [22]. Since we assume the time is slotted, when link l is active at time slot t and link $l' \in \mathcal{I}(l)$ is active at time slot $t + D(l, l')$, we say a *collision occurs*. In each time slot we use a binary number to indicate whether a link sends messages or not. Hence we use an infinite binary matrix $S : \mathcal{L} \times \mathbb{N} \to \{0, 1\}$ with rows indexed by $\mathcal{L}$ and columns indexed by $\mathbb{N}$ to specify a *schedule*: $S(l, t) = 1$ indicates that link l is active in time slot t, and $S(l, t) = 0$ indicates it is inactive. $S(l, t)$ has a collision in $\mathcal{N}$ if $S(l, t) = S(l', t + D(l, l')) = 1$ for a certain $l' \in \mathcal{I}(l)$. Otherwise $S(l, t)$ is *collision free*. A schedule S is collision free if $S(l, t)$ is collision free for all (l, t). There are different (though with similar ideas) definitions if delays are simply ignored, e.g., [4], but we aim to provide a framework general enough to cover the networks with non-negligible delays. For a collision-free schedule S and a link l, the link rate is

$$R_S(l) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} S(l, t). \tag{3}$$

If $R_S(l)$ exists for all $l \in \mathcal{L}$, we call $R_S = (R_S(l), l \in \mathcal{L})$ the *rate vector* of S. For a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I}, D)$, a rate vector $R = (R(l), l \in \mathcal{L})$ is said to be *achievable* if for all $\epsilon > 0$, there exists a collision-free schedule S such that $R_S(l) > R(l) - \epsilon$ for all $l \in \mathcal{L}$. For a link l, the rate $R(l)$ can stand for the maximum number of information symbols that can be sent on the channel per time slot. Then each achievable rate vector can be viewed as a rate constraint for the network.

The collection $\mathcal{R}(\mathcal{N})$ of all the achievable rate vectors is called the *(scheduling) rate region* of $\mathcal{N}$. We may use $\mathcal{R}$ instead of $\mathcal{R}(\mathcal{N})$ to simplify the notation when the context is clear. It is proved in [21] that $\mathcal{R}$ is a convex polytope, and can be achieved by using periodic collision-free schedules.

C. Problem Formulation

We define the maximum multiflow (MMF) and the maximum concurrent multiflow (MCMF) problems. Most existing literature consider *multiple unicast*, i.e., each source only has one corresponding sink with a certain demand [4], [12], [13]. However, we discuss general *multiple multicast*, with network coding [5]–[7] on intermediate nodes, where the nodes can encode their received data before passing them on.

For each multicast session, one source corresponds to multiple sinks. Though we use network coding to attain the maximum information flow in a session, we do not consider coding *between* sessions (i.e., inter-session network coding [26], [27]) in this paper for the sake of simplicity, since it is in general a hard problem [28] and can even be undecidable [29]. We say

$$F = (F(l) \in \mathbb{N}_{\geq 0} : l \in \mathcal{L}) \tag{4}$$

is a valid flow from source node $s \in \mathcal{V}$ to sink node $t \in \mathcal{V}$ with respect to a rate vector R if it satisfies:

- $0 \leq F(l) \leq R(l)$ for all $l \in \mathcal{L}$, i.e., the flow along link l does not exceed the rate constraint $R(l)$.
- The flow conservation equation $\sum_{l \in \mathrm{In}(v)} F(l) = \sum_{l \in \mathrm{Out}(v)} F(l)$ for all $v \in \mathcal{V} \backslash \{s, t\}$.

We see the flow $\sum_{l \in \mathrm{Out}(s)} F(l)$ out of s equals to the flow $\sum_{l \in \mathrm{In}(t)} F(l)$ into t, and this value is called the *value* of F, denoted as $\mathrm{val}(F)$. We say F is a *max-flow* from s to t with respect to R if F is a flow and has a value no smaller than the value of any other flow from s to t with respect to R.

