Maximum-Weighted Subset of Communication Requests Schedulable without Spectral Splitting

Peng-Jun Wan*†, Huaqiang Yuan*, Xiaohua Jia‡, Jiliang Wang§, and Zhu Wang¶
* School of Computer Science, Dongguan University of Technology, P. R. China
† Department of Computer Science, Illinois Institute of Technology
‡ Department of Computer Science, City University of Hong Kong, Hong Kong
§ School of Software, Tsinghua University, P. R. China
¶ Department of Mathematics, Computer Science, And Statistics, State University of New York at Oneonta

***Abstract*—Consider a set of point-to-point communication requests in a multi-channel multihop wireless network, each of which is associated with a traffic demand of at most one unit of transmission time, and a weight representing the utility if its demand is fully met. A subset of requests is said to be schedulable without spectral splitting if they can be scheduled within one unit of time subject to the constraint each request is assigned with a unique channel throughout its transmission. This paper develops efficient and provably good approximation algorithms for finding a maximum-weighted subset of communication requests schedulable without spectral splitting.**

I. Introduction

Two types of conflicts are present in the transmissions of point-to-point communication requests in a wireless network under protocol interference model. The *primary conflict* occurs between two communication sharing a common endpoint, and the *secondary conflict* occurs between two node-disjoint requests which are too spatially close to be able to transmit successfully at the same time over the same channel due to wireless interference. With the deployment of multiple channels, any pair of the communication requests with primary conflict are still prohibited to transmit concurrently regardless of the number of available channels, but a pair of communication requests with secondary conflict may now transmit concurrently by transmitting over different channels. Thus, the availability of multiple channels can effectively mitigate the secondary conflicts but is not beneficial to the primary conflicts. The extent to which the secondary conflicts can be mitigated depends heavily on the channel assignment to the communication requests.

The channel assignment can further be broaderly classified into two categories, with or without spectral splitting. With spectral splitting, a request is allowed to hop from one channel to the other throughout its transmission. Without spectral splitting, the transmission of each request has to stay over one channel throughout its transmission. The spectral splittability affects both the minimum schedule length and the maximum number of requests schedulable within a fixed period. Clearly, spectral non-splitting performs no better than spectral splitting. The following two instances quantitatively illustrate the performance disadvantage of spectral non-splitting in minimum schedule length and the maximum number of requests schedulable within a unit of time respectively.

- Minimum schedule length: Consider an instance of $\lambda > 1$ channels and $\lambda + 1$ node-disjoint but mutually conflicting requests each of which has traffic demand of transmission time $\frac{\lambda}{\lambda+1}$. Without spectral splitting, at least one channel is assigned by two requests by the pigeonhole principle, and hence the minimum schedule length is at least $\frac{2\lambda}{\lambda+1}$; and such length can be achieved trivially by assigning one channel to two requests and each other channels to exactly one request. With splitting overs channels, the minimum schedule length is at least
$$\frac{(\lambda+1)\frac{\lambda}{\lambda+1}}{\lambda} = 1;$$
and such length can be achieved by first splitting the traffic demand of one request into λ equal pieces and scheduling these pieces one after another over the λ channels respectively, and then filling the left vacancy of the λ channels by the remaining λ requests respectively. Thus, the ratio of the minimum schedule length without spectral splitting to the minimum schedule length with spectral splitting is
$$\frac{2\lambda}{\lambda+1} = 2 - \frac{2}{\lambda+1}.$$
- Maximum number of requests schedulable within a unit of time: Consider an instance of $\lambda > 1$ channels and $2\lambda - 1$ node-disjoint but mutually conflicting requests each of which has traffic demand of transmission time $\frac{\lambda}{2\lambda-1}$. Without spectral splitting over channels, each channel can accommodate one request within a unit of time, and hence the maximum number of schedulable requests within a unit of time is λ trivially. With splitting over channels, all the requests are schedulable within a unit of time according to the McNaughton's Wrap-Around Rule [7]: Pick an arbitrary order of the requests, and fill the channels successively with the requests in this order. Whenever the unit time bound is met, split the present request into two parts and schedule the second part of the split request on the next channel at zero time. Thus for this instance, the ratio of the maximum number of

feasible requests with spectral splitting to the maximum number of feasible requests without spectral splitting is

$$\frac{2\lambda-1}{\lambda}=2-\frac{1}{\lambda}.$$

On the other hand, the spectral non-splitting is advantageous in implementation simplicity. With spectral splitting, a communication request may split its traffic demand over multiple channels and hop frequently from one channel to another throughout the schedule. Such frequent channel-hopping would require extensive coordination between the senders and receivers of the request and may be time-consuming in practice. In addition, the intractability of finding a shortest transmission schedule of an arbitrary instance with spectral splitting [8] makes it impossible to fully exploit the power of spectral splitting by practical scheduling algorithms such as the greedy transmission scheduling [10]. In fact, a surprising discovery of this paper is that the best-known upper bound [10] on the length of the greedy transmission schedule with spectral splitting can be also matched by yet another greedy transmission schedule without spectral splitting.

Motivated by the above observations, this paper studies the (spectral) non-splitting variant of the problem **Maximum-Weighted Feasible Set** (**MWFS**) [9] in multihop multi-channel wireless networks. Consider a set of communication requests each of which is specified by the sender-receiver pair, the traffic demand in terms of transmission time which is at most one unit of time, and the weight representing the utility if the demand is fully met. A subset of them is said to be *feasible* without (respectively, with) spectral splitting if they can be scheduled within one unit of time but without (respectively, with) spectral splitting. The spectral non-splitting (respectively, splitting) variant of the problem **MWFS** seeks a subset of the given requests with maximum total weight which can be scheduled within one unit of time but without (respectively, splitting) spectral splitting. The non-splitting variant of **MWFS** has more stringent constraint on the channel assignment than the splitting variant of **MWFS**. When restricted to the instances of mutually disjoint and conflicting request, the schedulability test without splitting is *strictly harder* than the schedulability test with splitting: The former can be reduced from the NP-complete **Partition** problem [6] and hence is NP-complete itself; the latter can be solved in polynomial time by applying the McNaughton's Wrap-Around Rule [7].

To the best of our knowledge, the non-splitting variant of **MWFS** has not been studied before. It involves jointly selection of requests, non-splitting channel assignment to the selected requests, and the temporal transmission scheduling of the selected requests over the assigned channels. The main objective of this paper is to develop efficient and provably good approximation algorithms for the non-splitting variant of **MWFS**. Compared to the splitting variant [9], we are able to achieve better approximation bounds by a refined treatment of primary conflicts and secondary conflicts in both the design and analyses of our proposed algorithms.

