Scalable Algorithms for Wireless Link Schedulings in Multi-Channel Multi-Radio Wireless Networks

Peng-Jun Wan[*], Xiaohua Jia[†], Guojun Dai[‡], Hongwei Du[§], Zhiguo Wan[¶], and Ophir Frieder[‖]

[*] Department of Computer Science, Illinois Institute of Technology, wan@cs.iit.edu
[†] Department of Computer Science, City University of Hong Kong, csjia@cityu.edu.hk
[‡] School of Computer Science, Hangzhou DianZi University, daigj@hdu.edu.cn. Corresponding auhtor.
[§] School of Computer Science, Harbin Institute of Technology, hwdu@hitsz.edu.cn
[¶] School of software, Tsinghua University, wanzhiguo@tsinghua.edu.cn
[‖] Department of Computer Science, Georgetown University, ophir@ir.cs.georgetown.edu

Abstract—For wireless link scheduling in multi-channel multi-radio wireless networks aiming at maximizing (concurrent) multiflow, constant-approximation algorithms have recently been developed in [11]. However, the running time of those algorithms grows quickly with the number of radios per node (at least in the sixth order) and the number of channels (at least in the cubic order). Such poor scalability stems intrinsically from the exploding size of the fine-grained network representation upon which those algorithms are built. In this paper, we introduce a new structure, termed as concise conflict graph, on the node-level links directly. Such structure succinctly captures the essential advantage of multiple radios and multiple channels. By exploring and exploiting the rich structural properties of the concise conflict graphs, we are able to develop fast and scalable link scheduling algorithms for either minimizing the communication latency or maximizing the (concurrent) multiflow. These algorithms have running time growing linearly in both the number of radios per node and the number of channels, while not sacrificing the approximation bounds.

Index Terms—Link scheduling, multi-channel multi-radio, approximation algorithms.

I. INTRODUCTION

With the rapid technology advances, many off-the-shelf wireless transceivers (i.e., radios) are capable of operating on multiple channels. For example, the IEEE 802.11 b/g standard and IEEE 802.11a standard provide 3 and 12 channels respectively, and MICA2 sensor motes support more than 50 channels. The rapidly diminishing prices of the radios has also made it feasible to equip a wireless node with multiple radios. Providing each node with one or more multi-channel radios offers a promising avenue for enhancing the network capacity by simultaneously exploiting multiple non-overlapping channels through different radio interfaces and mitigating interferences through proper channel assignment. However, it is quite challenging to effectively utilize multiple channels and/or multiple radios to maximize throughput capacity. The major technical obstacle stems from the disruptive change of the network geometry in multi-channel multi-radio (MC-MR) wireless networks: the geometric closeness now does not necessarily imply interference, for it is possible for a pair of links to transmit concurrently over different channels without conflict even if they are very close to each other. As the result of such *non-locality* nature, it is even unknown whether a set of links in a MC-MR wireless network can transmit at the same time or not may be decidable in polynomial time. In contrast, the same decision problem in single-channel single-radio (SC-SR) wireless networks is easily solvable in polynomial time.

In order to overcome the technical obstacle caused by the non-locality nature, a *fine-grained network representation* of the MC-MR wireless networks was introduced in [11]. Consider a MC-MR wireless network in which V is the set of wireless nodes, A is the set of direct node-level communcation links, each node v has $\tau(v)$ radios, and there are λ non-overlapping channels. In its fine-grained network representation, each communication link is encoded by an ordered *quintuple* specifying the transmitting node, the receiver node, the radio at the transmitting node, the radio at the receiving node, and the channel. Specifically, for each node-level link (u, v) in A, we make $\lambda \cdot \tau(u) \cdot \tau(v)$ replications (u, v, i, j, k) for $1 \le i \le \tau(u)$, $1 \le j \le \tau(v)$, and $1 \le k \le \lambda$. A replication (u, v, i, j, k) always utilizes the i-th radio at u and the j-th radio at v over the k-th channel. Each of these replications is referred to as a fine-grained communication link. A set I of these replicated links can transmit at the same time if and only if (1) all replication links in I are radio-disjoint, in other words, no pair share a common radio, and (2) for each channel k, all the replication links in I transmitting over channel k are conflict-free. The first condition can be easily verified in polynomial time, and the second condition can be verified in the same way as in SC-SR wireless networks. Therefore, the decision problem whether a set of *replication* links can transmit at the same time can now be solvable in polynomial time.

The above fine-grained network representation allows for the conventional definition of the conflict graph of the replicated links. By leveraging the general techniques developed in [10], Wan et al. [11] obtained constant-approximation algorithms for the following link scheduling problems under the 802.11 interference model or the protocol interference model:

- **Shortest Weighted Link Schedule** (**SWLS**): Given a set of traffic demands on the *replicated* links, find a shortest fractional schedule of replicated links for these traffic demands.

- **Maximum Multiflow** (**MMF**): Given a set of end-to-end unicast communication requests, find a link schedule of length at most one which can carry on the maximum multiflow of these unicast requests.
- **Maximum Concurrent Multiflow** (**MCMF**): Given a set of end-to-end unicast communication requests together with their traffic demands, find a link schedule of length at most one which can carry on the maximum concurrent multiflow of these unicast requests.

A major drawback of those approximation algorithms is the poor scalability of the running time with respect to the number of radios and the number of channels. Those approximation algorithms for both **MMF** and **MCMF** involves solving a linear program with a huge number of variables. In fact, for each unicast request, there is a link-flow variable associated with each replicated link. Thus, the total number of variables in the linear program is the number of unicasts times the number of replicated links. On the other hand, the running time taken by the Karmarkar's interior point method for linear programming, the fastest one known so far, grows in the *cubic* order of the number of variables. Consequently, the running time of those algorithms grows quickly with the number of radios per node (at least in the *sixth* order) and the number of channels (at least in the *cubic* order). Such running time would be very slow even for a MC-MR wireless network of moderate size. For an instance, consider a MC-MR wireless network with 256 links, 4 radios per node, and 16 channels, and with 16 unicast requests, the number of link-flow variables is already

$$16 \times 4 \times 4 \times 16 \times 256 = 2^{20},$$

Thus, solving the LP with a mega (2^{20}) variables has a running time in the exa (2^{60}) order.

