Multiflows under Protocol Interference Model

(draft)

Consider a multihop wireless network $(V, A, \mathcal{I})$ under the protocol interference model, 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. Each link $a \in A$ has a communication data rate $\beta(a) > 0$. For any positive integer k, let $[k]$ denote the set $\{1, 2, \cdots, k\}$. 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 $f = \langle f_1, f_2, \cdots, f_k \rangle$ with $f_j \in \mathcal{F}_j$ for each $j \in [k]$. The transmission time of each link $a \in A$ required by a multiflow $f = \langle f_1, f_2, \cdots, f_k \rangle$ is

$$d_f(a) = \frac{1}{\beta(a)} \sum_{j \in [k]} f_j(a)$$

Historically, the following three parameters of a multiflow $f = \langle f_1, f_2, \cdots, f_k \rangle$ have been the subjects of the studies.

- Given that each request j has a weight w_j per unit of its flow, the total weight of f is $\sum_{j \in [k]} val(f_j) w_j$.
- Given that each request j has a traffic demand w_j, the concurrency of f is $\min_{j \in [k]} \frac{val(f_j)}{w_j}$.
- Given that each link $a \in A$ has an expense $g(a)$ per unit of flow through itself, the total expense of f is $\sum_{a \in A} g(a) \beta(a) d_f(a)$.

Among all variants of the multiflow problems, the following two are the most basic ones.

- **Maximum Weighted Multiflow** (**MWMF**): Given that each request has a weight per unit of flow, the problem **MWMF** seeks a multiflow f and a MAC-layer link schedule $\mathcal{S}$ of d_f such that the length of $\mathcal{S}$ is at most one and the total weight 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 d_f such that the length of $\mathcal{S}$ is at most one and the concurrency of f is maximized.

The following two are the budgeted extensions to the above two basic problems:

- **Budgeted MWMF**: Given that each request has a weight and each link has an expense per unit of flow, the problem **Budgeted MWMF** seeks a multiflow f and a MAC-layer link schedule $\mathcal{S}$ of d_f such that the length of $\mathcal{S}$ is at most one, the total expense of f is at most one, and the total weight of f is maximized.
- **Budgeted MCMF**: Given that each request has a traffic demand and each link has an expense per unit of flow, the problem **Budgeted MCMF** seeks a multiflow f and a MAC-layer link schedule $\mathcal{S}$ of d_f such that the length of $\mathcal{S}$ is at most one, the total expense of f is at most one, and the concurrency of f is maximized.

Both **MWMF** and **MCMF** under the protocol interference model are NP-hard even when all nodes have uniform (and fixed) communication radii and uniform (and fixed) interference radii and the positions of all nodes are available [1].

This chapter presents a number of practical approximation algorithms for them in addition to the PTAS. Section 1 gives approximation-preserving reductions of the four our variants of the multiflow problems to **MWISL** and polynomial capacity subregions. However, these reductions are quite slow, and faster and simpler approximation algorithms are developed subsequently based on an adaptive zero-sum game introduced in Section 2. Section 3 establishes approximate weak dualities of **MWMF** and **MCMF**, which reveal the intrinsic relations between them and shortest paths with respect to some link length function. Section 4 describes the design and analyses of the flow augmentation methods for **MWMF** and **MCMF**, which offer nice trade-off between accuracy in terms the approximation bound and efficiency in terms of the running time. Finally, Section 5 extends the flow augmentation methods to **Budgeted MWMF** and **Budgeted MCMF**.

1 Approximation-Preserving Reductions

In this section, we first present an approximation-preserving reductions from the four variants of the multiflow problems to **MWISL**.

Theorem 1.1 *Suppose that there is a polynomial (respectively, a polynomial μ-approximation) algorithm for **MWISL**. Then, all the four variants of the multiflow problems have a polynomial (respectively, a polynomial μ-approximation) algorithm.*

The proof of the above theorem is omitted. Since the problem **MWISL** under the protocol interference model has a PTAS, by Theorem 1.1 we have the following result.

Theorem 1.2 *all the four variants of the multiflow problems under protocol interference model have a PTAS.*

Next, we present first present an approximation-preserving reduction from **MWMF** and **MCMF** to polynomial approximate capacity subregions.

Theorem 1.3 *If there exists a polynomial μ-approximate capacity subregion, then there exist polynomial time μ-approximate algorithms for each of the all the four variants of the multiflow problems.*

Proof. We only provide the proof for **MWMF** while remarking that all other threes can be proved in the same way. Let Ω be the capacity region of the underlying network. The maximum multiflow can be formulated as the following linear program (LP)

$$\begin{array}{ll} \max & \sum_{j\in[k]} val\left(f_j\right) w_j \\ s.t. & f_j \in \mathcal{F}_j, \forall j \in [k]\,; \\ & d_f \in \Omega. \end{array}$$

Suppose that Φ is a polynomial μ-approximate capacity subregion. A k-flow $\langle f_1, f_2, \cdots, f_k\rangle$ is said to be Φ-*restricted* if $d_f \in \Phi$. The Φ-restricted maximum weighted multiflow is defined by the following LP

$$\begin{array}{ll} \max & \sum_{j\in[k]} val\left(f_j\right) w_j \\ s.t. & f_j \in \mathcal{F}_j, \forall j \in [k]\,; \\ & d_f \in \Phi. \end{array}$$

This LP is of polynomial size and we solve this Φ-restricted LP in polynomial time to obtain a k-flow f. Then, we compute a fractional link schedule for d_f. Finally, we scale the fractional link schedule to obtain a fractional stream schedule of length exactly one. This link schedule is a μ-approximate solution for **MWMF**. ■

By adopting the polynomial inward capacity subregion constructed in the previous chapter, we can achieve the approximation bounds $2\left(\lceil \pi/\arcsin\frac{c-1}{2c}\rceil - 1\right)$. In addition, in case uniform interference radii, we can adopt the strip-wise polynomial capacity subregion defined in the previous chapter to achieve improved approximation bound.

