

# From Least Interference-Cost Paths to Maximum (Concurrent) Multiflow in MC-MR Wireless Networks

Peng-Jun Wan\*, Zhu Wang\*, Lei Wang†, Zhiguo Wan‡, and Sai Ji§

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

† School of Software, Dalian University of Technology, lei.wang@ieee.org

‡ School of software, Tsinghua University, wanzhiguo@tsinghua.edu.cn

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

**Abstract**—Maximum multiflow and maximum concurrent multiflow in multi-channel multi-radio (MC-MR) wireless networks have been well-studied in the literature. They are NP-hard even in single-channel single-radio (SC-SR) wireless networks when all nodes have uniform (and fixed) interference radii and the positions of all nodes are available. While they admit a polynomial-time approximation scheme (PTAS) when the number of channels is bounded by a constant, such PTAS is quite infeasible practically. Other than the PTAS, all other known approximation algorithms, in both SC-SR wireless networks and MC-MR wireless networks, resorted to solve a polynomial-sized linear program (LP) exactly. The scalability of their running time is fundamentally limited by the general-purpose LP solvers. In this paper, we first introduce the concept of interference costs and prices of a path and explore their relations with the maximum (concurrent) multiflow. Then we develop purely combinatorial approximation algorithms which compute a sequence of least interference-cost routing paths along which the flows are routed. These algorithms are faster and simpler, and achieve nearly the same approximation bounds known in the literature.

## I. INTRODUCTION

Consider an instance of multi-channel multi-radio (MC-MR) multihop wireless network with a set  $V$  of networking nodes and a set  $A$  of node-level communication links. Each node  $v$  has  $\tau(v)$  radios, and there are  $\lambda$  non-overlapping channels. A set of links in  $A$  is said to be *independent* if its links are pairwise node-disjoint and can transmit successfully at the same time over the same channel under a pre-specified interference model. Let  $\mathcal{I}$  denote the collection of all independent subsets of  $A$ . For each node-level link  $a = (u, v)$  in  $A$ , we make  $\lambda \cdot \tau(u) \cdot \tau(v)$  replications  $(u, v, i, j, l)$  for  $1 \leq i \leq \tau(u)$ ,  $1 \leq j \leq \tau(v)$ , and  $1 \leq l \leq \lambda$ . A replication  $(u, v, i, j, l)$  always utilizes the  $i$ -th radio at  $u$  and the  $j$ -th radio at  $v$  over the  $l$ -th channel. These replications is referred to as replicated links of  $a$ . We use  $\{a\}^{\tau, \lambda}$  to denote the set of replicated links of  $a$ ; and in general, for each  $B \subseteq A$ , we use  $B^{\tau, \lambda}$  to denote the set of all replicated links of the links in  $B$ . Clearly, a set of replicated links can transmit successfully at the same time if and only if (1) they are pairwise radio-disjoint, and (2) for each channel  $l$ , all those replication links transmitting over channel  $l$  are replicated from an independent set of  $A$ . These set of replicated links are also referred to *independent*

sets of replicated links. Let  $\mathcal{I}^{\tau, \lambda}$  denote the collection of the independent sets of replicated links. Any set

$$\mathcal{S} = \{(I_j, l_j) \in \mathcal{I}^{\tau, \lambda} \times \mathbb{R}_+ : 1 \leq j \leq q\}$$

is referred to as a *link schedule*; the value  $\sum_{j=1}^q l_j$  is referred to as *length* (or *latency*) of  $\mathcal{S}$ . The link demand served by  $\mathcal{S}$  is the function  $x$  on  $A$  defined by

$$x(a) = \sum_{j=1}^q l_j |I_j \cap \{a\}^{\tau, \lambda}|$$

for each  $a \in A$ ;  $\mathcal{S}$  is also said to be a link schedule of this  $x$ .

Suppose that we are given  $k$  unicast requests specified by source-destination pairs. Let  $\mathcal{P}_j$  be the set of paths in  $(V, A)$  of the  $j$ -th request for all  $1 \leq j \leq k$ , and let  $\mathcal{P}$  be the union of  $\mathcal{P}_1, \dots, \mathcal{P}_k$ . Any set

$$\Gamma = \{(P_i, \delta_i) \in \mathcal{P} \times \mathbb{R}_+ : 1 \leq i \leq l\}$$

is referred to as a *path flow* of these requests; its *link flow* is a function  $x$  on  $A$  defined by

$$x(a) = \sum_{i=1}^l \delta_i |P_i \cap \{a\}|.$$

for each  $a \in A$ ; and the value  $\sum_{i=1}^l \delta_i$  is referred to as *flow value* of  $\Gamma$ . The problem **Maximum Multiflow (MMF)** seeks a path flow  $\Gamma$  of maximum flow value and a link schedule  $\mathcal{S}$  of length *one* such that  $\mathcal{S}$  serves the link flow of  $\Gamma$ .

Suppose that in addition that for each  $1 \leq j \leq k$  the  $j$ -th request has a positive demand  $d_j$ . Any set

$$\Pi = \{(P_{i1}, \dots, P_{ik}, \delta_i) \in \prod_{j=1}^k \mathcal{P}_j \times \mathbb{R}_+ : 1 \leq i \leq l\}$$

is referred to as a *concurrent  $k$ -path flow* of these requests; its *cumulative link flow* is a function  $x$  on  $A$  defined by

$$x(a) = \sum_{i=1}^l \delta_i \sum_{j=1}^k d_j |P_{ij} \cap \{a\}|.$$

for each  $a \in A$ ; and the value  $\sum_{i=1}^l \delta_i$  is referred to as *flow concurrency* of  $\Pi$ . The problem **Maximum Concurrent Multiflow (MCMF)** seeks a concurrent  $k$ -path flow  $\Pi$  of

maximum flow concurrency and a link schedule  $\mathcal{S}$  of length *one* such that  $\mathcal{S}$  serves the cumulative link flow of  $\Pi$ .

As fundamental problems in multihop wireless networking, both **MMF** and **MCMF** received much research interest in the past decade. Most of the existing studies (e.g., [4], [7], [8], [9], [10], [11], [12]) assumed some variants of the *protocol* (as opposed to *physical*) interference model. In general, a protocol interference model specifies a pairwise conflict relations among all links in  $A$ , and a subset of  $A$  is independent if its links are pairwise conflict-free. It is classified into two communication modes:

- **Unidirectional mode:** For each link  $a = (u, v) \in A$ , the communication between  $u$  and  $v$  occurs in the direction from  $u$  to  $v$ , and the endpoint  $u$  (respectively,  $v$ ) is referred to as the sender (respectively, receiver) of  $a$ . The sender  $u$  of the link  $a$  has an interference range, and the interference range of  $a$  is the interference range of its sender. Two links in  $A$  conflict with each other if and only if the receiver of at least one link lies in the interference range of the other link.
- **Bidirectional mode:** For each link  $a = (u, v) \in A$ , the communication between  $u$  and  $v$  occurs in both directions, and both  $u$  and  $v$  have an interference range. The interference range of  $a$  is the union of the interference ranges of its two endpoints. Two links in  $A$  conflict with each other if and only if at least one link has an endpoint lying in the interference range of the other link.

