Joint Scheduling and Multiflow Maximization in Wireless Networks

Yanxiao Liu, *Graduate Student Member, IEEE*, Shenghao Yang, *Member, IEEE* and Cheuk Ting Li, *Member, IEEE*

***Abstract*—Towards the development of 6G mobile, it is promising to integrate devices from satellites, aerial platforms, terrestrial base stations, and underwater environment. One critical issue in measuring the performance of a network is the maximum multiflow, which is usually solved in two steps: first find the scheduling rate region, and then find the maximum multiflow that can be supported by some achievable link rates. However, the scheduling problem is NP-hard, which makes solving the multiflow problem in a large-scale network computationally prohibitive. Moreover, another challenge that arises in multidimensional networks is the possibly longer propagation delays within and between different platforms. Though it has been shown that utilizing (instead of ignoring) the delays can enhance the network performance, this makes the scheduling problem even harder. In this paper, we provide a unified framework to efficiently solve the multiflow problem without the need of the whole scheduling rate region. This joint approach works in a general multi-source multi-sink network, with network coding performed on intermediate nodes, and propagation delays can be either ignored or non-negligible. We prove our algorithms output the optimal solutions (instead of approximations), and use experiments to show the advantages over conventional approaches.**

***Index Terms*—multihop network, maximum multiflow problem, network coding, wireless network scheduling, propagation delays**

I. Introduction

Over the past few years, both the number of users in the terrestrial wireless systems and the services to be provided have experienced significant growth. For the 6G mobile in the next generation wireless system, it is expected that a massive connectivity and emerging applications in various dimensions should be supported. Due to the limitation of the current terrestrial network capacity and coverage, an integrated network structure connecting segments from space, air, ground, ocean and underwater environments has attracted attention in recent years [1]–[3]. This structure is expected to achieve larger coverage and higher throughput. See [4]–[11] for other related literature. As an integrated ground-air-space-ocean network includes multidimensional segments, the performance will be simultaneously affected by all the segments in the coverage.

Y. Liu is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China. Email: yanxiaoliu@link.cuhk.edu.hk

S. Yang is with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Shenzhen 518172, China. Email: shyang@cuhk.edu.cn

C. T. Li is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China. Email: ctli@ie.cuhk.edu.hk.

To measure the performance of a large-scale, highly-connected network, it is crucial to find the maximum multiflow (MMF) and maximum concurrent multiflow (MCMF) that can be supported by some achievable collision-free link schedules of the network. The MMF (or MCMF) problem addresses the maximum total (or concurrent) throughput between multiple source nodes and sink nodes supported by the link rates. The key issues affecting the performance of multihop wireless networks are the wireless interference and the scheduling rate region. To solve the MMF (or MCMF) problem, we usually first use a *link conflict graph* to model the wireless interference [12], then convert the scheduling problem to the maximum independent set problem (which is a well-known NP-hard problem that is also hard to approximate [13]) in the graph to search collision-free schedules, and finally calculate the maximum (concurrent) throughput under the constraints described by the scheduling rate region. It has been shown in [14] that the MMF and MCMF problems are NP-hard even in very simple network settings. Although there are joint optimization methods with low complexity (see Section II-B for a review), most of them only approximate the throughput with certain constraints in a decentralized manner. However, we are interested in provably deriving the optimal solutions in a finite number of iterations.

In this paper, we provide both practical algorithms and theoretic analysis. Our algorithms can calculate the maximum (concurrent) multiflow values and the scheduling rates in multihop networks efficiently. By employing a decomposition method, we jointly solve the scheduling problem and the MMF (or MCMF) problem, and our algorithms can provably output the optimal results. Compared to the conventional two-step solution (first calculate the scheduling rate region, and then solve the MMF and MCMF problems), our algorithms only need a subset of the scheduling rate region and hence can be much more efficient in practice (see Section V for the performance evaluation). Moreover, most of the previous research on MMF (or MCMF) problems [14]–[16] only considers *multiple unicast* sessions, i.e., for each source, there is only one sink node with a certain demand. This assumption is not realistic in a large-scale ground-air-space-ocean network, which is expected to connect a massive number of devices/users from multidimensional platforms. As an advantage, our algorithms work on general multi-source multi-sink networks, and solve the *multiple multicast* case. It is well-known that in multihop networks, network coding [17]–[19] is a promising technique to improve the throughput and attain the maximum information rates in a multicast session. We also allow network coding to be performed on intermediate nodes to achieve the best performance. In a general multi-source multi-sink network with intra-session network coding, we prove the optimality of our algorithms.

Moreover, another critical challenge in developing a multidimensional network is the unavoidable longer propagation delay. In the existing theory of terrestrial wireless communications, though the communication media (e.g., radio, light, sound) have nonzero propagation delays, these delays are usually very short and are treated as a factor of interference [20]. However, in a network that connects devices from different platforms, e.g., satellites and underwater networks, the propagation delays can be significantly longer (see Section II-C for a review). Recent studies [21]–[27] have suggested that instead of mitigating the interference, we can actually utilize the delays to improve the throughput or the scheduling rates. See [21], [22] for a comprehensive study on scheduling wireless networks with propagation delays, where the conflict graph [12] is extended to a *weighted conflict graph*, and the scheduling problem could suffer even higher computational cost [21], [22]. Though the joint optimization of throughput, scheduling and network coding has been widely studied (see Section II-B), it is not straightforward to extend existing algorithms to networks with non-negligible delays. We will provide algorithms in a unified framework that can also solve the MMF and MCMF problems efficiently in networks with non-negligible delays.

In summary, in this paper, we provide efficient algorithms that can jointly solve the MMF (or MCMF) problem and the scheduling problem, with or without utilizing propagation delays, in a general multi-source multi-sink network with network coding performed on intermediate nodes, in a unified framework. The algorithms provably output the optimal solutions in a finite number of iterations. Compared to existing literature, the main contributions as follows:

- We consider the MMF and MCMF problems, which are important to measure the performance of an integrated network. Instead of solving them by a two-step scheme (calculating the scheduling rate region first, which is unrealistic due to the hardness of the scheduling problem), we directly solve the MMF and MCMF problems with only a subset of the scheduling rate region.
- Compared to the literature on the MMF (or MCMF) problem [12], [14], [15] for the multiple unicast case, we study *multiple multicast* in a general multi-source multi-sink network with network coding, which is more realistic in multidimensional networks.
- Instead of approximating the MMF (or MCMF) value by decentralized optimization methods, our algorithms provably output the exact optimal solutions in a finite number of iterations.
- For networks with non-negligible propagation delays, our algorithms can also utilize the delays in scheduling and solve the MMF (or MCMF) problem. With or without propagation delays in consideration, we solve the MMF and MCMF problems in a unified framework, indicating that our algorithms can be directly applied to the existing mobile systems.
- We prove the optimality of our algorithms on multi-source multi-sink networks with intra-session (not inter-session) network coding.

The remainder of the paper is organized as follows. We first review the related literature in Section II. In Section III, we first describe our network model, and then formulate the MMF and MCMF problems. We then propose two algorithms in Section IV for networks with negligible delays and with delays utilized in wireless scheduling, respectively, both of which can be proved to output optimal solutions in a finite number of iterations. We finally present the performance evaluation in Section V.