2 An Adaptive Zero-Sum Game

In this section, we introduce an adaptive zero-sum game in the context of learning or game theory. The game playing strategy to be described in this section makes both the algorithm designs and

analyses in the later part of this chapter fairly modular and clarifies the high-level structure of the argument. In the adaptive zero-sum game, a sequential game is played in rounds between an agent and a set E of m loss-incurring adversaries. At the beginning of each round, the agent declares an *adaptive binding strategy* in terms of probabilistic distributions on adversaries. Then the adversaries picks their losses subjected to the **Normalization Rule**: which dictates that the maximum value of the individual losses is exactly. Afterwards, the agent collects the profit according to the **Zero-Sum Rule**: which dictates that the profit is equal to the expected loss of the adversaries with respect to the agent's binding strategy. The objective of the agent is to realize as early as possible that the ratio of the maximum cumulative loss of adversaries to the cumulative profit of the agent reaches $1+\varepsilon$ for some $\varepsilon \in (0,1]$; the objective of the adversary is exactly the opposite.

Now, we describe the **Exponential Binding Strategies** for the agent, while leaving the strategies for the adversary to specific applications. Suppose at the beginning of a round each adversary e has the cumulative loss $L(e)$. Then, the exponential binding strategy on adversaries sets the probability of each adversary e proportional to $(1+\varepsilon)^{L(e)}$. An implementation of the game playing with this strategy is described as follows. Let $L(e)$ be cumulative loss of each adversary e, which is initially 0. Repeat following rounds for

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

times.

1. **Declaration of binding strategies**: The agent declares the exponential binding strategy on adversaries.
2. **Generation of losses**: The adversaries determines choose their non-negative losses in the present round subjected to the **Normalization Rule** and update $L(e)$ for each adversary e.

Since

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

The total number of rounds t is $O\left(\varepsilon^{-2} m \ln m\right)$. The correctness of above implementation of the game is asserted in the theorem below.

Theorem 2.1 *At the end of the game, the maximum cumulative loss of the adversaries is at most $1+\varepsilon$ times the cumulative profit of the agent.*

Proof. For each round $r \in [t]$ and each $e \in E$, $L_r(e)$ denotes his cumulative loss at the end of round r; in addition, let $L_0(e) = 0$ for each $e \in E$. For each round $r \in [t]$, ℓ_r denotes the cumulative profit of the agent; loss at the end of round r; in addition, in addition, let $\ell_0 = 0$. For each $0 \le r \le t$ and each $e \in E$, let

$$y_r(e) = (1+\varepsilon)^{L_r(e)}.$$

By the **Zero-Sum Rule**, for each round $r \in [t]$,

$$\ell_r - \ell_{r-1} = \sum_{e \in E} \frac{y_{r-1}(e)}{y_{r-1}(E)} \left(L_r(e) - L_{r-1}(e)\right).$$

Using the inequalities

$$(1+\varepsilon)^x \le 1 + \varepsilon x \le \exp(\varepsilon x)$$

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

$$\begin{aligned}
y_r(E) &= \sum_{e \in E} y_r(e) \\
&= \sum_{e \in E} y_{r-1}(e)(1+\varepsilon)^{L_r(e)-L_{r-1}(e)} \\
&\le \sum_{e \in E} y_{r-1}(e)\left(1+\varepsilon\left(L_r(e)-L_{r-1}(e)\right)\right) \\
&= \sum_{e \in E} y_{r-1}(e) + \varepsilon \sum_{e \in E} y_{r-1}(e)\left(L_r(e)-L_{r-1}(e)\right) \\
&= y_{r-1}(E)\left(1+\varepsilon \sum_{e \in E} \frac{y_{r-1}(e)}{y_{r-1}(E)}\left(L_r(e)-L_{r-1}(e)\right)\right) \\
&= y_{r-1}(E)\left(1+\varepsilon\left(\ell_r - \ell_{r-1}\right)\right) \\
&\le y_{r-1}(E)\exp\left(\varepsilon\left(\ell_r - \ell_{r-1}\right)\right).
\end{aligned}$$

So,

$$y_t(E) \le y_0(E)\prod_{r=1}^{t}\exp\left(\varepsilon\left(\ell_r - \ell_{r-1}\right)\right) = m\exp\left(\varepsilon \ell_t\right).$$

Therefore, for each $e \in E$ we have

$$\varepsilon \ell_t \ge \ln\frac{y_t(E)}{m} \ge \ln\frac{y_t(e)}{m} = L_t(e)\ln(1+\varepsilon) - \ln m.$$