Another limitation of the algorithmic studies in [11] on **SWLS** is the requirement on the traffic demands given on the replicated links, rather than on the node-level links. The works in [11] didn't explicitly address how to compute a link schedule for a given set of traffic demands on the *node-level* links. A viable two-phased approach is to first split the given node-level link demands to an appropriate set of traffic demands on the replicated links by solving a linear program similar to **MMF** or **MMCF**, and then apply the algorithms developed in [11] for **SWLS** to the resulting traffic demands on the replicated links to compute a link schedule. However, the linear program in the first phase also has variables as many as the replicated links, and hence also suffers from the poor scalability as discussed in the previous paragraph.

The main objective of this paper is to develop *scalable* approximation algorithms for **MMF**, **MMCF**, and the variant of **SWLS** with the traffic demands given on the *node-level* links, while *not* sacrificing the approximation bounds. For this purpose, we introduce a terse structure, termed by *concise conflict graph*, on the node-level links directly. Such structure succinctly captures the essential advantage of multiple radios and multiple channels. We will explore the rich structural properties of the concise conflict graphs. By exploiting these properties of the concise conflict graphs, we are able to develop scalable link scheduling algorithms for **MMF**, **MMCF**, and the variant of **SWLS** with the traffic demands given on the node-level links. These algorithms have running time growing *linearly* in both the maximum number of radios at individual nodes and the number of channels, while are still able to achieve the same approximation bounds as those achieved in [11].

The remainder of this paper is organized as follows. In Section II, we introduce the definition of concise conflict graph and explore its fundamental structural properties. In Section III, we develop a first-fit link scheduling algorithm for the variant of **SWLS** with the traffic demands given on the node-level links. In Section IV, we give two polynomial approximate capacity subregions defined on the node-level links directly. In Section V, we present the approximation algorithms for **MMF** and **MMCF**. In Section VI, we review the related works on link scheduling in MC-MR wireless networks. Finally, we conclude this paper in Section VII.

II. Concise Conflict Graph

Consider an instance of 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 λ non-overlapping channels. Two links in A are said to have a *conflict* if they cannot transmit at the same time over the same channel. Furthermore, a conflicting pair of distinct links in A are said to have *primary* conflict if there share one common end, and *secondary* conflict otherwise. For the sake of convenience, each link is said to have a *self-conflict* with itself. The *concise conflict graph* of the MC-MR wireless network is the edge-weighted graph G on A in which there is an edge between each conflicting pair of links (a, b) whose weight denoted by $c(a, b)$, is defined as follows:

- If $b = a$ (i.e., self-conflict), then

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

where u and v are the two end-nodes of a.

- If a and b have a common endpoint u (i.e., a and b have a primary conflict), then

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

- If a and b have the secondary conflict, then

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

Note that $c(a, b) = c(b, a)$. Let E denote the set of edges in G, and $\mathcal{I}$ denote the collection of the independent sets in G. In other words, $\mathcal{I}$ is the collection of the subsets of A which can transmit successfully at the same time over the same channel. Note that G can be regarded as a generalization of the conventional conflict graph of the underlying SC-SR wireless network by adding a self-loop at each link and assigning each edge a weight specified by the function c. Thus, $\mathcal{I}$ is essentially

the collection of the independent sets of links in the underlying SC-SR wireless network.

In this section, we explore the relevant topological properties of the concise conflict graph. We first introduce some terms and notations. For each subset B of A, we use $B^{\tau,\lambda}$ to denote the set of all replications of the links in B as described in the first section of this paper. In particular, $A^{\tau,\lambda}$ is the set of all replicated links of the links in A. For any $d \in \mathbb{R}_+^A$, a function (or vector) $x \in \mathbb{R}_+^{A^{\tau,\lambda}}$ is said to be *split* of d if for each $a \in A$,

$$d(a) = \sum_{e \in \{a\}^{\tau,\lambda}} x(e);$$

if x is a split of d, then d is also called the *aggregation* of x. We also distinguish three types of conflicts between a conflicting pair of replicated links e and e' in $A^{\tau,\lambda}$:

- self-conflict: this happens when $e = e'$.
- primary conflict: this happens when e and e' share exactly one common radio.
- secondary conflict: this happens when e and e' are radio-disjoint but share the same channel, and their node-level links conflict with each other.

We use $G^{\tau,\lambda}$ to denote the conflict graph of $A^{\tau,\lambda}$. We would like to emphasize that $G^{\tau,\lambda}$ also contains a self-loop at each vertex and its edges are unweighted. Let $\mathcal{I}^{\tau,\lambda}$ denote the collection of the independent sets in $G^{\tau,\lambda}$. In other words, $\mathcal{I}^{\tau,\lambda}$ is the collection of the sets of replicated links in $A^{\tau,\lambda}$ which can transmit successfully at the same time. For any subset S of $A^{\tau,\lambda}$, a subset I of S is said to be a *maximal independent set* of S if $I \in \mathcal{I}^{\tau,\lambda}$ and for any link $e \in S \setminus I$, $I \cup \{e\} \notin \mathcal{I}^{\tau,\lambda}$. Consider a link $a \in A$. An independent set $I \in \mathcal{I}^{\tau,\lambda}$ is said to be *a-tight* if each link $e \in \{a\}^{\tau,\lambda} \setminus I$ has a conflict with some link in $I \cap N_G^{\prec}(a)^{\tau,\lambda}$.

A. Interpretation of Edge-Weights

The lemma below reveals the physical meaning of the edge-weights in the concise conflict graph.

Lemma 2.1: Suppose that a and b are pair of conflicting links in A. Then, each replicated link of b conflicts with exactly $c(a,b)$ portion of replicated links of a.

Proof: Let $a = (u,v)$ and e be a replicated link of b.

