

# A New Paradigm for Multiflow in Wireless Networks: Theory And Applications

Peng-Jun Wan\*, Boliu Xu<sup>†</sup>, Lei Wang<sup>‡</sup>, Sai Ji<sup>§</sup>, and Ophir Frieder<sup>¶</sup>

\* Department of Computer Science, Illinois Institute of Technology, wan@cs.iit.edu

<sup>†</sup> Department of Computer Science, Illinois Institute of Technology, bxu8@iit.edu

<sup>‡</sup> School of Software, Dalian University of Technology, lei.wang@ieee.org

<sup>§</sup> College of Computer And Software, Nanjing University of Information Science And Technology, jisai@nuist.edu.cn

<sup>¶</sup> Department of Computer Science, Georgetown University, ophir@ir.cs.georgetown.edu

**Abstract**—Multiflow problems are one of the most fundamental problems in both wired networks and wireless networks. Due to the cross-layer nature, multiflow problems in wireless networks are significantly harder than their counterparts in wired networks and have received much research interest over the past decade. Common to most other early-staged research, the characterization of computational hardness and the “war” on achievable approximation bounds have been the priority to the existing studies of multiflow problems in wireless networks while their practical feasibility in both running time and memory requirement is ignored as long they are polynomial. In fact, almost all of the state-of-the-art approximation algorithms for multiflow problems in wireless networks are all resorted to the traditional linear programming (LP) methods exclusively. However, those traditional LP methods can require an inordinate amount of running time and memory even for a moderate sized input, and consequently they often prove unusable in practice. This paper presents a completely new paradigm for multiflow problems in general wireless networks which is radically different from the prevailing LP-based paradigm, and develops practical algorithmic solutions which are much faster and simpler.

## I. INTRODUCTION

Consider a multihop wireless network  $(V, A, \mathcal{I})$ , where  $V$  is the set of networking nodes,  $A$  is the set of communication links, and  $\mathcal{I}$  is the collection of independent set of links specified by an arbitrary interference model. Each link has a unit communication data rate. For any positive integer  $k$ , let  $[k]$  denote the set  $\{1, 2, \dots, k\}$ . Suppose that we are given  $k$  end-to-end unicast communication requests. For each  $j \in [k]$ ,  $\mathcal{P}_j$  denotes the set of simple paths of the request  $j$ ,  $\mathcal{F}_j$  denotes the set of flows of the request  $j$ , and the value of a flow  $f_j \in \mathcal{F}_j$  is denoted by  $val(f_j)$ . A multiflow is a sequence  $\vec{f} = \langle f_1, f_2, \dots, f_k \rangle$  with  $f_j \in \mathcal{F}_j$  for each  $j \in [k]$ . Let  $f = \langle f_1, f_2, \dots, f_k \rangle$  be a multiflow. The transmission time of each link  $a \in A$  required by  $f$  is  $\sum_{j \in [k]} f_j(a)$ . The total value of  $f$  is  $\sum_{j \in [k]} val(f_j)$ . Given that each request  $j$  has a positive traffic demand  $d_j$ , the concurrency of  $f$  is  $\min_{j \in [k]} val(f_j)/d_j$ . Among all variants of the multiflow problems [1], the following two are the most basic ones.

- **Maximum Multiflow (MMF)**: The problem MMF seeks a multiflow  $f$  and a MAC-layer link schedule  $\mathcal{S}$  of  $\sum_{j \in [k]} f_j$  such that the length of  $\mathcal{S}$  is at most one and the total value of  $f$  is maximized.
- **Maximum Concurrent Multiflow (MCMF)**: Given that each request has a traffic demand, the problem

MCMF seeks a multiflow  $f$  and a MAC-layer link schedule  $\mathcal{S}$  of  $\sum_{j \in [k]} f_j$  such that the length of  $\mathcal{S}$  is at most one and the concurrency of  $f$  is maximized.

Both MMF and MCMF admit polynomial-time approximation-preserving reduction [22], [25] to the following MAC-Layer optimization problem:

- **Maximum Weighted Independent Set (MWIS)**: Given a non-negative weight function  $w$  on  $A$ , the problem MWIS seeks an independent subset  $I$  of  $A$  with maximum total weight  $\sum_{a \in I} w(a)$ .

However, such reduction is based on ellipsoid method for linear programming (LP), which is quite inefficient in practice [19].

This paper develops a completely new paradigm for multiflow problems in general wireless networks which is radically different from the prevailing linear programming (LP) based paradigm. We first establish solid theoretic foundations consisting of the weak dualities of MMF and MCMF and a simple yet powerful game-theoretic framework. Upon them we design practical approximation algorithms for MMF and MCMF which involves only a sequence of computations of shortest paths and independent sets. These algorithms offer nice trade-off between accuracy in terms the approximation bound and efficiency in terms of the running time. Specifically, let  $\mathcal{A}$  be a  $\mu$ -approximation algorithm for MWIS, and  $\varepsilon \in (0, 1/2]$  be an accuracy-efficiency trade-off parameter. The approximation algorithms developed in this paper make  $O(\varepsilon^{-2})$  calls to  $\mathcal{A}$  and Dijkstra’s shortest path algorithm, and hence is faster and simpler. On the other hand, they achieve an approximation bound  $(1 + 2\varepsilon)\mu$ . Finally, we apply these general algorithms to derive effective and efficient approximation algorithms for MMF and MCMF in wireless networks under the physical interference model, and in wireless multiple-input multiple-output (MIMO) networks with receiver-side interference suppression under the protocol interference model.

The remainder of this paper is organized as follows. Section II reviews the technical approaches for multiflow problems in wireless networks and their drawbacks. Section III establishes weak dualities of MMF and MCMF, which reveal the intrinsic relations among them, shortest paths, and maximum-weighted independent sets. Section IV introduces a generic adaptive coupled game. Section V and Section VI describe the general design and analyses of the approximation algorithms

for **MCMF** and **MMF** respectively. Section VII presents the applications of these general algorithms to **MMF** and **MCMF** in wireless networks under the physical interference model, and in wireless MIMO networks with receiver-side interference suppression under the protocol interference model. Finally, we conclude this paper in Section VIII.

## II. RELATED WORKS

Most of the existing studies on multiflow problems in wireless networks assumed some variants of the protocol interference model (e.g., [11], [13], [14], [15], [22], [24], [26], [28]). Except for [28], all these studies are all resorted to the traditional linear programming (LP) methods *exclusively*. However, state-of-the-art implementations of the simplex method, interior-point method and ellipsoid method can require an inordinate amount of running time and memory even for a moderate sized input, and consequently they often prove unusable in practice [4]. This is mostly due to very expensive matrix computations, such as matrix inversions or Cholesky factorizations, that must be repeatedly carried out. As an example, consider a maximum concurrent flow problem of  $k$  communication requests on a simplest single-channel single-radio (SC-SR) wireless network with  $n$  nodes and  $m$  communication links. The overall constraint matrix for the linear program will have  $m + kn$  rows and  $km$  columns. Even when the network is very sparse ( $m = \Theta(n)$ ), if we have  $k = \Theta(n)$  the numerical linear algebra will be over matrices of dimension  $\Theta(n^2)$ ; and even though the constraint matrix may be very sparse, the matrix computations will usually experience “fill-in”. In [28], a fast combinatorial method was proposed which requires only a sequence of shortest path computations. This method takes the *unique* advantage of the protocol interference model which admits explicit polynomial capacity subregions [26]. The lack of such advantage in wireless networks under the physical interference model and wireless MIMO networks under the protocol inference model makes the method in [28] inapplicable to these wireless networks.