Let e be an adversary which incurs the largest cumulative loss. By the **Normalization Rule**, the total losses incurred by all adversaries in each round is at least one. Hence, at the end of the

the last round, the total cumulative losses incurred by all adversaries is at least t. Consequently,

$$L_t(e) \geq \frac{t}{m} \geq \frac{\ln m}{\ln(1+\varepsilon) - \frac{\varepsilon}{1+\varepsilon}}.$$

Hence,

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

Thus,

$$\varepsilon \ell_t \geq L_t(e)\ln(1+\varepsilon) - \ln m \geq \frac{\varepsilon}{1+\varepsilon} L_t(e)$$

which implies

$$L_t(e) \leq (1+\varepsilon)\,\ell_t.$$

So, the theorem holds. ■

3 Approximate Weak Dualities

Consider an orientation D of G with ILIN μ. Suppose that y is positive function on A. Let $\widehat{y}$ be the function on A defined by

$$\widehat{y}(a) = \frac{y\left(N_D^{out}[a]\right)}{\beta(a)}$$

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

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

The following invariant property holds.

Lemma 3.1 *For any-path flow decomposition x of a multiflow f,*

$$\sum_{j\in[k]} \sum_{P\in\mathcal{P}_j} x(P)\,\widehat{y}(P) = \sum_{a\in A} y(a)\, d_f\left(N_D^{in}[a]\right)$$

Proof. Since

$$
\begin{aligned}
&\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\,\widehat{y}(P)\\
&=\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\sum_{b\in P}\widehat{y}(b)\\
&=\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\sum_{b\in P}\frac{y\left(N_D^{out}[b]\right)}{\beta(b)}\\
&=\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\sum_{b\in A}|\{b\}\cap P|\,\frac{1}{\beta(b)}y\left(N_D^{out}[b]\right)\\
&=\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\sum_{b\in A}|\{b\}\cap P|\,\frac{1}{\beta(b)}\sum_{a\in N_D^{out}[b]}y(a)\\
&=\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\sum_{b\in A}|\{b\}\cap P|\,\frac{1}{\beta(b)}\sum_{a\in A}y(a)\left|\{b\}\cap N_D^{in}[a]\right|\\
&=\sum_{a\in A}y(a)\sum_{b\in A}\left|\{b\}\cap N_D^{in}[a]\right|\frac{1}{\beta(b)}\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j}|\{b\}\cap P|\,x(P)\\
&=\sum_{a\in A}y(a)\sum_{b\in A}\left|\{b\}\cap N_D^{in}[a]\right|\frac{1}{\beta(b)}\sum_{j\in[k]}f_j(b)\\
&=\sum_{a\in A}y(a)\sum_{b\in A}\left|\{b\}\cap N_D^{in}[a]\right|d_f(b)\\
&=\sum_{a\in A}y(a)\sum_{b\in N_D^{in}[a]}d_f(b)\\
&=\sum_{a\in A}y(a)\,d_f\left(N_D^{in}[a]\right),
\end{aligned}
$$

the lemma holds. ■

For each $j\in[k]$, let $dist_j(\widehat{y})$ be the length of a shortest j-path with respect to $\widehat{y}$. A multiflow f is said to be feasible if d_f lies the capacity region of the network.

Lemma 3.2 *For any feasible multiflow $f=(f_1,\cdots,f_k)$,*

$$\sum_{j\in[k]} dist_j(\widehat{y})\,val(f_j)\le \mu y(A).$$

Proof. Let x be a path-flow decomposition of f. On one hand,

$$\begin{aligned}
&\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\,\widehat{y}(P)\\
&\geq \sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\,dist_j(\widehat{y})\\
&= \sum_{j\in[k]} dist_j(\widehat{y})\sum_{P\in\mathcal{P}_j} x(P)\\
&= \sum_{j\in[k]} dist_j(\widehat{y})\,val(f_j).
\end{aligned}$$

On the other hand, by Lemma 3.1,

$$\begin{aligned}
&\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\,\widehat{y}(P)\\
&= \sum_{a\in A} y(a)\,d_f\left(N_D^{in}[a]\right)\\
&\leq \Delta_D^{in}(d_f)\sum_{b\in A} y(a)\\
&\leq \mu y(A).
\end{aligned}$$

Thus, the lemma holds. ■

The problem **MWMF** has the following weak duality.

Theorem 3.3 *The weight of the maximum weighted multiflow is at most*

$$\mu y(A)\max_{j\in[k]}\frac{w_j}{dist_j(\widehat{y})}.$$

Proof. Let $f=(f_1,\cdots,f_k)$ be a a maximum weighted multiflow. By Lemma 3.2,

$$\begin{aligned}
\mu y(A) &\geq \sum_{j\in[k]} dist_j(\widehat{y})\,val(f_j)\\
&= \sum_{j\in[k]}\frac{dist_j(\widehat{y})}{w_j}val(f_j)\,w_j\\
&\geq \left(\min_{j\in[k]}\frac{dist_j(\widehat{y})}{w_j}\right)\sum_{j\in[k]} val(f_j)\,w_j,
\end{aligned}$$

which implies

$$\sum_{j\in[k]} val(f_j)\,w_j \leq \mu\frac{y(A)}{\min_{j\in[k]}\frac{dist_j(\widehat{y})}{w_j}} = \mu y(A)\max_{j\in[k]}\frac{w_j}{dist_j(\widehat{y})}.$$

Thus, the theorem holds. ■

The problem **MCMF** has the following weak duality.