Case 1: b is a itself. A replicated link of a has no conflict with e if and only if it does not share any radio with e and it uses a different channel from e. The number of replicated links of a which have no conflict with e is

$$(\tau(u) - 1)(\tau(v) - 1)(\lambda - 1).$$

Hence, the number of replicated links of a which have conflict with e is

$$\tau(u)\tau(v)\lambda - (\tau(u) - 1)(\tau(v) - 1)(\lambda - 1)$$
$$= \tau(u)\tau(v)\lambda\left[1 - \left(1 - \frac{1}{\tau(u)}\right)\left(1 - \frac{1}{\tau(v)}\right)\left(1 - \frac{1}{\lambda}\right)\right]$$
$$= \tau(u)\tau(v)\lambda c(a,a).$$

Case 2: b shares exactly one end with a. By symmetry, we assume that u is the common end of a and b. A replicated link of a has no conflict with e if and only if it does not share any radio at u with e and it uses a different channel from e. The number of replicated links of a which have no conflict with e is

$$(\tau(u) - 1)\tau(v)(\lambda - 1).$$

Hence, the number of replicated links of a which have conflict with e is

$$\tau(u)\tau(v)\lambda - (\tau(u) - 1)\tau(v)(\lambda - 1)$$
$$= \tau(u)\tau(v)\lambda\left[1 - \left(1 - \frac{1}{\tau(u)}\right)\left(1 - \frac{1}{\lambda}\right)\right]$$
$$= \tau(u)\tau(v)\lambda c(a,b).$$

Case 3: b shares no end with a. A replicated link of a has conflict with e if and only if it it uses the same channel from e. Hence, the number of replicated links of a which have conflict with e is

$$\tau(u)\tau(v) = \tau(u)\tau(v)\lambda\frac{1}{\lambda} = \tau(u)\tau(v)\lambda c(a,b).$$

Therefore, the lemma holds in all cases. ∎

For any link $a \in A$, $N_G(a)$ denotes the set of neighbors of a in G. Since G has a self-loop at each vertex, a is a neighbor to itself, and hence $a \in N_G(a)$. Thus, $N_G(a)$ consisting of all links in A (including itself) having conflict with a. For any link a, any subset B of links, and any $d \in \mathbb{R}_+^A$, define

$$\Gamma(B, a; d) = \sum_{b \in N_G(a) \cap B} c(a,b)d(b).$$

we shall interpret the physical meaning of the above expression.

Consider any $x \in \mathbb{R}_+^{A^{\tau,\lambda}}$. For any pair of links e and e' in $A^{\tau,\lambda}$, the *x-weighted interference* received by e from e' is defined to be $x(e')$ if e and e' conflict to each other, and 0 otherwise. For any link e and and any subset S of $A^{\tau,\lambda}$, the x-weighted interference received by e from S is defined to be the total x-weighted interference received by e from all links in S. The lemma below presents an interesting *invariant* property.

Lemma 2.2: Consider a link a and a subset B of links. For any $d \in \mathbb{R}_+^A$ and any split x of d, the average x-weighted interference received by the replicated links of a received from $B^{\tau,\lambda}$ is exactly $\Gamma(B, a; d)$.

Proof: Let $a = (u,v)$ and consider any link $b \in B \cap N_G(a)$. By Lemma 2.1, the total weighted interference received by all replicated links of a from any replicated link e of b to is

$$\tau(u)\tau(v)\lambda c(a,b)x(e).$$

So, the total weighted interference by all replicated links of a from all replicated links of b is

$$\tau(u)\tau(v)\lambda c(a,b)d(b).$$

Thus, the total weighted interference by all replicated links of a from $B^{\tau,\lambda}$ is

$$\tau(u)\tau(v)\lambda\sum_{b \in B \cap N_G[a]} c(a,b)d(b)$$
$$= \tau(u)\tau(v)\lambda\Gamma(B, a; d).$$

As a has $\tau(u)\tau(v)\lambda$ replicated links, the lemma holds. ∎

For any subset S of $A^{\tau,\lambda}$, we use $\mathbf{1}^S$ to denote the indicator function of S. In other words, $\mathbf{1}^S(e)$ is one if $e \in S$ and zero otherwise.

Lemma 2.3: Let B be a non-empty subset of A, and $d \in \mathbb{R}_+^A$ be the aggregation of $\mathbf{1}^I$ for some maximal independent set I of a subset $B^{\tau,\lambda}$, Then, for any link $a \in B$, $\Gamma(B,a;d) \geq 1$.

Proof: Since I is a maximal independent set of a subset $B^{\tau,\lambda}$, the total $\mathbf{1}^I$-weighted interference received by any replicated link of a from $B^{\tau,\lambda}$ is at least one. Hence, the average $\mathbf{1}^I$-weighted interference received by the replicated links of a from $B^{\tau,\lambda}$, which is $\Gamma_G(B,a;d)$ by Lemma 2.2, is at least one. So, the lemma holds. ∎

Similarly, we can prove the following lemma.

Lemma 2.4: Let B be a non-empty subset of A, a be a link in B, and $d \in \mathbb{R}_+^A$ be the aggregation of $\mathbf{1}^I$ for some a-tight independent set I of $B^{\tau,\lambda}$. Then, $\Gamma(B,a;d) \geq 1$.

B. Weighted Inductivity

Consider a link ordering $\prec$ of A. For any link $a \in A$, $N_G^\prec(a)$ denotes the set of neighbors of a in G preceding a in the ordering $\prec$ plus a itself. For any $d \in \mathbb{R}_+^A$, the value

$$\max_{a \in A} \Gamma\left(N_G^\prec(a), a; d\right)$$

is referred to as *d-weighted inductivity* of $\prec$ and is denoted by $\Delta^\prec(d)$. The smallest d-weighted inductivity of all possible link orderings is called the *d-weighted inductivity* of the network. Now, we describe a special vertex ordering called *smallest-last ordering* and show that it achieves the smallest inductivity. It is produced successively as follows: Initialize B to A. For $i = |A|$ down to 1, let a_i be a link minimizing $\Gamma(B,b;d)$ among all links b in B, and delete a_i from B. Then the ordering $\langle a_1, a_2, \cdots, a_{|A|} \rangle$ is a smallest-last ordering. Similar to [7], we can implement this algorithm with time complexity $O(|E|)$ and space complexity $O(|E|)$.

