Journal of Interconnection Networks
Vol. 12, Nos. 1 & 2 (2011) 85–107

DOI: 10.1142/S021926591100285X

MINIMUM LATENCY LINK SCHEDULING FOR ARBITRARY DIRECTED ACYCLIC NETWORKS UNDER PRECEDENCE AND SINR CONSTRAINTS

QIANG-SHENG HUA*
Institute for Interdisciplinary Information Sciences,
Tsinghua University, 100084, Beijing, P.R. China
qshua@mail.tsinghua.edu.cn

YUEXUAN WANG
Institute for Interdisciplinary Information Sciences,
Tsinghua University, 100084, Beijing, P.R. China
wangyuexuan@mail.tsinghua.edu.cn

DONGXIAO YU
Department of Computer Science,
The University of Hong Kong, Pokfulam, Hong Kong, P.R. China
dxyu@cs.hku.hk

HAISHENG TAN
Institute for Interdisciplinary Information Sciences,
Tsinghua University, 100084, Beijing, P.R. China
tan@mail.tsinghua.edu.cn

Received 20 February 2011
Accepted 23 November 2011

Scheduling wireless links under the SINR model has attracted increasing attention in the past few years [1–6, 8–15, 18–20, 23–25, 27, 28, 33–36, 39, 41, 42, 45, 46]. However, most of previous work did not account for the precedence constraint that might exist among the wireless links. Precedence constraints are common in data aggregation problems where a sensor can not send data to its parent node before it has received data from all of its children. Existing solutions to the so-called minimum latency aggregation scheduling problem [7, 16, 21, 26, 29, 30, 32, 40, 43, 44] mainly focus on specific tree topologies rooted at the sink node. In this paper, we study the minimum latency link scheduling problem for arbitrary directed acyclic networks under both precedence and SINR constraints. Our formulation allows multiple sinks, and each sensor may transmit data to more than one parent node. We first show that the problem is NP-hard, and then propose a linear power assignment based polynomial time approximation algorithm and a dynamic labeling based heuristic algorithm. We have carried out extensive simulations for both dense and sparse

*This work was mainly done when the author was a researcher in Department of Computer Science, The University of Hong Kong.

arbitrary directed acyclic networks. The simulation results show that: (1) compared with both uniform and linear power assignments based algorithms, we can achieve much shorter scheduling lengths using our proposed labeling algorithm, and (2) the dynamic labeling based heuristic algorithm can lead to significantly shorter scheduling lengths than the heuristic algorithm which does not use labeling.

Keywords: Minimum latency link scheduling; SINR model; physical interference model; directed acyclic networks; precedence constraints.

1. Introduction

1.1. *Problem Motivation*

Recently there has been a surge of interest in studying the minimum length link scheduling problem under the SINR (Signal-to-Interference-plus-Noise-Ratio) model (also known as the physical interference model) [2, 3, 8, 11–13, 15, 24, 27, 34, 36]. Given a set of n arbitrarily constructed links on the plane, the objective of this problem is to use the minimum number of timeslots to schedule all the links such that the SINR ratios at all simultaneously scheduled links are greater than or equal to the threshold values as stipulated by the SINR model. All previous work, however, did not take the precedence constraints of the links into account. These precedence constraints exist naturally in data aggregation problems where a sensor can not send the data to its parent node if it has not received the data from all of its children. An example of precedence is when a node needs to compute the maximum or average value of all the collected data from other nodes. Very recently, there has been some work on this kind of minimum latency aggregation scheduling problem (**MLAS**) [7, 16, 21, 26, 29, 30, 32, 40, 43, 44], but (1) except the work in [16, 21, 29, 30, 32], most of them study the scheduling problem based on the graph based interference models instead of the SINR model; and (2) all of the current work study the joint problem of tree topology construction and link scheduling where each sensor only needs to send their aggregated data to one parent node. In this paper, we study the problem of minimum latency link scheduling for arbitrary directed acyclic networks under both precedence and SINR constraints (**MLSDAN**). Here by the minimum latency link scheduling, we mean to use the minimum number of time slots to schedule all the links in the arbitrary directed acyclic networks while satisfying the precedence and SINR constraints. Compared with the MLAS problem, the MLSDAN problem allows multiple sinks whereby each sensor may need to transmit data to more than one parent node. In addition, in the MLAS problem, given the nodes arbitrarily located on the plane, the algorithm can take advantage of the specifically constructed tree topology to facilitate the scheduling process (e.g., [16, 21, 29, 30, 32]). Whereas in our MLSDAN problem, an arbitrary network topology is given and thus the algorithm loses the freedom of constructing its own network topology to help the link scheduling process.

1.2. *Our Contribution*

First, we show that the MLSDAN problem is NP-hard. Second, we propose a polynomial time approximation algorithm with a provable performance guarantee. The approximation ratio is $O(min\{d(dan) \cdot C_{max}, CP \cdot d(dan)\})$ (min is a function to choose the minimum value). Here $d(dan)$ is the length diversity of all the links in the directed acyclic network dan(cf. Equation (2.2)); CP is the length of the critical path in dan (cf. subsection 3.2) and C_{max} means the maximum number of links whose receivers are located in a cell (cf. subsection 4.2). To our knowledge, this is the first polynomial time approximation algorithm for the MLSDAN problem. Third, we present a dynamic labeling based scheduling algorithm and extensive simulation results to show that our algorithm outperforms many frequently used link scheduling algorithms in terms of scheduling lengths.

1.3. *Related Work*

The MLAS problem which was first proposed by Chen, Hu and Zhu in [7]. Improvements were later given by Wan et al. in [26, 40] and Yu et al. in [44] which are based on either the unit disk graph model or the disk graph model. Xu et al. have also studied the MLAS problem in the protocol interference model [43]. Currently, the only results for the MLAS problem under the SINR model can be found in [16, 21, 29, 30, 32]. All of these work involve their own tree topology construction algorithms in order to expedite the link scheduling process.