Theorem 3.4 *The concurrency of the maximum concurrent multiflow is at most*

$$\mu \frac{y(A)}{\sum_{j\in[k]} w_j dist_j(\widehat{y})}.$$

Proof. Let $f = (f_1, \cdots, f_k)$ be a maximum concurrent multiflow. By Lemma 3.2,

$$\begin{aligned}
\mu y(A) &\geq \sum_{j\in[k]} dist_j(\widehat{y})\, val(f_j) \\
&= \sum_{j\in[k]} w_j dist_j(\widehat{y}) \frac{val(f_j)}{w_j} \\
&\geq \left(\min_{j\in[k]} \frac{val(f_j)}{w_j}\right) \sum_{j\in[k]} w_j dist_j(\widehat{y}),
\end{aligned}$$

which implies

$$\min_{j\in[k]} \frac{val(f_j)}{w_j} \leq \mu \frac{y(A)}{\sum_{j\in[k]} w_j dist_j(\widehat{y})}.$$

Thus, the theorem holds. ■

4 Flow Augmentation Methods

In this section, we develop purely combinatorial approximation algorithms for **MWMF** and **MCMF** in wireless networks using faster and simpler flow augmentation methods. We fix an orientation D of G with ILIN μ. The LP-based approach achieves an approximation bound 2μ in general and μ if D is acyclic. In contrast, for any $\varepsilon \in (0, 1]$ the flow augmentation method to be presented in this section achieves an approximation bound no more than $1+\varepsilon$ times those achieved by the LP-based approach in a running time growing with $1/\varepsilon$ in at most the square order. It thus provides a quantized trade-off between accuracy in terms the approximation bound and efficiency in terms of the running time.

The flow augmentation method for all multiflow problems in wireless networks runs in three stages:

- **Flow Augmentation Stage**: This stage computes a multiflow f from scratch with successive flow augmentations. Each augmenting flow is transported along the shortest paths of some requests.
- **Link-Scheduling Stage**: This stage computes a link schedule $\mathcal{S}$ of d_f.
- **Scaling Stage**: This stage scales both f and $\mathcal{S}$ down by a factor of the length of $\mathcal{S}$ and then return them as the final output.

Among the three phases, the **Scaling Stage** is straightforward, and the **Link-Scheduling Stage** can simply apply the greedy algorithm described in the previous chapter to compute a link schedule $\mathcal{S}$ of d_f, which is referred to as the *greedy link schedule* of d_f. On the other hand, despite of the conceptual simplicity the **Flow Augmentation Stage** has to address the following technical issues for an algorithmic implementation with targeted accuracy and efficiency:

- How to quantize the interference-aware link "congestions" and "lengths" with respect to a multiflow?
- Which shortest path(s) among the k shortest paths should be selected?
- How much flow should be routed along those selected shortest paths?
- When the augmentations should terminate?

We elaborate on the solutions to these technical issues subsequently.

In wired networks, the congestion of a link due to a multiflow is simply the total amount of flow through this link, and the length of a link capturing the edge congestion is simply an exponential function of the congestion of this link only. Such perception on link congestion and length has to be fundamentally changed in wireless networks due to the presence of wireless interference. Intuitively speaking, the "congestion" of a link a due to a multiflow should count both the flow amount through the link a itself and the flows through the links *interfering* the link a. Similarly, the "length" of a link a with respect to a multiflow should take into account both the flow amount through the link a itself and the flows through the links *interfered by* the link a. Subsequently, we will give precise quantizations of such interference-aware link congestions and lengths with respect to a multiflow f. The (interference-aware) *congestion* of a link a due to f is defined to be $d_f\left(N_D^{in}[a]\right)$. The (interference-aware) *bottleneck congestion* of f is defined to be the maximum congestion of f on all links, which is exactly $\Delta_D^{in}(d_f)$. Motivated by the agent's exponential binding strategy in the adaptive zero-sum game, the *congestion cost* of a link a due to f is defined to be

$$y_f(a) = (1+\varepsilon)^{d_f\left(N_D^{in}[a]\right)},$$

Motivated by the weak dualities, the (interference-aware) *length* of a link a with respect to f is defined to be

$$\widehat{y}_f(a) = \frac{y_f\left(N_D^{out}[a]\right)}{\beta(a)}.$$

For each $1 \leq j \leq k$, let $dist_j(\widehat{y}_f)$ be the length of the shortest path of the request j with respect to $\widehat{y}_f$.

The design of the **Flow Augmentation Stage** is based on the general framework of an adaptive zero-sum game. The adversaries are the links in A, and each flow-augmementing iteration of the **Flow Augmentation Stage** corresponds to a game round. Accordingly, the number of iterations or the rounds is

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

The agent plays exactly with the strategy described in Section 2. The loss generation strategy of the adversary is coupled with the multiflow augmentation: In each round of the game, the loss of each adversary is its interference-aware congestion due to the augmenting flow multiflow f. Consequently, at the end of the last round the cumulative loss of each link is its interference-aware congestion due to the multiflow f. The paths transporting the augmenting multiflow in each round are chosen according to the weak dualities. Specifically, in each round the adversaries first computes a shortest path P_j of each request $j \in [k]$ with respect to the interference-aware length function y_f. Depending on the variant of the multiflow problems, one or more of these k shortest paths are selected. The amount of flow routed along those selected shortest paths is fully determined by the **Normalization Rule** to ensure that the maximum link loss due to the augmenting flow is exactly one.