Multiflow problems in wireless networks under physical interference model is notoriously hard due to the non-locality and the additive nature of the wireless interference under the physical interference model. There are only a few existing works on multiflow problems in wireless networks under the physical interference model [6], [25], [27], all of which follow the prevailing LP-based paradigm. Again, these LP-based algorithms are inefficient in both running time and memory.

MIMO technology provides a means of boosting network capacity without requiring additional spectrum. It has received widespread attention over the past decade from both industry and academic researchers, now forming a key component of nearly all emerging wireless standards. However, the complicated constraints on independence imposes significant technical challenges on the effective and efficient utilization of the MIMO capability. Multiflow problems in wireless MIMO networks under protocol interference model has been studied in [9], [10], [16], [20], [29]. Except for [29], all these studies are purely heuristic without any provable performance guarantees. In [29], constant-approximation algorithms based on the ellipsoid method for LP were proposed for multiflow problems in wireless MIMO networks with receiver-side interference

suppression. However, these algorithms are quite inefficient in practice [19].

## III. WEAK DUALITIES

Suppose that  $y$  is positive function on  $A$ . For each  $j \in [k]$ , let  $dist_j(y)$  be the length of a shortest of the  $j$ -th request with respect to  $y$ . This section establishes the following weak dualities of **MMF** and **MCMF** respectively.

*Theorem 3.1:* The concurrency of the maximum concurrent multiflow is at most

$$\frac{\max_{I \in \mathcal{I}} y(I)}{\sum_{j \in [k]} d_j dist_j(y)}$$

*Theorem 3.2:* The total value of the maximum multiflow is at most

$$\frac{\max_{I \in \mathcal{I}} y(I)}{\min_{j \in [k]} dist_j(y)}$$

For each  $j \in [k]$ ,  $\mathcal{P}_j$  denotes the set of simple paths of the request  $j$ . Consider a multiflow  $f = \langle f_1, \dots, f_k \rangle$ . A non-negative function  $x$  on  $\bigcup_{j \in [k]} \mathcal{P}_j$  is said to be a path-flow decomposition of  $f$  if for each  $a \in A$  and each  $j \in [k]$ ,

$$f_j(a) = \sum_{P \in \mathcal{P}_j} |\{a\} \cap P| x(P).$$

A non-negative function  $z$  on  $\mathcal{I}$  is said to be a (fractional) link schedule of  $f$  if for each  $a \in A$ ,

$$\sum_{j=1}^k f_j(a) = \sum_{I \in \mathcal{I}} |\{a\} \cap I| z(I).$$

The following invariant property holds.

*Lemma 3.3:* For any path-flow decomposition  $x$  of  $f$  and any link schedule  $z$  of  $f$ ,

$$\sum_{j=1}^k \sum_{P \in \mathcal{P}_j} x(P) y(P) = \sum_{I \in \mathcal{I}} y(I) z(I)$$

*Proof:* Since

$$\begin{aligned} & \sum_{j=1}^k \sum_{P \in \mathcal{P}_j} x(P) y(P) \\ &= \sum_{j=1}^k \sum_{P \in \mathcal{P}_j} x(P) \sum_{a \in A} y(a) |\{a\} \cap P| \\ &= \sum_{a \in A} y(a) \sum_{j=1}^k \sum_{P \in \mathcal{P}_j} x(P) |\{a\} \cap P| \\ &= \sum_{a \in A} y(a) \sum_{j=1}^k f_j(a) \\ &= \sum_{a \in A} y(a) \sum_{I \in \mathcal{I}} |\{a\} \cap I| z(I) \\ &= \sum_{I \in \mathcal{I}} z(I) \sum_{a \in A} y(a) |\{a\} \cap I| \\ &= \sum_{I \in \mathcal{I}} y(I) z(I), \end{aligned}$$

the lemma holds. ■

*Lemma 3.4:* If  $f$  is feasible, then

$$\sum_{j=1}^k dist_j(y) val(f_j) \leq \max_{I \in \mathcal{I}} y(I)$$

*Proof:* Consider any path-flow decomposition  $x$  of  $f$  and any shortest link schedule  $z$  of  $f$ . On one hand,

$$\begin{aligned} & \sum_{j=1}^k \sum_{P \in \mathcal{P}_j} x(P) y(P) \\ & \geq \sum_{j=1}^k \sum_{P \in \mathcal{P}_j} x(P) dist_j(y) \\ & = \sum_{j=1}^k dist_j(y) \sum_{P \in \mathcal{P}_j} x(P) \\ & = \sum_{j=1}^k dist_j(y) val(f_j) \end{aligned}$$

On the other hand,

$$\sum_{I \in \mathcal{I}} y(I) z(I) \leq \left( \max_{I \in \mathcal{I}} y(I) \right) \sum_{I \in \mathcal{I}} z(I) \leq \max_{I \in \mathcal{I}} y(I).$$

By Lemma 3.3,

$$\sum_{j=1}^k dist_j(y) val(f_j) \leq \max_{I \in \mathcal{I}} y(I).$$

So, the lemma holds. ■

Next, we prove Theorem 3.1. Let  $f = \langle f_1, \dots, f_k \rangle$  be a maximum concurrent multifold. By Lemma 3.4,

$$\begin{aligned} \max_{I \in \mathcal{I}} y(I) & \geq \sum_{j=1}^k dist_j(y) val(f_j) \\ & = \sum_{j=1}^k d_j dist_j(y) \frac{val(f_j)}{d_j} \\ & \geq \left( \min_{j \in [k]} \frac{val(f_j)}{d_j} \right) \sum_{j=1}^k d_j dist_j(y). \end{aligned}$$

Thus,

$$\min_{j \in [k]} \frac{val(f_j)}{d_j} \leq \frac{\max_{I \in \mathcal{I}} y(I)}{\sum_{j=1}^k d_j dist_j(y)}.$$

So, Theorem 3.1 holds.

Finally, we prove Theorem 3.2. Let  $f = \langle f_1, \dots, f_k \rangle$  be a maximum multifold. By Lemma 3.4,

$$\begin{aligned} \max_{I \in \mathcal{I}} y(I) & \geq \sum_{j \in [k]} dist_j(y) val(f_j) \\ & \geq \left( \min_{j \in [k]} dist_j(y) \right) \sum_{j \in [k]} val(f_j). \end{aligned}$$

Thus,

$$\sum_{j \in [k]} val(f_j) \leq \frac{\max_{I \in \mathcal{I}} y(I)}{\min_{j \in [k]} dist_j(y)}.$$

So, Theorem 3.2 holds.

#### IV. AN ADAPTIVE COUPLED GAME

In this section, we introduce an adaptive coupled game, which generalizes the problem considered by Auer et al. [3], Vovk [21], Cesa-Bianchi et al. [5], Freund and Schapire [7], [8], Khandekar [12], and Arora et al. [2] in the context of learning or game theory. The game playing strategy to be described in this subsection makes both the algorithm designs and analyses proposed later in this paper fairly modular and clarifies the high-level structure of the argument. We believe that our general treatment on the game playing strategy will help to facilitate its application to other settings easily.