II. Related Works

In this section, we review the related literature, which are partitioned into three parts. Firstly, we discuss the literature on the MMF and MCMF problems. Secondly, we describe the existing joint optimization frameworks for throughput, scheduling, and/or network coding related to MMF and MCMF problems. Finally, we discuss the recent literature on improving the network performance by utilizing non-negligible propagation delays.

A. Maximum Multiflow Problem

We study two important factors for measuring the performance of a network: the maximum multiflow (MMF) and the maximum concurrent multiflow (MCMF). The maximum multiflow problem studies the maximum throughput between selected source nodes and sink nodes [14], and the maximum concurrent multiflow problem [28] models the case that every sender-receivers session transmits messages concurrently. The NP-hardness of both problems have been proved in [14], even in very simple network settings. In [29], [30], 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, [31] guarantees the schedulability and develops constant-approximation algorithms. More linear programming formulation and approximation algorithms with polynomial complexities can be found in [14], [28], [29]. In [16], 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 further extended to unicast networks with coding on intermediate nodes [15], where the network coding [17]–[19] is treated as a scheme to decrease the impact of wireless interference.

To support the (concurrent) multiflow in a network, it is usually required to find link rates that can be achieved by collision-free schedules, i.e., find the scheduling rate region (sometimes called the schedulability polytope [15]). This is called a *two-step* solution in this paper. To solve the scheduling problem, in [12], the effects of wireless interference between users have been modeled by a *conflict graph*. Then the problem of computing the scheduling rate region is equivalent to searching all the maximal independent sets in the conflict graph, which is an NP-hard problem that is also hard to approximate [13]. In a network, the model of the wireless interference is usually chosen to be the binary interference model [12], [15], [21] (which is also the case in this paper), though it is not difficult to cover other interference models, e.g., probabilistic interference model [32] can by covered by signal-to-interference-and-noise ratio, see [22], [32]. Due to the hardness of the scheduling problem, in practice it can be computationally prohibitive to calculate the scheduling rate region before solving the MMF (or MCMF) problem. We should note there exist optimization algorithms (possibly in a decentralized manner) that can approximate and optimize the solutions with a low complexity (see Section II-B), but our objective is to exactly find the (concurrent) multiflow value in an efficient way.

B. Joint Optimization Frameworks and Network Coding

Most of the literature on the MMF (or MCMF) problem [12], [14], [15] studied the multiple unicast case, i.e., each source node is paired with one sink node. However, we consider multiple multicast in a general multi-source multi-sink network, where each of a number of source nodes would like to transmit a message to a set of sink nodes. In the multicast scenario, usually it is expected to use network coding [17]–[19] to attain the maximum information flow rates. It allows a coding node to encode the received or overheard packets before transmitting (instead of directly routing) them to the next hop, and is an effective technique that can improve the network throughput in general. It enables the throughput to scale to dense large networks and gain increases up to several folds [33], [34]. In [15], network coding is used as a method to decrease the wireless interference for the MMF problem in multiple unicast case.

The joint consideration of throughput, scheduling and network coding has been widely studied in [32], [35]–[38] for various objectives, e.g., maximizing throughput, minimizing the energy consumption or performing decoding in the next hop, under different constraints, e.g., limited bandwidth, energy cost or particular coding schemes. The approaches in [32], [36]–[38] are either converging-to-optimal with respect to certain constraints or only approximations with decentralized/greedy optimization algorithms. Moreover, in [39] the network-coded multicast was formulated as searching edge-disjoint augmenting paths in the time-expanded graph to maximize the throughput.

In [35], the authors decompose the joint optimization of scheduling and network coding into two subproblems. We use a similar idea to decompose the joint MMF (or MCMF) and scheduling into two subproblems. Our approach is different in the following aspects. Except our approach can also cover the case where the propagation delays are non-negligible and utilized in scheduling (see Section II-C for a review and Section IV-B for the details of the corresponding algorithm) in a unified framework, 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. Moreover, the algorithm in [35] is an iterative algorithm that only converges to the optimum, but our algorithms can provably find the optimum in a finite number of iterations (also see Remark 3).

C. Propagation Delays in Multidimensional Networks

In the existing terrestrial wireless communication systems (e.g., 5G cellular network), the propagation delays are treated as a factor of interference since they are usually in the range of tens of microseconds and much shorter than the signal frame length [20]. However, in a multidimensional network that integrates components from satellites, aerial networks, terrestrial base stations and underwater systems, the propagation delays can be significantly longer. For example, the sound speed in underwater environment is about $1,500$ meters per second, and hence the delay is around 2 seconds for propagating over 3 kilometers; if the signal is transmitted in the outer space, the transmission is usually via multiple hops over a very long distance [40], also resulting in a relative longer delay. Even if only terrestrial radio communication is considered, with a large bandwidth, it is still practical to consider delays in orthogonal frequency-division multiplexing. See [22] for a related discussion on this topic.

Recently it has been observed that in scenarios where propagation delays are non-negligible, instead of ignoring the delays or employing guard intervals to mitigate the interference [28], [41]–[44], the delays can actually be *utilized* to improve the performance on energy consumption, throughput and link scheduling rate region [21]–[27], and some advantages can be even unbounded [23], [26]. In [24], [25], mixed integer linear programming has been used to study the collision constraints and provide heuristic algorithms. A dynamic-programming based algorithm has been proposed in [26]. In [21], [22], a graphical approaches has been proposed to characterize the scheduling rate region, which models the interference as an *weighted conflict graph* (in comparison, the conventional conflict graph is unweighted [12]). However, the scheduling problem becomes to suffer even higher computation cost. It is not straightforward to extend the existing algorithms for MMF (or MCMF) to networks with non-negligible delays. In this paper, we also provide efficient algorithms that can solve the MMF and MCMF problems in this scenario in a unified framework.

III. Network Model

In this section, we present the network model, the definitions about scheduling rates, and the formulation of the MMF and MCMF problems.

A. Network Model

We present a link-wise network model, similar to the model in [21]. The wireless interference between links is modeled by a general collision model, where each link is associated with a *collision set*, including all the links that can be interfered by it. Our collision model is called the binary collision model [12], [21]. It is not difficult to include the physical interference model by signal-to-interference-and-noise ratio, see [22], [32]. The collision graph in this model can also be extended to hypergraphs [22].

We assume the time is slotted and the link delays are multiples of a length of a time slot, and hence the network is called a *discrete network* [21]–[23]. The time-slotted assumption in scheduling has been justified in [23]. We assume the network is acyclic, and the intermediate nodes can wait until enough packets are collected before performing coding on the packet. The network coding with delays has also been considered in [45], [46].

We use a network model similar to [21]. The network is 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'). When delays are negligible, it reduces to an unweighted graph $(\mathcal{L}, \mathcal{I})$ [14], [15] and we denote the network by $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I})$. This graphical approach helps the discussion on algorithms for networks with non-negligible link delays.