1) Maximum Multiflow (MMF) Problem: Consider a source s_i multicasts a message to nodes in the set $\mathcal{D}_{s_i} = \{t_{i,1}, \ldots, t_{i,k_i}\}$, with network coding [5]–[7], at a rate

$$v_i = \min_j \mathrm{val}(F_{i,j}),$$

where $F_{i,j}$ is a flow from s_i to $t_{i,j}$ for each j and the rate of communication along link l is $\max_j F_{i,j}(l)$.

We now put the flows from the source nodes $s_1, \ldots, s_k$ together. Fix a rate vector R. Link l has to accommodate all these k flow requirements simultaneously, i.e., $\sum_{i=1}^{k} \max_j F_{i,j}(l) \leq R(l)$ for all $l \in \mathcal{L}$. We maximize the sum of the rates of multicasting these k sources, and it is called the maximum multiflow (MMF) problem, which is formulated by the following linear program, combining the linear program for multiple unicast [2], [4], [13] and the program for single multicast [16]:

$$\text{LP-MMF}(\tilde{\mathcal{R}}): \quad \text{maximize} \quad \sum_{i=1}^{k} v_i$$

subject to

$$\begin{aligned}
& F_{i,j} \text{ is a valid flow}, \forall i \in [k], j \in [k_i], \\
& \sum_{l \in \mathrm{Out}(s_i)} F_{i,j}(l) = \sum_{l \in \mathrm{In}(t_{i,j})} F_{i,j}(l) = v_i, \forall i \in [k], j \in [k_i], \\
& G_i(l) \geq F_{i,j}(l), \forall l \in \mathcal{L}, i \in [k], j \in [k_i], \\
& \sum_{i=1}^{k} G_i(l) \leq R(l), \forall l \in \mathcal{L}, \qquad (5) \\
& R \in \tilde{\mathcal{R}},
\end{aligned}$$

where $[k] := \{1, \ldots, k\}$, $k, k_i \in \mathbb{N}_+, \forall i$. The linear program takes a polytope $\tilde{\mathcal{R}}$ (a subset of the scheduling rate region) as an input. The variables $G_i(l)$, $i \in [k]$, $l \in \mathcal{L}$ are used to impose the constraint $\sum_{i=1}^{k} \max_j F_{i,j}(l) \leq R(l)$. The constraint (5) gives a dual vector that will be used in our algorithms, and the dual variable corresponding to link l means how sensitive the optimization objective is to the rate constraint $R(l)$.

2) Maximum Concurrent Multiflow (MCMF) Problem: While the MMF problem is to find the link schedule that can support the maximum total rate of transmission of the sources, the maximum concurrent multiflow (MCMF) problem is to find the link schedule such that all the sources can transmit concurrently at the maximum rate [4], [8], [9]. The settings of MCMF problems in [4], [9] are also for multiple unicast.

More generally, instead of maximizing the sum rate $\sum_{i=1}^{k} v_i$, we maximize ϕ such that the source s_i can multicast at a rate $v_i = \phi \gamma_i$, where γ_i is the desired traffic rate at s_i. The MCMF problem is formulated as follows [4], [16]:

LP-MCMF($\tilde{\mathcal{R}}$) :

$$\text{maximize} \quad \phi \tag{6}$$

subject to

$$\begin{aligned}
& F_{i,j} \text{ is a valid flow}, \forall i \in [k], j \in [k_i], \\
& \sum_{l \in \mathrm{Out}(s_i)} F_{i,j}(l) = \sum_{l \in \mathrm{In}(t_{i,j})} F_{i,j}(l) = \phi \gamma_i, \forall i \in [k], j \in [k_i], \\
& G_i(l) \geq F_{i,j}(l), \forall l \in \mathcal{L}, i \in [k], j \in [k_i], \\
& \sum_{i=1}^{k} G_i(l) \leq R(l), \forall l \in \mathcal{L}, \qquad (7) \\
& R \in \tilde{\mathcal{R}}.
\end{aligned}$$

IV. Algorithm

In this section, we present our main algorithm that jointly computes the MMF (or MCMF) and the scheduling rate region, thereby provably outputting the exact-optimal solution while requiring only a subset of the scheduling rate region.