The following notations and terms are adopted throughout this paper. Let λ be the number of available channels, and A be a set of point-to-point communication tasks. Each request a has a demand $d(a)\in(0,1]$ of transmission time and a positive weight $w(a)$ of utility. The weight of any subset B of A is

$$w(B):=\sum_{b\in B}w(b).$$

A request is said to be *light* if its demand is at most $1/2$, and *heavy* otherwise. Let $\prec$ be an ordering of A. For any $a,b\in A$, both $a\prec b$ and $b\succ a$ represent that a appears before b in the ordering $\prec$. For any $a\in A$ and any $B\subseteq A$, we use $B_{\prec a}$ (respectively, $B_{\succ a}$) to denote the set of $b\in B$ satisfying that $b\prec a$ (respectively, $b\succ a$); in addition, $B_{\preceq a}$ denotes $\{a\}\cup B_{\prec a}$, and $B_{\succeq a}$ denotes $\{a\}\cup B_{\succ a}$.

The remainder of this paper is organized as follows. In Section II we introduce a finer treatment of the conflicts. In Section III, we define a special ordering of the requests. In Section V, we present an approximation algorithm for the non-splitting variant of **MWFS** applicable to the case that all requests in A are light. In Section IV, we develop an approximation algorithm for the non-splitting variant of **MWFS** applicable to the case that all requests in A are heavy. In Section VI, we give an approximation algorithms for non-splitting variant **MWFS** applicable to the general case that A consists of both light requests and heavy requests. We conclude this paper in Section VII.

II. A Finer Treatment of Conflicts

Under the protocol interference model, each request a is associated with an interference range. A request a is said to interfere a request b if the receiver of b lies within the interference range of a. Consider a request a. The set of requests which interfere (respectively, are interfered by) a is denoted by $N^{in}(a)$ (respectively, $N^{out}(a)$), and the union of $N^{in}(a)$ and $N^{out}(a)$ is denoted by $N(a)$. Clearly, $N(a)$ consists of all requests which have conflict with a; $N^{in}(a)$ and $N^{out}(a)$ may have an non-empty intersection, and hence they may not form a partition of $N(a)$. Let $\dot{N}(a)$ (respectively, $\ddot{N}(a)$) denote the set of requests which have primary (respectively, secondary) conflict with a. Then, $\dot{N}(a)$ and $\ddot{N}(a)$ always form a partition of $N(a)$. Let $\dot{N}[a]$ denotes $\dot{N}(a)\cup\{a\}$. Define

$$\ddot{N}^{in}(a)=\ddot{N}(a)\cap N^{in}(a),$$
$$\ddot{N}^{out}(a)=\ddot{N}(a)\cap N^{out}(a).$$

The *inward local independence number* (ILIN) of A, denoted by μ, is defined to be the maximum number of pairwise conflict-free requests in $N^{in}(a)$ for all $a\in A$. In the plane geometric variant [8] of A, the ILIN is bounded by a constant. A set C of requests in A are said to be *compatible* if all requests in C can transmit successfully at the same time with some channel assignment. Denote

$$\mu_\lambda=\mu+\left(1-\frac{1}{\lambda}\right).$$

Then, the compatible subsets have the following local property.

Lemma 2.1: For any compatible $C \subseteq A$ and any $a \in A$,

$$\left| C \cap \dot{N}^{in}[a] \right| + \frac{2}{\lambda} \left| C \cap \ddot{N}^{in}(a) \right| \leq 2\mu_{\lambda}.$$

Proof. Consider an arbitrary channel assignment π to C under which all requests in C can transmit successfully at the same time.

Case 1: $a \in C$. Then, $C \cap \dot{N}^{in}[a] = \{a\}$ and hence $\left| C \cap \dot{N}^{in}[a] \right| = 1$. In addition, no request in $C \cap \ddot{N}^{in}(a)$ is assigned with the channel $\pi(a)$, and at most μ requests in $C \cap \ddot{N}^{in}(a)$ are assigned with each channel other than $\pi(a)$; thus

$$\left| C \cap \ddot{N}^{in}(a) \right| \leq (\lambda - 1)\mu.$$

So,

$$\begin{aligned} &\left| C \cap \dot{N}^{in}[a] \right| + \frac{2}{\lambda} \left| C \cap \ddot{N}^{in}(a) \right| \\ &\leq 1 + \frac{2}{\lambda}(\lambda - 1)\mu = 2\mu + 1 - \frac{2}{\lambda}\mu \\ &\leq 2\mu + 2\left(1 - \frac{1}{\lambda}\right) = 2\mu_{\lambda}. \end{aligned}$$

Case 2: $a \notin C$. Then,

$$\left| C \cap \dot{N}^{in}[a] \right| = \left| C \cap \dot{N}^{in}(a) \right| \leq 2.$$

We further consider three subcases.

Subcase 2.1: $\left| C \cap \dot{N}^{in}(a) \right| \leq 1$. Since at most μ requests in $C \cap \ddot{N}^{in}(a)$ are assigned with each channel, we have

$$\left| C \cap \ddot{N}^{in}(a) \right| \leq \lambda\mu.$$

So,

$$\begin{aligned} &\left| C \cap \dot{N}^{in}[a] \right| + \frac{2}{\lambda} \left| C \cap \ddot{N}^{in}(a) \right| \\ &\leq 1 + \frac{2}{\lambda}\lambda\mu = 2\mu + 1 \leq 2\mu + 2\left(1 - \frac{1}{\lambda}\right) = 2\mu_{\lambda}. \end{aligned}$$

Subcase 2.2: $\left| C \cap \dot{N}^{in}(a) \right| = 2$. Then, exactly one request $b \in C \cap \dot{N}^{in}(a)$ has the receiver of a as one of its endpoint. Since the receiver of a must lie within the interference range of b, at most $\mu - 1$ requests in $C \cap \ddot{N}^{in}(a)$ are assigned with the channel $\pi(a)$. As at most μ requests in $C \cap \ddot{N}^{in}(a)$ are assigned with each channel other than $\pi(a)$, we have

$$\left| C \cap \ddot{N}^{in}(a) \right| \leq (\mu - 1) + (\lambda - 1)\mu = \lambda\mu - 1.$$

So,

$$\begin{aligned} &\left| C \cap \dot{N}^{in}[a] \right| + \frac{2}{\lambda} \left| C \cap \ddot{N}^{in}(a) \right| \\ &\leq 2 + \frac{2}{\lambda}(\lambda\mu - 1) = 2\mu + 2\left(1 - \frac{1}{\lambda}\right) = 2\mu_{\lambda}. \end{aligned}$$

In all cases, the lemma holds. ■

More generally, the feasible subsets have the following local property.