In the adaptive coupled game, a sequential game is played between an adversary and an agent who is advised by a set  $E$  of experts. Prior to the first round of the game, the agent chooses a positive weight  $y(e)$  for each expert  $e \in E$ . In each round, the adversary determines a non-negative profit  $p(e)$  and a non-negative loss  $l(e)$  for each  $e \in E$  subject to two rules:

- **Normalization Rule:**

$$\max_{e \in E} |p(e) - l(e)| = 1.$$

- **Generalized Zero-Sum Rule:**

$$\sum_{e \in E} y(e) (p(e) - l(e)) \geq 0$$

The agent may then update the weight  $y(e)$  of each expert  $e \in E$  after observing the profits and loss of the experts. Given a parameter  $\eta \in (0, 1)$ , the objective of the agent is to maintain that at the end of each round for *each* expert  $e$ , the cumulative profit of  $e$  minus  $\eta$  times the cumulative loss of  $e$  is lower-bounded by some value invariant to the round number; the objective of the adversaries is exactly the opposite.

Now, we describe the **Multiplicative Weights Update (MWU)** strategy for the agent, while leaving the strategy for the adversary to specific applications. Fix an  $\varepsilon \in (0, 1)$ . For each expert  $e \in E$ ,

- $y(e) = 1$  initially;
- in each game round, if  $e$  earns a profit  $p(e)$  and incurs a loss  $l(e)$ ,  $y(e)$  is updated by a multiplicative factor

$$1 - \varepsilon (p(e) - l(e)).$$

An implementation of the game playing with this strategy is described as follows. Let  $P(e)$ ,  $L(e)$ , and  $y(e)$  be cumulative profit, cumulative loss, and weight respectively of each  $e \in E$ , which are initially 0, 0, and 1 respectively. Repeat following rounds:

- 1) **Generation of Profits/Losses:** The adversary determines a non-negative profit  $p(e)$  and loss  $l(e)$  for

each  $e \in E$  subject to the **Normalization Rule** and **Generalized Zero-Sum Rule**. As the result,

$$\begin{aligned} L(e) &\leftarrow L(e) + l(e); \\ P(e) &\leftarrow P(e) + p(e). \end{aligned}$$

- 2) **Multiplicative Weights Update:** The agent updates  $y(e)$  for each  $e \in E$  by setting

$$y(e) \leftarrow y(e) (1 - \varepsilon (p(e) - l(e))).$$

The effectiveness of above implementation of the game is asserted in the theorem below.

*Theorem 4.1:* At the end of each round, for any  $e \in E$ ,

$$\frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} L(e) \leq P(e) + \frac{\ln m}{\ln(1-\varepsilon)^{-1}}$$

*Proof:* For each round  $r$  and each  $e \in E$ , let  $p_r(e)$  (respectively,  $l_r(e)$ ) denote the profit (respectively, loss, weight) of  $e$  received in round  $r$ ; let  $P_r(e)$  (respectively,  $L_r(e)$ ,  $y_r(e)$ ) denote the cumulative profit (respectively, cumulative loss, weight) of  $e$  at the end of the round  $r$ . In addition, for each  $e \in E$ , let

$$\begin{aligned} P_0(e) &= L_0(e) = 0, \\ y_0(e) &= 1. \end{aligned}$$

We first claim that for any round  $t$ ,  $y_t(E) \leq y_{t-1}(E)$ . Indeed, by the **Generalized Zero-Sum Rule** we have

$$\begin{aligned} y_t(E) &= \sum_{e \in E} y_t(e) \\ &= \sum_{e \in E} y_{t-1}(e) (1 - \varepsilon ((p_t(e) - l_t(e)))) \\ &= \sum_{e \in E} y_{t-1}(e) - \varepsilon \sum_{e \in E} y_{t-1}(e) ((p_t(e) - l_t(e))) \\ &\leq \sum_{e \in E} y_{t-1}(e) \\ &\leq y_{t-1}(E). \end{aligned}$$

So, the claim holds.

Next, we claim that for any round  $t$  and any expert  $e$ ,

$$y_t(e) \geq (1 - \varepsilon)^{P_t(e)} (1 + \varepsilon)^{L_t(e)}.$$

Indeed, let

$$\begin{aligned} [t]^+ &= \{i \in [t] : p_i(e) \geq l_i(e)\}, \\ [t]^- &= \{i \in [t] : p_i(e) < l_i(e)\}. \end{aligned}$$

By the **Normalization Rule** and the inequality

$$(1 \pm \varepsilon)^x \leq 1 \pm \varepsilon x$$

for any  $x \in [0, 1]$ , we have

$$\begin{aligned} y_t(e) &= \prod_{i \in [t]} (1 - \varepsilon (p_i(e) - l_i(e))) \\ &= \prod_{i \in [t]^+} (1 - \varepsilon (p_i(e) - l_i(e))) \\ &\quad \cdot \prod_{i \in [t]^-} (1 + \varepsilon (l_i(e) - p_i(e))) \\ &\geq \prod_{i \in [t]^+} (1 - \varepsilon)^{p_i(e) - l_i(e)} \cdot \prod_{i \in [t]^-} (1 + \varepsilon)^{l_i(e) - p_i(e)} \\ &= (1 - \varepsilon)^{\sum_{i \in [t]^+} (p_i(e) - l_i(e))} (1 + \varepsilon)^{\sum_{i \in [t]^-} (l_i(e) - p_i(e))} \\ &= (1 - \varepsilon)^{\sum_{i \in [t]^+} p_i(e)} \left( \frac{1}{1 - \varepsilon} \right)^{\sum_{i \in [t]^+} l_i(e)} \\ &\quad \left( \frac{1}{1 + \varepsilon} \right)^{\sum_{i \in [t]^-} p_i(e)} (1 + \varepsilon)^{\sum_{i \in [t]^-} l_i(e)} \\ &\geq (1 - \varepsilon)^{\sum_{i \in [t]^+} p_i(e)} (1 + \varepsilon)^{\sum_{i \in [t]^+} l_i(e)} \\ &\quad (1 - \varepsilon)^{\sum_{i \in [t]^-} p_i(e)} (1 + \varepsilon)^{\sum_{i \in [t]^-} l_i(e)} \\ &= (1 - \varepsilon)^{\sum_{i \in [t]} p_i(e)} (1 + \varepsilon)^{\sum_{i \in [t]} l_i(e)} \\ &= (1 - \varepsilon)^{P_t(e)} (1 + \varepsilon)^{L_t(e)}. \end{aligned}$$

Thus, the claim holds.