Since a collision-free schedule can be found by searching an independent set in the graph $(\mathcal{L}, \mathcal{I})$ [21], we attach a weight $a_i \geq 0$ to link l_i, and would like to maximize the weighted total rate, i.e., for the scheduling rate region $\mathcal{R}$, we solve

$$\arg\max_{R \in \mathcal{R}} \langle \mathbf{a}, R \rangle, \tag{8}$$

where $\mathbf{a} = (a_i)_{i=1,\ldots,|\mathcal{L}|}$, and $\langle \cdot, \cdot \rangle$ is the inner product.

Remark 1. For (8), the objective is to maximize the weighted sum rate instead of just the sum rate, since a different weight vector could be used in each iteration of our algorithms (see step 4 of Algorithm 1 or step 5 of Algorithm 2). These weights are also crucial for the graphical approach we propose for Algorithm 2.

This corresponds to a weighted maximal independent set problem [4], [16] that can be solved by integer linear programming (ILP) by maximizing over $S(l_i) \in \{0, 1\}$ for $l_i \in \mathcal{L}$:

$$\text{ILP:} \quad \text{maximize} \quad \sum_{i=1}^{|\mathcal{L}|} a_i S(l_i)$$

Algorithm 1

Input: a network $(\mathcal{V}, \mathcal{L}, \mathcal{I})$

Output: maximum multiflow v

Start with any rate vector $R_1 \in \mathcal{R}$, $\mathcal{R}_1 \leftarrow \{R_1\}$

for $i = 1, 2, \dots$ **do**

  Run $v_i \leftarrow$LP-MMF($\mathcal{R}_i$) (or $v_i \leftarrow$LP-MCMF($\mathcal{R}_i$)) and obtain the dual vector μ_i

  Run ILP to find $R_{i+1} \leftarrow \arg\max_{R \in \mathcal{R}} \langle \mu_i, R \rangle$

  $\mathcal{R}_{i+1} \leftarrow \operatorname{conv}(\mathcal{R}_i \cup \{R_{i+1}\})$

  if $\langle \mu_i, R_{i+1} \rangle = \max_{R \in \mathcal{R}_i} \langle \mu_i, R \rangle$ **then**

    return v_i

$$\text{subject to} \quad S(l_i) + S(l_j) \le 1, \quad \forall\, l_i, l_j : l_j \in \mathcal{I}(l_i).$$

The solution gives us a maximal independent set of $(\mathcal{L}, \mathcal{I})$, and the corresponding achievable rate vector is $S = (S(l_i), l_i \in \mathcal{L})$, which is a vertex of the scheduling rate region $\mathcal{R}$.

Based on (8), we iteratively search the MMF or MCMF and the scheduling region. Even though our target is the optimal value instead of approximated or converging-to-optimal values, we show that it is unnecessary to find the entire scheduling region before solving the MMF or MCMF problem.

Suppose flow$(\cdot)$ is the function for calculating the MMF or MCMF in a given polytope, which can be a subset of the scheduling region. From $i = 1$, in each iteration, the algorithm works as follows:

1) We start with a subset of the scheduling region $\mathcal{R}_i$, which is formed by the vertices of $\mathcal{R}$ we have known (it is reasonable to assume some rate vectors are known, e.g., by activating the first link all the time and inactivating others, the vector $[1, 0, \dots, 0]^\intercal$ is achievable). In the first iteration, we start with an arbitrarily chosen rate vector R_1, i.e, $\mathcal{R}_1 = \{R_1\}$. We run the linear program LP-MMF($\mathcal{R}_i$) (or LP-MCMF($\mathcal{R}_i$)) to obtain the dual vector μ_i, corresponding to the constraint in (5) (or (7)).
2) By the ILP, we find a new rate vector R_{i+1} by
$$R_{i+1} = \arg\max_{R \in \mathcal{R}} \langle \mu_i, R \rangle. \tag{9}$$
3) We update the subset of the scheduling region by computing the convex hull $\mathcal{R}_{i+1} = \operatorname{conv}(\mathcal{R}_i \cup \{R_{i+1}\})$.
4) If $\langle \mu_i, R_{i+1} \rangle = \max_{R \in \mathcal{R}_i} \langle \mu_i, R \rangle$, the algorithm terminates and outputs the last optimal value of LP-MMF($\mathcal{R}_i$) (or LP-MCMF($\mathcal{R}_i$)); otherwise it comes back to step 1 and continues.

It can be proved that Algorithm 1 will terminate and output the maximum (concurrent) multiflow in finite iterations. The proof can be found in [1].

Theorem 1 (Optimality). *For a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I})$, Algorithm 1 will terminate and output the maximum multiflow (or the maximum concurrent multiflow).*

Remark 2. Our algorithms rely on an integer linear programming step and, therefore, do not theoretically guarantee polynomial-time complexity, which is expected due to the NP-hardness of the MMF or MCMF problem [2], [4]. This can be understood in that, in the worst case, one may still have to compute the entire scheduling rate region.

Remark 3. In [16], an algorithm based on subgradient optimization that decomposes the problem into two parts has been discussed. Though it shares some similarities with ours, our algorithm is guaranteed to find the optimum exactly in a finite number of steps (assuming access to an integer linear programming algorithm), whereas [16] is an iterative algorithm that only converges to the optimum. As we will see in the following sections, terminating in a small number of steps is especially important for networks with non-negligible delays, since the update of the subset $\mathcal{R}_i$ of the scheduling region is the bottleneck of the algorithm with exponential time complexity, and should be performed as little as possible.

V. Networks with Non-negligible Delays

We extend our previous discussions to networks with non-negligible propagation delays (e.g., underwater networks). Recent studies [21], [22], [24], [25], [30] have shown that, for such networks, instead of using guard intervals to mitigate the effects of delays, we can actually *utilize* the propagation delays to significantly improve the scheduling rate region.

We adopt a graphical approach, build upon the framework proposed by [21], [22], and provide an algorithm in the same spirit as Algorithm 1. The proposed Algorithm 2 also provably outputs exact-optimal solutions in a finite number of iterations, and maintains all the advantages.

The key to solve the MMF or MCMF problems in the networks with non-negligible delays is, we need a function similar to (8) that can output a vertex of the scheduling region in a time complexity at most exponential in $|\mathcal{L}|$ (which in turn will be polynomial in the size of the *scheduling graphs* below), which is more efficient than the cycle-enumeration approach [21] with complexity doubly exponential in $|\mathcal{L}|$.

We review the *scheduling graph* in [21] as follows, which will be generalized later: For a collision-free schedule matrix S and integers $T \in \mathbb{N}_+$, $k \in \mathbb{Z}$, $S[T, k]$ denotes the submatrix of S consisting of columns $kT, kT+1, \dots, (k+1)T - 1$. If a submatrix S' is formed by T columns of S, its columns are indexed by $0, 1, \dots, T-1$.

Definition 1 (Scheduling Graph [21]). Given a network $\mathcal{N}$ and an integer $T > 0$, the *scheduling graph* $(\mathcal{M}_T, \mathcal{E}_T)$ is a directed graph that is defined as follows: the vertex set $\mathcal{M}_T$ includes all the $|\mathcal{L}| \times T$ binary matrices A such that $A = S[T, 0]$ for a certain collision-free schedule S. The edge set $\mathcal{E}_T$ includes all the vertex pairs (A, B) such that $A = S[T, 0]$ and $B = S[T, 1]$ for a certain collision-free schedule S.