As an amalgamation of the above solutions, the two subsections below present the design and analyses of flow augmentation methods for an algorithmic implementation of **MWMF** and **MCMF** respectively.

4.1 Flow Augmentations for MWMF

An implementation of the flow augmentation method for **MWMF** is the algorithm **WMF**(ε) is outlined in Table 1. Each iteration of the **Flow Augmentation Stage** corresponds to a game round. In each round of the game, among all the k shortest paths P_j for $j \in [k]$ with respect to the interference-aware length function y_f, only the path P_j with best cost-efficiency

$$\frac{w_j}{\widehat{y}_f(P_j)} = \frac{w_j}{dist_j(\widehat{y}_f)}$$

is selected for flow augmentation. The value δ of the augmenting multiflow routed along P_j is determined by the **Normalization Rule** as follows. The loss of adversary a, which is the interference-aware congestion of a due to this augmenting multiflow, is equal to

$$\sum_{b \in N_D^{in}[a]} \frac{\delta \left| P_j \cap \{b\} \right|}{\beta(b)} = \delta \sum_{b \in N_D^{in}[a] \cap P_j} \frac{1}{\beta(b)}.$$

The **Normalization Rule** dictates that

$$\delta = \frac{1}{\max\limits_{a \in A} \sum_{b \in N_D^{in}[a] \cap P_j} \frac{1}{\beta(b)}}.$$

This completes the specification of the adversary's strategy on generating losses in each round. Accordingly, each iteration of the **Flow Augmentation Stage** proceeds as follows

- Compute the k shortest paths P_j for $j \in [k]$ with respect to y_f.
- Select the most cost-efficient path P_j among these k shortest paths.
- Compute the flow value δ.
- Augment f_j with a flow of value δ along P_j.

Algorithm **WMF**(ε)
// **Flow Augmentation Stage** $\forall j \in [k], f_j \leftarrow \mathbf{0}$; repeat $\left\lceil \frac{m \ln m}{\ln(1+\varepsilon) - \frac{\varepsilon}{1+\varepsilon}} \right\rceil$ times $\quad \forall j \in [k],\ P_j \leftarrow$ a shortest j-path w.r.t. $\widehat{y}_f$; $\quad j \leftarrow \arg\max_{j \in [k]} \frac{w_j}{\widehat{y}_f(P_j)}$; $\quad \delta \leftarrow \frac{1}{\max\limits_{a \in A} \sum_{b \in N_D^{in}[a] \cap P_j} \frac{1}{\beta(b)}}$; $\quad \forall a \in P_j,\ f_j(a) \leftarrow f_j(a) + \delta$;
// **Link-Scheduling Stage** $\mathcal{S} \leftarrow$ the greedy link schedule of d_f;
// **Scaling Stage** $f \leftarrow \frac{1}{\lVert \mathcal{S} \rVert} f$, $\mathcal{S} \leftarrow \frac{1}{\lVert \mathcal{S} \rVert} \mathcal{S}$; Output f and $\mathcal{S}$.

Table 1: Outline of the algorithm **WMF**(ε).

Next, we derive an approximation bound of the algorithm **WMF**(ε). Let *opt* be the total weight of a maximum-weighted multiflow.

Lemma 4.1 *At the end of the* ***Flow Augmentation Stage****,*

$$\frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\Delta_D^{in}\left(d_f\right)} \geq \frac{opt}{\left(1+\varepsilon\right)\mu}.$$

Proof. We first claim that in each round of the game, the profit of the agent is at most $\frac{\mu}{opt}$ times the weight of the augmenting flow selected in this round. Indeed, consider a particular round and let f be the multiflow at the beginning of this round. Suppose that the augmenting flow in this round routes a flow of value δ along the path P_j of some request j. Then, P_j is a least-priced path with respect to $\widehat{y}_f$. By the **Zero-Sum Rule** and Lemma 3.1, the profit earned by the agent in this round is

$$\frac{\delta}{y_f\left(A\right)}\widehat{y}_f\left(P_j\right) = \frac{\delta w_j}{y_f\left(A\right)}\frac{dist_j\left(\widehat{y}_f\right)}{w_j} \leq \frac{\mu}{opt}w_j\delta.$$

where the last inequality follows from Theorem 3.3. Note that $w_j\delta$ is exactly the weight of the augmenting flow selected in this round. Thus, the claim holds.

The above claim implies that the cumulative profit of the agent at the end of the last round is at most

$$\frac{\mu}{opt}\sum_{j\in[k]} w_j val\left(f_j\right).$$

On the other hand, the maximum cumulative loss of the adversaries at the end of the last round is exactly $\Delta_D^{in}\left(d_f\right)$. By Theorem 2.1,

$$\Delta_D^{in}\left(d_f\right) \leq \left(1+\varepsilon\right)\frac{\mu}{opt}\sum_{j\in[k]} w_j val\left(f_j\right).$$

Thus,

$$\frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\Delta_D^{in}\left(d_f\right)} \geq \frac{opt}{\left(1+\varepsilon\right)\mu}.$$

So, the lemma holds. ■

An approximation bound of the algorithm **MWF**(ε) is asserted in the theorem below.