We now describe the communication task over the network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{I}, D)$. We have $\mathcal{S} = \{s_1, \ldots, s_k\} \subseteq \mathcal{V}$ as the set of source nodes, $k \in \mathbb{N}_+$. We assume the information sources at different source nodes are mutually independent. For each source node s_i, it 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. Moreover, for $i \neq j$, we also may have $\mathcal{D}_{s_i} \cap \mathcal{D}_{s_j} \neq \emptyset$, i.e., different source nodes may share same sink nodes. In $\mathcal{L}$, 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.

We use a multihop line network to illustrate our model. The class of line networks has also been discussed in [21], [22], [26], [27].

Example 1 (Multihop line network)**.** Consider a multihop unicast line network with L hops: there are $L + 1$ nodes $\mathcal{V} = \{1, 2, \ldots, L + 1\}$, with link set

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

where each link is assumed to have a unit delay. Consider a K-hop interference model, where the reception of a node only has possible collisions from nodes within K hops distance. We assume the nodes are half-duplex. The collision set of link l_i is then

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

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

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

We denote such a network as $\mathcal{N}_{L,K}^{D=1}$. This class of networks can be represented by a graph. For example, for a line network with $L = 4$ links under 1-hop interference model, $\mathcal{N}_{4,1}^{D=1}$ is shown in Figure 1.

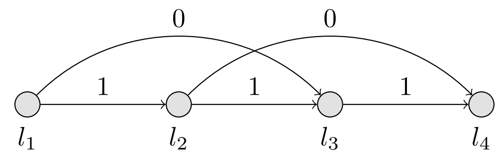


Fig. 1. In $\mathcal{N}_{4,1}^{D=1}$, each node represents a link, edges represent the collision relations and the link delays are represented by the weights on edges.

B. Collision-free Schedules and Rate Region

We review the concept of collision-free schedules in [21], [22] here. 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 can use a binary number to indicate whether a link is sending messages or not. Therefore, we can 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 in time slot t. $S(l, t)$ 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 of schedules if delays are not considered, e.g., see [14]. However, our definition would be helpful in the discussion of Algorithm 2 for networks with delays. For a collision-free schedule S and a link l, the link rate is

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

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 then define the flows, multiflows, the maximum multiflow (MMF) and the maximum concurrent multiflow (MCMF) problems. Most existing papers on MMF problems consider the multiple unicast setting, i.e., for each source node there is only one corresponding sink node with a certain demand [14]–[16]. However, we consider the general multi-source multi-sink network. We perform linear network coding [17]–[19] on intermediate nodes, and the nodes can encode its received data before passing it on.

In this paper, we consider multiple multicast sessions in the network, where for each multicast session, we have one source node and multiple corresponding sink nodes. Though we use network coding to attain the maximum information flow in a multicast session, we do not consider network coding between different sessions (i.e., inter-session network coding [47], [48]) in this paper for the sake of simplicity, since it is in general a hard problem [49] and can even be undecidable [50]. We say that

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

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 node s_i that multicasts a message to sink nodes in the set $\mathcal{D}_{s_i} = \{t_{i,1}, \ldots, t_{i,k_i}\}$. Suppose we have a flow $F_{i,j}$ from s_i to $t_{i,j}$ for each j. Network coding [17]–[19] allows us to multicast simultaneously to all these sink nodes at a rate

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

where 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. We do not consider coding between these flows. Hence 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 want to maximize the sum of the rates of multicasting these k sources. We call this the maximum multiflow (MMF) problem, which can be formulated as the following linear program that combines the MMF linear program for multiple unicast [12], [14], [15] and the linear program for single multicast [35]:

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

$$\text{maximize} \quad \sum_{i=1}^{k} v_i \tag{5}$$

subject to

$$\begin{aligned}
&F_{i,j} \text{ is a valid flow}, \forall i \in [k], j \in [k_i],\\
&\sum_{l \in \mathrm{Out}(s_i)} F_{i,j}(l) = \sum_{l \in \mathrm{In}(t_{i,j})} F_{i,j}(l) = v_i, \forall i \in [k], j \in [k_i],\\
&G_i(l) \geq F_{i,j}(l), \forall l \in \mathcal{L}, i \in [k], j \in [k_i],\\
&\sum_{i=1}^{k} G_i(l) \leq R(l), \forall l \in \mathcal{L}, \qquad (6)\\
&R \in \tilde{\mathcal{R}},
\end{aligned}$$

where $[k] := \{1, \ldots, k\}$, $k, k_i \in \mathbb{N}_+$ for any i. The linear program takes a polytope $\tilde{\mathcal{R}}$ (which may or may not be the whole scheduling rate region) as an input. Note that the variables $G_i(l)$, $i \in [k]$, $l \in \mathcal{L}$ are introduced to impose the constraint $\sum_{i=1}^{k} \max_j F_{i,j}(l) \leq R(l)$. The constraint (6) gives a dual vector that is going to be used in our algorithms, and the dual variable corresponding to the 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 [14], [28], [29]. The settings of MCMF problem in [14], [29] are also for the multiple unicast case. In this paper, we also study the MCMF problem jointly with scheduling in the multiple multicast setting.

More generally, instead of maximizing the sum rate $\sum_{i=1}^{k} v_i$ in the previous section, we maximize ϕ such that the source node 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 a linear program based on [14], [35] as follows.

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

$$\text{maximize} \quad \phi \tag{7}$$

subject to

$$\begin{aligned}
&F_{i,j} \text{ is a valid flow}, \forall i \in [k], j \in [k_i],\\
&\sum_{l \in \mathrm{Out}(s_i)} F_{i,j}(l) = \sum_{l \in \mathrm{In}(t_{i,j})} F_{i,j}(l) = \phi\gamma_i, \forall i \in [k], j \in [k_i],\\
&G_i(l) \geq F_{i,j}(l), \forall l \in \mathcal{L}, i \in [k], j \in [k_i],\\
&\sum_{i=1}^{k} G_i(l) \leq R(l), \forall l \in \mathcal{L}, \qquad (8)\\
&R \in \tilde{\mathcal{R}}.
\end{aligned}$$

Similarly, we will also use the dual vector given by the constraint (8) in our algorithms.

IV. ALGORITHMS

In this section, we start with discussing the conventional case, where link delays are assumed to be negligible. Then we discuss wireless networks with non-negligible link delays, which can also be a practical scenario in multidimensional networks integrating platforms from ocean to deep space. We introduce two algorithms for the two cases, respectively. The two algorithms are in a unified framework, indicating that our approaches can also be directly applied to the existing network architectures. For the wireless networks with non-negligible link delays, we use a similar graphical construction as used in [21], where the calculation of the scheduling rate region is formulated as a cycle-enumeration problem in a *scheduling graph*. In this paper, we model the search of rate vectors utilizing the dual vector in (6), (8) as a *maximum-mean-cycle problem* [51], [52]. We also prove the optimality of the algorithms, i.e., both algorithms output exact optimal solutions (instead of approximations) in a finite number of iterations.

A. Algorithm for Networks without Delays

We first discuss the case that link delays are negligible, which is the usual case assumed in literature. We propose an algorithm to calculate the maximum (concurrent) multiflow without calculating the whole scheduling rate region. We prove that our algorithm is guaranteed to find the maximum (concurrent) multiflow in a finite number of iterations (instead of only converging to the optimum). The algorithm developed for networks without delays will have a natural extension to the networks with delays.