We now briefly review the papers that study the minimum length link scheduling problem under the SINR model. For a more detailed survey of this line of research, please refer to [14, 25]. First, a line of heuristic algorithms aiming to minimize the scheduling lengths can be found in [2, 3, 8]. Tang et al. in [39] have formulated the link scheduling problem as an integer programming problem and proposed some heuristic algorithms based on the relaxed linear programming method. A seminal paper which uses an elegant non-linear power assignment was proposed by Moscibroda and Wattenhofer in [33]. Basically, this paper shows that a connected wireless topology can be constructed in $O(log^4 n)$ timeslots where n is the number of nodes. This paper has spawned a large body of following works that either aim to reduce the scheduling lengths [4, 13, 16, 21, 23, 34–36] or to analyze the hardness of the scheduling problem [12, 15, 27]. In addition, since the non-linear power assignment based scheduling algorithm needs the information of all the links such as their lengths, there are some recent works that focus on oblivious power assignment methods. For example, Fanghänel et al. have proposed a so-called square-root or mean power assignment that is based on each link's own length [11]. This square-root power assignment has also been used by Halldorsson in [18]. Note that, both constant [4, 13, 15] and linear power assignments [10] are also oblivious power assignments since these strategies also only need to know each link's own length information. Different from heuristic algorithms, all these non-linear or oblivious power assignment based algorithms have worst-case performance guarantees. In addition to heuristic and approximation algorithms, an exponential time exact algorithm has been proposed by Hua and Lau in [24]. By also considering routing in the minimum length link scheduling scenario, a so-called cross-layer latency minimization problem in the SINR model was first investigated by Chafekar et al. in [5] and an improved algorithm has been given by Fanghänel et al. in [10].

Besides the multitude of studies on the minimum length link scheduling problem, there have also been many recent works on the (weighted) capacity maximization problem in the SINR model. The capacity maximization problem asks for the maximum number of links that can be simultaneously scheduled in the same timeslot under the SINR model. The weighted version assigns some weight value to each link. Note that, by using the master-slave strategy [13], an approximation algorithm for the capacity maximization problem with approximation ratio ρ translates automatically to an approximation algorithm for the minimum length link scheduling problem with approximation ratio $logn/\rho$. The papers $[1, 6, 9, 20, 28, 41, 42]$ belong to this category.

2. Preliminaries and Problem Definition

2.1. *System Model*

We have the following assumptions. First, by a wireless link, we mean a wireless transmission comprising a source node (transmitter or sender) and a destination node (receiver); second, we assume all the stationary wireless nodes are arbitrarily located on a plane, and each node is equipped with an omni-directional antenna; third, we assume a single channel and half-duplex mode, which means each node can not send to or receive from more than one node, nor to receive and send at the same time; fourth, we employ the physical interference model, or the signal-to-interference-plus-noise ratio (SINR) model [17]. The SINR model requires that only when the SINR ratios at all the receivers are above some threshold value can these links be scheduled in the same timeslot. More specifically, the SINR ratio at the receiver of a link $i = (i_s, i_r)$ can be represented as:

$$SINR_i = \frac{g_{ii} \cdot p_i}{n_i + \sum_{j \neq i} g_{ij} \cdot p_j} \geq \beta \tag{2.1}$$

Briefly speaking, the numerator $g_{ii} \cdot p_i$ means the received power at i_r and $g_{ij} \cdot p_j$ in the denominator means the attenuated power of p_j at i_r and it is regarded as the interference power for link i. Thus $\sum_{j \neq i} g_{ij} \cdot p_j$ is the accumulated interference caused by all the other simultaneous transmissions. More specifically, p_i denotes the transmission power of link i's transmitter i_s; n_i is the background noise at link i's receiver i_r (Note that, throughout of the paper, we will assume all the links' receivers have the same background noise n_0); $g_{ii} = 1/d^{\alpha}(i_s, i_r)$ and $g_{ij} = 1/d^{\alpha}(j_s, i_r)$ are the link gain from i_s to i_r, and that from the transmitter j_s of link j to i_r, respectively; here $d(,)$ is the Euclidean distance function and α is the path loss exponent which ranges between 2 and 6; β is the SINR threshold which is larger than 1.

Based on the SINR inequality, we define a non-negative link gain matrix $H = (h_{ij})$ such that $h_{ij} = \beta \cdot g_{ij}/g_{ii}$ for $i \neq j$ and $h_{ij} = 0$ for $i = j$. We also define a noise vector $\eta = (\eta_i)$ such that $\eta_i = \beta \cdot n_i/g_{ii}$. With these definitions, we can rewrite the SINR inequality as $p_i \geq \sum_{j=1} h_{ij} \cdot p_j + \eta_i$. By using a vector-matrix notation, the above inequality becomes $P \geq HP + \eta$ or $(I - H)P \geq \eta$ where $P = \{p_i\}$. Finally we define the spectral radius $\rho(H) = \max_i |\lambda_i(H)|$ where $\lambda_i(H)$ stands for the ith eigenvalue of H. Now according to [22, 38], we know that the matrix H is a non-negative irreducible matrix, and the following useful properties of the H matrix hold:

Property 1: $(I - H)^{-1} > 0$ if and only if $\rho(H) < 1$;

Property 2: The power vector $P^* = (I - H)^{-1} \cdot \eta$ is Pareto-optimal in the sense that $P^* \geq P$ component-wise for any other nonnegative P satisfying $(I - H)P \geq \eta$.

2.2. *Problem Definition*

The Minimum Latency Link Scheduling for Arbitrary Directed Acyclic Networks with Precedence and SINR Constraints Problem (MLSDAN): First we are given an arbitrary directed network that consists of n wireless links (directed edges); second, the directed edges or links represent the execution dependencies of the wireless nodes. For a link i, we call its source node (transmitter) i_s as the child node and its destination node (receiver) i_r as the parent node. Then the precedence constraints of the wireless links require that each wireless node can not transmit data to any of its parent node until it has received the data from all of its children. The MLSDAN problem is to decide in each timeslot which links to transmit and at what power levels so that the totally used timeslots to schedule all the links are minimized. Note that all the links transmitted in the same timeslot must satisfy the SINR constraints. An example of the directed network that comprises 9 links is given in Fig. 1. In this example, the link $(3, 1)$ can not be scheduled before links $(8, 3)$, $(9, 3)$ and $(4, 3)$. Also the link $(4, 3)$ can not be scheduled before links $(8, 5)$ and $(5, 4)$.