In the plane geometric variant, the interference range of an endpoint  $u$  of a link  $a$  is assumed to be a disk centered at  $u$  of radius  $r_a(u)$ , which is also known as the interference radius. Under the plane geometric variant of the protocol interference model in either unidirectional mode or bidirectional mode, the computational hardness of **MMF** and **MCMF** was well characterized in both single-channel single-radio (SC-SR) setting [10] and MC-MR setting [11]. On one hand, they are NP-hard in even in SC-SR networks in which all nodes have uniform (and fixed) communication radii and uniform (and fixed) interference radii and the positions of all nodes are available. On the other hand, they admit a polynomial-time approximation scheme (PTAS) when the number of channels is bounded by a constant. In other words, for any fixed  $\varepsilon > 0$ , it has polynomial-time (depending on  $\varepsilon$ )  $(1 + \varepsilon)$ -approximation algorithm. Such PTAS is of theoretical interest only and is quite infeasible practically as it involves very time-consuming exhaustive enumerations and ellipsoid method. Other than the PTAS, all other known approximation algorithms for **MMF** and **MCMF**, in both SC-SR wireless networks (e.g., [7], [8], [9], [10]) and in MC-MR wireless networks (e.g., [4], [11], [12]), resorted to solve a polynomial-sized linear program (LP) exactly. However, such LP-based methods can require an inordinate amount of running time and memory even for a moderate sized input [1].

In this paper, we take a purely combinatorial approach for approximating **MMF** and **MCMF** in MC-MR wireless networks (and in SC-SR wireless networks as well as a

special case) under the protocol interference model. A subtle definition of the interference cost of a path in the node-level communication topology  $(V, A)$  is introduced. Our algorithms iteratively compute a sequence of cheapest paths in terms of interference cost along which the flows are routed. They are much faster and conceptually simpler (than the general-purpose LP solvers). They also present a trade-off between the targeted approximation bound and the running time: for any fixed  $\varepsilon > 0$ , they may achieve approximation bounds at most  $1 + \varepsilon$  times those achieved in [11] at the growth of the running time in at most the square order of  $1/\varepsilon$ . Both the designs and analyses of our algorithms are applicable to general protocol interference model. When applied to the same variants of the protocol interference model, they achieve nearly the same approximation bounds as those achieved in [11].

The remainder of this paper is organized as follows. In Section II, we introduce the relevant known facts and results to be exploited by the subsequent sections of this paper. In Section III, we define the interference costs and prices of a path and exploring their relations with the maximum (concurrent) multiflow. In Section IV and Section V, we present the designs and analyses of our approximation algorithms for **MMF** and **MCMF** respectively. In Section VI, we apply the approximations algorithms developed in the previous two sections to the specific variants of the protocol interference model. We conclude this paper in Section VII.

The following standard terms and notations are adopted throughout this paper. Let  $n = |V|$  and  $m = |A|$ .  $G$  denotes the link-conflict graph of  $A$  under the given protocol interference model. In other words,  $A$  is the vertex set of  $G$  and two links in  $A$  are adjacent in  $G$  if and only if they conflict with each other. An *orientation* of  $G$  is a digraph  $D$  obtained from  $G$  by imposing an orientation on each edge of  $G$ . For any  $a \in A$ ,  $N[a]$  denotes the set of links in  $A$  conflicting with  $a$  plus  $a$  itself. Suppose that  $D$  is an orientation. For each  $a \in A$ ,  $N_D^{in}[a]$  (respectively,  $N_D^{out}[a]$ ) denotes the set of in-neighbors (respectively, out-neighbors) of  $a$  in  $D$  plus  $a$  itself. For any real-valued function  $f$  on  $A$  and any  $B \subseteq A$ ,  $f(B)$  represents  $\sum_{b \in B} f(b)$ ; For any real-valued function  $f$  on  $A \times A$ , and any  $a \in A$ , and any  $B \subseteq A$ ,  $f(a, B)$  represents  $\sum_{b \in B} f(a, b)$ . Let  $\prec$  be an ordering of  $A$ . For any pair of links  $a, b \in A$ , both  $a \prec b$  and  $b \succ a$  represent that  $a$  appears before  $b$  in the ordering  $\prec$ .

## II. PRELIMINARIES

For MC-MR wireless networks, Wan et al. [12] introduced the *interference factor*  $\rho(a, b)$  of two conflicting links  $a$  and  $b$ , which captures the essential advantage of multiple radios and multiple channels. If  $a$  and  $b$  share a common endpoint  $u$ , then each replicated link of  $b$  conflicts with exactly

$$\rho(a, b) = 1 - \left(1 - \frac{1}{\tau(u)}\right) \left(1 - \frac{1}{\lambda}\right)$$

portion of replicated links of  $a$ ; otherwise, each replicated link of  $b$  conflicts with exactly

$$\rho(a, b) = \frac{1}{\lambda}$$

portion of replicated links of  $a$ . Note that  $\rho(a, b) = \rho(b, a)$  in either case. In addition, each replicated link of a link  $a = (u, v)$  conflicts with exactly

$$\rho(a, b) = 1 - \left(1 - \frac{1}{\tau(u)}\right) \left(1 - \frac{1}{\tau(v)}\right) \left(1 - \frac{1}{\lambda}\right)$$

portion of replicated links of  $a$  (include itself).

Consider any non-negative function  $x$  on  $A$ . The  $x$ -weighted inductivity of  $A$  is the value

$$\Delta^*(x) = \max_{\emptyset \neq B \subseteq A} \min_{a \in B} \sum_{b \in N[a] \cap B} \rho(a, b) x(b)$$

A link schedule of  $x$  of length at most  $\Delta^*(x)$  can be produced by a simple greedy algorithm **GLS** developed by Wan et al. [12] in  $O(m^2\lambda + mm' \max_{v \in V} \tau(v))$  time, where  $m'$  is the total number of conflicting pairs of links in  $A$ . Note that this running time grows linearly with the number of channels and the maximum number of radios of all nodes. Consider an orientation  $D$  of  $G$ . The  $x$ -weighted inward interference of  $a$  with respect to  $D$  is defined to be

$$\Delta_D^{in}(a; x) = \sum_{b \in N_D^{in}[a]} \rho(a, b) x(b).$$

and the  $x$ -weighted inward interference of  $D$  is defined to be

$$\Delta_D^{in}(x) = \max_{a \in A} \Delta_D^{in}(a; x).$$

The following relation between  $\Delta^*(x)$  and  $\Delta_D^{in}(x)$  was proved in [12].

*Lemma 2.1:*  $\Delta^*(x) \leq 2\Delta_D^{in}(x)$ . If  $D$  is acyclic, then  $\Delta^*(x) \leq \Delta_D^{in}(x)$ .

The *inward local independence number* (ILIN) of  $D$  is defined to be

$$\max_{a \in A} \max_{I \in \mathcal{I}} |I \cap N_D^{in}[a]|.$$

The following upper bound on  $\Delta_D^{in}(x)$  was proved in [10] in the SC-SR setting and in [12] in the MC-MR setting.

*Lemma 2.2:* Suppose  $x$  has a link schedule of length at most one. Then, for any orientation  $D$  with ILIN  $\mu$ ,

$$\Delta_D^{in}(x) \leq \hat{\mu}$$

where  $\hat{\mu} = \mu$  in SC-SR setting and  $\hat{\mu} = \mu + 2$  in MC-MR setting.