The previous two claims yield that for any round  $t$  and any expert  $e$ ,

$$(1 - \varepsilon)^{P_t(e)} (1 + \varepsilon)^{L_t(e)} \leq y_t(e) \leq y_t(E) \leq y_0(E) = m.$$

Taking the logarithm of both sides, we have

$$P_t(e) \ln(1 - \varepsilon) + L_t(e) \ln(1 + \varepsilon) \leq \ln m,$$

Hence,

$$\frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} L_t(e) \leq P_t(e) + \frac{\ln m}{\ln(1-\varepsilon)^{-1}}.$$

So, the theorem holds.  $\blacksquare$

## V. MAXIMUM CONCURRENT MULTIFLOW

Let  $\mathcal{A}$  be a  $\mu$ -approximation algorithm for **MWIS**, and  $\varepsilon \in (0, 1/2]$  be an accuracy-efficiency trade-off parameter. This section presents a  $(1 + 2\varepsilon)\mu$ -approximation algorithm **CMF-LS**( $\varepsilon$ ) for **MCMF**. We first give an overview on the design of the algorithm **CMF-LS**( $\varepsilon$ ). Let  $opt$  be the concurrency of a maximum concurrent multiflow. The algorithm is iterative. Each iteration first computes a multiflow  $f$  of concurrency  $\ell$ , a scaling factor  $\lambda \geq \frac{opt}{\mu}$ , and a “primary” link schedule  $\Gamma$  of length  $\ell$  and its corresponding link transmission-time function  $g$ . With respect to the triple  $(f, \lambda, g)$ , the *deficit* of a link  $a$  is defined to be

$$d'(a) = \max \left\{ 0, \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda \sum_{j \in [k]} f_j(a) - g(a) \right\}.$$

For such deficit demand  $d'$ , a “complementary” link schedule  $\Gamma'$  is then computed. Clearly,  $\Gamma \cup \Gamma'$  is a link schedule of the

multiflow

$$\frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda f.$$

Let  $\ell'$  be the length of  $\Gamma'$ . If

$$\frac{\ell'}{\ell} \leq \varepsilon' := \frac{(1+2\varepsilon)\ln(1+\varepsilon) + \ln(1-\varepsilon)}{\ln(1-\varepsilon)^{-1}},$$

then the scaled multiflow

$$\frac{1}{\ell + \ell'} \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda f$$

and the scaled schedule

$$\frac{1}{\ell + \ell'} (\Gamma \cup \Gamma')$$

are returned as the output. The concurrency of the returned multiflow is

$$\begin{aligned} \frac{1}{\ell + \ell'} \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda \ell &= \frac{1}{1 + \ell'/\ell} \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda \\ &\geq \frac{\lambda}{1+2\varepsilon} \geq \frac{opt}{(1+2\varepsilon)\mu}. \end{aligned}$$

In order to compute the complementary link schedule quickly, a preprocessing is performed to partition all links into independent sets. On such partition  $\mathcal{J}$  can be obtained by applying the algorithm **Greedy Scheduling** given in [30], and let  $l = |\mathcal{J}|$ . Then, a complementary link schedule  $\Gamma'$  of a deficit demand  $d'$  is computed as follows. Initially,  $\Gamma'$  is empty. For each  $J \in \mathcal{J}$ , let

$$\delta' = \max_{a \in J} d'(a),$$

and if  $\delta' > 0$  then add the pair  $(J, \delta')$  to  $\Gamma'$ . The length of  $\Gamma'$  is

$$\ell' = \sum_{J \in \mathcal{J}} \max_{a \in J} d'(a) \leq |\mathcal{J}| \max_{a \in A} d'(a) = l \max_{a \in A} d'(a).$$

The algorithm **CMF-LS**( $\varepsilon$ ) is outlined in Table I. The computation of the primary schedule  $\Gamma$  (together with its length  $\ell$  and its link transmission time function  $g$ ), the multiflow  $f$ , and the scaling factor  $\lambda$  in each round follows the general framework of the coupled game introduced in Section IV. Each link is regarded as an expert, and each iteration corresponds to a game round. The agent plays exactly with the strategy on maintaining the expert/link weight function  $y$  described in Section IV. The profit (respectively, loss) generation strategy of the adversary is coupled with the primary schedule (respectively, multiflow) augmentation with the following *invariant* property maintained throughout the game: At the end of each round, the cumulative profit of each link  $a$  is exactly  $g(a)$ , and the cumulative loss of each link is at least  $\lambda \sum_{j \in [k]} f_j(a)$ . Initially, both the multiflow  $f$  and the primary schedule  $\Gamma$  are empty, and the scaling factor  $\lambda$  is infinity. In each round, the adversary generates the profits/losses as follows. Let  $I$  be the independent set w.r.t.  $y$  output by the algorithm  $\mathcal{A}$ , and  $P_j$  be a shortest path of the  $j$ -th request w.r.t.  $y$  for each  $j \in [k]$ . If  $\lambda$  is greater than

$$\frac{y(I)}{\sum_{j \in [k]} d_j y(P_j)} = \frac{y(I)}{\sum_{j \in [k]} d_j \text{dist}_j(y)},$$

Algorithm **CMF-LS**( $\varepsilon$ ):

<pre> // initialization <math>\forall j \in [k], f_j \leftarrow \mathbf{0}; \Gamma \leftarrow \emptyset, \ell \leftarrow 0, g \leftarrow \mathbf{0}; \lambda \leftarrow \infty;</math> <math>y \leftarrow \mathbf{1}, \varepsilon' \leftarrow \frac{(1+2\varepsilon)\ln(1+\varepsilon) + \ln(1-\varepsilon)}{\ln(1-\varepsilon)^{-1}};</math> <math>\mathcal{J} \leftarrow</math> a link partition output by <b>Greedy Scheduling</b>; repeat forever   // primary schedule and flow augmentation   <math>I \leftarrow</math> a <math>y</math>-weighted IS in <math>\mathcal{I}</math> output by <math>\mathcal{A}</math>;   <math>\forall j \in [k], P_j \leftarrow</math> a shortest <math>j</math>-path w.r.t. <math>y</math>;   <math>\lambda \leftarrow \min \left\{ \lambda, \frac{y(I)}{\sum_{j \in [k]} d_j y(P_j)} \right\}</math>; //truncation   <math>\delta \leftarrow \frac{1}{\max_{a \in A} \left(  \{a\} \cap I  - \lambda \sum_{j=1}^k d_j  \{a\} \cap P_j  \right)}</math>;   <math>\Gamma \leftarrow \Gamma \cup \{(I, \delta)\}, \ell \leftarrow \ell + \delta</math>;   for each <math>a \in I</math> do <math>g(a) \leftarrow g(a) + \delta</math>;   for each <math>j \in [k]</math> do     for each <math>a \in P_j</math> do <math>f_j(a) \leftarrow f_j(a) + \delta d_j</math>;   // complementary schedule   <math>\Gamma' \leftarrow \emptyset, \ell' \leftarrow 0</math>;   for each <math>J \in \mathcal{J}</math> do     <math>\delta' \leftarrow \max_{a \in J} \max \left\{ 0, \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda \sum_{j \in [k]} f_j(a) - g(a) \right\}</math>;     if <math>\delta' &gt; 0</math> then <math>\Gamma' \leftarrow \Gamma' \cup \{(J, \delta')\}, \ell' \leftarrow \ell' + \delta'</math>;   // test of termination   if <math>\ell' \leq \varepsilon' \ell</math> then return <math>\frac{1}{\ell + \ell'} \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} f</math> and <math>\frac{1}{\ell + \ell'} (\Gamma \cup \Gamma')</math>.   // multiplicative weights update   for each <math>a \in A</math> do     <math>y(a) \leftarrow y(a) \left( 1 - \varepsilon \delta \left(  \{a\} \cap I  - \lambda \sum_{j \in [k]} d_j  \{a\} \cap P_j  \right) \right)</math>; </pre>
--