Lemma 2.2: For any feasible subset F and any $a \in A$,

$$d\left(\dot{N}[a] \cap F\right) + \frac{2}{\lambda} d\left(\ddot{N}^{in}(a) \cap F\right) \leq 2\mu_{\lambda}.$$

Proof. Consider a shortest schedule $\mathcal{S}$ of F. Let $\mathcal{C}$ be the collection of compatible sets of requests in F transmitting concurrently in $\mathcal{S}$. For each $C \in \mathcal{C}$, let $l(C)$ be the total transmission time by C. Then,

$$\textstyle\sum_{C \in \mathcal{C}} l(C) \leq 1.$$

and for each $b \in F$,

$$d(b) = \textstyle\sum_{C \in \mathcal{C}} l(C) \left| C \cap \{b\} \right|.$$

Thus,

$$\begin{aligned} &d\left(\dot{N}[a] \cap F\right) + \frac{2}{\lambda} d\left(\ddot{N}^{in}(a) \cap F\right) \\ &= \textstyle\sum_{C \in \mathcal{C}} l(C) \left| \dot{N}[a] \cap C \right| + \frac{2}{\lambda} \textstyle\sum_{C \in \mathcal{C}} l(C) \left| \ddot{N}(a) \cap C \right| \\ &= \textstyle\sum_{C \in \mathcal{C}} l(C) \left(\left| \dot{N}[a] \cap C \right| + \frac{2}{\lambda} \left| \ddot{N}(a) \cap C \right| \right) \\ &\leq 2\mu_{\lambda} \textstyle\sum_{C \in \mathcal{C}} l(C) \\ &\leq 2\mu_{\lambda}. \end{aligned}$$

So, the lemma holds. ■

III. Surplus-Preserving Ordering

Consider a non-negative function x on A. An ordering $\prec$ of A is said to be *surplus-preserving* with respect to x if for each $a \in A$,

$$x\left(\ddot{N}^{in}(a) \cap A_{\prec a}\right) \geq x\left(\ddot{N}^{out}(a) \cap A_{\prec a}\right).$$

An essential property of a surplus-preserving ordering $\prec$ with respect to x is that for each $a \in A$,

$$\begin{aligned} &x\left(\ddot{N}(a) \cap A_{\prec a}\right) \\ &\leq x\left(\ddot{N}^{in}(a) \cap A_{\prec a}\right) + x\left(\ddot{N}^{out}(a) \cap A_{\prec a}\right) \\ &= 2x\left(\ddot{N}^{in}(a) \cap A_{\prec a}\right) \\ &\leq 2x\left(\ddot{N}^{in}(a)\right). \end{aligned}$$

The existence of a surplus-preserving ordering with respect to x is implied by the following fact: For any $B \subseteq A$, a link $a \in B$ is said to be a *x-surplus request* in B if

$$x\left(\ddot{N}^{in}(a) \cap B\right) \geq x\left(\ddot{N}^{out}(a) \cap B\right).$$

It was proved in [8] that any non-empty subset B of A has at least one x-surplus request. Based on this fact, a surplus-preserving ordering of A can be computed by a simple *greedy* strategy as follows.

- Initialize B to A.
- For $i = |A|$ down to 1, let a_i be an x-surplus request in B and delete a_i from B.

Then, the ordering $\langle a_1, a_2, \cdots, a_{|A|} \rangle$ is surplus-preserving.

IV. HEAVY REQUESTS

In this section, we assume all requests in A are heavy, i.e. $d(a) \in (1/2, 1]$ for each $a \in A$. For heavy requests, the feasibility has a much simplified characterization.

Theorem 4.1: A set F of heavy requests is feasible if and only if F is compatible.

Proof. The sufficient part of this equivalence is trivial, and the necessary part of this equivalence is argued as follows. Suppose that F is feasible. Since any pairs of requests in A with primary conflict cannot transmit at the same time and their demands sum up more than one, F contains no pair of requests with primary conflict. Let π be a channel assignment to F under which there is a scheduling of F with length is at most one. Then, no pair of requests in F with secondary conflict in F have the same channel under π for otherwise their demands sum up more than one and their transmission cannot completed within one time unit. This means with the channel assignment π all requests over each channel are conflict-free. Thus, all requests in F can transmit successfully at the same time under the channel assignment π. So, F is compatible. ■

The above equivalence between feasibility and compatibility effectively eliminates the need to handling the specific demands as we only have to select a maximum-weighted compatible subset. Given a set F of compatible requests with a compatible channel assignment π, the transmission scheduling is trivial. Each request a simply transmits over channel $\pi(a)$ for a time period of $d(a)$ starting from the time zero. The length of this schedule is $\max_{a \in F} d(a)$, which is at most one. However, the test of compatibility is still NP-complete. So, we further introduce a tractable and strong type of compatibility below. For better understanding of the proposed algorithm, we shall explore the rationale of the algorithm design progressively in the *reverse* order of the algorithm execution. The description and analysis of the full algorithm will be put forward at the end of this section.

A. Inductive Compatibility

Consider a subset F of requests. Given an ordering $\prec$ of F, F is said to be *inductively compatible* in $\prec$ if for each $a \in F$, $F_{\prec a}$ contains no primary neighbor of a and less than λ secondary neighbors of a. Equivalently, F is inductively compatible in $\prec$ if and only if for each $a \in F$,

$$\left|\dot{N}(a) \cap F_{\prec a}\right| + \frac{1}{\lambda}\left|\ddot{N}(a) \cap F_{\prec a}\right| < 1.$$

An inductively compatible set F in $\prec$ is always compatible; indeed, a compatible channel assignment to F can be *greedily* produced as follows:

- The first task a in F receives the first channel.
- For each subsequent task $a \in F$ in the ordering $\prec$, it receives the first channel which is not used by any (preceding) secondary neighbor of a in F.

Such channel assignment is referred to as the *greedy channel assignment* to F in $\prec$.