Theorem 2.5: The smallest-last ordering achieves the smallest d-weighted inductivity

$$\Delta^*(d) = \max_{\emptyset \neq B \subseteq A} \min_{b \in B} \Gamma(B,b;d).$$

Proof: We first show that the d-weighted inductivity of any link ordering $\prec$ is at least $\Delta^*(d)$. Let B be an non-empty subset of A such that

$$\Delta^*(d) = \min_{b \in B} \Gamma(B,b;d).$$

Let a be the last link in this ordering $\prec$ such that $a \in B$. Then,

$$N_G(a) \supseteq N_G^\prec(a) \supseteq N_G(a) \cap B,$$

which implies

$$\Gamma\left(N_G^\prec(a), a; d\right) \geq \Gamma(N_G(a) \cap B, a; d)$$
$$= \Gamma(B,a;d) \geq \Delta^*(d).$$

Thus, the d-weighted inductivity of $\prec$ is at least $\Delta^*(d)$.

Next, let $\prec$ be the smallest-last ordering $\langle a_1, a_2, \cdots, a_{|A|} \rangle$ and we show that its d-weighted inductivity is at most $\Delta^*(d)$. Consider any $1 \leq i \leq A$. Let $A_i = \{a_1, a_2, \cdots, a_i\}$. Then,

$$N_G^\prec(a_i) = N_G(a_i) \cap A_i,$$

and hence

$$\Gamma\left(N_G^\prec(a_i), a_i; d\right) = \Gamma(A_i, a_i; d).$$

By the selection criteria of a_i, we have

$$\Gamma(A_i, a_i; d) = \min_{b \in V_i} \Gamma(A_i, b; d) \leq \Delta^*(d),$$

which implies

$$\Gamma\left(N_G^\prec(a_i), a_i; d\right) \leq \Delta^*(d).$$

Thus, the d-weighted inductivity of $\prec$ is at most $\Delta^*(d)$.

Therefore, the smallest-last ordering achieves the smallest inductivity $\Delta^*(d)$ among all link orderings. So, the theorem follows. ∎

Next, we present an upper bound on $\Delta^*(d)$. An *orientation* of G is a digraph obtained from G by imposing an orientation on each edge of G. Consider an orientation D of G. For any link $a \in A$, $N_D^{in}(a)$ (respectively, $N_D^{out}(a)$) denotes the set of in-neighbors (respectively, out-neighbors) of a in D. Note that D also has a self-loop at each vertex, and consequently, $a \in N_D^{in}(a) \cap N_D^{out}(a)$ for each $a \in A$. Define

$$\Delta_D^{in}(d) = \max_{a \in A} \Gamma\left(N_D^{in}(a), a; d\right).$$

Theorem 2.6: For any orientation D of G and any $d \in \mathbb{R}_+^A$,

$$\Delta^*(d) \leq 2\Delta_D^{in}(d).$$

The proof of the above theorem utilizes the following lemma, which is a generalization of Lemma 3 in [12].

Lemma 2.7: Suppose that D is an orientation D of G. Then, for any $d \in \mathbb{R}_+^A$ and any nonempty subset B of A, there exists at least one link $a \in B$ satisfying

$$\Gamma\left(N_D^{in}(a) \cap B, a; d\right) \geq \Gamma\left(N_D^{out}(a) \cap B, a; d\right).$$

Now we prove Theorem 2.6. Let B be an non-empty subset of A such that

$$\Delta^*(d) = \min_{b \in B} \Gamma(B,b;d).$$

By Lemma 2.7, there is at least one link $a \in B$ such

$$\Gamma\left(N_D^{in}(a) \cap B, a; d\right) \geq \Gamma\left(N_D^{out}(a) \cap B, a; d\right).$$

Thus,

$$\Gamma(B,a;d)$$
$$= \Gamma\left(N_D^{in}(a) \cap B, a; d\right) + \Gamma\left(N_D^{out}(a) \cap B, a; d\right)$$
$$\quad - c(a,a)d(a)$$
$$\leq 2\Gamma\left(N_D^{in}(a) \cap B, a; d\right) \leq 2\Gamma\left(N_D^{in}(a), a; d\right) \leq 2\Delta_D^{in}(d).$$

Thus,

$$\Delta^*(d) \leq \Gamma(B,a;d) \leq 2\Delta_D^{in}(d).$$

This completes the proof of Theorem 2.6.

C. Weighted Link Schedule

Suppose that $d \in \mathbb{R}_+^A$ is a node-level link-demand function. A *(fractional) link schedule* of $d \in \mathbb{R}_+^A$ is a set

$$\Pi = \left\{ (I_j, \ell_j) \in \mathcal{I}^{\tau,\lambda} \times \mathbb{R}_+ : 1 \le j \le k \right\}$$

satisfying that for each link $a \in A$,

$$d(a) = \sum_{j=1}^k \ell_j \left| I_j \cap \{a\}^{\tau,\lambda} \right|;$$

the two values k and $\sum_{j=1}^k \ell_j$ are referred to as the *size* and *length* (or *latency*) of Π respectively. The minimum length of all link schedules of d is denoted by $\chi^*(d)$. In this subsection, we derive lower bounds on $\chi^*(d)$.

Given a link ordering $\prec$ of A, its *backward local independence number* (BLIN) is defined to be

$$\max_{a \in A} \max \left\{ |I| : I \subseteq N_G^\prec(a), I \in \mathcal{I} \right\}.$$

Given an orientation D of G, its *inward local independence number* (ILIN) is defined to be

$$\max_{a \in A} \max \left\{ |I| : I \subseteq N_D^{in}(a), I \in \mathcal{I} \right\}.$$

Then, we have the following lower bounds on $\chi^*(d)$.

Lemma 2.8: Consider any $d \in \mathbb{R}_+^A$.