TABLE I. OUTLINE OF THE ALGORITHM **CMF-LS**( $\varepsilon$ ).

then  $\lambda$  is reset to

$$\frac{y(I)}{\sum_{j \in [k]} d_j \text{dist}_j(y)}.$$

Thus,  $\lambda$  is non-increasing with the round number. For some positive length  $\delta$  to be determined shortly,  $\Gamma$  is augmented by the pair  $(I, \delta)$ , and both  $\ell$  and  $g$  are updated accordingly. The profit of each link  $a$  is exactly its transmission time received from the augmenting pair  $(I, \delta)$ , which is equal to  $\delta |\{a\} \cap I|$ . Thus, the cumulative profit of each link  $a$  is exactly  $g(a)$ . For each  $j \in [k]$ ,  $f_j$  is augmented by a flow of value  $\delta d_j$  along the path  $P_j$ . The loss of each link  $a$  is exactly  $\lambda$  times the total amount of the augmenting flow through  $a$ , which is equal to

$$\delta \lambda \sum_{j=1}^k d_j |\{a\} \cap P_j|.$$

Since  $\lambda$  is non-increasing with the round number, the cumulative loss of each link  $a$  is at least  $\lambda \sum_{j \in [k]} f_j(a)$ . The **Normalization Rule** dictates that

$$\delta = \frac{1}{\max_{a \in A} \left( |\{a\} \cap I| - \lambda \sum_{j=1}^k d_j |\{a\} \cap P_j| \right)}.$$

Since

$$\begin{aligned} &\sum_{a \in A} y(a) \left( \delta |\{a\} \cap I| - \delta \lambda \sum_{j \in [k]} d_j |\{a\} \cap P_j| \right) \\ &= \delta \left( \sum_{a \in A} y(a) |\{a\} \cap I| - \lambda \sum_{a \in A} y(a) \sum_{j=1}^k d_j |\{a\} \cap P_j| \right) \\ &= \delta \left( y(I) - \lambda \sum_{j=1}^k d_j \sum_{a \in A} y(a) |\{a\} \cap P_j| \right) \end{aligned}$$

$$= \delta \left( y(I) - \lambda \sum_{j=1}^k d_j y_{t-1}(P_j) \right) \geq 0,$$

the **Generalized Zero-Sum Rule** is also satisfied by the profit/loss assignment in this round.

Subsequently, the complementary link schedule  $\Gamma'$  of the deficit demand is computed. By Theorem 4.1, for each link  $a$ ,

$$\frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda \sum_{j \in [k]} f_j(a) \leq g(a) + \frac{\ln m}{\ln(1-\varepsilon)^{-1}}.$$

This means that the deficit of each link is at most

$$\frac{\ln m}{\ln(1-\varepsilon)^{-1}}.$$

Hence,

$$\ell' \leq \frac{l \ln m}{\ln(1-\varepsilon)^{-1}}.$$

If  $\ell' \leq \varepsilon' \ell$ , both

$$\frac{1}{\ell + \ell'} \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} f$$

and

$$\frac{1}{\ell + \ell'} (\Gamma \cup \Gamma')$$

are returned as the output. Otherwise, the agent updates the weight  $y(a)$  of each link  $a$  by the multiplicative factor

$$1 - \varepsilon \delta \left( |\{a\} \cap I| - \lambda \sum_{j \in [k]} d_j |\{a\} \cap P_j| \right)$$

and the algorithm moves on to the next round.

We proceed to derive the upper bounds on both the number of iterations and the approximation ratio of the algorithm **CMF-LS**( $\varepsilon$ ). Let  $h_j$  be the hop number of a minimum-hop path of the  $j$ -th request for each  $j \in [k]$ , and  $\alpha$  be the size of a maximum-sized independent set.

*Theorem 5.1:* The algorithm **CMF-LS**( $\varepsilon$ ) runs in

$$O \left( \varepsilon^{-2} \max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\} l \ln m \right).$$

iterations and has an approximation bound  $(1 + 2\varepsilon)\mu$ .

The following essential properties of  $\lambda$  and  $\delta$  are the cornerstone to the above theorem

*Lemma 5.2:* The scaling factor  $\lambda$  and the length  $\delta$  computed in each iteration satisfy that

$$\begin{aligned} \frac{opt}{\mu} \leq \lambda \leq \frac{\alpha}{\sum_{j \in [k]} d_j h_j}, \\ \delta \geq \frac{1}{\max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\}}. \end{aligned}$$

*Proof:* Let  $\lambda_0$  and  $y_0$  be the initial values of  $\lambda$  and  $y$ . For each round  $t \geq 1$ , let  $\lambda_t$ ,  $\delta_t$  and  $y_t$  be the value of  $\lambda$ ,  $\delta$  and  $y$

respectively computed in the round  $t$ ; let  $I_t$  be the independent set  $I$ , computed in the iteration  $t$ . Then,

$$\lambda_1 = \frac{y_0(I_1)}{\sum_{j \in [k]} d_j dist_j(y_0)} = \frac{|I_1|}{\sum_{j \in [k]} d_j h_j} \leq \frac{\alpha}{\sum_{j \in [k]} d_j h_j}.$$

Since  $\lambda_t$  is non-increasing with  $t$ , for each iteration  $t$  we have

$$\lambda_t \leq \frac{\alpha}{\sum_{j \in [k]} d_j h_j}.$$

Next, we prove by induction on  $t$  that  $\lambda_t \geq \frac{opt}{\mu}$ . Consider the (first) iteration  $t = 1$ . By the weak duality given in Theorem 3.1,

$$\lambda_1 = \frac{y_0(I_1)}{\sum_{j \in [k]} d_j dist_j(y_0)} \geq \frac{1}{\mu} \frac{\max_{I \in \mathcal{I}} y_0(I)}{\sum_{j \in [k]} d_j dist_j(y_0)} \geq \frac{opt}{\mu}.$$

Now consider any iteration  $t > 1$  and assume that  $\lambda_{t-1} \geq \frac{opt}{\mu}$  and show that  $\lambda_t \geq \frac{opt}{\mu}$ . If  $\lambda_t = \lambda_{t-1}$ , then it is trivial that  $\lambda_t \geq \frac{opt}{\mu}$ . If  $\lambda_t < \lambda_{t-1}$ , then by the weak duality given in Theorem 3.1,

$$\begin{aligned} \lambda_t &= \frac{y_{t-1}(I_t)}{\sum_{j \in [k]} d_j dist_j(y_{t-1})} \geq \frac{1}{\mu} \frac{\max_{I \in \mathcal{I}} y_{t-1}(I)}{\sum_{j \in [k]} d_j dist_j(y_{t-1})} \\ &\geq \frac{opt}{\mu}. \end{aligned}$$

So, in either case,  $\lambda_t \geq \frac{opt}{\mu}$ .