Given an ordering $\prec$ of A and a subset $S \subseteq A$, an inductively feasible subset F of S can be computed in a *greedy* manner as follows:

- Initially, F is empty.
- For each $a \in S$ in the ordering $\prec$, a is added to F if and only if

$$\left|\dot{N}(a) \cap F_{\prec a}\right| + \frac{1}{\lambda}\left|\ddot{N}(a) \cap F_{\prec a}\right| \leq 1.$$

The final set F computed in this greedy manner is referred to as the *maximal inductively compatible subset* of S in $\prec$. It is maximal in the sense that for each $a \in S \setminus F$,

$$\left|\dot{N}(a) \cap F_{\prec a}\right| + \frac{1}{\lambda}\left|\ddot{N}(a) \cap F_{\prec a}\right| \geq 1.$$

However, how to select the "candidate" subset S is very essential. In fact, when $S = A$ or $S = \emptyset$ at the two opposite extremes, the maximal inductively compatible subset of S in $\prec$ is almost surely to have poor performance in general. This motivates us to utilize in the next subsection variants of the local-ratio scheme [1], [2], [3], [5], [11], [13], which is equivalent to the primal-dual scheme [4], for selecting candidate set S properly.

B. Selection of Candidates

Given an ordering $\prec$ of A, the local-ratio scheme computes a candidate set S in the following *greedy* manner:

- S is initially empty.
- For each $a \in A$ in the *reverse* order of $\prec$, a *discounted* weight $\overline{w}(a)$ of a is computed by

$$\overline{w}(a) = w(a) - \overline{w}\left(\dot{N}(a) \cap S\right) - \frac{1}{\lambda}\overline{w}\left(\ddot{N}(a) \cap S\right);$$

  and if $\overline{w}(a) > 0$, a is added to S.

The final set S computed in this greedy manner is referred to as the *greedy candidate subset* in $\prec$. For each $a \in A$, the following relations between its original weight and its discounted weight has the following relation:

$$w(a) = \overline{w}(a) + \overline{w}\left(\dot{N}(a) \cap S_{\succ a}\right) + \frac{1}{\lambda}\overline{w}\left(\ddot{N}(a) \cap S_{\succ a}\right)$$

In addition, the maximal inductively compatible subset F of S in $\prec$ has the following property.

Lemma 4.2: The maximal inductively compatible subset F of S in $\prec$ satisfies that $w(F) \geq \overline{w}(S)$.

Proof. By the greedy selection of F, for each $a \in S \setminus F$,

$$\left|\dot{N}(a) \cap S\right| + \frac{1}{\lambda}\left|\ddot{N}(a) \cap S\right| \geq 1.$$

Using the relations between the original weights and the discounted weights, we have

$$\begin{aligned}
&w(F)\\
&=\overline{w}(F)+\sum_{a\in S}\overline{w}(a)\left[\left|\dot{N}(a)\cap F_{\prec a}\right|+\frac{1}{\lambda}\left|\ddot{N}(a)\cap F_{\prec a}\right|\right]\\
&\geq\overline{w}(F)+\sum_{a\in S\setminus F}\overline{w}(a)\left[\left|\dot{N}(a)\cap F_{\prec a}\right|+\frac{1}{\lambda}\left|\ddot{N}(a)\cap F_{\prec a}\right|\right]\\
&\geq\overline{w}(F)+\sum_{a\in S\setminus F}\overline{w}(a)\\
&=\overline{w}(F)+\overline{w}(S\setminus F)\\
&=\overline{w}(S).
\end{aligned}$$

So, the claim holds. ■

By now, all the design issues boils down to the choice of proper ordering $\prec$ such that the total discounted weight of the greedy candidate set in $\prec$ is sufficiently large. This last but most critical design issue will be explored in the next subsection.

C. Selection of Ordering

Our choice of request ordering is motivated by the following property of surplus-preserving orderings.

Lemma 4.3: Suppose that $\prec$ is a surplus-preserving ordering of A with respect to some non-zero $x\in\mathbb{R}_+^A$. Then, the greedy candidate subset S in $\prec$ satisfies that

$$\overline{w}(S)\geq\frac{\sum_{a\in A}w(a)\,x(a)}{\max_{a\in A}\left[x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)\right]}.$$

Proof. We first show that

$$\sum_{a\in A}x(a)\,\overline{w}\left(\dot{N}(a)\cap S_{\succ a}\right)\leq\sum_{a\in S}\overline{w}(a)\,x\left(\dot{N}(a)\right).$$

Note that for any $a\in A$ and $b\in S$,

$$b\in\dot{N}(a)\cap A_{\succ a}\Leftrightarrow a\in\dot{N}(b)\cap A_{\prec b};$$

Thus,

$$\begin{aligned}
&\sum_{a\in A}x(a)\,\overline{w}\left(\dot{N}(a)\cap S_{\succ a}\right)\\
&=\sum_{a\in A}x(a)\sum_{b\in\dot{N}(a)\cap S_{\succ a}}\overline{w}(b)\\
&=\sum_{b\in S}\overline{w}(b)\sum_{a\in\dot{N}(b)\cap A_{\prec b}}x(a)\\
&=\sum_{b\in S}\overline{w}(b)\,x\left(\dot{N}(b)\cap A_{\prec b}\right)\\
&=\sum_{a\in S}\overline{w}(a)\,x\left(\dot{N}(a)\cap A_{\prec a}\right)\\
&\leq\sum_{a\in S}\overline{w}(a)\,x\left(\dot{N}(a)\right).
\end{aligned}$$

Next, we show that

$$\sum_{a\in A}x(a)\,\overline{w}\left(\ddot{N}(a)\cap S_{\succ a}\right)\leq2\sum_{a\in S}\overline{w}(a)\,x\left(\ddot{N}(a)\right).$$

Note that for any $a\in A$ and $b\in S$,

$$b\in\ddot{N}(a)\cap A_{\succ a}\Leftrightarrow a\in\ddot{N}(b)\cap A_{\prec b};$$

and for any $a\in A$

$$x\left(\ddot{N}(a)\cap A_{\prec a}\right)\leq2x\left(\ddot{N}^{in}(a)\right)$$

as $\prec$ is a surplus-preserving ordering of A with respect to x. Thus,

$$\begin{aligned}
&\sum_{a\in A}x(a)\,\overline{w}\left(\ddot{N}(a)\cap S_{\succ a}\right)\\
&=\sum_{a\in A}x(a)\sum_{b\in\ddot{N}(a)\cap S_{\succ a}}\overline{w}(b)\\
&=\sum_{b\in S}\overline{w}(b)\sum_{a\in\ddot{N}(b)\cap A_{\prec b}}x(a)\\
&=\sum_{b\in S}\overline{w}(b)\,x\left(\ddot{N}(b)\cap A_{\prec b}\right)\\
&=\sum_{a\in S}\overline{w}(a)\,x\left(\ddot{N}(a)\cap A_{\prec a}\right)\\
&\leq2\sum_{a\in S}\overline{w}(a)\,x\left(\ddot{N}(a)\right).
\end{aligned}$$