For networks without delays, 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., we would like to solve

$$\arg\max_{R \in \mathcal{R}} \langle \mathbf{a}, R \rangle, \tag{9}$$

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

Remark 1. For (9), 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 to the graphical approach for solving the MMF and MCMF problems in networks with non-negligible delays, see Definition 2.

This corresponds to a weighted maximal independent set problem [14], [35], which can be solved via integer linear programming (ILP) as follows.

$$\begin{aligned} \text{ILP:} \quad & \text{maximize} \quad \sum_{i=1}^{|\mathcal{L}|} a_i S(l_i) \\ & \text{subject to} \quad S(l_i) + S(l_j) \leq 1, \quad \forall\, l_i, l_j : l_j \in \mathcal{I}(l_i). \end{aligned}$$

where we maximize over the variables $S(l_i) \in \{0, 1\}$ for $l_i \in \mathcal{L}$. 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 $\mathcal{R}$ (which is a convex polytope [12]).

Algorithm 1 Algorithm for Networks without Delays

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

1: Start with any rate vector $R_1 \in \mathcal{R}$, $\mathcal{R}_1 \leftarrow \{R_1\}$
2: **for** $i = 1, 2, \dots$ **do**
3: 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
4: Run ILP to find $R_{i+1} \leftarrow \arg\max_{R \in \mathcal{R}} \langle \mu_i, R \rangle$
5: $\mathcal{R}_{i+1} \leftarrow \mathrm{conv}\,(\mathcal{R}_i \cup \{R_{i+1}\})$
6: **if** $\langle \mu_i, R_{i+1} \rangle = \max_{R \in \mathcal{R}_i} \langle \mu_i, R \rangle$ **then**
7: **return** v_i

Based on (9), we iteratively search the maximum (concurrent) multiflow and the scheduling rate region. We will show that even though our target is the optimal value instead of approximated or converging-to-optimal values, it is unnecessary to find the whole scheduling rate region before solving the maximum (concurrent) multiflow problem. Suppose $\mathrm{flow}(\cdot)$ is the function for calculating maximum (concurrent) multiflow in a given polytope, which might be a subset of the rate region. Starting with $i = 1$, in each iteration, the algorithm works as follows:

1) We start with a subset of the scheduling rate region, denoted by $\mathcal{R}_i$, which is formed by the vertices of $\mathcal{R}$ we have known (it is reasonable to assume some achievable rate vectors are known, e.g., by activating the first link all the time and inactivating all the other links, we know that the rate vector $[1, 0, \dots, 0]^\intercal$ is achievable). In the first iteration, we start with an arbitrarily chosen achievable 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 (6) (or (8)).
2) We calculate a new achievable rate vector R_{i+1} by
$$R_{i+1} = \arg\max_{R \in \mathcal{R}} \langle \mu_i, R \rangle, \tag{10}$$
that is solved by the integer program ILP.
3) We update the subset of scheduling rate region for the next iteration by computing the convex hull $\mathcal{R}_{i+1} = \mathrm{conv}\,(\mathcal{R}_i \cup \{R_{i+1}\})$.
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, we continue and come back to step 1 for the next iteration.

The algorithm is formally described in Algorithm 1. We then prove it will terminate and output the maximum multilflow (or the maximum concurrent multiflow) in a finite number of iterations.

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).*

Proof. Since the scheduling rate region is a convex polytope [12], it has a finite number of vertices. Therefore the Algorithm 1 will eventually terminate, since the worst case is that all the vertices are found to solve the maximum (concurrent) multiflow problem.

We then show the output is optimal, i.e., when the algorithm terminates, the output is exactly the maximum multiflow (or the maximum concurrent multiflow). We use $\mathrm{flow}(R)$ to denote the function that outputs the maximum (concurrent) multiflow with respect to a rate vector R.

Denote $\mathcal{R}$ as the entire rate region (which may not need to be found), and $\mathcal{R}_i$ is the subset of $\mathcal{R}$ found until iteration i. Suppose the algorithm terminates at iteration i', i.e., $\langle \mu_{i'}, R_{i'+1} \rangle = \max_{R \in \mathcal{R}_{i'}} \langle \mu_{i'}, R \rangle$. By substituting (10), we have

$$\max_{R \in \mathcal{R}} \langle \mu_{i'}, R \rangle = \max_{R \in \mathcal{R}_{i'}} \langle \mu_{i'}, R \rangle. \tag{11}$$

Suppose the optimal R in LP-MMF($\mathcal{R}_{i'}$) (or LP-MCMF($\mathcal{R}_{i'}$)) is R^*. Note that LP-MMF($\mathcal{R}_{i'}$) can be written as

$$\begin{aligned} &\text{maximize} \;\; \mathrm{flow}(R) - \chi_{\mathcal{R}_{i'}}(\bar{R}) \\ &\text{subject to} \;\; R = \bar{R}, \end{aligned} \tag{12}$$

where $\chi_{\mathcal{R}_{i'}}(\bar{R})$ is the characteristic function (which is 0 if $\bar{R} \in \mathcal{R}_{i'}$, or ∞ otherwise), which forces $\bar{R} \in \mathcal{R}_{i'}$. The dual vector $\mu_{i'}$ corresponding to the constraint in (6) (or (8)) is the same as the dual vector corresponding to the constraint $R = \bar{R}$ in (12). Considering the Lagrangian of (12), at the optimum $(R^*, \bar{R}^*)$, the subgradient satisfies $0 \in \partial_{R^*} \mathrm{flow}(R^*) - \partial_{R^*} \chi_{\mathcal{R}_{i'}}(\bar{R}^*) - \mu_{i'}$ and $0 \in \partial_{\bar{R}^*} \mathrm{flow}(R^*) - \partial_{\bar{R}^*} \chi_{\mathcal{R}_{i'}}(\bar{R}^*) + \mu_{i'}$, and we have $R^* = \bar{R}^*$. Hence, R^* maximizes $\mathrm{flow}(R) - \langle \mu_{i'}, R \rangle$ for $R \in \mathbb{R}_{\geq 0}^{|\mathcal{L}|}$, and maximizes $\langle \mu_{i'}, R \rangle$ for $R \in \mathcal{R}_{i'}$.

Fix any $R' \in \mathcal{R}$. Since R^* maximizes $\mathrm{flow}(R) - \langle \mu_{i'}, R \rangle$ for $R \in \mathbb{R}_{\geq 0}^{|\mathcal{L}|}$,

$$\begin{aligned} &\mathrm{flow}(R^*) - \langle \mu_{i'}, R^* \rangle \\ &\geq \mathrm{flow}(R') - \langle \mu_{i'}, R' \rangle \\ &\geq \mathrm{flow}(R') - \langle \mu_{i'}, R^* \rangle, \end{aligned}$$

where the last inequality is by (11) and the fact that R^* maximizes $\langle \mu_{i'}, R \rangle$ for $R \in \mathcal{R}_{i'}$. Therefore, R^* maximizes $\mathrm{flow}(R)$ for $R \in \mathcal{R}$. When the algorithm terminates, the output is the maximum (concurrent) multiflow.