Consider a link schedule

$$\mathcal{S} = \{(I_j, l_j) \in \mathcal{I}^{\tau, \lambda} \times \mathbb{R}_+ : 1 \leq j \leq q\}.$$

We use  $\|\mathcal{S}\|$  to denote its length  $\sum_{j=1}^q l_j$ . Scaling  $\mathcal{S}$  by a positive factor  $c$  results in the link schedule

$$c\mathcal{S} := \{(I_j, cl_j) : 1 \leq j \leq q\};$$

clearly,  $\|c\mathcal{S}\| = c\|\mathcal{S}\|$ .

Consider a path flow

$$\Gamma = \{(P_i, \delta_i) : 1 \leq i \leq l\}.$$

We use  $\|\Gamma\|$  to denote its flow value  $\sum_{i=1}^l \delta_i$ . Scaling  $\Gamma$  by a positive factor  $c$  results in the path flow

$$c\Gamma := \{(P_i, c\delta_i) : 1 \leq i \leq l\};$$

clearly,  $\|c\Gamma\| = c\|\Gamma\|$ .

Consider a concurrent  $k$ -path flow

$$\Pi = \{(P_{i1}, \dots, P_{ik}, \delta_i) : 1 \leq i \leq l\}$$

We use  $\|\Pi\|$  to denote its flow concurrency  $\sum_{i=1}^l \delta_i$ . Scaling  $\Pi$  by a positive factor  $c$  results in the concurrent  $k$ -path flow

$$c\Pi := \{(P_{i1}, \dots, P_{ik}, c\delta_i) : 1 \leq i \leq l\};$$

clearly,  $\|c\Pi\| = c\|\Pi\|$ .

### III. LEAST INTERFERENCE-COST PATHS

Consider an orientation  $D$  of  $G$  with ILIN  $\mu$ , and let  $\hat{\mu} = \mu$  in SC-SR setting and  $\hat{\mu} = \mu + 2$  in MC-MR setting. Suppose that  $y$  is positive function on  $A$ . Let  $\hat{y}$  be the function on  $A$  defined by

$$\hat{y}(a) = \sum_{b \in N_D^{out}[a]} \rho(a, b) y(b)$$

for each  $a \in A$ . The value  $\hat{y}(a)$  is interpreted as the  $y$ -weighted interference cost brought by  $a$  to its closed out-neighborhood in  $D$ . For each  $1 \leq j \leq k$ , let  $dist_j(\hat{y})$  be the (interference) cost of the cheapest  $(s_j, t_j)$ -path with respect to  $\hat{y}$ . In this section, we establish the following intrinsic relations between the least interference-cost paths and maximum (concurrent) multiflows.

*Theorem 3.1:* Let  $opt$  be the value of the maximum multiflow. Then,

$$\min_{1 \leq j \leq k} dist_j(\hat{y}) \leq \hat{\mu} \frac{y(A)}{opt}.$$

*Theorem 3.2:* Let  $opt$  be the concurrency of the maximum concurrent multiflow. Then,

$$\sum_{j=1}^k d_j dist_j(\hat{y}) \leq \hat{\mu} \frac{y(A)}{opt}.$$

We begin with following property of a singleton path flow.

*Lemma 3.3:* Suppose that  $x$  is the link flow of a singleton path flow  $\{(P, \delta)\}$ . Then, for any  $a \in A$ ,

$$\Delta_D^{in}(a; x) = \delta \rho(a, N_D^{in}[a] \cap P)$$

and

$$\sum_{a \in A} y(a) \Delta_D^{in}(a; x) = \delta \hat{y}(P).$$

*Proof:* For any  $a \in A$ ,

$$\begin{aligned} \Delta_D^{in}(a; x) &= \sum_{b \in N_D^{in}[a]} \rho(a, b) x(b) \\ &= \sum_{b \in N_D^{in}[a]} \rho(a, b) \delta |P \cap \{a\}| \\ &= \delta \sum_{b \in N_D^{in}[a]} \rho(a, b) |P \cap \{b\}| \\ &= \delta \rho(a, N_D^{in}[a] \cap P). \end{aligned}$$

Thus, the first part holds.

By the first part, we have

$$\begin{aligned}
 & \sum_{a \in A} y(a) \Delta_D^{in}(a; x) \\
 &= \delta \sum_{a \in A} y(a) \rho(a, N_D^{in}[a] \cap P) \\
 &= \delta \sum_{a \in A} y(a) \sum_{b \in P} |\{b\} \cap N_D^{in}[a]| \rho(a, b) \\
 &= \delta \sum_{b \in P} \sum_{a \in A} |\{b\} \cap N_D^{in}[a]| \rho(a, b) y(a) \\
 &= \delta \sum_{b \in P} \sum_{a \in N_D^{out}[b]} \rho(a, b) y(a) \\
 &= \delta \widehat{y}(P).
 \end{aligned}$$

So, the second part also holds.  $\blacksquare$

Lemma 3.3 gives an alternative interpretation on the interference cost of a path  $P$ : Let  $x$  be the link flow of a path flow  $\{(P, 1)\}$ .

$$\widehat{y}(P) = \sum_{a \in A} y(a) \Delta_D^{in}(a; x).$$

From Lemma 3.3, we also derive the following property of a singleton concurrent  $k$ -path flow.

*Lemma 3.4:* Suppose that  $x$  is the cumulative link flow of a singleton concurrent  $k$ -path flow

$$\{(P_1, \dots, P_k, \delta)\}.$$

Then, for any  $a \in A$ ,

$$\Delta_D^{in}(a; x) = \delta \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j)$$

and

$$\sum_{a \in A} y(a) \Delta_D^{in}(a; x) = \delta \sum_{j=1}^k d_j \widehat{y}(P_j).$$

*Proof:* The concurrent  $k$ -path flow

$$\{(P_1, \dots, P_k, \delta)\}.$$

is equivalent to the path flow

$$\{(P_j, \delta d_j) : 1 \leq j \leq k\}.$$

For each  $1 \leq j \leq k$ , let  $x_j$  be the link flow of the singleton path flow  $(P_j, \delta d_j)$ . Then,  $x = \sum_{j=1}^k x_j$ . By Lemma 3.3, for any  $a \in A$ ,

$$\begin{aligned}
 \Delta_D^{in}(a; x) &= \sum_{j=1}^k \Delta_D^{in}(a; x_j) \\
 &= \sum_{j=1}^k \delta d_j \rho(a, N_D^{in}[a] \cap P_j) \\
 &= \delta \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j).
 \end{aligned}$$

Again by Lemma 3.3, we have

$$\begin{aligned}
 & \sum_{a \in A} y(a) \Delta_D^{in}(a; x) \\
 &= \sum_{a \in A} y(a) \sum_{j=1}^k \Delta_D^{in}(a; x_j) \\
 &= \sum_{j=1}^k \sum_{a \in A} y(a) \Delta_D^{in}(a; x_j) \\
 &= \delta \sum_{j=1}^k d_j \widehat{y}(P_j)
 \end{aligned}$$

So, the lemma holds.  $\blacksquare$

Now, we are ready to prove Theorem 3.1.