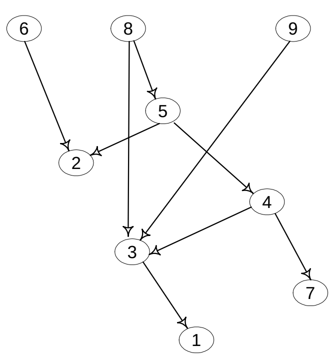

Fig. 1. A directed acyclic network with 3 sinks and 9 links.

2.3. *Other Definitions*

First, we define the length diversity $d(L)$ of a family of links $L = \{i\}$. Basically, it defines the number of length magnitudes of all the links in this collection of links. Let c denote any positive constant greater than 1, then the definition of length diversity is given below:

$$d(L) = |\{m|\exists i \in L : \lfloor \log_c(d(i_s, i_r)) \rfloor = m\}| \tag{2.2}$$

Second, let ρ be some constant greater than or equal to βn_i where β is the SINR threshold and n_i is the background noise at link i's receiver i_r, we then formally define linear power assignment for each link i as follows:

$$p_i = \rho \cdot (d(i_s, i_r))^\alpha \tag{2.3}$$

Third, let L be a set of concurrently scheduled links, then we formally define uniform (constant) power assignment for each link i as follows:

$$p_i = max_{j \in L}\{\rho \cdot (d(j_s, j_r))^\alpha\} \tag{2.4}$$

Finally, if each link i assigns its power based on the Pareto-optimal power vector of the link gain matrix H which is calculated according to Property 2, we call it a Pareto-optimal power assignment.

3. Problem Hardness and Basic Labeling Algorithms

3.1. *Problem Hardness*

Theorem 3.1. The decision version of the MLSDAN problem is NP-complete.

Proof. First we show that the decision version of the MLSDAN problem is in NP. In other words, given an instance of the MLSDAN problem and a number T, we can verify the given answer in polynomial time. The verification process takes four steps: (1)to check whether the used timeslots is less than or equal to T; (2)to check whether the given scheduling process satisfy the precedence constraints of the wireless links; (3)to check whether all the links scheduled in each timeslot meet the SINR requirement; and (4)all the links have been scheduled once. It's easy to know that all these four steps take polynomial time.

Now we show that, no matter whether we allow power control, the MLSDAN problem is NP-hard. First, by using a reduction from the Partition problem, Goussevskaia, Oswald and Wattenhofer have shown that the minimum length link scheduling problem under constant (uniform) power assignment (no power control) is NP-hard [15]. Second, by extending the proof in [15], Katz, Völker and Wagner proved that the minimum length link scheduling problem under power control is also NP-hard [27]. Since the minimum length link scheduling problem is only a special case of the MLSDAN problem, we can see that, no matter we allow power control or not, the MLSDAN problem is NP-hard. □

3.2. *Basic Labeling Algorithms*

In this subsection, we will give three basic labeling algorithms which assign each wireless link in the directed acyclic network some specific value. But first, we need to give an algorithm, Algorithm 1, which tells how to sort all the wireless links based on the topological order of the wireless nodes in the network. In this algorithm, topological_sort(dan) is a frequently used topological sorting procedure that returns a linear ordering of the nodes in the directed acyclic network dan. Similar to the topological ordering of the nodes, the topological ordering of the links, i.e., Algorithm 1, tells us that in the linear ordering of the links, each precedence constraint between the links (cf. the precedence constraint example in the MLSDAN problem definition) in the network must also be obeyed in the linear ordering of the links.

Algorithm 1 Topological Order of the Wireless Links.

Input: An arbitrary directed acyclic network called dan
Output: $lorder$: topological order of the wireless links

```
lorder := ∅;
norder :=topological_sort(dan); //topological order of the nodes
For each node i in norder do
   For each child node j of norder(i) do
      if j is not empty then lorder := [lorder; [i, j]];
   End For
End For
```

Now according to Algorithm 1, we give three labeling algorithms in order. Algorithm 2 computes each link's $tvalue$ and all the links with $tvalue = 1$ are called entry links. Since the transmitters in the entry links do not have child nodes, these entry links can start scheduling immediately. Formally, the $tvalue$ of a link i means the length of the longest path from an entry link to link i, and the length of a path is the number of links along the path. Thus the $tvalue(i)$ highly correlates with link i's earliest scheduling time. Algorithm 3 computes each link's $bvalue$ and all the links with $bvalue = 1$ are called exit links. All the receivers in the exit links do not need to transmit since they have no receivers. Formally, the $bvalue$ of a link i means the length of a longest path from link i to an exit link. From these definitions, we can see that both $tvalue$ and $bvalue$ are bounded by the length of the critical path (denoted as CP), i.e., the longest path in the whole directed acyclic network. Thus we have the following Lemma 3.1 and Table 1 gives the corresponding $tvalue$ and $bvalue$ for the network topology example given in Fig. 1.

Lemma 3.1. *The scheduling length of the MLSDAN problem is lower bounded by the length of the critical path of the directed acyclic network, i.e.,* $CP = max_i\{tvalue(i)\} = max_i\{bvalue(i)\}$.

Algorithm 2 Labeling each wireless link with its *tvalue*.

Input: An arbitrary directed acyclic network called *dan*
Output: Each link is labeled with its *tvalue*

```
Computing lorder of the wireless links using Algorithm 1;
For each link i in lorder do
   tvalue(i):=0;
End For
For each link i in lorder do
   max:=1;
   For each link j whose receiver is link i's transmitter do
      if tvalue(j)+1≥ max
         max:=tvalue(j)+1;
      end if
   End For
   tvalue(i):=max;
End For
```

Algorithm 3 Labeling each wireless link with its *bvalue*.

Input: An arbitrary directed acyclic network called *dan*
Output: Each link is labeled with its *bvalue*

```
Computing lorder of the wireless links using Algorithm 1;
Reverse lorder.
For each link i in lorder do
   bvalue(i):=0;
End For
For each link i in lorder do
   max:=1;
   For each link j whose transmitter is link i's receiver do
      if bvalue(j)+1≥ max
         max:=bvalue(j)+1;
      end if
   End For
   bvalue(i):=max;
End For
```

The first two labeling algorithms are based on the observations from the precedence constraints of the directed acyclic network. Now we give the third labeling algorithm, Algorithm 4, that is based on the following proposition of infeasible power assignment.