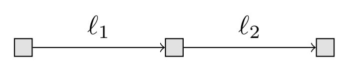
□

Remark 2. The Algorithm 1 requires integer linear programming, hence it does not have a polynomial time complexity, which is expected since this problem is NP-hard [12], [14]. Algorithm 1 and Theorem 1 pave the way to the algorithm for networks with non-negligible propagation delays in the next subsection, where the optimality can also be proved using the same arguments as Theorem 1.

Remark 3. In [35], an algorithm based on subgradient optimization, which also decomposes the problem into two parts, has been discussed. Although their algorithm shares some similarities with ours, one difference is that our algorithm is guaranteed to find the optimum exactly in a finite number of steps (assuming access to an integer linear programming algorithm), whereas [35] is an iterative algorithm that only converges to the optimum. As we will see in the next section, terminating in a finite 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 rate region is the bottleneck of the algorithm with exponential time complexity, and should be performed as little as possible.

We use a simple example to illustrate the Algorithm 1, which is also in the class of multihop line networks introduced in Example 1.

Example 2 (2-hop Line Network)**.** Suppose we have a 3-node, 2-link unicast line network $\mathcal{N}_{2,1}^{D=0}$ under the 1-hop interference model, and the propagation delay on each link is negligible (i.e., zero).



Starting with $R_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^\intercal$, we solve

$$\begin{aligned} \max \quad & v = R(l_1) = R(l_2) \\ \text{s.t.} \quad & R(l_1) \leq 1, \\ & R(l_2) \leq 0, \\ & R(l_1), R(l_2) \geq 0, \end{aligned}$$

which gives us a maximum flow of value $v = 0$ in $\mathcal{R}_i = \{R_1\}$. We then use the ILP to find another rate vector:

$$\begin{aligned} \max \quad & \sum_{i=1}^{2} R(l_i) \\ \text{s.t.} \quad & R(l_1) + R(l_2) \leq 1, \end{aligned}$$

which gives $R_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\intercal$ and $\mathcal{R}_2 = \mathrm{conv}(R_1, R_2)$. Next,

$$\begin{aligned} \max \quad & v = R(l_1) = R(l_2) \\ \text{s.t.} \quad & R(l_1) + R(l_2) \leq 1, \\ & R(l_1), R(l_2) \geq 0 \end{aligned}$$

gives us the maximum flow in $\mathcal{R}_2$, $v = 1/2$. We can verify that the resulting dual vector μ_2 in this iteration gives

$$f(\mathcal{R}_2, \mu_2) = \arg\max_{R \in \mathcal{R}_2} f(\mathcal{R}_2, \mu_2),$$

which meets the condition that the algorithm terminates. It is also straightforward to verify that $1/2$ is indeed the maximum flow value.

B. Algorithm for Networks with Delays

We then extend the discussion to networks with non-negligible delays, which is practical in an integrated networks consisting of components from ocean, air and deep space. We propose an algorithm to solve this case, which makes use of a graphical characterization from [21], [22] and also in a same framework with Algorithm 1.

By utilizing link delays in networks, the throughput or the scheduling rate region may increase [21], [22], [26]. The key of solving the MMF (or MCMF) problem in the network with link delays is, we require a function similar with (9) that can output a vertex of the scheduling rate region in a time complexity at most exponential in the number of links $|\mathcal{L}|$ (which in turn will be polynomial in the size of the *scheduling graphs* shown below), which is more efficient than the cycle-enumeration approach in [21] (which is doubly exponential in $|\mathcal{L}|$).

We first review a graph proposed in [21], called the *scheduling graph* as follows. For a collision-free schedule matrix S and for integers $T \in \mathbb{N}_+$, $k \in \mathbb{Z}$, we use $S[T, k]$ to denote the submatrix of S with its columns $kT, kT+1, \ldots, (k+1)T-1$. If a submatrix S' is formed by T columns of S, we index the columns of S' by $0, 1, \ldots, T-1$.

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.

Remark 4. For a network with negligible propagation delays, the scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$ is a complete graph.

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

Instead of enumerating simple cycles for the scheduling rate region, we search *maximum-mean-cycles* (which can be solved in a time complexity polynomial in the size of the graph) in a new graph, to find some vertices of the scheduling rate region. Then we may only need to find 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 [53], 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 *minimum-mean-cycle* is the cycle in the given weighted, directed graph with the minimum average weight over all directed cycles in the given graph, and the average weight is called the *minimum cycle mean*. The *maximum-mean-cycle* is defined similarly.

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 (9).

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.

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*

$$R_{S'} \in \arg\max_{R \in \mathcal{R}} \langle \mathbf{a}, R \rangle.$$

Proof. For any $R \in \mathcal{R}$, suppose it is achieved by a schedule S that corresponds to a cycle $(v_0, v_1, \ldots, v_m)$ in $(\mathcal{M}_T, \mathcal{E}_T)$, by the definition of rate vectors (3). Given $\mathbf{a} = (a_i)_{i=1,\ldots,|\mathcal{L}|}$,

$$\begin{aligned}
\langle \mathbf{a}, R \rangle &= \sum_{i=1}^{|\mathcal{L}|} a_i R(l_i) \\
&= \sum_{i=1}^{|\mathcal{L}|} a_i \cdot \frac{1}{mT} \sum_{t=0}^{mT-1} S(l_i, t) \\
&= \frac{1}{mT} \sum_{i=1}^{|\mathcal{L}|} \sum_{t=0}^{mT-1} a_i \cdot S(l_i, t) \\
&= \sum_{j=0}^{m-1} \left(\frac{1}{mT} \sum_{i=1}^{|\mathcal{L}|} \sum_{k=0}^{T-1} a_i \left(v_j(i, k) \right) \right) \\
&= \frac{1}{mT} \sum_{j=0}^{m-1} \mathbf{a}^\intercal v_j \mathbf{1},
\end{aligned}$$

which states that the inner product $\langle \mathbf{a}, R \rangle$ equals to the average over values $\mathbf{a}^\intercal v_j \mathbf{1}$ for $j = 0, 1, \ldots, m-1$. Therefore, it justifies the construction of $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$ and shows that it is indeed the maximum-mean-cycle that attains the maximum of $\langle \mathbf{a}, R \rangle$. $\square$

Therefore, given a vector $\mathbf{a}$, finding a rate vector that solves (9) is equivalent to finding the maximum-mean-cycle in $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$. The maximum (or minimum) mean cycle problem has been widely studied [51], [52], which 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 with polynomial time complexity is Karp's algorithm [51]. We briefly review the Karp's algorithm as follows for the sake of completeness.

Given the graph with vertex set $\mathcal{V}$ and a source node $s \in \mathcal{V}$, for each $v \in \mathcal{V}$ and every non-negative integer k, suppose $F_k(v)$ denotes the maximum weight of a length-k path from s to v, and we say $F_k(v) = -\infty$ if such a path does not exist. Then the maximum cycle mean λ^* can be derived from the following theorem, whose proof can be found in [51].

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) of $(\mathcal{M}_T, \mathcal{E}_T, w_{\mathbf{a}})$, search the maximum-mean-cycle in each component and choose the one with the largest maximum cycle mean.