*Proof of Theorem 3.1:* Let

$$\{(P_i, \delta_i) \in \mathcal{P} \times \mathbb{R}_+ : 1 \leq i \leq l\}$$

be a maximum multiflow, and  $x$  be its link flow. For each  $1 \leq i \leq l$ , let  $x_i$  be the link flow of the singleton path flow  $\{(P_i, \delta_i)\}$ . Then,  $x = \sum_{i=1}^l x_i$ . By Lemma 3.3,

$$\begin{aligned}
 & \sum_{a \in A} y(a) \Delta_D^{in}(a; x) \\
 &= \sum_{a \in A} y(a) \sum_{i=1}^l \Delta_D^{in}(a; x_i) \\
 &= \sum_{i=1}^l \sum_{a \in A} y(a) \Delta_D^{in}(a; x_i) \\
 &= \sum_{i=1}^l \delta_i \widehat{y}(P_i) \\
 &\geq \sum_{i=1}^l \delta_i \text{dist}_j(\widehat{y}) \\
 &\geq \left( \min_{1 \leq j \leq k} \text{dist}_j(\widehat{y}) \right) \sum_{i=1}^l \delta_i \\
 &= \left( \min_{1 \leq j \leq k} \text{dist}_j(\widehat{y}) \right) \text{opt}.
 \end{aligned}$$

On one hand, by Lemma 2.2,

$$\sum_{a \in A} y(a) \Delta_D^{in}(a; x) \leq \Delta_D^{in}(x) \sum_{b \in A} y(b) \leq \widehat{\mu} y(A).$$

Thus, the theorem holds.  $\blacksquare$

Finally, we prove Theorem 3.2.

*Proof of Theorem 3.2:* Let

$$\{(P_{i1}, \dots, P_{ik}, \delta_i) : 1 \leq i \leq l\}$$

be a maximum concurrent multiflow, and  $x$  be its cumulative link flow. For each  $1 \leq i \leq l$ , let  $x_i$  be the link flow of the singleton concurrent  $k$ -path flow

$$\{(P_{i1}, \dots, P_{ik}, \delta_i)\}.$$

Then,  $x = \sum_{i=1}^l x_i$ . By Lemma 3.4,

$$\begin{aligned}
 & \sum_{a \in A} y(a) \Delta_D^{in}(a; x) \\
 &= \sum_{a \in A} y(a) \sum_{i=1}^l \Delta_D^{in}(a; x_i) \\
 &= \sum_{i=1}^l \sum_{a \in A} y(a) \Delta_D^{in}(a; x_i) \\
 &= \sum_{i=1}^l \delta_i \sum_{j=1}^k d_j \hat{y}(P_{ij}) \\
 &\geq \sum_{i=1}^l \delta_i \sum_{j=1}^k d_j \text{dist}_j(\hat{y}) \\
 &\geq \left( \sum_{j=1}^k d_j \text{dist}_j(\hat{y}) \right) \left( \sum_{i=1}^l \delta_i \right) \\
 &= \left( \sum_{j=1}^k d_j \text{dist}_j(\hat{y}) \right)_{opt}
 \end{aligned}$$

On the other hand, by Lemma 2.2,

$$\sum_{a \in A} y(a) \Delta_D^{in}(a; x) \leq \Delta_D^{in}(x) \sum_{b \in A} y(b) \leq \hat{\mu} y(A).$$

Thus, the theorem holds.  $\blacksquare$

#### IV. MAXIMUM MULTIFLOW

Suppose that  $D$  is an orientation of  $G$  with ILIN  $\mu$ , and denote  $\hat{\mu} = \mu$  in SC-SR setting and  $\hat{\mu} = \mu + 2$  in MC-MR setting. Consider an arbitrary parameter  $\varepsilon \in (0, 1]$ . In this section, we present an efficient algorithm  $\mathbf{MF}(\varepsilon)$  algorithm for the problem  $\mathbf{MMF}$ , which achieves an approximation bound  $2(1 + \varepsilon)\hat{\mu}$  in general and  $(1 + \varepsilon)\hat{\mu}$  if  $D$  is acyclic, and has running time growing with  $1/\varepsilon$  in at most the square order.

The algorithm  $\mathbf{MF}(\varepsilon)$  is outlined in Table I. It runs in three phases:

- **Flow Phase:** This phase computes a path flow  $\Gamma$  and its link flow  $x$  iteratively by computing a sequence of cheapest paths in terms of interference costs.
- **Link-Scheduling Phase:** This phase computes a link schedule  $S$  of  $x$  by simply applying the greedy algorithm **GLS** developed in [12].
- **Scaling Phase:** This phase scales both  $\Gamma$  and  $S$  by a factor  $1/\|S\|$  and then return them as the final output.

Both the **Link-Scheduling Phase** and the **Scaling Phase** are straightforward. The **Flow Phase** is quite subtle and is elaborated below.

The **Flow Phase** builds up a path flow  $\Gamma$  incrementally and updates its link flow  $x$  accordingly throughout this phase. The **Flow Phase** maintains a flow cost variable  $\gamma$ , which are the cumulative costs incurred by  $\Gamma$ . Initially,  $\Gamma$  is empty,  $x$  is zero-valued, and  $\gamma$  is zero. The **Flow Phase** also maintains a positive weight function  $y$  on  $A$  to help the building up  $\Gamma$ . Initially,  $y$  is one-valued. The **Flow Phase** runs in

Algorithm $\mathbf{MF}(\varepsilon)$
<b>// Flow Phase</b> $\Gamma \leftarrow \emptyset, x \leftarrow \mathbf{0}, \gamma \leftarrow 0; y \leftarrow \mathbf{1};$ while $\Delta_D^{in}(x) \geq (1 + \varepsilon)\gamma$ do $P \leftarrow$ a cheapest path in $\mathcal{P}$ w.r.t. $\hat{y};$ $\delta \leftarrow \frac{1}{\max_{a \in A} \rho(a, N_D^{in}[a] \cap P)};$ $\Gamma \leftarrow \Gamma \cup \{(P, \delta)\};$ $\gamma \leftarrow \gamma + \frac{\hat{y}(P)}{y(A)} \delta;$ $\forall a \in P, x(a) \leftarrow x(a) + \delta;$ $\forall a \in A, y(a) \leftarrow y(a) (1 + \varepsilon \delta \rho(a, N_D^{in}[a] \cap P));$
<b>// Link-Scheduling Phase</b> $S \leftarrow$ the link schedule of $x$ output by <b>GLS</b> ;
<b>// Scaling Phase</b> $\Gamma \leftarrow \frac{1}{\ S\ } \Gamma, S \leftarrow \frac{1}{\ S\ } S;$ Output $\Gamma$ and $S$ .

TABLE I  
OUTLINE OF THE ALGORITHM  $\mathbf{MF}(\varepsilon)$ .

iterations until the flow cost  $\gamma$  exceeds  $\Delta_D^{in}(x)/(1 + \varepsilon)$ . In each iteration, a cheapest path  $P \in \mathcal{P}$  with respect to  $\hat{y}$  is computed, and  $\delta$  units of flow are routed along  $P$  where

$$\delta = \frac{1}{\max_{a \in A} \rho(a, N_D^{in}[a] \cap P)}.$$

The  $\delta$  is selected such that after  $\Gamma$  is augmented by  $(P, \delta)$ , the maximum increment on  $\Delta_D^{in}(a; x)$  for all  $a \in A$  is exactly one by Lemma 3.3. The pair  $(P, \delta)$  is then added to  $\Gamma$ , and all other variables are subsequently updated. Specifically, the new flow  $(P, \delta)$  incurs a flow cost  $\frac{\hat{y}(P)}{y(A)} \delta$ , which is the product of the  $y$ -weighted interference cost of  $P$  and the flow amount  $\delta$  routed along  $P$ . Such cost is charged to  $\gamma$ . The update on  $x$  is straightforward:  $x(a)$  is incremented by  $\delta$  for each link  $a \in P$ . For each link  $a \in A$ ,  $y(a)$  is increased by a factor

$$1 + \varepsilon \delta \rho(a, N_D^{in}[a] \cap P).$$

Note that  $\delta \rho(a, N_D^{in}[a] \cap P)$  is the increment of  $\Delta_D^{in}(a; x)$  due to the growth of  $\Gamma$  by Lemma 3.3. Such update on  $y$  ensures that if a link  $a$  receives a larger increment on  $\Delta_D^{in}(a; x)$ , then  $y(a)$  grows faster which in turn makes it less likely to appear in a cheapest path in the future iteration.