Finally, we show that for each iteration  $t \geq 1$ ,

$$\delta_t \geq \frac{1}{\max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\}}.$$

For each round  $t \geq 1$ , let  $P_j^{(t)}$  be the shortest path of the  $j$ -th request computed in the round  $t$  for each  $j \in [k]$ . Then, for each link  $a$ ,

$$\begin{aligned} &\left| |\{a\} \cap I_t| - \lambda_t \sum_{j=1}^k d_j |\{a\} \cap P_j^{(t)}| \right| \\ &\leq \max \left\{ |\{a\} \cap I_t|, \lambda_t \sum_{j=1}^k d_j |\{a\} \cap P_j| \right\} \\ &\leq \max \left\{ 1, \lambda_t \sum_{j=1}^k d_j \right\} \\ &\leq \max \left\{ 1, \frac{\alpha}{\sum_{j \in [k]} d_j h_j} \sum_{j=1}^k d_j \right\} \\ &\leq \max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\} \end{aligned}$$

Thus,

$$\begin{aligned} \delta_t &= \frac{1}{\max_{a \in A} \left| |\{a\} \cap I_t| - \lambda_t \sum_{j=1}^k d_j |\{a\} \cap P_j^{(t)}| \right|} \\ &\geq \frac{1}{\max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\}}. \end{aligned}$$

This completes the proof of the lemma.  $\blacksquare$

Next, we prove by contradiction that the number of iterations is at most

$$t = \left\lceil \frac{\max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\} l \ln m}{(1 + 2\varepsilon) \ln(1 + \varepsilon) + \ln(1 - \varepsilon)} \right\rceil.$$

Assume to the contrary the number of iterations is more than  $t$ . Consider the iteration  $t$ . By Lemma 5.2, the primary schedule  $\Gamma$  in the iteration  $t$  has length

$$\begin{aligned} \ell &\geq \frac{t}{\max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\}} \\ &\geq \frac{l \ln m}{(1 + 2\varepsilon) \ln(1 + \varepsilon) + \ln(1 - \varepsilon)} \\ &= \frac{l \ln m}{\varepsilon' \ln(1 - \varepsilon)^{-1}}. \end{aligned}$$

On the other hand, the complementary schedule  $\Gamma'$  in the iteration  $t$  has length

$$\ell' \leq \frac{l \ln m}{\ln(1 - \varepsilon)^{-1}}.$$

Thus,  $\ell' \leq \varepsilon' \ell$ . This implies that the algorithm would terminate at the end the iteration  $t$ , which is a contradiction. So, the number of iterations is at most  $t$ , which is

$$O \left( \varepsilon^{-2} \max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\} l \ln m \right).$$

Finally, the approximation bound  $(1 + 2\varepsilon) \mu$  follows from the property  $\lambda \geq \frac{opt}{\mu}$  in each iteration which is asserted in Lemma 5.2 and the argument given in the overview of the algorithm at the beginning of this section. This completes the proof of Theorem 5.1.

## VI. MAXIMUM MULTIFLOW

Let  $\mathcal{A}$  be a  $\mu$ -approximation algorithm for **MWIS**, and  $\varepsilon \in (0, 1/2]$  be an accuracy-efficiency trade-off parameter. This section presents a  $(1 + 2\varepsilon) \mu$ -approximation algorithm **MF-LS**( $\varepsilon$ ) for **MMF**. We first give an overview on the design of the algorithm **MF-LS**( $\varepsilon$ ). Let  $opt$  be the total value of a maximum concurrent multifold. The algorithm is iterative. Each iteration first computes a multifold  $f$  of total value  $\ell$ , a scaling factor  $\lambda \geq \frac{opt}{\mu}$ , and a “primary” link schedule  $\Gamma$  of length  $\ell$  and its corresponding link transmission-time function  $g$ . With respect to the triple  $(f, \lambda, g)$ , the *deficit* of a link  $a$  is defined to be

$$d'(a) = \max \left\{ 0, \frac{\ln(1 + \varepsilon)}{\ln(1 - \varepsilon)^{-1}} \lambda \sum_{j \in [k]} f_j(a) - g(a) \right\}.$$

For such deficit demand  $d'$ , a “complementary” link schedule  $\Gamma'$  of length  $\ell'$  is then computed by the same method as in the algorithm **MF-LS**( $\varepsilon$ ). If

$$\frac{\ell'}{\ell} \leq \varepsilon' := \frac{(1 + 2\varepsilon) \ln(1 + \varepsilon) + \ln(1 - \varepsilon)}{\ln(1 - \varepsilon)^{-1}},$$

then the scaled multifold

$$\frac{1}{\ell + \ell'} \frac{\ln(1 + \varepsilon)}{\ln(1 - \varepsilon)^{-1}} \lambda f$$

and the scaled schedule

$$\frac{1}{\ell + \ell'} (\Gamma \cup \Gamma')$$

are returned as the output. The total value of the returned multifold is

$$\begin{aligned} \frac{1}{\ell + \ell'} \frac{\ln(1 + \varepsilon)}{\ln(1 - \varepsilon)^{-1}} \lambda \ell &= \frac{1}{1 + \ell'/\ell} \frac{\ln(1 + \varepsilon)}{\ln(1 - \varepsilon)^{-1}} \lambda \\ &\geq \frac{\lambda}{1 + 2\varepsilon} \geq \frac{opt}{(1 + 2\varepsilon) \mu}. \end{aligned}$$

The algorithm **MF-LS**( $\varepsilon$ ) for **MMF** is outlined in Table II. It is very similar to the algorithm **CMF-LS**( $\varepsilon$ ) for **MCMF**, and consequently we only highlight the difference in this section. The computation of the primary schedule  $\Gamma$  (together with its length  $\ell$  and its link transmission time function  $g$ ), the multifold  $f$ , and the scaling factor  $\lambda$  in each round also follows the general framework of the coupled game introduced in Section IV. Each link is regarded as an expert, and each iteration corresponds to a game round. The agent plays exactly with the strategy on maintaining the expert/link weight function  $y$  described in Section IV. The profit (respectively, loss) generation strategy of the adversary is coupled with the primary schedule (respectively, multifold) augmentation with the following *invariant* property maintained throughout the game: At the end of each round, the cumulative profit of each link  $a$  is exactly  $g(a)$ , and the cumulative loss of each link is at least  $\lambda \sum_{j \in [k]} f_j(a)$ . Initially, both the multifold  $f$  and the primary schedule  $\Gamma$  are empty, and the scaling factor  $\lambda$  is infinity. In each round, the adversary generates the profits/losses as follows. Let  $I$  be the independent set w.r.t.  $y$  output by the algorithm  $\mathcal{A}$ , and  $P_j$  be the shortest one among the  $k$  shortest paths with respect to  $y$ . Then, only  $f_j$  is augmented along  $P_j$  in this iteration. Accordingly,  $\lambda$  is computed by

$$\lambda \leftarrow \min \left\{ \lambda, \frac{y(I)}{y(P_j)} \right\};$$

and  $\delta$  is chosen to be

$$\frac{1}{\max_{a \in A} \left| |\{a\} \cap I| - \lambda |\{a\} \cap P_j| \right|}.$$