Theorem 3 (Karp's Theorem [51])**.** *Given a strongly connected graph, the maximum cycle mean λ^* is given by*

$$\lambda^* = \max_{v \in \mathcal{V}} \min_{0 \le k \le n-1} \frac{F_n(v) - F_k(v)}{n-k}, \tag{13}$$

where $\mathcal{V}$ is the vertex set of the graph.

The value of $F_k(v)$ can be given by a recurrence relation in [51], where $\mathcal{E}$ denotes the edge set and $w(u,v)$ denotes the weight on the edge (u,v):

$$F_k(v) = \max_{(u,v) \in \mathcal{E}} [F_{k-1}(u) + w(u,v)], \qquad k = 1, 2, \ldots, n$$

with the initial conditions $F_0(s) = 0$ and $F_0(v) = -\infty$, $v \neq s$. The Karp's algorithm computes $F_k(v)$ recurrently for $k = 0, 1, \ldots, n$ and $v \in \mathcal{V}$. For a more discussion on this algorithm, we refer the readers to [51]. In [52], the authors further improve the Karp's algorithm by using a graph-unfolding scheme.

Finally, we present our algorithm for the networks with non-negligible link delays in Algorithm 2. The scheduling rate region of a network with non-negligible propagation delays can also be proved to be a convex polytope [21], [22] which has a finite number of vertices, and hence the Algorithm 2 will also terminate eventually.