Proposition 1 ([23]) *For any two wireless links* $i = (i_s, i_r)$ *and* $j = (j_s, j_r)$*, if* $d(i_s, j_r) \cdot d(j_s, i_r) \leq \beta^{\frac{2}{\alpha}} \cdot d(i_s, i_r) \cdot d(j_s, j_r)$*, then there are no feasible power assignments to simultaneously schedule these two links.*

Table 1. Topological ordering of the wireless links in Fig. 1 and their two corresponding labeling values.

Links with topological order	tvalue	bvalue
9,3	1	2
8,3	1	2
8,5	1	4
5,2	2	1
5,4	2	3
4,3	3	2
4,7	3	1
3,1	4	1
6,2	1	1

The first node is the link's transmitter and the second node is the link's receiver.

Algorithm 4 Computing each link's $conflict$ value.

Input: An arbitrary directed acyclic network called dan
Output: The $conflict$ value for each link

Computing $lorder$ of the wireless links using Algorithm 1;
For each link i in $lorder$ **do**
  $conflict(i) := 0;$
  For each link j in $lorder$ **do**
    if $d(i_s, j_r) \cdot d(j_s, i_r) \leq \beta^{\frac{2}{\alpha}} \cdot d(i_s, i_r) \cdot d(j_s, j_r)$
      $conflict(i) := conflict(i) + 1;$
    end if
  End For
End For

3.3. *Labeling Algorithms Time Complexities*

In this subsection, we will give the time complexities for the four algorithms presented in the last subsection. Since there are n links and at most n nodes in the directed acyclic network, we can easily know that the time complexities for Algorithms 1, 2, 3 are all $O(n)$, where n is the number of links. Similarly we can conclude the time complexity for Algorithm 4 is $O(n^2)$, where n is the number of links.

4. LPA: A Linear Power Assignment based Approximation Algorithm

In this section, we will give a linear power assignment based polynomial time approximation algorithm for the MLSDAN problem. Basically, this algorithm works as follows:

(1) We first pick all the entry links, i.e., the links $L = \{i|tvalue(i) = 1\}$ (cf. line 4 in Algorithm 5), which guarantees there are no precedence constraints among the links;

(2) For all the links in L, the algorithm first picks the wireless link $i \in L$ that has the largest $bvalue(i)$ value since this link is in the critical path of the network(cf. line 6 in Algorithm 5); then it will schedule all the other links in L that (a) have similar lengths to link i and (b) sufficiently apart from link i. This is realized by 3-coloring of all the links in L_k (cf. line 6 in Algorithm 5) and picking the links meeting conditions (a) and (b) will guarantee they satisfy the SINR constraints by using the linear power assignment (cf. lines 7-15 in Algorithm 5);

(3) Repeat step (2) until all links in L have been scheduled (cf. lines 16,17 in Algorithm 5);

(4) Remove the links L in the directed acyclic network dan and repeat all the above three steps until all links have been scheduled (cf. lines 18,19 in Algorithm 5).

The complete algorithm can be seen in Algorithm 5.

4.1. *Correctness Analysis*

According to line 12 in Algorithm 5, we need to show that all links in S_t satisfy SINR constraints by using the linear power assignment. As mentioned earlier, in order to guarantee all the SINR constraints, the simultaneously scheduled links must have similar lengths and all of them are sufficiently apart. For links with similar lengths, in each timeslot, the algorithm chooses to schedule the links L_k such that each link $i \in L_k$ satisfies $c^k \leq d(i_s, i_r) < c^{k+1}$ (cf. line 6). In order to make the simultaneously scheduled links in L_k are sufficiently apart, similar to [12], we will 3-color of all the links in L_k with hexagons of side length $W = \mu \cdot c^k$ (cf. line 7). The color of each link in L_k equals the color of the cell (hexagon) that the link's receiver belongs to (cf. line 11). Now we show that all links in S_t do not share a common node and also meet the SINR requirements.

First, since (1)each link $j \in L_k$ satisfies $c^k \leq d(j_s, j_r) < c^{k+1}$; and (2)the side length of all the hexagons $W = \mu \cdot c^k = \frac{4}{\sqrt{3}} c(12\beta\frac{\alpha-1}{\alpha-2})^{\frac{1}{\alpha}} \cdot c^k \geq 2 \cdot c^{k+1}$ since $2 < \alpha \leq 6$ and $\beta > 1$, we know that all these links can not share a common node.

Now we turn to prove that all the links do satisfy the SINR constraints. Namely, we need to show that each link $i \in S_t$ satisfies the following inequality (4.1):

$$SINR_i = \frac{\frac{p_i}{(d(i_s,i_r))^\alpha}}{n_0 + \sum_{j \neq i} \frac{p_j}{(d(j_s,i_r))^\alpha}} \geq \beta \tag{4.1}$$

Since we employ linear power assignment, i.e., $p_i = \rho \cdot (d(i_s, i_r))^\alpha$ and $p_j = \rho \cdot (d(j_s, j_r))^\alpha$, the inequality (4.1) becomes

$$SINR_i = \frac{\rho}{n_0 + \sum_{j \neq i} \frac{\rho \cdot (d(j_s,j_r))^\alpha}{(d(j_s,i_r))^\alpha}} \geq \beta \tag{4.2}$$

Algorithm 5 LPA: An Oblivious Linear Power Assignment based Algorithm for the MLSDAN Problem.