Theorem 4.2 *The algorithm* ***MWF***(ε) *has an approximation bound* $2\left(1+\varepsilon\right)\mu$ *in general and a* $\left(1+\varepsilon\right)\mu$ *if* D *is acyclic.*

Proof. It is sufficient to show that at the end of the **Link-Scheduling Phase**,

$$\frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\|\mathcal{S}\|} \geq \frac{opt}{2\left(1+\varepsilon\right)\mu},$$

and if D is acyclic, then

$$\frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\|\mathcal{S}\|} \geq \frac{opt}{\left(1+\varepsilon\right)\mu}.$$

Indeed, since

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

Lemma 4.1 implies that

$$\frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\|\mathcal{S}\|} \geq \frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{2\Delta_D^{in}\left(d_f\right)} \geq \frac{opt}{2\left(1+\varepsilon\right)\mu}.$$

If D is acyclic, then

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

and by Lemma 4.1 we have

$$\frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\|\mathcal{S}\|} \geq \frac{\sum_{j\in[k]} w_j val\left(f_j\right)}{\Delta_D^{in}\left(d_f\right)} \geq \frac{opt}{\left(1+\varepsilon\right)\mu}.$$

So, the theorem holds. ■

We conclude this subsection by a remark on an efficient computation of the k shortest paths in each iteration of the **Augmentation Stage**. An attractive feature of Dijkstra's shortest-path algorithm is that it is able to compute all shortest paths from a common node to all other nodes in a graph. Such feature can be been exploited to reduce the number of shortest-path computations. We first group requests by a common source node. The number of groups is at more $\min\{n,k\}$. Then, for each group we compute shortest paths from the common source node to the sinks of all requests in this group by a single call to the Dijkstra's algorithm. As the Dijkstra's algorithm has running time $O\left(m+n\log n\right)$ based on Fibonacci heap, we can compute the k shortest paths

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

time. For $k=\Theta\left(n^2\right)$, this implementation is a linear factor speedup over the naive k separate shortest path computations for the k requests respectively.

4.2 Flow Augmentations for MCMF

An implementation of the flow augmentation method for **MCMF** is the algorithm **CMF**(ε) outlined in Table 1. Each iteration of the **Flow Augmentation Stage** corresponds to a game round. In each round of the game, all the k shortest paths P_j for $j\in[k]$ with respect to the interference-aware length function y_f are selected for flow augmentation. The augmenting multiflow routes a flow of value σw_j along P_j for each $j\in[k]$ for some concurrency σ to be determined by the **Normalization**

Rule as follows. The loss of each adversary a, which is the interference-aware congestion of a due to this augmenting multiflow, is equal to

$$\sum_{b\in N_D^{in}[a]} \frac{\sum_{j\in[k]} \sigma w_j \left|P_j \cap \{b\}\right|}{\beta\left(b\right)} = \sigma \sum_{j\in[k]} w_j \sum_{b\in N_D^{in}[a]\cap P_j} \frac{1}{\beta\left(b\right)}.$$

The **Normalization Rule** dictates that

$$\sigma = \frac{1}{\max\limits_{a\in A} \sum\limits_{j\in[k]} w_j \sum_{b\in N_D^{in}[a]\cap P_j} \frac{1}{\beta(b)}}.$$

This completes the specification of the adversary's strategy on generating losses in each round. Accordingly, In each round of the game, the adversary first computes a shortest path P_j of each request $j \in [k]$ with respect to the interference-aware length function $\widehat{y}_f$.

$$\sigma = \frac{1}{\max\limits_{a\in A} \sum\limits_{j\in[k]} w_j \sum_{b\in N_D^{in}[a]\cap P_j} \frac{1}{\beta(b)}}.$$

For each $j \in [k]$, the flow f_j is updated accordingly. Accordingly, each iteration of the **Flow Augmentation Stage** proceeds as follows

- Compute the k shortest paths P_j for $j \in [k]$ with respect to y_f.
- Compute the multiflow concurrency σ.
- Augment f_j with a flow of value σw_j along P_j for each $j \in [k]$.

In the remaining of this subsection, we derive an approximation bound of the algorithm **WMF**(ε). Let *opt* be the concurrency of a maximum-weighted multiflow.

Lemma 4.3 *At the end of the **Flow Augmentation Stage**,*

$$\frac{\min_{j\in[k]} \frac{val(f_j)}{w_j}}{\Delta_D^{in}\left(d_f\right)} \geq \frac{opt}{\left(1+\varepsilon\right)\mu}.$$

Proof. We first claim that in each round the profit of the agent is at most $\frac{\mu}{opt}$ times the concurrency of the augmenting flow selected in this round. Indeed, consider a particular round and let f be the multiflow at the beginning of this round. Suppose that the augmenting flow in this round routes a flow of value δ along the path P_j of some request j. Then, P_j is a least-priced path

Algorithm **CMF**(ε)
// **Flow Augmentation Stage** $\forall j \in [k], f_j \leftarrow \mathbf{0}$; repeat $\left\lceil \frac{m \ln m}{\ln(1+\varepsilon) - \frac{\varepsilon}{1+\varepsilon}} \right\rceil$ times $\quad \forall j \in [k]$, $P_j \leftarrow$ a shortest j-path w.r.t. $\widehat{y}_f$; $\quad \sigma \leftarrow \frac{1}{\max\limits_{a \in A} \sum\limits_{j \in [k]} w_j \sum_{b \in N_D^{in}[a] \cap P_j} \frac{1}{\beta(b)}}$; $\quad \forall j \in [k]$ and $a \in P_j$, $f_j(a) \leftarrow f_j(a) + \sigma w_j$;
// **Link-Scheduling Stage** $\mathcal{S} \leftarrow$ the greedy link schedule of $\sum_{j \in [k]} f_j$;
// **Scaling Stage** $f \leftarrow \frac{1}{\lVert \mathcal{S} \rVert} f$, $\mathcal{S} \leftarrow \frac{1}{\lVert \mathcal{S} \rVert} \mathcal{S}$; Output f and $\mathcal{S}$.