Next, we analyze the performance of the algorithm  $\mathbf{MF}(\varepsilon)$ .

*Theorem 4.1:* The **Flow Phase** of the algorithm  $\mathbf{MF}(\varepsilon)$  terminates in at most

$$\left\lceil \frac{m \ln m}{\ln(1 + \varepsilon) - \frac{\varepsilon}{1 + \varepsilon}} \right\rceil$$

iterations, and the final output of the algorithm  $\mathbf{MF}(\varepsilon)$  is a  $2(1 + \varepsilon)\hat{\mu}$ -approximate solution in general and a  $(1 + \varepsilon)\hat{\mu}$ -approximate solution if  $D$  is acyclic.

*Proof:* We introduce the following notations in this proof. Let  $opt$  be the value of the maximum multiflow.  $\Gamma_0, x_0, y_0$ , and  $\gamma_0$  denote initial values of  $x, y$ , and  $\gamma$  respectively in the **Flow Phase**. For each iteration  $i \geq 1$  of the **Flow Phase**,  $\Gamma_i, x_i, y_i$  and  $\gamma_i$  denote the values of  $x, y$ , and  $\gamma$  respectively at the end of the  $i$ -th iteration;  $(P_i, \delta_i)$  denotes the flow added in

the  $i$ -th iteration. Consider any iteration  $i$  of the **Flow Phase**. For each  $a \in A$ , since

$$\begin{aligned} \Delta_D^{in}(a; x_i) - \Delta_D^{in}(a; x_{i-1}) \\ = \delta_i \rho(a, N_D^{in}[a] \cap P_i) \leq 1, \end{aligned}$$

we have

$$\begin{aligned} y_i(a) &= y_{i-1}(a) (1 + \varepsilon \delta_i \rho(a, N_D^{in}[a] \cap P_i)) \\ &\geq y_{i-1}(a) (1 + \varepsilon)^{\delta_i \rho(a, N_D^{in}[a] \cap P_i)} \\ &= y_{i-1}(a) (1 + \varepsilon)^{\Delta_D^{in}(a; x_i) - \Delta_D^{in}(a; x_{i-1})}. \end{aligned}$$

Furthermore,

$$\begin{aligned} y_i(A) &= \sum_{a \in A} y_i(a) \\ &= \sum_{a \in A} y_{i-1}(a) (1 + \varepsilon \delta_i \rho(a, N_D^{in}[a] \cap P_i)) \\ &= y_{i-1}(A) + \varepsilon \delta_i \sum_{a \in A} \rho(a, N_D^{in}[a] \cap P_i) y_{i-1}(a) \\ &= y_{i-1}(A) + \varepsilon \delta_i \widehat{y}_{i-1}(P_i) \\ &= y_{i-1}(A) \left( 1 + \varepsilon \delta_i \frac{\widehat{y}_{i-1}(P_i)}{y_{i-1}(A)} \right) \\ &= y_{i-1}(A) (1 + \varepsilon (\gamma_i - \gamma_{i-1})) \\ &\leq y_{i-1}(A) \exp(\varepsilon (\gamma_i - \gamma_{i-1})), \end{aligned}$$

where the fourth equality follows from Lemma 3.3. Therefore, by induction on the iterations, for each iteration  $l$  of the **Flow Phase**, for each  $a \in A$ ,

$$\begin{aligned} y_l(a) &\geq y_0(a) (1 + \varepsilon)^{\sum_{i=1}^l (\Delta_D^{in}(a; x_i) - \Delta_D^{in}(a; x_{i-1}))} \\ &= (1 + \varepsilon)^{\Delta_D^{in}(a; x_l)}; \end{aligned}$$

and

$$y_l(A) \leq y_0(A) \exp\left(\varepsilon \sum_{i=1}^l (\gamma_i - \gamma_{i-1})\right) = m \exp(\varepsilon \gamma_l).$$

Hence,

$$\gamma_l \geq \frac{1}{\varepsilon} \ln \frac{y_l(A)}{m} \geq \frac{1}{\varepsilon} \ln \frac{(1 + \varepsilon)^{\Delta_D^{in}(x_l)}}{m}.$$

On the other hand, by Theorem 3.1, we have

$$\gamma_l = \sum_{i=1}^l \delta_i \frac{\widehat{y}_{i-1}(P_i)}{y_{i-1}(A)} \leq \frac{\widehat{\mu} \sum_{i=1}^l \delta_i}{opt} = \frac{\widehat{\mu} \|\Gamma_l\|}{opt}.$$

Now, we bound the number of iterations of the **Flow Phase**. Assume to the contrary that the **Flow Phase** didn't terminate after

$$\left\lceil \frac{m \ln m}{\ln(1 + \varepsilon) - \frac{\varepsilon}{1 + \varepsilon}} \right\rceil$$

iterations. Let

$$l = \left\lceil \frac{m \ln m}{\ln(1 + \varepsilon) - \frac{\varepsilon}{1 + \varepsilon}} \right\rceil.$$

By the choice of  $\delta_i$ ,  $\sum_{a \in A} \Delta_D^{in}(a; x_i)$  strictly increases by at least one with the iteration  $i$ . So,

$$\sum_{a \in A} \Delta_D^{in}(a; x_l) \geq l,$$

which implies

$$\Delta_D^{in}(x_l) \geq \frac{\sum_{a \in A} \Delta_D^{in}(a; x_l)}{m} \geq l/m > \frac{\ln m}{\ln(1 + \varepsilon) - \frac{\varepsilon}{1 + \varepsilon}}.$$

Hence,

$$\frac{1}{\varepsilon} \ln \frac{(1 + \varepsilon)^{\Delta_D^{in}(x_l)}}{m} > \frac{\Delta_D^{in}(x_l)}{1 + \varepsilon}.$$

Thus,

$$\gamma_l \geq \frac{1}{\varepsilon} \ln \frac{(1 + \varepsilon)^{\Delta_D^{in}(x_l)}}{m} > \frac{\Delta_D^{in}(x_l)}{1 + \varepsilon}.$$

This means that the number of iterations is at most  $l$ , which is a contradiction.