So,  $\Gamma$  is augmented by the pair  $(I, \delta)$ , and both  $\ell$  and  $g$  are updated accordingly;  $f_j$  is augmented by a flow of value  $\delta$  along  $P_j$ . Only each link  $a \in I$  earns a profit  $\delta$ , and only each link  $a \in P_j$  incurs a loss  $\delta \lambda$ . It is easy to verify that such profit/loss assignment in this round maintains the invariant property and satisfies both the **Normalization Rule** and the **Generalized Zero-Sum Rule**. Subsequently, the algorithm computes the complementary schedule  $\Gamma'$  and its length  $\ell'$ , and takes a test of termination, i.e., whether  $\ell' \leq \varepsilon' \ell$ . If the test is passed, then the algorithm returns the scaled multifold

$$\frac{1}{\ell + \ell'} \frac{\ln(1 + \varepsilon)}{\ln(1 - \varepsilon)^{-1}} \lambda f$$

and the scaled schedule

$$\frac{1}{\ell + \ell'} (\Gamma \cup \Gamma');$$

otherwise, the agent updates the weight  $y(a)$  of each link  $a$  by the multiplicative factor

$$1 - \varepsilon \delta (|\{a\} \cap I| - \lambda |\{a\} \cap P_j|),$$

and the algorithm moves on to the next round.

Algorithm <b>MF-LS</b> ( $\varepsilon$ ):
<b>// initialization</b> $\forall j \in [k], f_j \leftarrow 0; \Gamma \leftarrow \emptyset, \ell \leftarrow 0, g \leftarrow 0; \lambda \leftarrow \infty;$ $y \leftarrow \mathbf{1}, \varepsilon' \leftarrow \frac{(1+2\varepsilon)\ln(1+\varepsilon)+\ln(1-\varepsilon)}{\ln(1-\varepsilon)^{-1}};$ $\mathcal{J} \leftarrow$ a link partition output by <b>Greedy Scheduling</b> ; repeat forever
<b>// primary schedule and flow augmentation</b> $I \leftarrow$ a $y$ -weighted IS in $\mathcal{I}$ output by $\mathcal{A}$ ; $\forall j \in [k], P_j \leftarrow$ a shortest $j$ -path w.r.t. $y$ ; $j \leftarrow \arg \min_{j \in [k]} y(P_j);$ $\lambda \leftarrow \min \left\{ \lambda, \frac{y(I)}{y(P_j)} \right\};$ $\delta \leftarrow \frac{\max_{a \in A} \ \{a\} \cap I - \lambda \{a\} \cap P_j\ }{\max_{a \in A} \ \{a\} \cap I - \lambda \{a\} \cap P_j\ };$ $\Gamma \leftarrow \Gamma \cup \{(I, \delta)\}, \ell \leftarrow \ell + \delta;$ for each $a \in I$ do $g(a) \leftarrow g(a) + \delta;$ for each $a \in P_j$ do $f_j(a) \leftarrow f_j(a) + \delta;$
<b>// complementary schedule</b> $\Gamma' \leftarrow \emptyset, \ell' \leftarrow 0;$ for each $J \in \mathcal{J}$ do $\delta' \leftarrow \max_{a \in J} \max \left\{ 0, \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} \lambda \sum_{j \in [k]} f_j(a) - g(a) \right\};$ if $\delta' > 0$ then $\Gamma' \leftarrow \Gamma' \cup \{(J, \delta')\}, \ell' \leftarrow \ell' + \delta';$
<b>// test of termination</b> if $\ell' \leq \varepsilon' \ell$ then return $\frac{1}{\ell + \ell'} \frac{\ln(1+\varepsilon)}{\ln(1-\varepsilon)^{-1}} f$ and $\frac{1}{\ell + \ell'} (\Gamma \cup \Gamma')$ .
<b>// multiplicative weights update</b> for each $a \in A$ do $y(a) \leftarrow y(a) (1 - \varepsilon \delta ( \{a\} \cap I  - \lambda  \{a\} \cap P_j ));$

TABLE II. OUTLINE OF THE ALGORITHM **MF-LS**( $\varepsilon$ ).

For the performance analysis, we can show that in each iteration

$$\begin{aligned} \frac{opt}{\mu} &\leq \lambda \leq \frac{\alpha}{\min_{j \in [k]} h_j}, \\ \delta &\geq \min \left\{ 1, \frac{\min_{j \in [k]} h_j}{\alpha} \right\}, \\ \ell' &\leq l \frac{\ln m}{\ln(1-\varepsilon)^{-1}}. \end{aligned}$$

Using these properties, we can prove the following performance of the algorithm **MF-LS**( $\varepsilon$ ). The proof detail is omitted due to the space limitation.

*Theorem 6.1:* The algorithm **MF-LS**( $\varepsilon$ ) runs in

$$O \left( \varepsilon^{-2} \max \left\{ 1, \frac{\alpha}{\min_{j \in [k]} h_j} \right\} l \ln m \right).$$

iterations and has an approximation bound  $(1 + 2\varepsilon)\mu$ .

## VII. APPLICATIONS

In this section, we apply the two general algorithms **MCF-LS**( $\varepsilon$ ) and **MF-LS**( $\varepsilon$ ) to derive effective and efficient approximation algorithms for **MMF** and **MCMF** in wireless networks under the physical interference model, and in wireless MIMO networks with receiver-side interference suppression under the protocol interference model.

Consider an instance of wireless MIMO network under the physical interference model. In the setting of no power control, an assignment of transmission power to links is pre-specified, and a set  $I$  of links is independent if and only if all links in  $I$  can communicate successfully at the same time under the physical interference model. A power assignment is said to be *monotone* if the transmission power of a link is non-decreasing with the link length, to be *sub-linear* if the received power by a link is non-increasing with the link length, and to be a *linear* if all links have the same received power. In the setting of power control, a set  $I$  of links is independent if and only if there exists a transmission power assignment to  $I$  at which all links in can communicate successfully at the same time under the physical interference model. With linear power assignment, constant-approximation algorithms for **MWIS** have been developed in [27]; with any other fixed monotone and sublinear power assignment or with power control, logarithmic approximation algorithms for **MWIS** have been developed in [17], [17], [23], [25], [27]. By utilizing these approximation algorithms for **MWIS**, the two algorithms **MCF-LS**( $\varepsilon$ ) and **MF-LS**( $\varepsilon$ ) produce constant approximate solutions for **MCMF** and **MMF** respectively with linear power assignment, and logarithmic approximate solutions for **MCMF** and **MMF** respectively with any other fixed monotone and sublinear power assignment or with power control.

Consider an instance of wireless MIMO network on a set  $V$  of networking nodes. Each node  $v$  has  $\tau(v)$  antennas and operates in the half-duplex mode, i.e. it cannot transmit and receive at the same time. Along each node-level directed communication link  $(u, v)$ ,  $\min\{\tau(u), \tau(v)\}$  streams can be multiplexed. Let  $A$  denote the set of streams of all directed node-level communication links. The directed multigraph  $(V, A)$  is referred to as the stream-level communication topology. Under a protocol interference model, each node-level communication link is associated with an interference range and all its streams inherit the same interference range from it. When a set  $I$  of streams in  $A$  transmit at the same time, the transmission by a stream  $a \in I$  from a sender  $u$  to a receiver  $v$  succeeds with the receiver-side interference suppression if all the following three constraints are satisfied:

- 1) **Half-Duplex Constraint:**  $u$  is not the receiver of any other stream in  $I$ , and  $v$  is not the sender of any other stream in  $I$ .
- 2) **Sender Constraint:**  $u$  is the sender is at most  $\tau(u)$  streams in  $I$ .
- 3) **Receiver Constraint:**  $v$  lies in the interference range of at most  $\tau(v)$  streams in  $I$ .