In [21], it has been shown that by choosing $T \ge \max_{l \in \mathcal{L}} \max_{l' \in \mathcal{I}(l)} |D(l, l')|$, calculating the scheduling region is equivalent to searching all the simple cycles in the scheduling graph, which is NP-hard. The scheduling problem then may then even have doubly exponential complexity, since the number of vertices in $(\mathcal{M}_T, \mathcal{E}_T)$ increases exponentially with respect to $|\mathcal{L}|$, and the cycle enumeration in $(\mathcal{M}_T, \mathcal{E}_T)$ is also NP-hard in general.

Instead of enumerating cycles for the scheduling region, we search *maximum-mean-cycles* (which can be solved in a time complexity *polynomial* in the graph size) in a new graph, to find the vertices of the scheduling region. We may only need a few rate vectors to solve the MMF or MCMF problem.

Before describing our approach, we need some graphical concepts. In a directed graph $\mathcal{G}$, a *path* is a sequence of vertices $v_0, v_1 \ldots, v_m$ where for $i = 0, 1, \ldots, m-1$, (v_i, v_{i+1}) is a directed edge. A path is *closed* if $v_0 = v_m$. A *cycle* in $\mathcal{G}$ is a closed path $(v_0, v_1 \ldots, v_m)$ such that $m \geq 1$, $v_i \neq v_j$ for any $0 \leq i \neq j \leq m-1$ and $v_0 = v_m$, i.e., in such a sequence, the only repeated vertices are the first and the last vertices. Note a closed path can be decomposed into a sequence of cycles [31], and this has been used in proving that it suffices to enumerate all the simple cycles for calculating the scheduling rate region [21]. In a graph where each edge is associated with a weight, we say the weight of a directed cycle is the total weight on the edges in the cycle. Then we say the average weight of a directed cycle is the total weight divided by the number of edges in the cycle. The *maximum-mean-cycle* is the cycle in the given weighted, directed graph with the maximum average weight over all directed cycles in the given graph.

It has been proved in [21], [22] that a collision-free, periodic schedule is equivalent to a closed path (which can be decomposed to multiple simple cycles) in $(\mathcal{M}_T, \mathcal{E}_T)$ and vice versa, i.e., the concatenation of a sequence of vertices in $(\mathcal{M}_T, \mathcal{E}_T)$ (which are matrices of size $|\mathcal{L}| \times T$) forms a periodic, collision-free schedule. In this paper, we define a *weighted scheduling graph* and use the maximum-mean-cycle in it to solve (8).

Definition 2 (Weighted Scheduling Graph)**.** Given a weight vector $\mathbf{a} \in \mathbb{R}^{|\mathcal{L}|}$ and a scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$ whose vertices are matrices of size $|\mathcal{L}| \times T$, a *weighted scheduling graph* $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$ is a directed, weighted graph defined as follows: the vertex set is still $\mathcal{M}_T$, and each edge is associated with a weight. For a directed edge (v_1, v_2) in $(\mathcal{M}_T, \mathcal{E}_T)$, there is a weighed, directed edge (v_1, v_2) in $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$ with weight $w_{\mathbf{a}}(v_1, v_2) = \mathbf{a}^\intercal v_2 \mathbf{1}$, where $\mathbf{1} = [1, \ldots, 1]^\intercal \in \mathbb{R}^T$.

Since each achievable rate vector can be achieved by a periodic, collision-free schedule, which corresponds to a cycle in $(\mathcal{M}_T, \mathcal{E}_T)$ [21], we have the following result, whose proof can be found in [1].