Next, we show that at the end of the of the **Flow Phase**,

$$\frac{\|\Gamma\|}{\Delta_D^{in}(x)} \geq \frac{opt}{(1 + \varepsilon) \widehat{\mu}}.$$

Suppose that the **Flow Phase** runs in  $l$  iterations. By the stopping rule of the **Flow Phase**,

$$\Delta_D^{in}(x_l) \leq (1 + \varepsilon) \gamma_l \leq \frac{(1 + \varepsilon) \widehat{\mu} \|\Gamma_l\|}{opt}.$$

Thus,

$$\frac{\|\Gamma_l\|}{\Delta_D^{in}(x_l)} \geq \frac{opt}{(1 + \varepsilon) \widehat{\mu}}.$$

Finally, we prove the approximation bound of the output by the algorithm. It is sufficient to show that at the end of the **Link-Scheduling Phase**,

$$\frac{\|\Gamma\|}{\|\mathcal{S}\|} \geq \frac{opt}{2(1 + \varepsilon) \widehat{\mu}},$$

and if  $D$  is acyclic, then

$$\frac{\|\Gamma\|}{\|\mathcal{S}\|} \geq \frac{opt}{(1 + \varepsilon) \widehat{\mu}}.$$

By Lemma 2.2, we have

$$\|\mathcal{S}\| \leq \Delta^*(x) \leq 2\Delta_D^{in}(x),$$

which implies that

$$\frac{\|\Gamma\|}{\|\mathcal{S}\|} \geq \frac{\|\Gamma\|}{2\Delta_D^{in}(x)} \geq \frac{opt}{2(1 + \varepsilon) \widehat{\mu}}.$$

If  $D$  is acyclic, again by Lemma 2.2 we have

$$\|\mathcal{S}\| \leq \Delta^*(x) \leq \Delta_D^{in}(x),$$

which implies that

$$\frac{\|\Gamma\|}{\|\mathcal{S}\|} \geq \frac{\|\Gamma\|}{\Delta_D^{in}(x)} \geq \frac{opt}{(1 + \varepsilon) \widehat{\mu}}.$$

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

The running time of  $\mathbf{MF}(\varepsilon)$  increases with  $1/\varepsilon$  in at most the square order because

$$\begin{aligned} \ln(1+\varepsilon) - \frac{\varepsilon}{1+\varepsilon} &= -\ln\left(1 - \frac{\varepsilon}{1+\varepsilon}\right) - \frac{\varepsilon}{1+\varepsilon} \\ &\geq \frac{1}{2} \left(\frac{\varepsilon}{1+\varepsilon}\right)^2 = \frac{1}{2} (1+1/\varepsilon)^{-2}. \end{aligned}$$

The running time of the the **Flow Phase** does not depend on the the number of channels and the number of radios at each node. Within each iteration of the **Flow Phase**, a cheapest path  $P \in \mathcal{P}$  with respect to  $\hat{y}$  has to be computed. A naive implementation is first making  $k$  separate computations of cheapest paths  $P_j \in \mathcal{P}_j$  for each  $1 \leq j \leq k$  and then choosing the cheapest one among them as  $P$ . However, by observing that Dijkstra's algorithm for computing shortest paths gives the shortest path to every node in the graph, we adopt the following more economic implementation. We first group unicasts by a common source node. The number of groups is at more  $\min\{n, k\}$ . Then, for each group we compute cheapest paths from the common source node to the sinks of all unicast in this group by a single call to the Dijkstra's algorithm. As the Dijkstra's algorithm has running time  $O(m+n \log n)$  based on Fibonacci heap, we can compute the cheapest path  $P \in \mathcal{P}$  in

$$O(\min\{n, k\} (m+n \log n))$$

time. For  $k = \Theta(n^2)$ , this implementation is a linear factor speedup over the naive implementation.

## V. MAXIMUM CONCURRENT MULTIFLOW

Suppose that  $D$  is an orientation of  $G$  with ILIN  $\mu$ , and denote  $\hat{\mu} = \mu$  in SC-SR setting and  $\hat{\mu} = \mu + 2$  in MC-MR setting. Consider an arbitrary parameter  $\varepsilon \in (0, 1]$ . In this section, we present an efficient algorithm  $\mathbf{CMF}(\varepsilon)$  algorithm for the problem **MCMF**, which achieves an approximation bound  $2(1+\varepsilon)\hat{\mu}$  in general and  $(1+\varepsilon)\hat{\mu}$  if  $D$  is acyclic, and has running time growing with  $1/\varepsilon$  in at most the square order.

The algorithm  $\mathbf{CMF}(\varepsilon)$  is outlined in Table II. Similar to the algorithm  $\mathbf{MF}(\varepsilon)$ , it runs in three phases:

- **Flow Phase**: This phase computes a concurrent  $k$ -path flow  $\Pi$  and its cumulative link flow  $x$  iteratively by computing a sequence of cheapest paths in terms of interference costs.
- **Link-Scheduling Phase**: This phase computes a link schedule  $\mathcal{S}$  of  $x$  by simply applying the greedy algorithm **GLS** developed in [12].
- **Scaling Phase**: This phase scales both  $\Pi$  and  $\mathcal{S}$  by a factor  $1/\|\mathcal{S}\|$  and then return them as the final output.

The **Flow Phase** of  $\mathbf{CMF}(\varepsilon)$  is more intricate than that of  $\mathbf{MF}(\varepsilon)$  and is elaborated below.

The **Flow Phase** builds up a concurrent  $k$ -path flow  $\Pi$  incrementally and updates its cumulative link flow  $x$  accordingly throughout this phase. The **Flow Phase** maintains a flow cost variable  $\gamma$  storing the cumulative costs incurred by  $\Pi$ . Initially,  $\Pi$  is empty,  $x$  is zero-valued, and  $\gamma$  is zero. The **Flow Phase**

Algorithm $\mathbf{CMF}(\varepsilon)$
<p><b>// Flow Phase</b>  <math>\Pi \leftarrow \emptyset, x \leftarrow \mathbf{0}, \gamma \leftarrow 0; y \leftarrow \mathbf{1};</math>                      while <math>\Delta_D^{in}(x) \geq (1+\varepsilon)\gamma,</math></p> <p style="padding-left: 2em;"><math>(P_1, \dots, P_k) \leftarrow</math> cheapest paths in <math>\prod_{j=1}^k \mathcal{P}_j</math> w.r.t. <math>\hat{y};</math></p> <p style="padding-left: 2em;"><math>\delta \leftarrow \frac{1}{\max_{a \in A} \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j)};</math>  <math>\Pi \leftarrow \Pi \cup \{(P_1, \dots, P_k, \delta)\};</math>  <math>\gamma \leftarrow \gamma + \frac{\sum_{j=1}^k d_j \hat{y}(P_j)}{y(A)} \delta;</math>  <math>\forall a \in A,</math></p> <p style="padding-left: 4em;"><math>x(a) \leftarrow x(a) + \delta \sum_{j=1}^k d_j  P_j \cap \{a\} ;</math></p> <p style="padding-left: 4em;"><math>y(a) \leftarrow y(a) (1 + \varepsilon \delta \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j));</math></p>
<p><b>// Link-Scheduling Phase</b>  <math>\mathcal{S} \leftarrow</math> the link schedule of <math>x</math> output by <b>GLS</b>;</p>
<p><b>// Scaling Phase</b>  <math>\Pi \leftarrow \frac{1}{\ \mathcal{S}\ } \Pi, \mathcal{S} \leftarrow \frac{1}{\ \mathcal{S}\ } \mathcal{S};</math>                      Output <math>\Pi</math> and <math>\mathcal{S}.</math></p>

TABLE II  
OUTLINE OF THE ALGORITHM  $\mathbf{CMF}(\varepsilon)$ .