A set  $I$  of streams is said to be *independent* if all streams in  $I$  succeed when they transmit at the same time. Let  $\mathcal{I}$  denote the collection of all independent subsets of  $A$ . Constant-approximation algorithms for the problem **MWIS** have been developed in [28] in the following three settings:

- Constant bounded number of antennas at all nodes.
- Uniform interference radii but arbitrary number of antennas.
- Uniform number of antennas but arbitrary interference radii.

By utilizing these algorithms, the two algorithms **MCF-LS**( $\epsilon$ ) and **MF-LS**( $\epsilon$ ) produce constant approximation solutions for **MCMF** and **MMF** respectively in the above three settings as well.

### VIII. CONCLUSION

This paper presents a completely new paradigm for multiflow problems in general wireless networks under an arbitrary interference model which is radically different from the prevailing LP-based paradigm. The multiflow problems are reduced to sequence of a sequence of computations of shortest paths and independent sets. These algorithms offer nice trade-off between accuracy in terms the approximation bound and efficiency in terms of the running time, and are much simpler and faster. The link schedule generated by these algorithms is split into a primary link schedule and a complementary link schedule. The computation of the primary link schedule follows a simple yet powerful framework of the adaptive coupled game introduced in Section IV. The computation of the complementary link schedule is greedy and is introduced for the purpose of the speed-up of the algorithms. The broader applicability of these algorithms are demonstrated by their applications to **MMF** and **MCMF** in wireless networks under the physical interference model and wireless MIMO networks with receiver-side interference suppression under the protocol interference model. We also believe that the new paradigm developed in this paper has wide applications in application domains beyond wireless networks.

**ACKNOWLEDGEMENTS:** This work was supported in part by the National Science Foundation of USA under grants CNS-1454770, CNS-1219109, and CNS-1217309, by the National Natural Science Foundation of P. R. China under grant 61272524, and by Dalian University of Technology under the Haitian Scholar grant.

### REFERENCES

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, 1993.
- [2] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: A meta-algorithm and application. *Theory of Computing* 8(1):121–164, 2012.
- [3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science*, pages 322–331, 1995.
- [4] D. Bienstock. *Potential function methods for approximately solving linear programs: theory and practice*. Kluwer Academic Publisher, 2002.
- [5] N. Cesa-Bianchi, Y. Freund, D. Helmbold, D. Haussler, R. Schapire, and M. Warmuth. How to use expert advice. *Journal of the Association for Computing Machinery*, 44(3):427–485, 1997.
- [6] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Approximation algorithms for computing capacity of wireless networks with SINR constraints, *IEEE INFOCOM* 2008, pages 1166–1174.
- [7] Y. Freund and R. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [8] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.
- [9] A. Gkelias, F. Boccardi, C.H. Liu, and K.K. Leung. MIMO routing with QoS provisioning, *Proc. ISWPC* 2008, pages 46–50.
- [10] B. Hamdaoui, K. Shin, D. Jiang. Characterization and Analysis of Multi-Hop Wireless MIMO Network Throughput, *Proc. ACM MobiHoc* 2007, pp. 120–129.
- [11] B. Han, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Distributed Strategies for Channel Allocation and Scheduling in Software-Defined Radio Networks. *Proc. IEEE INFOCOM* 2009, pp. 1521-1529.
- [12] R. Khandekar, Lagrangian relaxation based algorithms for convex programming problems, *PhD thesis*, Indian Institute of Technology, Delhi, 2004.
- [13] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. *Proc. of ACM MobiCom* 2003.
- [14] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. *Proc. of ACM MobiCom*, 2005.
- [15] V.S.A. Kumar, M.V. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic aspects of capacity in wireless networks. *SIGMETRICS Perform. Eval. Rev.* 33(1):133–144, 2005.
- [16] J. Liu, Y.T. Hou, Y. Shi, and H. Sherali. Cross-Layer Optimization for MIMO-Based Wireless Ad Hoc Networks: Routing, Power Allocation, and Bandwidth Allocation, *IEEE Journal on Selected Areas in Communications* (6):913–926, August 2008.
- [17] C. Ma, F. Al-dhelaan, and P.-J. Wan. Maximum Independent Set of Links with a Monotone and Sublinear Power Assignment. *WASA* 2013: 64-75.
- [18] C. Ma, F. Al-dhelaan, and P.-J. Wan. Maximum Independent Set of Links with Power Control. *WASA* 2013: 474-485.
- [19] A. Schrijver. *Combinatorial Optimization*. Number 24 in Algorithms and Combinatorics. Springer, 2003.
- [20] K. Sundaresan and R. Sivakumar, Routing in ad-hoc networks with MIMO links, *Proc. IEEE ICNP* 2005, pages 85–98.
- [21] V. Vovk, A game of prediction with expert advice. In *Proceedings of the 8th Annual Conference on Computational Learning Theory*, pages 51–60, 1995.
- [22] P.-J. Wan, Multiflows in Multihop Wireless Networks, *ACM MOBIHOC* 2009, pp. 85-94.
- [23] P.-J. Wan, D. Chen, G. Dai, Z. Wang, and F. Yao, Maximizing Capacity with Power Control under Physical Interference Model in Duplex Mode, *IEEE INFOCOM* 2012: 415-423.
- [24] P.-J. Wan, Y. Cheng, Z. Wang, and F. Yao, Multiflows in Multi-Channel Multi-Radio Multihop Wireless Networks, *IEEE INFOCOM* 2011: 846-854.
- [25] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X.-H. Xu, S.-J. Tang, Wireless Link Scheduling under Physical Interference Model, *IEEE INFOCOM* 2011.
- [26] P.-J. Wan, X. Jia, G. Dai, H. Du, Z.G. Wan, and O. Frieder, Scalable Algorithms for Wireless Link Scheduling in Multi-Channel Multi-Radio Wireless Networks, *IEEE INFOCOM* 2013.
- [27] P.-J. Wan, L. Wang, C. Ma, Z. Wang, B. Xu, and M. Li, Maximizing Wireless Network Capacity with Linear Power: Breaking The Logarithmic Barrier, *IEEE INFOCOM* 2013.
- [28] P.-J. Wan, Z. Wang, L. Wang, Z.G. Wan, and S. Ji, From Least Interference-Cost Paths to Maximum (Concurrent) Multiflow in MC-MR Wireless Networks, *IEEE INFOCOM* 2014.
- [29] P.-J. Wan, B. Xu, O. Frieder, S. Ji, B. Wang, and X. Xu, Capacity Maximization in Wireless MIMO Networks with Receiver-Side Interference Suppression, *ACM MOBIHOC* 2014.
- [30] P.-J. Wan, X.-H. Xu, and O. Frieder, Shortest Link Scheduling with Power Control under Physical Interference Model, *The 6th International Conference on Mobile Ad-hoc and Sensor Networks (MSN'10)*, 2010.