Lemma 2. *For a weighted scheduling graph $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$ and its maximum-mean-cycle $\mathcal{C} = (v_0, v_1, \ldots, v_m)$ with $m \geq 0$ and $v_0 = v_m$, the concatenation of the vertices in $\mathcal{C}$ gives a periodic schedule S' such that $R_{S'} \in \arg\max_{R \in \mathcal{R}} \langle \mathbf{a}, R \rangle$.*

Therefore, given a vector $\mathbf{a}$, finding a vector that solves (8) is equivalent to finding the maximum-mean-cycle in $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$, which is a widely studied problem [32], [33] that can be solved with time complexity $\Theta(nm)$, where n is the number of nodes and m is the number of edges in the graph. A classical algorithm is the Karp's algorithm [32].

Algorithm 2 Algorithm for Networks with Delays

Input: a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I}, D)$
Output: maximum multiflow v

Start with any rate vector $R_1 \in \mathcal{R}$, $\mathcal{R}_1 \leftarrow \{R_1\}$
Construct $(\mathcal{M}_T, \mathcal{E}_T)$ from $\mathcal{N}$
for $i = 1, 2, \ldots$ **do**
  Run $v_i \leftarrow$LP-MMF$(\mathcal{R}_i)$ (or $v_i \leftarrow$LP-MCMF$(\mathcal{R}_i)$) and obtain the dual vector μ_i
  Construct $(\mathcal{M}_T, \mathcal{E}_T, w_{\mu_i})$ by $(\mathcal{M}_T, \mathcal{E}_T)$ and μ_i
  Find maximum-mean-cycle in $(\mathcal{M}_T, \mathcal{E}_T, w_{\mu_i})$ and obtain $R_{i+1} \leftarrow \arg\max_{R \in \mathcal{R}} \langle \mu_i, R \rangle$
  $\mathcal{R}_{i+1} \leftarrow \operatorname{conv}(\mathcal{R}_i \cup \{R_{i+1}\})$
  if $\langle \mu_i, R_{i+1} \rangle = \max_{R \in \mathcal{R}_i} \langle \mu_i, R \rangle$ **then**
    return v_i

Note we assume the graph $(\mathcal{M}_T, \mathcal{E}_T)$ is strongly connected, and hence $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$ is also strongly connected. Otherwise, we find the strongly connected components (with linear time complexity), search for the maximum-mean-cycle in each component and choose the one with the largest cycle mean.

Our approach is formally described in Algorithm 2, whose optimality can be proved similar to the proof of Theorem 1.

Theorem 3 (Optimality)**.** *For a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I}, D)$, Algorithm 2 will terminate and output the maximum multiflow (or the maximum concurrent multiflow).*

VI. Concluding Remarks

In large-scale wireless systems, the maximum (concurrent) multiflow problem is important for understanding network capacity; however, it is NP-hard even in simple network settings, making it computationally prohibitive to solve exactly. In this paper, we provide algorithms that jointly solve the MMF and MCMF problems, as well as the scheduling problem, in a general multi-source multi-sink network with network coding allowed and propagation delays potentially utilized in scheduling. Our algorithms use only a subset of the scheduling rate region for the MMF and MCMF problems, making them much more efficient without sacrificing solution accuracy or scheduling performance. We theoretically prove that our algorithms output optimal solutions in a finite number of iterations. The complete proof, as well as various simulation results demonstrating the advantages of our approaches, can be found in the full version of this paper [1].

VII. Acknowledgments

The work of Cheuk Ting Li was partially supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No.s: CUHK 24205621 (ECS), CUHK 14209823 (GRF)]. The work of Shenghao Yang was funded by the Shenzhen Fundamental Research Program (Grant No. JCYJ20241202124023031). This work was mostly completed while Yanxiao Liu was at The Chinese University of Hong Kong. He is currently a postdoctoral research associate at Imperial College London.

References

[1] Y. Liu, S. Yang, and C. T. Li, "Joint scheduling and multiflow maximization in wireless networks," *arXiv preprint arXiv:2509.14582*, 2025.

[2] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.

[3] B. S. Baker, "Approximation algorithms for np-complete problems on planar graphs," *Journal of the ACM (JACM)*, vol. 41, no. 1, pp. 153–180, 1994.