1) For any link ordering $\prec$ of A with BLIN μ,

$$\chi^*(d) \ge \frac{\Delta^\prec(d)}{\mu + 2}.$$

2) For any orientation D of G with ILIN μ,

$$\chi^*(d) \ge \frac{\Delta_D^{in}(d)}{\mu + 2}.$$

Proof: We only give the proof of the second part, while remarking that the proof the first part is literally the same. Consider a split x of d given by a shortest link schedule of d. Let $a \in A$ be such that

$$\Delta_D^{in}(d) = \Gamma\left(N_D^{in}(a), a; d \right).$$

By Lemma 2.2, the total x-weighted interference received by some replicated link e of a from all replicated links of $N_D^{in}(a)$ is at least $\Gamma\left(N_D^{in}(a), a; d \right)$. In other words, the total x-weights of all replicated links of $N_D^{in}(a)$ which have interference with e is at least $\Gamma\left(N_D^{in}(a), a; d \right)$. On the other hand, the transmission concurrency of replicated links is limited by $\mu + 2$ as any independent set consisting of these replicated links contains at most μ of those which have the same channel as e and at most two others which have different channel from e but share an radio with e. Thus, the minimum latency required by these links is at least

$$\frac{\Gamma\left(N_D^{in}(a), a; d \right)}{\mu + 2} = \frac{\Delta_D^{in}(d)}{\mu + 2}.$$

This implies that

$$\chi^*(d) \ge \frac{\Delta_D^{in}(d)}{\mu + 2}.$$

So, the second part of the lemma holds. ∎

The corollary below follows immediately from Theorem 2.5, Theorem 2.6, and Lemma 2.8,

Corollary 2.9: Consider any $d \in \mathbb{R}_+^A$.

1) If there is a link ordering of A with BLIN μ, then

$$\chi^*(d) \ge \frac{\Delta^*(d)}{\mu + 2}.$$

2) If there is an orientation of G with ILIN μ,

$$\chi^*(d) \ge \frac{\Delta^*(d)}{2(\mu + 2)}.$$

Since BLIN and ILIN only depend on the topology of G rather than the edge weight function c, the following properties which hold in the convectional conflict graph of the underlying SC-SR wireless network also hold in G:

- Under the 802.11 interference model with uniform interference radii, the lexicographic ordering of A has BLIN at most 6 [3] and it can be computed in in $O(|A| \log |A|)$ time.
- Under the 802.11 interference model with arbitrary interference radii, there is an orientation D of G with ILIN at most 8 which can be constructed in in $O(|A|)$ time [12].
- Under the protocol interference model in which the interference radius of the sender of each link is at least φ times the link length for some $\varphi > 1$, there is an orientation D of G with ILIN $\left\lceil \pi / \arcsin \frac{\varphi - 1}{2\varphi} \right\rceil - 1$ which can be constructed in $O(|A|)$ time [10].

III. Greedy Link Scheduling

In this section, we present a scalable greedy scheduling algorithm for the variant of **SWLS** with the traffic demands given on the *node-level* links. Consider an instance of MC-MR wireless network specified in Section II, and let $G = (A, E; c)$ be its concise conflict graph. For any subset B of A, we use $V(B)$ to denote the set of endpoints of the links in B, and $E(B)$ to denote the set of edges in the subgraph of G induced by B. Note that $E(B)$ contains all the self-loop edges at B.

We first describe an algorithm *GMIS* which takes as input a list B of links $b_i = (u_i, v_i)$ for $1 \le i \le |B|$ in A and outputs a maximal independent I of $B^{\tau,\lambda}$, the subset B' of links in B which have at least one replicated link in I, and the function $g \in \mathbb{R}_+^{B'}$ which counts the number of replicated links in I of each link in B'. Conceptually, the algorithm is a simple greedy one: For each link in B from the head to the tail, the algorithm selects as many replicated links of this link as possible using the radios at its two endpoints which have not been used by any preceding links and the channels which have not been used by any preceding *conflicting* links. An efficient implementation of such greedy approach is described as follows. Both I and B' are empty initially. Each node $v \in V(B)$ maintains a list $R(v)$ of available radios, which is initialized to the list

$$\langle 1, 2, \cdots, \tau(v) \rangle,$$

and a variable to store $|R(v)|$. Each link $a \in B$ maintains a list $C(a)$ of the available channels, which is initialized to the list

$$\langle 1, 2, \cdots, \lambda \rangle,$$

and a variable to store $|C(a)|$. For the purpose of the fast update on the channel lists, each link also maintains a companion array $C'(a)$ indexed by the channel numbers. The j-th element of $C'(a)$ consists of two entries: the first entry is a 0/1 variable, which indicates the availability (respectively, unavailability) of channel j with value 1 (respectively, 0); the second entry is a pointer, which points to null if the first entry is 0, or otherwise points to the channel number j in the list $C(a)$. Initially, for each $1 \le j \le \lambda$, the first entry in the j-th element is 1, and the second entry of the j-th element points to the channel number j in the list $C(a)$. The algorithm runs iteratively for each i incrementally from 1 to $|B|$. In the i-th iteration, we first compute

$$g(b_i) = \min \{|R(u_i)|, |R(v_i)|, |C(a_i)|\}.$$

If $g(b_i) = 0$, we move on to the next iteration. Otherwise, we proceed as follows:

- Add b_i to B'.
- Repeat the following for $g(b_i)$ times: Remove the first radio from $R(u_i)$, the first radio from $R(v_i)$, and the first channel from $C(b_i)$, and then add the replicated link of b_i determined by these three elements together I.
- Decrease both $|R(u_i)|$ and $|R(v_i)|$ by $g(b_i)$.
- For each link b_j with $j > i$ which conflicts with b_i, we update the unavailability of the $g(b_i)$ channels of the above selected replicated links of b_i and then remove them from $C(b_j)$.

After the last iteration, we output the triple (I, g, B).

The correctness of the algorithm *GMIS* is obvious. We show that its running time is

$$O\left(|B|\lambda + |E(B)| \max_{v \in V(B)} \tau(v)\right).$$

Indeed, the total running time taken by the initialization is

$$O\left(\sum_{v \in V(B)} \tau(v) + |B|\lambda\right)$$
$$= O\left(|V(B)| \max_{v \in V(B)} \tau(v) + |B|\lambda\right).$$

The total running time taken by the computation of g is $O(|B|)$. The total running time taken by the computation of I and the updates of the radio lists of nodes is

$$O\left(\sum_{i=1}^{|B|} g(b_i)\right) = O\left(|B| \max_{v \in V(B)} \tau(v)\right).$$

Since the total number of times for which the channels lists of the links are updated is at most $|E(B)|$ and each update takes $O(\max_{v \in V} \tau(v))$ time, the total running time taken by updating the channel lists of the links in B is