also maintains a positive weight function  $y$  on  $A$  to help the building up  $\Gamma$ . Initially,  $y$  is one-valued. The **Flow Phase** runs in iterations until the flow cost  $\gamma$  exceeds  $\Delta_D^{in}(x) / (1+\varepsilon)$ . In each iteration,  $k$  cheapest paths  $P_j \in \mathcal{P}_j$  for  $1 \leq j \leq k$  with respect to  $\hat{y}$  are computed. Along these  $k$  paths, a concurrent flow of concurrency  $\delta$  is routed where

$$\frac{1}{\max_{a \in A} \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j)}.$$

The  $\delta$  is selected such that after  $\Pi$  is augmented by the new flow

$$(P_1, \dots, P_k, \delta),$$

the maximum increment on  $\Delta_D^{in}(a; x)$  for all  $a \in A$  is exactly one by Lemma 3.4. The new flow is then added to  $\Pi$ , and all other variables are subsequently updated. Specifically, the  $j$ -th new flow incurs a cost  $\frac{\hat{y}(P_j)}{y(A)} \delta d_j$ , which is the product of the  $y$ -weighted interference prices of  $P_j$  and the flow amount  $\delta d_j$  routed along  $P_j$ . The total cost of the new flow is  $\frac{\sum_{j=1}^k d_j \hat{y}(P_j)}{y(A)} \delta$ , and it is added to  $\gamma$ . The update on  $x$  is straightforward: for each link  $a \in A$ ,  $x(a)$  is incremented by the total new flow through  $a$ , which is equal to  $\delta \sum_{j=1}^k d_j |P_j \cap \{a\}|$ . For each link  $a \in A$ ,  $y(a)$  is increased by a factor

$$1 + \varepsilon \delta \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j).$$

Note that the term  $\delta \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_j)$  is the increment of  $\Delta_D^{in}(a; x)$  due to the growth of  $\Pi$  by Lemma 3.4. Such update on  $y$  ensures that if a link  $a$  receives a larger increment on  $\Delta_D^{in}(a; x)$ , then  $y(a)$  grows faster which in turn makes it less likely to appear in a cheapest path in the future iteration.

Next, we analyze the performance of the algorithm  $\mathbf{CMF}(\varepsilon)$ .

*Theorem 5.1:* The algorithm  $\mathbf{CMF}(\varepsilon)$  terminates in at most

$$\left\lceil \frac{m \ln m}{\ln(1+\varepsilon) - \frac{\varepsilon}{1+\varepsilon}} \right\rceil$$

iterations, and outputs  $\Pi$  and  $x$  satisfying that

$$\frac{\|\Pi\|}{\Delta_D^{in}(x)} \geq \frac{opt}{(1+\varepsilon)\hat{\mu}}.$$

*Proof:* We introduce the following notations in this proof. Let  $opt$  be the concurrency of the maximum concurrent multifold.  $\Pi_0, x_0, y_0$ , and  $\gamma_0$  denote initial values of  $x, y$ , and  $\gamma$  respectively in the **Flow Phase**. For each iteration  $i \geq 1$  of the **Flow Phase**,  $\Pi_i, x_i, y_i$  and  $\gamma_i$  denote the values of  $x, y$ , and  $\gamma$  respectively at the end of the  $i$ -th iteration;

$$(P_{i1}, \dots, P_{ik}, \delta_i)$$

denotes the concurrent flow added in the  $i$ -th iteration. Consider any iteration  $i$  of the **Flow Phase**. For each  $a \in A$ , since

$$\begin{aligned} & \Delta_D^{in}(a; x_i) - \Delta_D^{in}(a; x_{i-1}) \\ &= \delta_i \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_{ij}) \leq 1, \end{aligned}$$

we have

$$\begin{aligned} y_i(a) &= y_{i-1}(a) \left( 1 + \varepsilon \delta_i \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_{ij}) \right) \\ &\geq y_{i-1}(a) (1 + \varepsilon)^{\delta_i \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_{ij})} \\ &= y_{i-1}(a) (1 + \varepsilon)^{\Delta_D^{in}(a; x_i) - \Delta_D^{in}(a; x_{i-1})}. \end{aligned}$$

Furthermore,

$$\begin{aligned} y_i(A) &= \sum_{a \in A} y_{i-1}(a) (1 + \varepsilon \delta_i \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_{ij})) \\ &= y_{i-1}(A) + \varepsilon \delta_i \sum_{a \in A} y_{i-1}(a) \sum_{j=1}^k d_j \rho(a, N_D^{in}[a] \cap P_{ij}) \\ &= y_{i-1}(A) + \varepsilon \delta_i \sum_{j=1}^k d_j \sum_{a \in A} \rho(a, N_D^{in}[a] \cap P_{ij}) y_{i-1}(a) \\ &= y_{i-1}(A) + \varepsilon \delta_i \sum_{j=1}^k d_j \text{dist}_j(\hat{y}_{i-1}) \\ &= y_{i-1}(A) \left( 1 + \varepsilon \delta_i \frac{\sum_{j=1}^k d_j \text{dist}_j(\hat{y}_{i-1})}{y_{i-1}(A)} \right) \\ &= y_{i-1}(A) (1 + \varepsilon(\gamma_i - \gamma_{i-1})) \\ &\leq y_{i-1}(A) \exp(\varepsilon(\gamma_i - \gamma_{i-1})), \end{aligned}$$

where the fourth equality follows from Lemma 3.3.

The rest of the proof is almost the same as that in the proof of Theorem 4.1. We only outline the steps of the argument

and skip the details. First, we can show that for any iteration number  $l$  of the **Flow Phase**, at the end of the  $l$ -th iteration,

$$\frac{1}{\varepsilon} \ln \frac{(1+\varepsilon)^{\Delta_D^{in}(x_l)}}{m} \leq \gamma_l \leq \frac{\hat{\mu} \|\Pi_l\|}{opt}.$$

Then, using the above inequality and the fact that  $\sum_{a \in A} \Delta_D^{in}(a; x_i)$  strictly increases by at least one with the iteration  $i$ , we can prove the upper bound on the number of iterations of the **Flow Phase** by contradiction. Using the above inequality again and the stopping rule of the **Flow Phase**, we can show that at the end of the of the **Flow Phase**,

$$\frac{\|\Pi\|}{\Delta_D^{in}(x)} \geq \frac{opt}{(1+\varepsilon)\hat{\mu}}.$$

Using this inequality and Lemma 2.2, we can conclude that at the end of the **Link-Scheduling Phase**,

$$\frac{\|\Pi\|}{\|\mathcal{S}\|} \geq \frac{opt}{2(1+\varepsilon)\hat{\mu}},$$

and if  $D$  is acyclic, then

$$\frac{\|\Pi\|}{\|\mathcal{S}\|} \geq \frac{opt}{(1+\varepsilon)\hat{\mu}}.$$

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

The same remark at the end of Section IV also applies to the algorithm  $\mathbf{CMF}(\varepsilon)$  and is omitted here.

## VI. APPLICATIONS

In this section, we apply the algorithms  $\mathbf{MF}(\varepsilon)$  and  $\mathbf{CMF}(\varepsilon)$  to the plane geometric variants of the protocol interference model by choosing an appropriate orientation  $D$ . We remark that any ordering  $\prec$  of  $A$  naturally induces an orientation  $D$  by orienting every edge  $\{a, b\}$  in the link-conflict graph  $G$  for  $a$  to  $b$  if  $a \prec b$ , or in the reverse direction otherwise.