Theorem 4 (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).*

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 ∈ 𝓡, 𝓡_1 ← {R_1}
Construct (𝓜_T, 𝓔_T) from 𝓝
for i = 1, 2, ... do
    Run v_i ← LP-MMF(𝓡_i) (or v_i ← LP-MCMF(𝓡_i)) and obtain the dual vector μ_i
    Construct (𝓜_T, 𝓔_T, w_{μ_i}) by (𝓜_T, 𝓔_T) and μ_i
    Find maximum-mean-cycle in (𝓜_T, 𝓔_T, w_{μ_i}) and obtain R_{i+1} ← arg max_{R∈𝓡} ⟨μ_i, R⟩
    𝓡_{i+1} ← conv(𝓡_i ∪ {R_{i+1}})
    if ⟨μ_i, R_{i+1}⟩ = max_{R∈𝓡_i} ⟨μ_i, R⟩ then
        return v_i
```

We can also prove the Algorithm 2 outputs the maximum (concurrent) multiflow in a way similar to the proof of Theorem 1. The only difference is, we use Lemma 2 to justify that in the iteration i' when the algorithm terminates, the maximum-mean-cycle in $(\mathcal{M}_T, \mathcal{E}_T, \mu_{i'})$ gives

$$R_{i'+1} = \arg\max_{R \in \mathcal{R}} \langle \mu_{i'}, R \rangle,$$

and then we substitute it into

$$\langle \mu_{i'}, R_{i'+1} \rangle = \max_{R \in \mathcal{R}_{i'}} \langle \mu_{i'}, R \rangle,$$

and then we have the same result with (11). The remaining steps are similar with the proof of Theorem 1 and hence are omitted.

Remark 5. Karp's algorithm can find one maximum-mean-cycle in a time complexity polynomial in $|\mathcal{E}_T|$, the number of edges in the weighted scheduling graph. Nevertheless, $|\mathcal{E}_T|$ is exponential in $|\mathcal{L}|$, the number of links in the communication network. Therefore, the overall running time is at least exponential in $|\mathcal{L}|$. We note that finding the entire scheduling rate region generally has a time complexity exponential in $|\mathcal{E}_T|$, which should be doubly exponential in $|\mathcal{L}|$. As demonstrated in experiments in Section V, our algorithms can be significantly faster than the two-step algorithm for both MMF and MCMF problems, since we do not require finding the entire scheduling rate region.

We use a 4-link line network as introduced in Example 1 to illustrate our approach, and similar examples have also been discussed in [21], [22], [26], [27].

Example 3 (4-hop line network with unit link delay)**.** We use $\mathcal{N}_{4,1}^{D=1}$ as an example (see Figure 1). At first, we construct the scheduling graph. We find $T = \max_{l \in \mathcal{L}} \max_{l' \in \mathcal{I}(l)} |D(l, l')| = 1$, and hence denote the scheduling graph as $(\mathcal{M}_1, \mathcal{E}_1)$. The vertex set includes matrices of size 4×1 such that each vertex v can be a column of a certain collision-free schedule S.

We have a vertex set $\mathcal{M}_1$ $\{v_0, v_1, \ldots, v_8\}$, where

$$\begin{aligned}
v_0 &= \begin{bmatrix}0\\0\\0\\0\end{bmatrix}, v_1 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, v_2 = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \\
v_3 &= \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, v_4 = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}, v_5 = \begin{bmatrix}1\\0\\0\\1\end{bmatrix}, \\
v_6 &= \begin{bmatrix}1\\1\\0\\0\end{bmatrix}, v_7 = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}, v_8 = \begin{bmatrix}0\\0\\1\\1\end{bmatrix}.
\end{aligned} \tag{14}$$

The edge set $\mathcal{E}_1$ can be described as an adjacent matrix as follows.

$$\begin{array}{c c}
 & \begin{array}{ccccccccc} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\
\begin{array}{c} v_0\\ v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6\\ v_7\\ v_8 \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1\\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1\\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0\\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}. \tag{15}$$

By implementing the algorithms in [21] we can find the scheduling rate region $\mathcal{R}$, whose vertices are:

$$\begin{bmatrix}1/2\\1/2\\1/2\\1/2\end{bmatrix}, \begin{bmatrix}1/2\\1/2\\1/2\\0\end{bmatrix}, \begin{bmatrix}0\\1/2\\1/2\\1/2\end{bmatrix}, \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\0\\0\\0\end{bmatrix}. \tag{16}$$

Then we elaborate that we can find the maximum flow without the need of finding the whole scheduling rate region shown in (16).

Suppose we start with a rate vector $R_1 = \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}^\intercal$, and in the first iteration we start with solving the following linear program

$$\begin{aligned}
\max \quad & v = R(l_1) = R(l_2) = R(l_3) = R(l_4) \\
\text{s.t. } & R(l_i) \leq 0 \qquad i = 1, 3, 4 \\
& R(l_2) \leq 1 \\
& R(l_i) \geq 0 \qquad i = 1, 2, 3, 4
\end{aligned}$$

We obtain the maximum flow $v = 0$ and the corresponding dual vector is $\mu_1 = \begin{bmatrix}\epsilon & 0 & \epsilon & 0\end{bmatrix}^\intercal$ with some $\epsilon > 0$. We then use μ_1 to construct the corresponding weighted scheduling graph. Consider the vertex set (14) and the edge set (15), for $i = 0, 1, \ldots, 8$, the weights on the edges of the weighted scheduling graph are defined as

$$\begin{aligned}
w(v_i, v_0) = 0,\ \ w(v_i, v_1) = \epsilon,\ \ w(v_i, v_2) = 0,\\
w(v_i, v_3) = \epsilon,\ \ w(v_i, v_4) = 0,\ \ w(v_i, v_5) = \epsilon,\\
w(v_i, v_6) = \epsilon,\ \ w(v_i, v_7) = \epsilon,\ \ w(v_i, v_8) = \epsilon.
\end{aligned}$$

We can verify that there are 9 vertices and 56 edges in the weighted scheduling graph.

We use the maximum-mean-cycle algorithm to solve $\arg\max_{R \in \mathcal{R}} \langle \mu_1, R_1 \rangle$, and we find $R_2 = \begin{bmatrix}1/2 & 1/2 & 1/2 & 1/2\end{bmatrix}^\intercal$. It can be checked that the algorithm terminates in the next step.

Compared to the vertices of $\mathcal{R}$ shown in (16), our approach only need to find two of them. The initial starting point (R_1) is easy to find by searching an arbitrary maximal independent set of $\mathcal{N}_{4,1}^{D=1}$ (shown in Figure 1). Moreover, the maximum flow value $1/2$ indeed achieves the upper bound of line networks with delays proved in [27].

V. Performance Evaluation

In this section, we evaluate our algorithms on different wireless networks for MMF and MCMF problems. We compare our simulation results to the performance of the two-step solutions (i.e., first calculate the scheduling rate region and then find the maximal multiflow value). We note again that although there exist joint optimization algorithms that can approximate the maximum (concurrent) multiflow values with low complexity (see Section II-B), we are interested in efficiently deriving the exactly optimal solution in a finite number of iterations.

We choose simple network settings for the sake of simplicity, which are enough to show the advantages of our algorithms. For both the multiflow problem and the scheduling problem, since the networks without delays can be treated as special cases of the networks with non-negligible propagation delays, we mainly discuss the latter in the performance evaluation. Due to the hardness of the scheduling problem when propagation delays are taken into account, the two-step solution can only handle networks with a small number of links, thus limiting our evaluation on networks with simple topologies. Therefore, we select the network settings from the class of line networks (also see Example 1, 2 and 3), which have also been used for illustration and performance evaluation in [21], [22], [26]. It is worth noting that the choice of $D = 1$ is for the purpose of simplification: for networks with wither larger or smaller delays, the same methodology can be applied in a straightforward way (see [21], [22] for a detailed discussion).

We evaluate our algorithms on three problems that have been discussed through the paper: single flow maximization, the MMF problem and the MCMF problem. They are illustrated as follows.

1) **Single flow maximization**: We start with the simplest case. We consider a unicast multihop line network under the 1-hop interference model, i.e., $\mathcal{N}_{L,1}^{D=1}$ introduced in

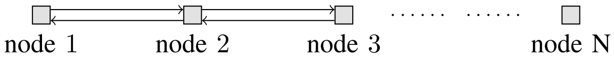


Fig. 2. For the MMF and MCMF problems, we consider bi-directional line networks. For $i = 1, \ldots, N-1$, there are two links with opposite directions between node i and node $i+1$.

Example 1 (also see Figure 1). The only source in this network is the first node, and the only sink is the last node.

2) **Multiflow maximization**: We then utilize bi-directional multihop line networks for evaluating the MMF problem, as shown in Figure 2. We assume the interference model is the *single collision domain* [23], [26], i.e., in the network only one link can be active in a time slot. There are two information flows in the network, one is from node 1 to node N, and another one is from node N to node 1.
3) **Concurrent multiflow maximization**: We then use the same bi-directional multihop line network in Figure 2 to evaluate the MCMF problem, under the same interference model. For the two flows with opposite directions, we require that one's desired traffic rate is the half of another one's desired traffic rate.

For all the networks in this section, we assume each link has a unit capacity and a unit delay. Though the network topology is simple, it is sufficient for us to illustrate the advantages of our algorithms over conventional two-step methods. Though our algorithms can be applied to more complicated networks, the conventional two-step methods are restricted to networks with small sizes. The multihop line networks have also been discussed in [21], [22], [26], [27] to demonstrate the advantages (throughput and the scheduling rate region) of utilizing link delays in link scheduling.

We consider two approaches:

- **Two-step algorithm**: To find the optimal solution (instead of approximations), the conventional method is to first calculate the entire scheduling rate region, and then solve the maximum (concurrent) flow of the network. We use the algorithms proposed in [21] to calculate the scheduling rate region.