$O(|E(B)| \max_{v \in V} \tau(v))$. As $|B| \le |E(B)|$ and $V(B) = O(E(B))$, the total running time of the algorithm is

$$O\left(|B|\lambda + |E(B)| \max_{v \in V(B)} \tau(v)\right).$$

Next, we describe an algorithm *GLS* which takes as input a list of A sorted in some ordering $\prec$ and a link demand function $d \in \mathbb{R}_+^A$ specified by an array, and outputs a (fractional) link schedule Π of d. Initially, Π is empty. The algorithm *GLS* repeats the following iteration until $d = 0$. Let B be the subset of links $a \in A$ with $d(a) > 0$. We apply the algorithm *GMIS* to B sorted in the ordering $\prec$ to output a triple (I, g, B'). Let

$$\ell = \min_{a \in B'} \frac{d(a)}{g(a)},$$

and add (I, ℓ) to Π. For each $a \in B'$, replace $d(a)$ by $d(a) - \ell g(a)$. When $d = 0$, the algorithm outputs Π. We refer to the output Π as the *first-fit fractional schedule of d in the ordering $\prec$*.

The next theorem gives an upper bound on the length of Π.

Theorem 3.1: The algorithm *GLS* has running time

$$O\left(|A|^2 \lambda + |A||E| \max_{v \in V} \tau(v)\right),$$

and the output schedule Π has length at most $\Delta^\prec(d)$.

Proof: Suppose that the algorithm runs in k iterations. For each $1 \le j \le k$, let B_j be the subset B (of links with residue demands) at the beginning of the j-th iteration, and $(I_j, \ell_j) \in \mathcal{I}^{\tau, \lambda} \times \mathbb{R}_+$ be the pair selected in the j-th iteration. Since at least one link gets satisfied in each iteration, the k subsets $B_1, B_2, \cdots, B_k$ are strictly decreasing. Hence,

$$k \le |B_1| \le |A|.$$

Since each iteration has running time at most

$$O\left(|A|\lambda + |E| \max_{v \in V} \tau(v)\right),$$

the total running time of the algorithm is

$$O\left(|A|^2 \lambda + |A||E| \max_{v \in V} \tau(v)\right).$$

So, the first part of the theorem holds.

Consider an arbitrary link $a \in B_k$. Then, $a \in B_j$ for each $1 \le j \le k$. For each $1 \le j \le k$, let A_j be the set of links in B_j preceding (and including) a, and denote $I'_j = I_j \cap A_j^{\tau, \lambda}$. Then, I'_j is also a maximal independent set of $A_j^{\tau, \lambda}$. Let g'_j be the aggregation of $\mathbf{1}^{I'_j}$. By Lemma 2.3,

$$\Gamma(A_j, a; g'_j) \ge 1.$$

Since

$$N_G^\prec(a) \supseteq N_G(a) \cap A_j$$

we have

$$\Gamma(N_G^\prec(a), a; g'_j) \ge \Gamma(A_j, a; g'_j) \ge 1.$$

Note that for each link $b \in N_G^\prec (a)$,

$$d(b) = \sum_{j=1}^k \ell_j g_j'(b).$$

Thus,

$$
\begin{aligned}
\Delta^\prec (d) &\geq \Gamma\left(N_G^\prec(a), a; d\right) = \sum_{b \in N_G^\prec(a)} c(b,a) d(b) \\
&= \sum_{b \in N_G^\prec(a)} c(b,a) \sum_{j=1}^k \ell_j g_j'(b) \\
&= \sum_{j=1}^k \ell_j \sum_{b \in N_G^\prec(a)} c(b,a) g_j'(b) \\
&= \sum_{j=1}^k \ell_j \Gamma\left(N_G^\prec(a), a; g_j'\right) \geq \sum_{j=1}^k \ell_j.
\end{aligned}
$$

So, the second part of the theorem holds. ∎

Finally, we are ready to describe our approximation algorithm for **SWLS** with the traffic demands given on the node-level links. Let $d \in \mathbb{R}_+^A$ be the input link demand function. Our algorithm consists of two steps:

- Step 1: Compute the smallest-last ordering $\prec$ of (G, d).
- Step 2: Apply the algorithm *GLS* to compute a first-fit fractional schedule of d in the ordering $\prec$.

By Theorem 3.1 and Theorem 2.5, the output link schedule has length at most $\Delta^*(d)$. By Corollary 2.9, it achieves the following approximation bounds:

- 8 under the 802.11 interference model with uniform interference radii;
- 20 under the 802.11 interference model with arbitrary interference radii;
- $2\left(\left\lceil \pi / \arcsin \frac{\varphi - 1}{2\varphi} \right\rceil + 1\right)$ under the protocol interference model in which the interference radius of the sender of each link is at least φ times the link length for some $\varphi > 1$.

The running time of Step 1 is $O(E)$, and by Theorem 3.1 the running time of the second step is

$$O\left(|A|^2 \lambda + |A| |E| \max_{v \in V} \tau(v)\right).$$

Thus, the total running time is

$$O\left(|A|^2 \lambda + |A| |E| \max_{v \in V} \tau(v)\right),$$

which grows *linearly* with the number of channels and the maximum number of radios of all nodes.

IV. POLYNOMIAL APPROXIMATE CAPACITY SUBREGIONS

Consider an instance of MC-MR wireless network specified in Section II. Its capacity region is defined to be

$$P = \left\{d \in \mathbb{R}_+^A : \chi^*(d) \leq 1\right\}.$$

In general, the membership of the capacity region P is NP-complete. A subregion Q of P is called a *polynomial μ-approximate capacity subregion* for some $\mu \geq 1$ if it satisfies the following three conditions: (1) Q has an explicit representation by a polynomial number of linear inequalities or equalities, (2) there is a polynomial algorithm which produces a fractional schedule of length at most one for any $d \in Q$,

and (3) Q is a μ-approximation of P i.e., $P \subseteq \mu Q$. The first condition ensures the membership of Q is polynomial, the second condition implies that $Q \subseteq P$, and the third condition ensures that Q is "close" to P. In this section, we present two polynomial capacity subregions directly on A.