Input: An arbitrary directed acyclic network called dan

Output: The timeslots t to schedule all links in dan

Define two constants c and μ such that c is a constant greater than 1 and $\mu = \frac{4}{\sqrt{3}}c(12\beta\frac{\alpha-1}{\alpha-2})^{\frac{1}{\alpha}}$; $t = 0$;
While not all links in dan have been scheduled **do**
    Computing $tvalue(i)$, $bvalue(i)$ and $conflict(i)$ values for each link $i \in dan$;
    Let $L := \{i|tvalue(i) = 1\}$;
    While $L \neq \emptyset$ **do**
        Pick the link $i \in L$ with the largest $bvalue(i)$ value and break ties with the largest $conflict(i)$ value. Suppose link i's length satisfies $c^k \leq d(i_s, i_r) < c^{k+1}$ (k is a nonnegative integer), then group all the other links $j \in L$ with similar lengths as i, i.e., $c^k \leq d(j_s, j_r) < c^{k+1}$. Call this group of links as link category L_k.
        As Fig. 2 shows, we three-color all the links in link category L_k;
        For each of the three colors **do**
            While not all links whose receivers are located in the cells with the same color have been scheduled **do**
                $S_t := \emptyset$;
                For each cell with the same color, we pick one link j (if there is one) whose receiver is located in the cell and $S_t := \{S_t \cup \{j\}\}$;
                For all the links in S_t, we assign the power levels based on the linear power assignment(cf. Equation (2.3), and we set $\rho = 2n_0\beta$);
                t:=t+1;
            End While
        End For //all links in L_k have been scheduled
        $L := L \backslash L_k$;
    End While //all links in L have been scheduled
    $dan := dan \backslash L$;
End While //all links in dan have been scheduled
Return t.

By observing the three-coloring method shown in Fig. 2, we have

$$\begin{aligned} d(j_s, i_r) &\geq d(j_r, i_r) - d(j_s, j_r) && (4.3)\\ &\geq (\frac{3}{2}\sqrt{3}m - \sqrt{3})W - c^{k+1} && (4.4)\\ &\geq (\frac{\sqrt{3}}{2}m)W - c^{k+1} && (4.5)\\ &\geq (\frac{\sqrt{3}}{2}m \cdot \mu - c) \cdot c^k && (4.6)\\ &\geq \frac{\sqrt{3}}{4}m \cdot \mu \cdot c^k && (4.7) \end{aligned}$$

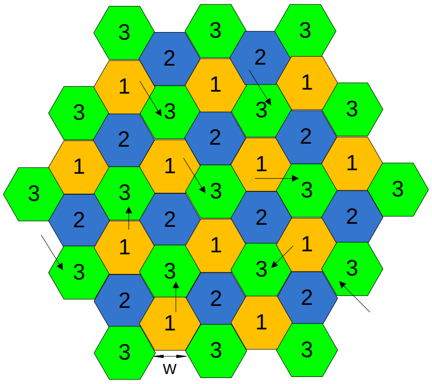

Fig. 2. Three shading of the plane with hexagons of side length $W = \mu c^k$. The arrows mean all the links in link category L_k and all their receivers are located in the cells with shade 3. Here each link $j \in L_k$ satisfies $c^k \le d(j_s, j_r) < c^{k+1}$.

Here the inequality (4.3) is obtained through triangle inequality; inequality (4.4) is derived from the three shading of the hexgons and m stands for link i's mth nearest ring; since $m \ge 1$, we have $\frac{3}{2}\sqrt{3}m - \sqrt{3} \ge \frac{\sqrt{3}}{2}m$, thus we can achieve inequality (4.5). Inequality (4.6) is obtained by plugging into the W value. Finally, since $\mu = \frac{4}{\sqrt{3}}c(12\beta\frac{\alpha-1}{\alpha-2})^{\frac{1}{\alpha}} \ge \frac{4}{\sqrt{3}}c$ and $m \ge 1$, we have $\frac{\sqrt{3}}{2}m \cdot \mu - c \ge \frac{\sqrt{3}}{4}m \cdot \mu$. From this we obtain inequality (4.7).

Now based on inequality (4.7),we have

$$\sum_{j \neq i} \frac{\rho \cdot (d(j_s, j_r))^\alpha}{(d(j_s, i_r))^\alpha} \le \sum_{m=1}^{\infty} \frac{\rho \cdot c^{(k+1)\alpha} \cdot 6m}{(\frac{\sqrt{3}}{4})^\alpha \cdot m^\alpha \cdot \mu^\alpha \cdot c^{k\alpha}} \tag{4.8}$$

$$\le \sum_{m=1}^{\infty} \frac{\rho \cdot c^\alpha \cdot (\frac{4}{\sqrt{3}})^\alpha 6m}{m^\alpha \cdot \mu^\alpha}$$

$$\le \sum_{m=1}^{\infty} \frac{6\rho \cdot c^\alpha \cdot (\frac{4}{\sqrt{3}})^\alpha}{m^{\alpha-1} \cdot \mu^\alpha}$$

$$\le \frac{6\rho \cdot c^\alpha \cdot (\frac{4}{\sqrt{3}})^\alpha}{\mu^\alpha} \cdot \frac{\alpha - 1}{\alpha - 2} \tag{4.9}$$

Here the inequality (4.8) is obtained by observing that there are at most $6m$ links in link i's m-th nearest ring. For example, as seen from Fig. 2, there are 12 hexagons(cells) that have color 3. And inequality (4.9) is derived from the Riemann Zeta Function.

Now since $\mu = \frac{4}{\sqrt{3}}c(12\beta\frac{\alpha-1}{\alpha-2})^{\frac{1}{\alpha}}$, we have

$$\begin{aligned} SINR_i &= \frac{\rho}{n_0 + \sum_{j\neq i}\frac{\rho\cdot(d(j_s,j_r))^\alpha}{(d(j_s,i_r))^\alpha}} \\ &\geq \frac{\rho}{n_0 + \frac{\rho}{2\beta}} \\ &= \frac{1}{\frac{n_0}{2n_0\beta} + \frac{1}{2\beta}} && (4.10) \\ &= \beta && (4.11) \end{aligned}$$

Here the equation (4.10) is obtained by plugging into the $\rho = 2n_0\beta$ value (cf. line 12 in Algorithms 5).