Finally, we prove the inequality stated in the lemma. Using the relations between the original weights and the discounted weights, we have

$$\begin{aligned}
&\sum_{a\in A}w(a)\,x(a)\\
&=\sum_{a\in A}\overline{w}(a)\,x(a)+\sum_{a\in A}x(a)\,\overline{w}\left(\dot{N}(a)\cap S_{\succ a}\right)\\
&+\frac{1}{\lambda}\sum_{a\in A}x(a)\,\overline{w}\left(\ddot{N}(a)\cap S_{\succ a}\right)\\
&\leq\sum_{a\in S}\overline{w}(a)\,x(a)+\sum_{a\in S}\overline{w}(a)\,x\left(\dot{N}(a)\right)\\
&+\frac{2}{\lambda}\sum_{a\in S}\overline{w}(a)\,x\left(\ddot{N}^{in}(a)\right)\\
&=\sum_{a\in S}\overline{w}(a)\left[x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)\right]\\
&\leq\max_{a\in S}\left[x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)\right]\sum_{a\in S}\overline{w}(a)\\
&=\overline{w}(S)\max_{a\in S}\left[x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)\right]\\
&\leq\overline{w}(S)\max_{a\in A}\left[x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)\right].
\end{aligned}$$

Therefore, the lemma holds. ■

Note that a non-zero $x\in\mathbb{R}_+^A$ maximizing the value

$$\frac{\sum_{a\in A}w(a)\,x(a)}{\max_{a\in A}\left[x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}_D^{in}(a)\right)\right]}$$

is achieved by an optimal solution to the following linear program (LP):

$$\begin{array}{ll} \max & \sum_{a\in A} w(a)\, x(a) \\ s.t. & x\left(\dot{N}[a]\right) + \frac{2}{\lambda} x\left(\ddot{N}^{in}(a)\right) \leq 1, \forall a \in A \\ & x(a) \geq 0, \forall a \in A \end{array} \quad (1)$$

The value of the above LP and the weight of a maximum-weighted feasible subset O are related as follows.

Lemma 4.4: The value of the LP in equation (1) is at least $\frac{1}{2\mu_\lambda} w(O)$.

Proof. Let y be the function on A defined by

$$y(a) = \begin{cases} \frac{1}{2\mu_\lambda}, & \text{if } a \in O; \\ 0, & \text{if } a \notin O. \end{cases}$$

By Theorem 4.1 and Lemma 2.1, for each $a \in A$,

$$\begin{aligned} & y\left(\dot{N}[a]\right) + \frac{2}{\lambda} y\left(\ddot{N}^{in}(a)\right) \\ &= \frac{1}{2\mu_\lambda}\left[\left|\dot{N}[a] \cap O\right| + \frac{2}{\lambda}\left|\ddot{N}^{in}(a) \cap O\right|\right] \\ &\leq \frac{1}{2\mu_\lambda} 2\mu_\lambda \\ &= 1. \end{aligned}$$

Thus, y is a feasible solution to the LP in equation (1), and consequently the value of the LP in equation (1) is at least

$$\sum_{a\in A} w(a)\, y(a) = \frac{1}{2\mu_\lambda} \sum_{a\in O} w(a) = \frac{1}{2\mu_\lambda} w(O).$$

So, the lemma holds. ■

The above two lemmas imply immediately the following corollary.

Corollary 4.5: Suppose that $\prec$ is a surplus-preserving ordering of A with respect to an optimal solution x to the LP in equation (1). Then, the greedy candidate subset S in $\prec$ satisfies that $\overline{w}(S) \geq \frac{1}{2\mu_\lambda} w(O)$.

Motivated by the above corollary, we produce the request ordering in two steps: First compute an optimal solution x to the LP in equation (1), and then compute a surplus-preserving ordering of A with respect to x.

D. Putting Together

By integrating all the algorithm ingredients explored above together, the full algorithm runs in six steps:

1) Compute an optimal solution x to the LP in equation (1).
2) Compute a surplus-preserving ordering $\prec$ of A with respect to x.
3) Compute the greedy candidate subset S of A in $\prec$.
4) Compute the maximal inductively compatible subset F of S in $\prec$.
5) Compute the greedy channel assignment π to F in $\prec$.
6) Compute the trivial transmission schedule of F under π.

The first step requires solving a linear programming, the last step is trivial, and the intermediate four steps are greedy in nature and have very simple implementations. The approximation bound of this algorithm follows immediately from Lemma 4.2 and Corollary 4.5.

Theorem 4.6: F is a $2\mu_\lambda$-approximate solution.

V. Light Requests

In this section, we assume all requests in A are light, i.e. $d(a) \in (0, 1/2]$ for each $a \in A$. Again, the feasibility test is NP-complete. We shall also explore the rationale of the algorithm design progressively in the *reverse* order of the algorithm execution. The full description and analysis of the algorithm will be put forward at the end of this section.

A. Greedy Schedule

Consider a subset F of requests. Give an ordering $\prec$ of F and a channel assignment π to F, a transmission schedule $\mathcal{S}_\pi^\prec$ of F can be computed efficiently by a simple greedy algorithm developed in [8]. Initially, $\mathcal{S}_\pi^\prec$ is empty and let F' be the set of tasks $a \in F$ with $d(a) > 0$. The algorithm repeats the following iteration until F' is empty:

- Compute the maximal subset C of F in $\prec$ which can be transmit at the same time.
- Let $\ell = \min_{a\in C} d(a)$, and add (C, ℓ) to $\mathcal{S}_\pi^\prec$.
- For each $a \in C$, replace $d(a)$ by $d(a) - \ell$ and remove a from F' if $d(a) = 0$.