Let G be the concise conflict graph of the MC-MR wireless network. For any link ordering $\prec$ of A, its *backward capacity subregion* is defined to be

$$Q^\prec = \left\{d \in \mathbb{R}_+^A : \Delta_G^\prec(d) \leq 1\right\}.$$

Theorem 4.1: For any link ordering $\prec$ of A with BLIN μ, $Q^\prec$ is a polynomial $(\mu + 2)$-approximation capacity subregion.

Proof: Clearly, $Q^\prec$ is defined by $|A|$ linear inequalities. By Theorem 3.1, for any $d \in Q$, the algorithm *GLS* in the ordering $\prec$ would produce a fractional link schedule of length at most $\Delta^\prec(G, d) \leq 1$. In addition, for any $d \in P$, by Lemma 2.8,

$$\Delta^\prec(d) \leq (\mu + 2) \chi^*(d) \leq \mu + 2,$$

which implies $d \in (\mu + 2) Q^\prec$. Thus, $P \subseteq (\mu + 2) Q^\prec$. So, the theorem follows. ∎

For any orientation D of G, its *inward capacity subregion* of G is defined to be

$$Q^D = \left\{d \in \mathbb{R}_+^A : \Delta_D^{in}(d) \leq 1/2\right\}.$$

Theorem 4.2: For any orientation D of G with ILIN μ, Q^D is a polynomial $2(\mu + 2)$-approximation capacity subregion.

Proof: Clearly, Q^D is defined by $|A|$ linear inequalities. By Theorem 2.6, $\Delta^*(d) \leq 1$ for any $d \in Q$. By Theorem 3.1 and Theorem 2.5, for any $d \in Q$, the algorithm *GLS* in the smallest-last ordering would produce a fractional link schedule of length at most $\Delta^*(d) \leq 1$. In addition, for any $d \in P$, by Lemma 2.8,

$$\Delta_D^{in}(d) \leq (\mu + 2) \chi^*(d) \leq \mu + 2,$$

which implies $d \in 2(\mu + 2) Q^D$. Thus, $P \subseteq 2(\mu + 2) Q^D$. So, the theorem follows. ∎

Theorem 4.1 and Theorem 4.2 imply the following specific polynomial approximate capacity subregions:

- Under the 802.11 interference model with uniform interference radii, the backward capacity subregion of the lexicographic ordering of A is a polynomial 8-approximate capacity subregion.
- Under the 802.11 interference model with arbitrary interference radii, the inward capacity subregion of the orientation D of G described in [12] is a polynomial 20-approximate capacity subregion.
- Under the protocol interference model in which the interference radius of the sender of each link is at least φ times the link length for some $\varphi > 1$, the inward capacity subregion of the orientation D of G described in [10] is a polynomial $2\left(\left\lceil \pi / \arcsin \frac{\varphi - 1}{2\varphi} \right\rceil + 1\right)$-approximate capacity subregion.

V. RESTRICTED MULTIFLOW

In this section, we present scalable approximation algorithms for **MMF** and **MCMF**. Consider an instance of MC-MR wireless network specified in Section II. For each node $v \in V$, we use $\delta^{in}(v)$ (respectively, $\delta^{out}(v)$) to denote the set of links in A entering (respectively, leaving) v. Consider two distinct nodes $s, t \in V$. A vector $f \in \mathbb{R}_+^A$ is called a *flow* from s to t, or simply an *s–t flow*, if for each $v \in V \setminus \{s, t\}$,

$$f\left(\delta^{out}(v)\right) = f\left(\delta^{in}(v)\right).$$

This condition is called the *flow conservation law*: the amount of flow entering a vertex $v \neq s, t$ should be equal to the amount of flow leaving v. The *value* of a flow f from s to t is, by definition:

$$val(f) = f\left(\delta^{out}(s)\right) - f\left(\delta^{in}(s)\right).$$

So, the value is the net amount of flow leaving s, which is also equal to the net amount of flow entering t. Suppose that we are given k unicasts with s_i, t_i being the source and sink, respectively, for commodity i. We use $\mathcal{F}_i$ to denote the set of s_i–t_i flows. A k-flow is a sequence of flows $\langle f_1, f_2, \cdots, f_k \rangle$ with $f_i \in \mathcal{F}_i$ for each $1 \leq i \leq k$.

Suppose that Q is a polynomial μ-approximate capacity subregion. A k-flow $\langle f_1, f_2, \cdots, f_k \rangle$ is said to be *Q-restricted* if $\sum_{i=1}^k f_i \in Q$. The maximum Q-restricted multiflow is defined by the following LP

$$\begin{array}{ll} \max & \sum_{i=1}^k val(f_i) \\ s.t. & f_i \in \mathcal{F}_i, \forall 1 \leq i \leq k; \\ & \sum_{i=1}^k f_i \in Q. \end{array}$$

This LP is of polynomial size and we solve this Q-restricted LP in polynomial time to obtain a k-flow $\langle f_1, f_2, \cdots, f_k \rangle$. Then we compute a fractional link schedule of length at most one for $\sum_{i=1}^k f_i$. Such link schedule is a μ-approximate solution. Similarly, the maximum concurrent Q-restricted multiflow with demands d_i for $1 \leq i \leq k$ is defined by the following LP

$$\begin{array}{ll} \max & \phi \\ s.t. & f_i \in \mathcal{F}_i, \forall 1 \leq i \leq k; \\ & val(f_i) \geq \phi d_i, \forall 1 \leq i \leq k; \\ & \sum_{i=1}^k f_i \in Q. \end{array}$$

We first solve this Q-restricted LP of polynomial size in polynomial time to obtain a k-flow $\langle f_1, f_2, \cdots, f_k \rangle$, and then compute a fractional link schedule of length at most one for $\sum_{i=1}^k f_i$. This link schedule is a μ-approximate solution.

We choose the polynomial approximate capacity subregion Q constructed in Section IV:

- Under the 802.11 interference model with uniform interference radii, Q is the backward capacity subregion of the lexicographic ordering of A.
- Under the 802.11 interference model with arbitrary interference radii, Q is the inward capacity subregion of the orientation D of G described in [12].

- Under the protocol interference model, Q is the inward capacity subregion of the orientation D of G described in [10].