4.2. *Approximation Ratio Analysis*

First, we need to know the scheduling length upper bound of our LPA algorithm (Algorithm 5). Let dan denote the set of all links in the directed acyclic network. According to this algorithm, we have $L \subseteq dan$ and $d(L) \leq d(dan)$ (recall that $d(L)$ is the length diversity of all links in L). Now let C^k_{max} ($C^k_{max}(L)$) denote the maximum number of links in dan (L) whose receivers belong to the same cell. Here k means the link's category number in the current timeslot, i.e., each link i to be scheduled in this timeslot satisfies $c^k \leq d(i_s, i_r) < c^{k+1}$. Now, (1) According to line 11, since we only pick one link from each cell of the same color, the while loop between line 9 and line 14 will be iterated at most $C^k_{max}(L)$ times; (2) Since there are 3 colors, we know the for loop between line 8 and line 15 will be iterated 3 times and all links in L_k have been scheduled after the for loop; (3) According to line 6, since each L_k contains all the links with similar lengths in L, we know that the while loop between line 5 and line 17 will be iterated $d(L)$ times; (4) According to line 18, since each removal of all the links in L will make the length of the critical path of dan minus 1, we know the while loop from line 2 and line 19 will be iterated at most CP times (recall that CP is the length of the critical path of dan). ¿From the above analysis, by representing $C_{max} = max_{k,L}\{C^k_{max}(L)\}$, we can obtain that the scheduling length upper bound of the LPA algorithm is

$$C^k_{max}(L) \cdot 3 \cdot d(L) \cdot CP = O(d(dan) \cdot C_{max} \cdot CP) \quad (4.12)$$

Let Opt denote the minimum number of timeslots to schedule all the links in C_{max} (the cell with the maximum number of links whose receivers are located in this cell) using linear power assignment. We now turn to compute a lower bound of Opt. First, due to triangle inequality, we have

$$\begin{aligned} d(j_s, i_r) &\leq d(j_s, j_r) + d(j_r, i_r) \\ &\leq c^{k+1} + 2W \\ &\leq c^{k+1} + 2\mu c^k && (4.13) \end{aligned}$$

Suppose there are at most t links that can be scheduled in the same timeslot. Then according to the SINR inequality, we have:

$$SINR_i = \frac{\rho}{n_0 + t \cdot \frac{\rho \cdot (d(j_s, j_r))^\alpha}{(d(j_s, i_r))^\alpha}} \geq \beta \tag{4.14}$$

According to inequalities (4.14) and (4.13) and since $d(j_s, j_r) \geq c^k$, we know that

$$\begin{aligned} t &\leq \frac{1}{2\beta} \cdot \frac{(d(j_s, i_r))^\alpha}{(d(j_s, j_r))^\alpha} \\ &\leq \frac{1}{2\beta} \cdot (c + 2\mu)^\alpha \end{aligned} \tag{4.15}$$

According to inequality (4.15), we know that the minimum number of timeslots to schedule all the links in C_{max} is

$$opt \geq C_{max}/t \geq \frac{2\beta \cdot C_{max}}{(c + 2\mu)^\alpha} \tag{4.16}$$

Then the approximation ratio of the LPA algorithm is less than

$$\begin{aligned} \frac{O(d(dan) \cdot C_{max} \cdot CP)}{opt} &\leq \frac{O(d(dan) \cdot C_{max} \cdot CP)(c + 2\mu)^\alpha}{2\beta \cdot C_{max}} \\ &= O(d(dan) \cdot CP) \end{aligned} \tag{4.17}$$

In addition, according to Lemma 3.1, we know that the scheduling length of the MLSDAN problem is lower bounded by the length CP of the critical path in *dan*. By combing the scheduling length upper bound of the LPA algorithm (cf. Equation (4.12)), we have another approximation ratio for the LPA algorithm as $O(d(dan) \cdot C_{max})$. Thus we have the following Theorem 4.1.

Theorem 4.1. The approximation ratio of the LPA algorithm is $O(min\{d(dan) \cdot CP, d(dan) \cdot C_{max}\})$.

Note that, in practice, the length diversity of all the links in *dan* ($d(dan)$), the length of the critical path in *dan* (CP) and the maximum number of links whose receivers are located in a cell (C_{max}) are all usually some small constant values, so our approximation ratio of the LPA algorithm is also a constant in most realistic scenarios.

4.3. *Time Complexity Analysis*

According to Section 3.3, we know that the time complexity for computing *tvalue*,*bvalue* and *conflict* values are $O(n)$, $O(n)$ and $O(n^2)$, respectively. From this we know that the time complexity for Algorithm 5 is $O(n^3)$ where n is the number of links.

5. DLS: A Dynamic Labeling Based Link Scheduling Algorithm

In this section, we will present a heuristic algorithm that is based on dynamically updating the labeling values for each link. Similar to Algorithm 5, among all the links $L = \{i|tlevel(i) = 1\}$, we first pick the link i with the largest $bvalue(i)$ value. Then instead of picking the links that have similar lengths with link i, we turn to try picking the link in L that has the second largest $bvalue(i)$ value. We then repeat this process until there are no links in L that can be concurrently scheduled with the picked links. At the beginning of the next timeslot, we update all the labeling values and then repeat until all links have been scheduled. Note that, we adopt the Pareto-optimal power assignment in the DLS algorithm (Algorithm 6).

Algorithm 6 DLS: Dynamic Labeling based Link Scheduling Algorithm for the MLSDAN Problem.

Input: An arbitrary directed acyclic network called *dan*

Output: The timeslots t to schedule all links in *dan*

$t := 0$;
While not all links in *dan* have been scheduled **do**
  Computing $tvalue(i)$, $bvalue(i)$ and $conflict(i)$ for each link $i \in dan$;
  Let $L := \{i|tvalue(i) = 1\}$;
  $S_t := \emptyset$;
  Consider all $i \in L$ in a decreasing order of $bvalue(i)$; break ties via a decreasing order of $conflict(i)$;
    if $allowed(i, S_t)$ **then**
      $S_t := S_t \cup \{i\}$; $dan := dan \backslash \{i\}$
    end if
  For all the links in S_t, we assign the power levels based on the Pareto-optimal power assignment;
  t:=t+1;
End While
Return t.