The output $\mathcal{S}_\pi^\prec$ is referred to as the *greedy schedule* of F in $\prec$ under the channel assignment π. It was shown in [8] that the length of the schedule $\mathcal{S}_\pi^\prec$ is at most

$$\max_{a\in F}\left[d\left(\dot{N}[a] \cap F_{\preceq a}\right) + \sum_{b\in \ddot{N}(a)\cap F_{\prec a}:\pi(b)=\pi(a)} d(b)\right],$$

which is denoted by $\Delta_\pi^\prec(F)$ and referred to as *the d-weighted inductivity of F in $\prec$ under π.*

B. Greedy Channel Assignment

Consider a subset F of requests. Given an ordering $\prec$ of F, we would like to find a channel assignment π such that $\Delta_\pi^\prec(F)$ is small. More precisely, let

$$\Delta^\prec(F) := \max_{a\in F}\left[d\left(\dot{N}[a] \cap F_{\preceq a}\right) + \frac{1}{\lambda} d\left(\ddot{N}(a) \cap F_{\prec a}\right)\right],$$

which is referred to as *the d-weighted inductivity of F in $\prec$*. The channel assignment π shall satisfies that $\Delta_\pi^\prec(F) \leq \Delta^\prec(F)$.

We observe that inside the operand of the maximization in the expression of $\Delta_\pi^\prec(F)$, only the second term depends on the channel assignment π and represents the total secondary conflicts received by a request from preceding requests. This observation naturally motivate us to propose a greedy channel assignment π in $\prec$.

- The first request in F receives the first channel.
- For each other request $a \in F$ in the ordering $\prec$, the *secondary congestion* of a over a channel l is defined to be total demands of the requests $b \in \ddot{N}(a) \cap F_{\prec a}$ which are assigned with the channel l, and a is assigned with the

first channel $\pi(a)$ over which it has the least secondary congestion.

Such channel assignment π is referred to as the *greedy channel assignment* to A in $\prec$. It has the following property:

Lemma 5.1: The greedy channel assignment π to A in $\prec$ satisfies that $\Delta_{\pi}^{\prec}(F) \leq \Delta^{\prec}(F)$.

Proof. Consider any $a \in F$. The total secondary congestion of $a \in F$ over *all* channels is exactly the total demands of the requests $\ddot{N}(a) \cap F_{\prec a}$. By the pigeonhole principle, the secondary congestion of a over the channel $\pi(a)$ is at most $\frac{1}{\lambda} d\left(\ddot{N}(a) \cap F_{\prec a}\right)$. Therefore,

$$\begin{aligned} \Delta_{\pi}^{\prec}(F) &\leq \max_{a \in F}\left[d\left(\dot{N}[a] \cap F_{\preceq a}\right) + \frac{1}{\lambda} d\left(\ddot{N}(a) \cap F_{\prec a}\right)\right] \\ &= \Delta^{\prec}(F). \end{aligned}$$

So, the lemma holds. ■

C. Selection of A Feasible Subset

Our selection of a feasible subset takes a *restriction* approach. A set $F \subseteq A$ is said to be *inductively feasible* if $\Delta^{\prec}(F) \leq 1$. The inductive feasibility of F implies the feasibility of F and can be tested in polynomial time. It also has the following equivalent characterization. Consider any link $a \in A$, for each $b \in \dot{N}(a)$, define

$$\varrho(a, b) = \frac{d(a)}{1 - d(b)};$$

for each $b \in \ddot{N}(a)$, define

$$\varrho(a, b) = \frac{1}{\lambda} \frac{d(a)}{1 - d(b)}.$$

Then, it is easy to verify that $\Delta^{\prec}(F) \leq 1$ if and only if

$$\max_{a \in F} \sum_{b \in N(a) \cap F_{\prec a}} \varrho(b, a) \leq 1.$$

Given a subset $S \subseteq A$ and an ordering $\prec$ of A, an inductively feasible subset F of S can be computed in a *greedy* manner as follows:

- Initially, F is empty.
- For each $a \in S$ in the ordering $\prec$, a is added to F if and only if
$$\sum_{b \in N(a) \cap F} \varrho(b, a) \leq 1.$$

The final set F is referred to as the *maximal inductively feasible subset* of S in $\prec$. It is maximal in the sense that for each $a \in S \setminus F$,

$$\sum_{b \in N(a) \cap F_{\prec a}} \varrho(b, a) > 1.$$

Again, the "candidate" subset S is very essential, and its selection will be explored further in the next subsection.

D. Selection of Candidates

Given an ordering $\prec$ of A, the local-ratio scheme computes a candidate set S in the following *greedy* manner:

- S is initially empty.
- For each $a \in A$ in the *reverse* order of $\prec$, a *discounted* weight $\overline{w}(a)$ of a is computed by
$$\overline{w}(a) = w(a) - \sum_{b \in N(a) \cap S} \varrho(a, b) \overline{w}(b);$$
and if $\overline{w}(a) > 0$, a is added to S.

The final set S computed in this greedy manner is referred to as the *greedy candidate subset* of A in $\prec$. For each $a \in A$, its original weight and its discounted weight have the following relation:

$$w(a) = \overline{w}(a) + \sum_{b \in N(a) \cap S_{\succ a}} \varrho(a, b) \overline{w}(b).$$

In addition, the maximal inductively compatible subset F of S in $\prec$ has the following property

Lemma 5.2: The maximal inductively compatible subset F of S in $\prec$ satisfies that $w(F) \geq \overline{w}(S)$.

Proof. By the greedy selection of F, for each $a \in S \setminus F$,

$$\sum_{b \in N(a) \cap F_{\prec a}} \varrho(b, a) > 1.$$

Thus,

$$\begin{aligned} w(F) &= \sum_{a \in F} w(a) \\ &= \sum_{a \in F} \overline{w}(a) + \sum_{a \in F} \sum_{b \in N(a) \cap S_{\succ a}} \varrho(a, b) \overline{w}(b) \\ &= \overline{w}(F) + \sum_{a \in S} \overline{w}(a) \sum_{b \in N(a) \cap F_{\prec a}} \varrho(b, a) \\ &\geq \overline{w}(F) + \sum_{a \in S \setminus F} \overline{w}(a) \sum_{b \in N(a) \cap F_{\prec a}} \varrho(b, a) \\ &\geq \overline{w}(F) + \sum_{a \in S \setminus F} \overline{w}(a) \\ &= \overline{w}(S). \end{aligned}$$

So, the lemma holds. ■

Next, we proceed to compute a link ordering $\prec$ such that the discounted weight of the greedy candidate subset S in $\prec$ is close to the original weight of an optimal solution.

E. Selection of Ordering

Our choice of request ordering is motivated by the following property of surplus-preserving orderings.