Table 2: Outline of the algorithm **CMF**(ε).

with respect to $\widehat{y}_f$. By the **Zero-Sum Rule** and Lemma 3.1, the profit earned by the agent in this round is

$$\frac{\sigma}{y_f(A)} \sum_{j \in [k]} d_j \widehat{y}_f(P_j) = \frac{\sigma}{y_f(A)} \sum_{j \in [k]} d_j dist_j(\widehat{y}_f) \leq \frac{\mu}{opt} \sigma.$$

where the last inequality follows from Theorem 3.3. Thus, the claim holds.

The above claim implies that the cumulative profit of the agent at the end of the last round is at most

$$\frac{\mu}{opt} \min_{j \in [k]} \frac{val(f_j)}{w_j},$$

On the other hand, the maximum cumulative loss of the adversaries at the end of the last round is exactly $\Delta_D^{in}(d_f)$. By Theorem 2.1,

$$\Delta_D^{in}(d_f) \leq (1+\varepsilon) \frac{\mu}{opt} \min_{j \in [k]} \frac{val(f_j)}{w_j}$$

and hence

$$\frac{\min_{j \in [k]} \frac{val(f_j)}{w_j}}{\Delta_D^{in}(d_f)} \geq \frac{opt}{(1+\varepsilon)\mu}.$$

So, the lemma holds. ■

An approximation bound of the algorithm **CWF**(ε) is asserted in the theorem below.

Theorem 4.4 *The algorithm **CWF**(ε) has an approximation bound $2(1+\varepsilon)\mu$ in general and a $(1+\varepsilon)\mu$ if D is acyclic.*

The proof of the above theorem is the same as the proof of Theorem 4.2 and is thus omitted. The same remark on efficient computations of the k shortest paths at the end of previous subsection also applies to the algorithm **CMF**(ε).

5 Extensions to Budgeted Multiflows

In this section, we highlight the extensions of the flow augmentation methods to **Budgeted MWMF** and **Budgeted MCMF** and leave the proofs of the lemmas and theorems in this section as exercises. Consider an orientation D of G with ILIN μ. For the simplicity of the presentation, we introduce a "virtual" link a^+ on behalf of the budget constraint which is disjoint from all links in A, and let $A^+ = A \cup \{a^+\}$. Suppose that y is positive function on A^+. Let $\widehat{y}$ be the function on A defined by

$$\widehat{y}(a) = \frac{y\left(N_D^{out}[a]\right)}{\beta(a)} + g(a)\, y\left(a^+\right)$$

for each $a \in A$. The following lemma is an extension to Lemma 3.1.

Lemma 5.1 *For any path-flow decomposition x of a multiflow $f = (f_1, \cdots, f_k)$,*

$$\begin{aligned}&\sum_{j\in[k]}\sum_{P\in\mathcal{P}_j} x(P)\,\widehat{y}(P)\\ &= \sum_{a\in A} y(a)\, d_f\left(N_D^{in}[a]\right) + y\left(a^+\right)\sum_{a\in A} g(a)\,\beta(a)\, d_f(a).\end{aligned}$$

A multiflow f is said to be feasible if d_f lies the capacity region of the network and

$$\sum_{a\in A} g(a)\,\beta(a)\, d_f(a) \le 1.$$

The following lemma is an extension to Lemma 3.2.

Lemma 5.2 *For any feasible multiflow $f = (f_1, \cdots, f_k)$,*

$$\sum_{j\in[k]} dist_j(\widehat{y})\, val(f_j) \le \mu y\left(A^+\right).$$

Using the above lemma, we can establish the following approximate weak dualities of **Budgeted MWMF** and **Budgeted MCMF**.

Theorem 5.3 *The weight of the maximum weighted multiflow is at most*

$$\mu y\left(A^{+}\right)\max_{j\in[k]}\frac{w_j}{dist_j\left(\widehat{y}\right)}.$$

Theorem 5.4 *The concurrency of the maximum concurrent multiflow is at most*

$$\mu\frac{y\left(A^{+}\right)}{\sum_{j\in[k]}w_j dist_j\left(\widehat{y}\right)}.$$

For a wireless network, the (interference-aware) *congestion* of a link $a\in A$ due to f is defined to be $d_f\left(N_D^{in}[a]\right)$, and the *congestion* of a^{+} due to f is defined to be

$$\sum_{a\in A}g\left(a\right)\beta\left(a\right)d_f\left(a\right).$$

The (interference-aware) *bottleneck congestion* of f is defined to be the maximum congestion of f on all links, which is exactly

$$\max\left\{\Delta_D^{in}\left(d_f\right),\sum_{a\in A}g\left(a\right)\beta\left(a\right)d_f\left(a\right)\right\}.$$

The *congestion cost* of a link $a\in A$ due to f is defined to be

$$y_f\left(a\right)=\left(1+\varepsilon\right)^{d_f\left(N_D^{in}[a]\right)},$$

and the *congestion cost* of a^{+} due to f is defined to be

$$y_f\left(a^{+}\right)=\left(1+\varepsilon\right)^{\sum_{a\in A}g(a)\beta(a)d_f(a)}.$$

The (interference/budget-aware) *length* of a link a with respect to f is defined to be

$$\widehat{y}_f\left(a\right)=\frac{y_f\left(N_D^{out}[a]\right)}{\beta\left(a\right)}+g\left(a\right)y_f\left(a^{+}\right).$$

For each $1\leq j\leq k$, let $dist_j\left(\widehat{y}_f\right)$ be the length of the shortest path of the request j with respect to $\widehat{y}_f$.