$\mathbf{allowed(i, S_t)}$

if $\rho(H_{S_t \cup \{i\}}) \geq 1$ **then**
  return false
else if any $p_j \in (I - H_{S_t \cup \{i\}})^{-1} \eta_{S_t \cup \{i\}}$ exceeds the power limit **then**
  return false
else
  return true
end if

5.1. *Time Complexity Analysis*

The time complexity of the DLS algorithm is dominated by checking whether a set of links can be simultaneously scheduled, i.e. checking the spectral radius of the link gain matrix H is smaller than 1. According to [37], we know that the time complexity of checking an $n * n$ matrix H takes $O(n^3)$ time. Then for our DLS algorithm, the worst case is to schedule only one link in each timeslot. Thus each timeslot can take $O(n)$ eigenvaule computations and the total number of eigenvalue computations is $O(n^2)$. So the worst case time complexity for our algorithm is $O(n^2 * n^3) = O(n^5)$ where n is the number of links in the directed acyclic network.

6. Simulation Results

In this section, we present the simulation results for our scheduling algorithms. First, we generate two kinds of topologies: the dense link topologies and the sparse link topologies (cf. Fig. 3 and Fig. 4). In the dense link topologies, n nodes are randomly distributed on a plane of size 200m*200m. Then we generate a link by randomly picking its sender and receiver from the n nodes under the constraints that all the picked links must form a directed acyclic network. We repeat this process until n links have been generated. From this link topology construction, we can see that: (1) the generated link topology is a very dense link topology, or a directed acyclic network topology with very high disturbances (cf. [36]). For the sparse link topology, we construct it with a nearest neighbor algorithm, i.e., each node will pick its nearest neighbor as the receiver.

Remark: As discussed in the Problem Motivation (cf. subsection 1.1) and Related Work (cf. subsection 1.3), although there have been some works on the minimum latency aggregation scheduling problem (MLAS) [7, 16, 21, 26, 29, 30, 32, 40, 43, 44], our proposed algorithms are not comparable to the MLAS ones since we are considering arbitrarily given network topologies whereas the MLAS algorithms can take full advantage of the topology construction procedure to facilitate the scheduling process. So in the following simulations, we will only compare various algorithms for arbitrary network topologies.

For the LPA algorithm, we first compare it with the LPA(FirstFit) algorithm which is an adaptation of LPA in the sense that we do not label the wireless links before scheduling them. Here FirstFit means that we use a first fit policy to pick the links.

The simulation results for the LPA algorithm can be seen in Fig. 5 and Fig. 6. Here in order to help distinguish the scheduling lengths for different scheduling algorithms, we have chosen a relatively large path loss exponent value $\alpha = 5$. In addition, we have also chosen a higher SINR threshold value β for the sparse link topologies ($\beta = 40$ for sparse topologies and $\beta = 5$ for dense topologies). This is because either a lower β value in sparse topologies or a higher β value in dense topologies will make all the scheduling lengths almost incomparable (much more links will be simultaneously scheduled in sparse topologies with a smaller β value and much less links will be simultaneously scheduled in dense topologies with a larger β value).

From these two figures, we can see that a labeling based algorithm can always help to reduce the scheduling length for the dense link topology. But for the sparse link topologies, the LPA algorithm without first labeling the wireless links (using a first fit policy) may generate shorter scheduling lengths for a very small fraction of the link topologies. This may be due to the fact that more links could be scheduled in the same timeslot in sparse link topologies. In addition, we can achieve much shorter scheduling lengths for sparse link topologies.

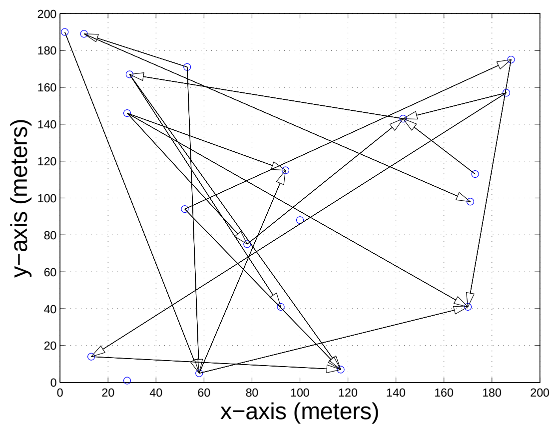

Fig. 3. A dense directed acyclic link topology with 20 links.

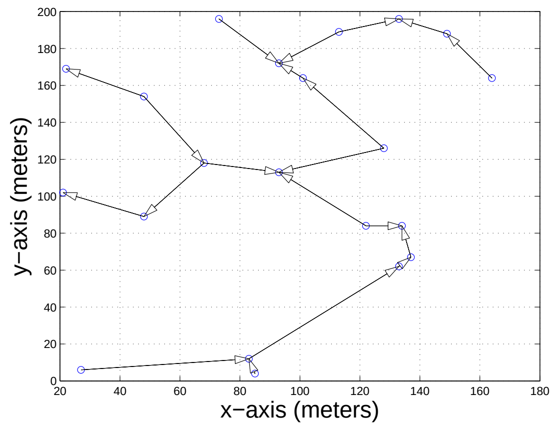

Fig. 4. A sparse directed acyclic link topology with 20 links.

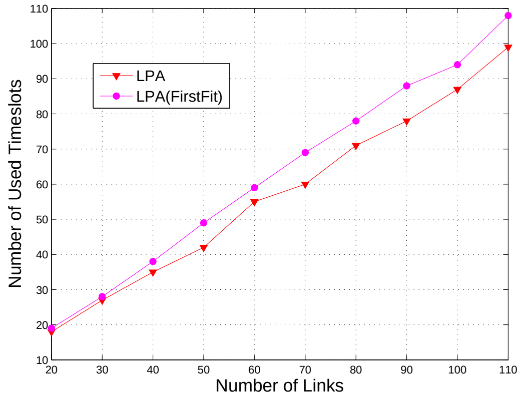

Fig. 5. Scheduling lengths for dense link topologies at $\alpha = 5$ $\beta = 5$.

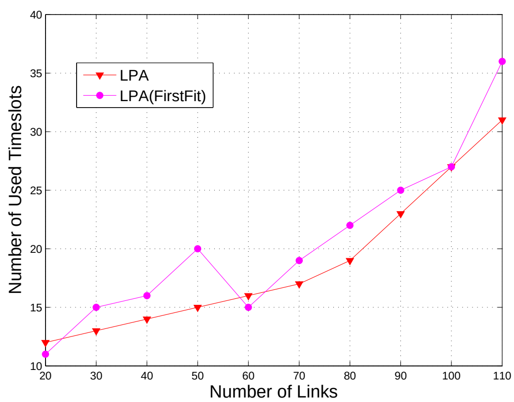

Fig. 6. Scheduling lengths for sparse link topologies at $\alpha = 5$ $\beta = 40$.