Lemma 5.3: Suppose that $\prec$ is a surplus-preserving ordering of A with respect to some non-zero $x \in \mathbb{R}_{+}^{A}$. Then, the greedy candidate subset S in $\prec$ satisfies that

$$\overline{w}(S) \geq \frac{\sum_{a \in A} \frac{w(a)}{d(a)} x(a)}{2\delta(x)}$$

where

$$\delta(x)=\max_{a\in A}\max\left\{x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right),\frac{x(a)}{d(a)}\right\}.$$

Proof. We first show that for each $a\in A$,

$$\frac{x(a)}{d(a)}+\sum_{b\in N(a)\cap A_{\prec a}}\varrho(b,a)\frac{x(b)}{d(b)}\leq 2\delta(x).$$

Indeed, since $\prec$ is a surplus-preserving ordering of A with respect to x and $d(a)\leq 1/2$, we have

$$\begin{aligned}
&\frac{x(a)}{d(a)}+\sum_{b\in N(a)\cap A_{\prec a}}\varrho(b,a)\frac{x(b)}{d(b)}\\
&=\frac{x(a)}{d(a)}+\frac{x\left(\dot{N}(a)\cap A_{\prec a}\right)+\frac{1}{\lambda}x\left(\ddot{N}(a)\cap A_{\prec a}\right)}{1-d(a)}\\
&\leq\frac{x(a)}{d(a)}+\frac{x\left(\dot{N}(a)\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)}{1-d(a)}\\
&\leq\frac{x(a)}{d(a)}+\frac{\delta(x)-x(a)}{1-d(a)}\\
&=2\frac{x(a)}{d(a)}+\frac{\delta(x)-\frac{x(a)}{d(a)}}{1-d(a)}\\
&\leq 2\frac{x(a)}{d(a)}+2\left(\delta(x)-\frac{x(a)}{d(a)}\right)\\
&=2\delta(x).
\end{aligned}$$

Next, we prove the inequality stated in the lemma. Using the relations between the original weights and the discounted weights, we have

$$\begin{aligned}
&\sum_{b\in A}\frac{w(a)}{d(a)}x(a)\\
&=\sum_{a\in A}\overline{w}(a)\frac{x(a)}{d(a)}+\sum_{a\in A}\frac{x(a)}{d(a)}\sum_{b\in N(a)\cap S_{\succ a}}\varrho(a,b)\overline{w}(b)\\
&\leq\sum_{a\in S}\overline{w}(a)\frac{x(a)}{d(a)}+\sum_{a\in A}\frac{x(a)}{d(a)}\sum_{b\in N(a)\cap S_{\succ a}}\varrho(a,b)\overline{w}(b)\\
&=\sum_{a\in S}\overline{w}(a)\frac{x(a)}{d(a)}+\sum_{b\in S}\overline{w}(b)\sum_{a\in N(b)\cap A_{\prec b}}\varrho(a,b)\frac{x(a)}{d(a)}\\
&=\sum_{a\in S}\overline{w}(a)\frac{x(a)}{d(a)}+\sum_{a\in S}\overline{w}(s)\sum_{b\in N(a)\cap A_{\prec a}}\varrho(b,a)\frac{x(b)}{d(b)}\\
&=\sum_{a\in S}\overline{w}(a)\left(\frac{x(a)}{d(a)}+\sum_{b\in N(a)\cap A_{\prec a}}\varrho(b,a)\frac{x(b)}{d(b)}\right)\\
&\leq 2\delta(x)\sum_{a\in S}\overline{w}(a)\\
&=2\delta(x)\overline{w}(S).
\end{aligned}$$

So, the lemma holds. ■

Note that a non-zero $x\in\mathbb{R}_+^A$ maximizing the value

$$\frac{\sum_{a\in A}\frac{w(a)}{d(a)}x(a)}{\delta(x)}$$

is achieved by an optimal solution to the following linear program (LP):

$$\begin{array}{ll}\max & \sum_{a\in A}\frac{w(a)}{d(a)}x(a)\\ s.t. & x\left(\dot{N}[a]\right)+\frac{2}{\lambda}x\left(\ddot{N}^{in}(a)\right)\leq 1,\forall a\in A\\ & 0\leq x(a)\leq d(a),\forall a\in A\end{array}\qquad(2)$$

The value of the above LP and the weight of a maximum-weighted feasible subset O are related as follows.

Lemma 5.4: The value of the LP in equation (2) is at least $\frac{1}{2\mu_\lambda}w(O)$.

Proof. Let y be the function on A defined by

$$y(a)=\begin{cases}\frac{1}{2\mu_\lambda}d(a), & \text{if } a\in O;\\ 0, & \text{if } a\notin O.\end{cases}$$

By Lemma 2.2, for each $a\in A$, $y(a)\leq d(a)$ and

$$\begin{aligned}
&y\left(\dot{N}[a]\right)+\frac{2}{\lambda}y\left(\ddot{N}^{in}(a)\right)\\
&=\frac{1}{2\mu_\lambda}\left[d\left(\dot{N}[a]\cap O\right)+\frac{2}{\lambda}d\left(\ddot{N}^{in}(a)\cap O\right)\right]\\
&\leq\frac{1}{2\mu_\lambda}2\mu_\lambda=1.
\end{aligned}$$

Thus, y is a feasible solution to the LP in equation (2). So, the value of the LP in equation (2) is at least

$$\sum_{a\in A}\frac{w(a)}{d(a)}y(a)=\frac{1}{2\mu_\lambda}\sum_{a\in O}\frac{w(a)}{d(a)}d(a)=\frac{1}{2\mu_\lambda}w(O).$$

Therefore, the lemma holds. ■

The above two lemmas imply immediately the following corollary.

Corollary 5.5: Suppose that $\prec$ is a surplus-preserving ordering of A with respect to an optimal solution x to the LP in equation (2). Then, the greedy candidate subset S in $\prec$ satisfies that $\overline{w}(S)\geq\frac{1}{4\mu_\lambda}w(O)$.

Motivated by the above corollary, we produce the request ordering in two steps: First compute an optimal solution x to the LP in equation (2), and then compute a surplus-preserving ordering of A with respect to x.

F. Putting Together

Finally, we are ready to describe the full algorithm. The algorithm proceeds in six steps:

1) Compute an optimal solution x to the LP in equation (2).
2) Compute a surplus-preserving ordering $\prec$ of A with respect to x.
3) Compute the greedy candidate subset S of A in $\prec$.
4) Compute the maximal inductively feasible subset F of S in $\prec$.
5) Compute the greedy channel assignment π to F in $\prec$.
6) Compute the greedy schedule of F in $\prec$ under π.