[4] P.-J. Wan, "Multiflows in multihop wireless networks," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, 2009, pp. 85–94.

[5] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[6] S.-Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE transactions on information theory*, vol. 49, no. 2, pp. 371–381, 2003.

[7] R. W. Yeung, *Information theory and network coding*. Springer Science & Business Media, 2008.

[8] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *Journal of the ACM (JACM)*, vol. 37, no. 2, pp. 318–334, 1990.

[9] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 42–54.

[10] ——, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, 2005, pp. 73–87.

[11] V. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "Algorithmic aspects of capacity in wireless networks," in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2005, pp. 133–144.

[12] H. Su and X. Zhang, "Characterizing the throughput gain of network coding in wireless ad hoc networks," in *MILCOM 2008-2008 IEEE Military Communications Conference*. IEEE, 2008, pp. 1–7.

[13] J. Zhou, S. Xia, Y. Jiang, H. Zheng, and L. Cui, "Maximum multiflow in wireless network coding," *IEICE transactions on communications*, vol. 96, no. 7, pp. 1780–1790, 2013.

[14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, 2006, pp. 243–254.

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

[16] D. Traskov, M. Heindlmaier, M. Médard, and R. Koetter, "Scheduling for network-coded multicast," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1479–1488, Oct. 2012.

[17] N. M. Jones, B. Shrader, and E. Modiano, "Optimal routing and scheduling for a simple network coding scheme," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 352–360.

[18] T. Cui, L. Chen, and T. Ho, "Distributed optimization in wireless networks using broadcast advantage," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 5839–5844.

[19] T. Wiese, M. Riemensberger, and W. Utschick, "Scheduling for network-coded multicast with interference," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2245–2254, 2016.

[20] R. Niati, A. H. Banihashemi, and T. Kunz, "Throughput and energy optimization in wireless networks: Joint mac scheduling and network coding," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1372–1382, 2012.

[21] J. Ma, Y. Liu, and S. Yang, "Rate region of scheduling a wireless network with discrete propagation delays," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[22] S. Yang, J. Ma, and Y. Liu, "Wireless network scheduling with discrete propagation delays: Theorems and algorithms," *arXiv 2107.03083*, 2023.

[23] Y. Fan, Y. Liu, and S. Yang, "Continuity of link scheduling rate region for wireless networks with propagation delays," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 730–735.

[24] M. Chitre, M. Motani, and S. Shahabudeen, "Throughput of networks with large propagation delays," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 645–658, Jul 2012.

[25] W. Bai, M. Motani, and H. Wang, "On the throughput of linear unicast underwater networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[26] A. Eryilmaz and D. S. Lun, "Control for inter-session network coding," in *Proc. Workshop on Network Coding, Theory & Applications*. Citeseer, 2007, pp. 1–9.

[27] M. Yang and Y. Yang, "A linear inter-session network coding scheme for multicast," in *2008 Seventh IEEE International Symposium on Network Computing and Applications*. IEEE, 2008, pp. 177–184.

[28] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1008–1014, 2011.

[29] C. T. Li, "Undecidability of network coding, conditional information inequalities, and conditional independence implication," *IEEE Transactions on Information Theory*, vol. 69, no. 6, pp. 3493–3510, 2023.

[30] C.-C. Hsu, K.-F. Lai, C.-F. Chou, and K.-J. Lin, "ST-MAC: Spatial-temporal MAC scheduling for underwater sensor networks," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 1827–1835.

[31] P. M. Gleiss, J. Leydold, and P. F. Stadler, "Circuit bases of strongly connected digraphs," 2001.

[32] R. M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete mathematics*, vol. 23, no. 3, pp. 309–311, 1978.

[33] A. Dasdan and R. K. Gupta, "Faster maximum and minimum mean cycle algorithms for system-performance analysis," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 17, no. 10, pp. 889–899, 1998.