Let $\varepsilon\in(0,1]$ be a parameter representing a targeted trade-off between accuracy and efficiency. An implementation of the flow augmentation method for **Budgeted MWMF** is the algorithm **BWMF**(ε) outlined in Table 3. Its approximation bound is asserted in the theorem below.

Algorithm **BWMF**(ε)
// **Flow Augmentation Stage** $\forall j \in [k]\,, f_j \leftarrow \mathbf{0}$; repeat $\left\lceil \frac{(m+1)\ln(m+1)}{\ln(1+\varepsilon)-\frac{\varepsilon}{1+\varepsilon}} \right\rceil$ times $\quad \forall j \in [k],\ P_j \leftarrow$ a shortest j-path w.r.t. $\widehat{y}_f$; $\quad j \leftarrow \arg\max_{j\in[k]} \frac{w_j}{\widehat{y}_f(P_j)}$; $\quad \delta \leftarrow \dfrac{1}{\max\left(\max\limits_{a\in A} \sum\limits_{b\in N_D^{in}[a]\cap P_j} \frac{1}{\beta(b)}, g(P_j)\right)}$; $\quad \forall a \in P_j,\ f_j(a) \leftarrow f_j(a) + \delta$;
// **Link-Scheduling Stage** $\mathcal{S} \leftarrow$ the greedy link schedule of d_f;
// **Scaling Stage** $f \leftarrow \frac{1}{\max\left\{\lVert\mathcal{S}\rVert, \sum_{a\in A} g(a)\beta(a)d_f(a)\right\}} f$; $\mathcal{S} \leftarrow \frac{1}{\max\left\{\lVert\mathcal{S}\rVert, \sum_{a\in A} g(a)\beta(a)d_f(a)\right\}} \mathcal{S}$; Output f and $\mathcal{S}$.

Table 3: Outline of the algorithm **BWMF**(ε).

Algorithm **BCMF**(ε)
// **Flow Augmentation Stage** $\forall j \in [k]\,, f_j \leftarrow \mathbf{0}$; repeat $\left\lceil \frac{(m+1)\ln(m+1)}{\ln(1+\varepsilon)-\frac{\varepsilon}{1+\varepsilon}} \right\rceil$ times $\quad \forall j \in [k],\ P_j \leftarrow$ a shortest j-path w.r.t. $\widehat{y}_f$; $\quad \sigma \leftarrow \dfrac{1}{\max\left\{\max\limits_{a\in A} \sum\limits_{j\in[k]} w_j \sum\limits_{b\in N_D^{in}[a]\cap P_j} \frac{1}{\beta(b)}, \sum\limits_{j\in[k]} w_j g(P_j)\right\}}$; $\quad \forall j \in [k]$ and $a \in P_j,\ f_j(a) \leftarrow f_j(a) + \sigma w_j$;
// **Link-Scheduling Stage** $\mathcal{S} \leftarrow$ the greedy link schedule of d_f;
// **Scaling Stage** $f \leftarrow \frac{1}{\max\left\{\lVert\mathcal{S}\rVert, \sum_{a\in A} g(a)\beta(a)d_f(a)\right\}} f$; $\mathcal{S} \leftarrow \frac{1}{\max\left\{\lVert\mathcal{S}\rVert, \sum_{a\in A} g(a)\beta(a)d_f(a)\right\}} \mathcal{S}$; Output f and $\mathcal{S}$.

Table 4: Outline of the algorithm **BCMF**(ε).

Theorem 5.5 *The algorithm* ***BWMF***(ε) *has an approximation bound* $2(1+\varepsilon)\mu$ *in general and a* $(1+\varepsilon)\mu$ *if* D *is acyclic.*

An implementation of the flow augmentation method for **Budgeted MWMF** is the algorithm **BWMF**(ε) outlined in Table 3. Its approximation bound is asserted in the theorem below.

Theorem 5.6 *The algorithm* ***BCWF***(ε) *has an approximation bound* $2(1+\varepsilon)\mu$ *in general and a* $(1+\varepsilon)\mu$ *if* D *is acyclic.*

6 Chapter Notes

The NP-hardness of both **MWMF** and **MCMF** under the protocol interference model even when all nodes have uniform (and fixed) communication radii and uniform (and fixed) interference radii and the positions of all nodes are available is proved in [1]. The polynomial-time approximation-preserving reductions from **MWMF** and **MCMF** to **MWISL** and polynomial capacity subregions were also given [1]. The flow augmentation methods for **MWMF** and **MCMF** were developed in [2].

References

[1] P.-J. Wan, Multiflows in Multihop Wireless Networks, *ACM MOBIHOC* 2009, pp. 85-94.

[2] 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.