The first step requires solving a linear programming, and all other steps are greedy in nature and have very simple implementations. The approximation bound of this algorithm follows immediately from Lemma 5.2 and Corollary 5.5.

Theorem 5.6: F is a $4\mu_\lambda$-approximate solution.

VI. Arbitrary Requests

In this section, we assume that A contains both light requests and heavy requests. The algorithm described below selects a feasible subset of A by following the divide-and-conquer scheme:

- **Division**: Let A_1 (respectively, A_2) denote the set of heavy (respectively, light) requests in A. Then, A_1 and A_2 form a partition of A.
- **Conquer**: Compute a feasible subset F_1 (respectively, F_2) of A_1 (respectively, A_2) using the the first four steps of the algorithm proposed in Section V (respectively, Section IV).
- **Combination**: If F_1 has larger weight than F_2, then select F_1 and compute the channel assignment and transmission schedule of F_1 using the last two steps of the algorithm proposed in Section V; otherwise select F_2 and compute the channel assignment and transmission schedule of F_2 using the last two steps of the algorithm proposed in Section IV.

Let F be the feasible set selected by the algorithm,. Then, we have the following approximation bounds:

Theorem 6.1: F is a $6\mu_\lambda$-approximate solution.

Proof. Let O be a maximum-weighted subset of A. Since $O \cap A_1$ is a feasible subset of A_1, by Theorem 4.6 we have

$$w(O \cap A_1) \leq 2\mu_\lambda w(F_1) \leq 2\mu_\lambda w(F).$$

Since $O \cap A_2$ is a feasible subset of A_2, by Theorem 5.6 we have

$$w(O \cap A_2) \leq 4\mu_\lambda w(F_1) \leq 4\mu_\lambda w(F).$$

Thus,

$$\begin{aligned} w(O) &= w(O \cap A_1) + w(O \cap A_2) \\ &\leq 2\mu_\lambda w(F) + 4\mu_\lambda w(F) \\ &\leq 6\mu_\lambda w(F). \end{aligned}$$

So, the theorem holds. ■

VII. Conclusion

In this paper, we have exploited a rich set of algorithm design strategies including greedy method, local-ratio (primal-dual) scheme, divide and conquer, restriction, and linear program relaxation for approximately solving the (spectral) non-splitting variant of the problem **MWFS**. The approximation bounds of our algorithms in the cases of all heavy requests, all light requests, and arbitrary requests respectively are summarized in Table I. For the purpose of comparison, the respective approximation bounds for the (spectral) splitting variant of the problem **MWFS** [9] are also listed. The strictly better approximation bounds are made possible by the finer treatment of primary conflicts and secondary conflicts in our algorithm design and analyses.

	Non-splitting [This Paper]	Splitting [9]
All heavy	$2\left(\mu + \left(1 - \frac{1}{\lambda}\right)\right)$	$4\left(\mu + 2\left(1 - \frac{1}{\lambda}\right)\right)$
All light	$4\left(\mu + \left(1 - \frac{1}{\lambda}\right)\right)$	$4\left(\mu + 2\left(1 - \frac{1}{\lambda}\right)\right)$
General	$6\left(\mu + \left(1 - \frac{1}{\lambda}\right)\right)$	$8\left(\mu + 2\left(1 - \frac{1}{\lambda}\right)\right)$

TABLE I
THE APPROXIMATION BOUNDS FOR THE PROBLEM **MWFS**: μ IS THE ILIN, AND λ IS THE NUMBER OF CHANNELS.

We conclude this paper with a comment on the problem of minimum-length scheduling. Consider a set A of communication requests in a multihop wireless network with λ channels. Each task a has a demand $d(a)$ of transmission time. Let $\Delta^*(A)$ be the minimum of $\Delta^{\prec}(A)$ over all possible orderings $\prec$ of B. A *smallest-last ordering* $\prec$ of A [10] achieves the least inductivity $\Delta^*(A)$ and can be computed by a simple greedy strategy. With spectral splitting, the best-known upper bound on the length of the schedules produced by a polynomial time algorithm is $\Delta^*(A)$. Without spectral splitting, the greedy channel assignment and transmission scheduling developed in Section V in the smallest-last ordering also produce a schedule of length at most $\Delta^*(A)$.

ACKNOWLEDGEMENTS: This work was supported in part by the National Natural Science Foundation of P. R. China under grants 61529202, 61572131, 61572277, and 61672195, by the National Science Foundation of USA under grants CNS-1219109 CNS-1454770, and EECS-1247944, and by SUNY Oneonta Faculty Development Funds.

References

[1] K. Akcoglu, J. Aspnes, B. DasGupta, and M-Y. Kao, Opportunity Cost Algorithms for Combinatorial Auctions, *Applied Optimization 74: Computational Methods in Decision-Making, Economics and Finance*, 455-479, 2002.

[2] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor and B. Schieber, A Unified Approach to Approximating Resource Allocation And Scheduling, *Journal of ACM* 48(5): 1069-1090, 2001.

[3] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber, Approximating The Throughput of Multiple Machines In Real-Time Scheduling, *SIAM Journal on Computing* 31(2): 331-352, 2001.

[4] R. Bar-Yehuda and D. Rawttz, On The Equivalence between The Primal-Dual Schema And The Local Ratio Technique, *SIAM J. Disc. Math.* 19: 762–797, 2005.

[5] R. Bar-Yehuda and D. Rawttz, Using Fractional Primal-Dual to Schedule Split Intervals with Demands, *Journal Discrete Optimization* 3(4): 275-287, 2006.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[7] R. McNaughton, Scheduling with deadlines and loss functions. *Management Science* 12:1–12, 1959.

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

[9] P.-J. Wan, Joint Selection And Transmission Scheduling of Point-to-Point Communication Requests in Multi-Channel Wireless Networks, *ACM Mobihoc* 2016.

[10] P.-J. Wan, X. Jia, G. Dai, H. Du, Z.G. Wan, and O. Frieder, Scalable Algorithms for Wireless Link Schedulings in Multi-Channel Multi-Radio Wireless Networks, *Proc. IEEE INFOCOM* 2013.

[11] P.-J. Wan, X. Jia, G. Dai, H. Du, and O. Frieder, Fast And Simple Approximation Algorithms for Maximum Weighted Independent Set of Links, *Proc. IEEE INFOCOM* 2014: 1653-1661.

[12] 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.

[13] Y. Ye and A. Borodin, Elimination graphs, *ACM Transactions on Algorithms* 8(2):14:1–14:23, 2012.