- **Our algorithm**: We implement Algorithm 2 to solve three cases: the single flow maximization, the MMF problem and the MCMF problem, for all of which we do not need to calculate the entire scheduling rate region first.

We compare the two approaches by the time required to calculate the maximum (concurrent) multiflow.[1] We should note that the comparison is between the time required by the two approaches to find the optimal (concurrent) multiflow values, therefore the reduction in search time by our algorithms does not sacrifice any scheduling performance.

[1]For each L, Algorithm 2 starts with $R_1 = [1, 0, 0, \ldots, 0]^\intercal$. The algorithm is implemented in Python, and executed on a laptop computer with i7-8550u CPU and Python 3.7.

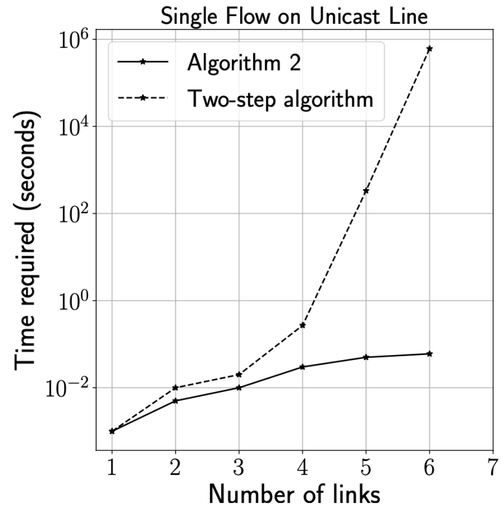


Fig. 3. Single flow maximization in unicast line network $\mathcal{N}_{L,K}^{D=1}$. The x-axis is the number of links and the y-axis (in logarithmic scale) is the required time (seconds) to calculate the maximum flow value.

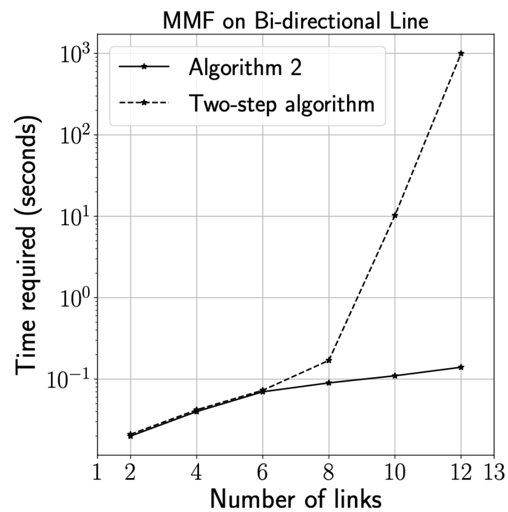


Fig. 4. MMF problem in bi-directional line networks. The x-axis is the number of links and the y-axis (in logarithmic scale) is the required time (seconds) to calculate the maximum multiflow value.

The performance evaluation on single flow maximization, the MMF problem and the MCMF problem are shown in Figure 3, Figure 4 and Figure 5, respectively. The running time (in logarithmic scale) is plotted against the number of links L. It can be found that due to the NP-hardness of the scheduling problem, the required time for solving the multiflow problem by the two-step solution increases doubly-exponentially fast, which matches the theoretic study in [21] and shows that calculating the scheduling rate region is the key bottleneck of this problem.

In comparison, the figures show that the time required by our algorithms to achieve the optimal (concurrent) multiflow values is significantly lower. This highlights the advantage of searching only a few rate vectors. For example, the algorithm

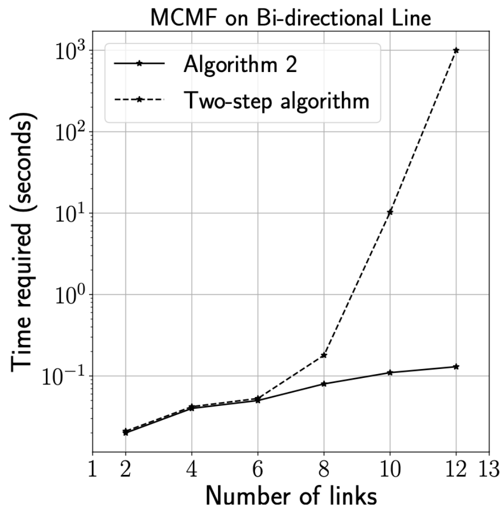


Fig. 5. MCMF problem in bi-directional line networks, respectively. The x-axis is the number of links and the y-axis (in logarithmic scale) is the required time (seconds) to calculate the maximum concurrent multiflow value.

in [21] needs to search 7653 achievable rate vectors to characterize the scheduling rate region of $\mathcal{N}_{4,1}^{D=1}$, which has 9 vertices as shown in (16), while we only need to find 2 of them for solving the MMF (or MCMF) problem. For $\mathcal{N}_{6,1}^{D=1}$ whose rate region has 57 vertices, we only need 4 of them. The two-step method requires more than a week to calculate the scheduling rate region of $\mathcal{N}_{6,1}^{D=1}$, but our Algorithm 2 can find the maximum flow value in less than one second. In more complicated network settings, the scheduling problem will be even harder to solve, which would make our joint algorithms more preferable.

VI. Conclusion

In a real-world multidimensional wireless system, though calculating the maximum (concurrent) multiflow value is important for measuring the performance of this network, it is computationally prohibitive to calculate the scheduling rate region for such problems, and most of the linear programming algorithms with low complexities only consider either approximated solutions or specific networks under certain constraints. In this paper, we jointly solve the MMF (or MCMF) and scheduling problems, in a general multi-source multi-sink network with network coding allowed (which is almost the most general setting). Our algorithms may only need to find a very small subset of the scheduling rate region to solve the MMF and MCMF problems, which makes our approaches efficient.

To integrate multi-platform networks with components from ground, air, ocean and space, it is also important to consider the unavoidable longer propagation delays within and between different platforms in certain environments. We provide algorithms to solve the MMF and MCMF problems in networks with or without long propagation delays, in a unified framework. Moreover, we prove that our algorithms output optimal solutions in a finite number of iterations. Experiments show that even in very simple network settings, our algorithms are significantly faster than the two-step methods without sacrificing any accuracy of the solution.

VII. Future Works

In this paper, we utilize the maximum-mean-cycle algorithms in solving the MMF (or MCMF) problem. We find that it can also be used to calculate the throughput (the objective in [26], [27]) or the entire scheduling rate region (the objective in [21], [22]) as follows. The convex hull method [54] is an algorithm for finding the vertices of an unknown polytope $\mathcal{P} \subseteq \mathbb{R}^n$, given an oracle that can find $\operatorname{argmax}_{x\in\mathcal{P}}\langle x, a\rangle$ for $a \in \mathbb{R}^n$. By using the maximum-mean-cycle algorithm as the oracle to the convex hull method, we can iteratively search the vertices of the entire scheduling rate region. This method can be more efficient than the cycle-enumeration method in $(\mathcal{M}_T, \mathcal{E}_T)$ [21], since we do not need to enumerate all the simple cycles in $(\mathcal{M}_T, \mathcal{E}_T)$. Detailed implementations of such algorithms for solving the scheduling rate region is left for future study.

VIII. Acknowledgement

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 partially supported by the National Key R&D Program of China under Grant 2022YFA1005000.

References

[1] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[2] N. Cheng, H. Jingchao, Y. Zhisheng, Z. Conghao, W. Huaqing, L. Feng, Z. Haibo, and S. Xuemin, "6g service-oriented space-air-ground integrated network: A survey," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 1–18, 2022.

[3] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6g," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 53–87, 2021.

[4] J. Cui, S. X. Ng, D. Liu, J. Zhang, A. Nallanathan, and L. Hanzo, "Multiobjective optimization for integrated ground-air-space networks: Current research and future challenges," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 88–98, 2021.

[5] X. Cao, B. Yang, C. Yuen, and Z. Han, "Hap-reserved communications in space-air-ground integrated networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8286–8291, 2021.

[6] P. K. Chowdhury, M. Atiquzzaman, and W. Ivancic, "Handover schemes in satellite networks: State-of-the-art and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 2–14, 2006.

[7] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2442–2473, 2016.

[8] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[9] C. Niephaus, M. Kretschmer, and G. Ghinea, "Qos provisioning in converged satellite and terrestrial networks: A survey of the state-of-the-art," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2415–2441, 2016.

[10] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "Optimizing computation offloading in satellite-uav-served 6g iot: A deep learning approach," *IEEE Network*, vol. 35, no. 4, pp. 102–108, 2021.

[11] H. Guo, X. Zhou, J. Liu, and Y. Zhang, "Vehicular intelligence in 6g: Networking, communications, and computing," *Vehicular Communications*, vol. 33, p. 100399, 2022.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] J. G. Proakis, "Intersymbol interference in digital communication systems," *Wiley Encyclopedia of Telecommunications*, 2003.

[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.

[24] 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.

[25] Y. Guan, C.-C. Shen, and J. Yackoski, "MAC scheduling for high throughput underwater acoustic networks," in *Proc. IEEE WCNC 2011*, Cancún, Mexico, Mar 2011, pp. 197–202.

[26] 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.

[27] 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.

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

[29] 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.

[30] ——, "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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] K. Rajawat and G. B. Giannakis, "Joint scheduling and network coding for multicast in delay-constrained wireless networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6186–6196, 2011.

[40] Z. Huakai, D. Guangliang, and L. Haitao, "Simplified bats codes for deep space multihop networks," in *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*. IEEE, 2016, pp. 311–314.

[41] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 910–917, Sept 1988.

[42] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on communications*, vol. 38, no. 4, pp. 456–460, 1990.

[43] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec 1992.

[44] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006, pp. 227–238.

[45] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM transactions on networking*, vol. 11, no. 5, pp. 782–795, 2003.

[46] Y. Wu, "A trellis connectivity analysis of random linear network coding with buffering," in *2006 IEEE International Symposium on Information Theory*. IEEE, 2006, pp. 768–772.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

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

[52] 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.

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

[54] C. Lassez, "Quantifier elimination for conjunctions of linear constraints via a convex hull algorithm," *Symbolic and Numerical Computation for Artificial Intelligence*, pp. 103–122, 1992.