In either case, the LP for the maximum Q-restricted multiflow has $k|A|$ flow variables, and the LP for the maximum concurrent Q-restricted multiflow has $k|A|$ flow variables and an additional variable ϕ. These LP can be solved in $O\left((k|A|)^3\right)$ time, which depends on neither the number of radios or nor the number of channels. We apply the algorithm described in Section III to compute a fractional link schedule of length at most one for $\sum_{i=1}^k f_i$, which takes

$$O\left(|A|^2 \lambda + |A|(|V| + |E|) \max_{v \in V} \tau(v)\right)$$

time. Therefore, the overall running time is

$$O\left((k|A|)^3 + |A|^2 \lambda + |A|(|V| + |E|) \max_{v \in V} \tau(v)\right),$$

which grows linearly with λ and $\max_{v \in V} \tau(v)$. The output link schedule is an 8-approximation solution under the 802.11 interference model with uniform interference radii, a 20-approximate solution under the 802.11 interference model with arbitrary interference radii, and a $2\left(\left\lceil \pi/\arcsin \frac{\varphi-1}{2\varphi} \right\rceil + 1\right)$-approximate solution under the protocol interference model in which the interference radius of each node is at least φ times its communication radius for some $\varphi > 1$.

We remark that our algorithms have the same approximation bound as those developed in [11], but are significantly faster. Indeed, the running time of the algorithms in [11] is at least $O\left(\left(k\left|A^{\tau,\lambda}\right|\right)^3\right)$, where $A^{\tau,\lambda}$ is the number of replicated links of all links in A. For the illustrative purpose, we consider the uniform radio setting in which each node has τ radios. Then, $\left|A^{\tau,\lambda}\right| = \lambda \tau^2 |A|$, and

$$O\left(\left(k\left|A^{\tau,\lambda}\right|\right)^3\right) = \lambda^3 \tau^6 O\left((k|A|)^3\right).$$

Thus, the running time of the algorithms in [11] grows with the *product* of the *cubic* order of the number of channels and the *sixth* order of the number of radios per node. In contrast, the running time of our algorithm grows *linearly* with the number of channels and the number of radios per node.

VI. RELATED WORKS

Wireless link scheduling in MC-MR multihop wireless networks have been studied recently by a few research works [1], [2], [4], [5], [6], [8], [9]. Among these works, Lin and Rasool [6] studied the queuing-theoretic aspects of link scheduling in MC-MR multihop wireless networks under the 802.11 interference model with uniform interference/communication radii. Alicherry et al. [1] proved an approximation bound $8\lambda/\min_{v \in V} \tau(v)$ on their approximation algorithms for maximum concurrent multiflow under the 802.11 interference model with uniform interference radii. Han et al. [2] gave an approximation algorithm for the maximum (concurrent) multiflow in MC-MR multihop wireless networks, which achieves an approximation bound of 122 under the 802.11 interference

model with arbitrary interference/communication radii. Wan et al. [11] developed approximation algorithms for maximum (concurrent) multiflow in MC-MR wireless networks under either 802.11 or protocol interference model. The approximation bounds of their algorithms are at most 8 under the 802.11 interference model with uniform interference radii, at most 20 under the 802.11 interference model with arbitrary interference radii, and at most $2\left(\left\lceil \pi / \arcsin \frac{\varphi-1}{2\varphi} \right\rceil + 1\right)$ under the protocol interference model in which the interference radius of the sender of each link is at least φ times the link length for some $\varphi > 1$. In addition, if the number of channels λ is bounded by a constant, they presented a PTAS under the 802.11 interference model or under the protocol interference model with some additional mild conditions.

VII. Conclusion

In this paper, we introduced the concise conflict graph of the node-level links in a MC-MR wireless network. Instead of creating many replications at the radio-channel level of all node-level links as in the fine-grained conflict graph [11], the concise conflict graph simply assigns a weight to each conflicting pair of links which captures the essential benefit of multiple radios and multiple channels. We have explored a number of fundamental structural properties of the concise conflict graph. By exploring these properties, we developed a first-fit link scheduling algorithm for **SWLS** whose running time grows linearly with the number of channels and the maximum number of radios at the individual nodes. We also constructed two polynomial approximate capacity subregions defined on the node-level links directly. By restricting to these polynomial approximate capacity subregions, the maximum (concurrent) multiflow can be approximated within the same approximation bound as achieved in [11] but in a running time growing linearly with the number of channels and the maximum number of radios at individual nodes. In contrast, the algorithms for maximum (concurrent) multiflow developed in [11] have running time growing in the cubic order of the number of channels and the sixth order of the maximum number of radios at individual nodes. We expect that the concise conflict graph and its structural properties established in this paper have applications in the design and analyses of other optimization problems in MC-MR wireless networks.

References

[1] M. Alicherry, R. Bhatia, and L.E. Li, Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks, IEEE Journal on Selected Areas in Communications 24(11):1960–1971, Nov. 2006. Also appeared in *Proc. of ACM Mobi-Com*, 2005.

[2] 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.

[3] C. Joo, X. Lin, and N. B. Shroff, Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks, in *Proc. IEEE INFOCOM* 2008, April 2008.

[4] M. Kodialam and T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, *Proc. of ACM MobiCom*, 2005.

[5] P. Kyasanur and N. Vaidya, Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks, *IEEE Wireless Communications and Networking Conference*, New Orleans, LA, March 2005.

[6] X. Lin and S. Rasool, A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad-hoc Wireless Networks, *Proc. IEEE INFOCOM* 2009, pp. 1118-1126.

[7] D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms, *Journal of the Association of Computing Machinery* 30(3):417-427, 1983.

[8] A. Raniwala and T.-C. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in *Proc. IEEE INFOCOM*, 2005, pp. 2223–2234.

[9] A. Raniwala, K. Gopalan, and T.-C. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, *ACM Mobile Comput. Commun. Rev. (MC2R)*, vol. 8, no. 2, pp. 50–65, 2004.

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

[11] P.-J. Wan, Y. Cheng, Z. Wang, and F. Yao, Multiflows in Multi-Channel Multi-Radio Multihop Wireless Networks, *Proc. IEEE INFOCOM* 2011, pp. 846-854.

[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, in *Proc. IEEE INFOCOM* 2011, pp. 2327-2335.