In the unidirectional mode, Wan [10] introduced the following orientation  $D$ . Consider any conflicting pair of links  $a$  and  $b$  in  $A$ . If the receiver of  $a$  is within the interference range of the sender of  $b$ , then we take the orientation from  $b$  to  $a$ ; otherwise, we take the orientation from  $a$  to  $b$ . Ties are broken arbitrarily. Suppose that for each link  $a \in A$ , the interference radius of its sender is at least  $c$  times its length for some constant  $c > 1$ . It was shown in [13] that the ILIN of such  $D$  is at most  $\lceil \pi / \arcsin \frac{1-c}{2} \rceil - 1$ . By Theorem 4.1 and Theorem 5.1, we have the following corollary.

*Corollary 6.1:* Under the plane geometric variant of the protocol interference model in the unidirectional mode in which the interference radius of its sender of each link is at least  $c$  times the link length for some constant  $c > 1$ , by adopting the orientation  $D$  given in [10], both  $\mathbf{MF}(\varepsilon)$  and  $\mathbf{CMF}(\varepsilon)$  have an approximation bound

$$2(1+\varepsilon) \left( \left\lceil \pi / \arcsin \frac{c-1}{2c} \right\rceil + 1 \right)$$

in MC-MR wireless networks, and

$$2(1+\varepsilon) \left( \left\lceil \pi / \arcsin \frac{c-1}{2c} \right\rceil - 1 \right)$$

in SC-SR wireless networks,

In the bidirectional mode, Wan et al. [13] introduced the following orientation  $D$ . Consider any conflicting pair of links  $a$  and  $b$  in  $A$ . If  $a$  has an endpoint  $u$  and  $b$  has an endpoint  $v$  satisfying that  $u$  is within the interference range of  $v$  and its interference radius is no more than that of  $v$ , then we take the orientation from  $b$  to  $a$ ; otherwise, we take the orientation from  $a$  to  $b$ . Ties are broken arbitrarily. It was shown in [13] that the ILIN of such  $D$  is at most 8. By Theorem 4.1 and Theorem 5.1, we have the following corollary.

*Corollary 6.2:* Under the plane geometric variant of the protocol interference model in the bidirectional mode, by adopting the orientation  $D$  given in [13], both  $\mathbf{MF}(\varepsilon)$  and  $\mathbf{CMF}(\varepsilon)$  have an approximation bound  $18(1 + \varepsilon)$  in MC-MR wireless networks, and  $16(1 + \varepsilon)$  in SC-SR wireless networks,

In the bidirectional mode with symmetric interference radii (i.e., for each link, its two endpoints have equal interference radii), we sort the links in the decreasing order of the interference radii of their endpoints and ties are broken arbitrarily. Such ordering is referred to as the *interference radius decreasing ordering*. It was shown in [13] that for the acyclic orientation  $D$  induced by the interference radius decreasing ordering, its ILIN is at most 8. By Theorem 4.1 and Theorem 5.1, we have the following corollary.

*Corollary 6.3:* Under the plane geometric variant of the protocol interference model in the bidirectional mode with symmetric interference radii, by adopting the orientation  $D$  induced by the interference radius decreasing ordering, both  $\mathbf{MF}(\varepsilon)$  and  $\mathbf{CMF}(\varepsilon)$  have an approximation bound  $10(1 + \varepsilon)$  in MC-MR wireless networks, and  $8(1 + \varepsilon)$  in SC-SR wireless networks,

In the bidirectional mode with uniform interference radii (i.e., the endpoints of all links have equal interference radii), we sort the links in the lexicographic order of their left endpoints and ties are broken arbitrarily. Such ordering is referred to as the *lexicographic ordering*. It was shown in [5] that for the acyclic orientation  $D$  induced by the lexicographic ordering, its ILIN is at most 6. By Theorem 4.1 and Theorem 5.1, we have the following corollary.

*Corollary 6.4:* Under the plane geometric variant of the protocol interference model in the bidirectional mode with uniform interference radii, by adopting the orientation  $D$  induced by the lexicographic ordering, both  $\mathbf{MF}(\varepsilon)$  and  $\mathbf{CMF}(\varepsilon)$  have an approximation bound  $8(1 + \varepsilon)$  in MC-MR wireless networks, and  $6(1 + \varepsilon)$  in SC-SR wireless networks,

## VII. CONCLUSION

In this paper, we have developed purely combinatorial approximation algorithms for both maximum multifold and maximum concurrent multifold in MC-MR wireless networks. These algorithms adopt an orientation  $D$  of the conflict graph and a parameter  $\varepsilon \in (0, 1]$  which represents the trade-off between the approximation ratio and the running time. Let  $\mu$  be the ILIN of  $D$ , and denote  $\hat{\mu} = \mu$  in SC-SR setting and  $\hat{\mu} = \mu + 2$  in MC-MR setting. At the running time growing with  $1/\varepsilon$  in at most the square order, they produce

$2(1 + \varepsilon)\hat{\mu}$ -approximate solutions in general, and  $(1 + \varepsilon)\hat{\mu}$ -approximate solutions if  $D$  is acyclic. Compared to the existing approximation algorithms based on the LP approach, these approximation bounds are only slightly larger by a factor  $1 + \varepsilon$ , but they are much faster and simpler. Underlying the combinatorial approach followed by these algorithms is the intrinsic relation between interference costs and prices of a path and the maximum (concurrent) multifold. These algorithms iteratively compute a sequence of least interference-cost routing paths along which the flows are routed. By leveraging the techniques in [3], [2], [6], we can further speed up these algorithms and the details will be reported in the full version of this paper.

**ACKNOWLEDGEMENTS:** This work was supported in part by the National Science Foundation of USA under grants CNS-0916666 and CNS-1219109, and by the National Natural Science Foundation of P. R. China under grants 61128005.

## REFERENCES

- [1] D. Bienstock. *Potential function methods for approximately solving linear programs: theory and practice*. Kluwer Academic Publisher, 2002.
- [2] L. K. Fleischer, Approximating fractional multicommodity flow independent of the number of commodities, *SIAM Journal of Discrete Mathematics* 13(4):505–520, 2000.
- [3] N. Garg and J. Könemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, *SIAM Journal on Computing* 37: 630–652, 2007. Conference version appeared in *Proc. IEEE FOCS* 1998, pp. 300–309.
- [4] 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.
- [5] C. Joo, X. Lin, and N. B. Shroff, Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-hop Wireless Networks, *Proc. IEEE INFOCOM* 2008.
- [6] G. Karakostas, Faster approximation schemes for fractional multicommodity flow problems, *Proc. ACM-SIAM SODA* 2002.
- [7] M. Kodialam and T. Nandagopal, Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem, *Proc. of ACM MobiCom* 2003.
- [8] M. Kodialam and T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, *Proc. ACM MobiCom* 2005.
- [9] 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.
- [10] P.-J. Wan, Multiflows in Multihop Wireless Networks, *Proc. ACM MOBIHOC* 2009.
- [11] P.-J. Wan, Y. Cheng, Z. Wang, and F. Yao, Multiflows in Multi-Channel Multi-Radio Multihop Wireless Networks, *Proc. IEEE INFOCOM* 2011.
- [12] 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, *Proc. IEEE INFOCOM* 2013.
- [13] P.-J. Wan, C. Ma, Z. Wang, B. Xu, M. Li, and X. Jia, Weighted Wireless Link Scheduling without Information of Positions And Interference/Communication Radii, *Proc. IEEE INFOCOM* 2011.