For the DLS algorithm, we compare it with the LPA algorithm, the UPA algorithm as well as the FirstFit algorithm. Here the UPA algorithm is an adaptation for the LPA algorithm in the sense that we employ uniform power assignment (please see Section 2.3) instead of the linear power assignment. The FirstFit algorithm is an adaptation for the DLS algorithm in the sense that we do not label the wireless links before scheduling them. Instead, we use a first fit policy to choose the links.

The simulation results for these algorithms are shown in Fig. 7 and Fig. 8. From these two figures, we can see that a labeling based algorithm can always help to reduce the scheduling lengths for both the dense and sparse link topologies. Also the dynamic labeling based scheduling algorithms greatly outperform the linear and constant power assignment based scheduling algorithms in terms of scheduling lengths. In addition, we can achieve much shorter scheduling lengths for sparse link topologies.

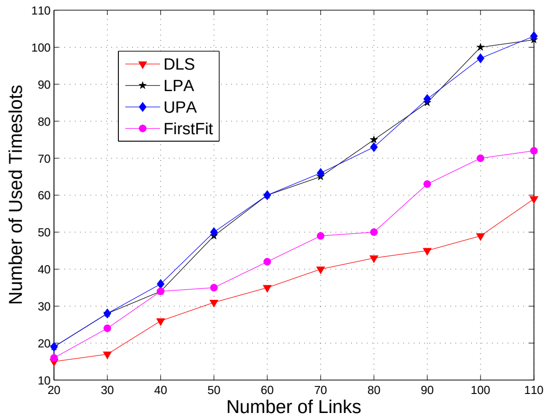

Fig. 7. Scheduling lengths for dense link topologies at $\alpha = 5$ $\beta = 5$.

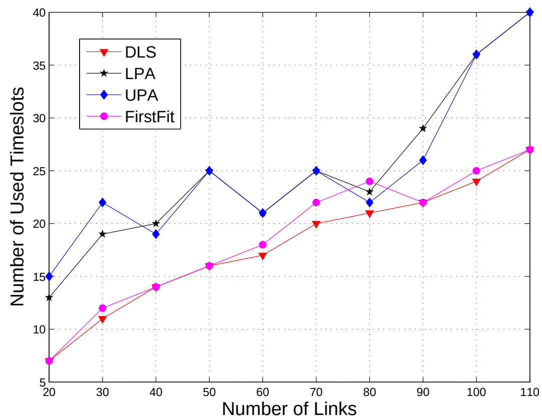

Fig. 8. Scheduling lengths for sparse link topologies at $\alpha = 5$ $\beta = 40$.

Finally we also simulated the DLS algorithm for various α and β values. The results can be found in Fig. 9 and Fig. 10. From these two figures, we can see that the the scheduling lengths can be greatly shortened by either using a larger α value or a lower β value. In addition, we can achieve much shorter scheduling lengths for sparse link topologies.

7. Conclusion

In this paper, we have formulated the MLSDAN problem, i.e., the problem of Minimum Latency Link Scheduling for Arbitrary Directed Acyclic Networks under both precedence and SINR constraints. Although link scheduling under SINR

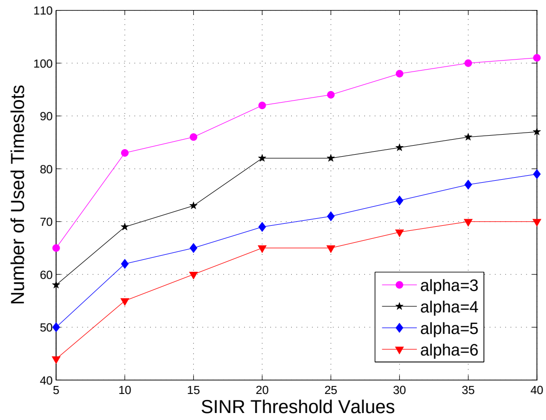

Fig. 9. Scheduling lengths for a dense link topology with 110 links.

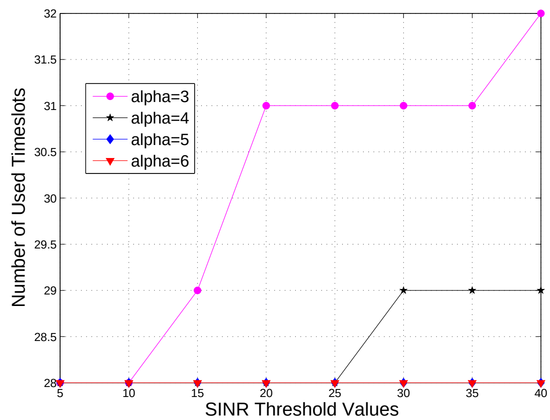

Fig. 10. Scheduling lengths for a sparse link topology with 110 links.

constraints is not a new problem, to our best knowledge, this is the first work to address arbitrary networks link scheduling under both precedence and SINR constraints. We have shown that the MLSDAN problem is NP-hard, and have proposed both polynomial time approximation and heuristic algorithms. In realistic scenarios, the achieved approximation ratio is a constant value. For the heuristic algorithms, the extensive simulations have demonstrated that the presented dynamic labeling based algorithms outperform the frequently used link scheduling algorithms, such as the constant and linear power assignments based scheduling algorithms. In addition, the simulation results indicate that labeling the wireless links before scheduling is of paramount importance in reducing the scheduling lengths. There are many other issues that warrant further investigation. For example, it would be interesting to design an approximation algorithm with an approximation ratio independent of the three parameters in our approximation ratio (cf. Theorem 4.1). For designing an approximation algorithm with an approximation ratio independent of the length diversity $d(dan)$, a possibility is to consider some other power assignment strategies, such as the square-root power assignment [11], the non-linear power assignment [33] or the iterative power assignment [28]. In addition, designing a fully distributed algorithm for the MLSDAN problem should also be a great but meaningful challenge. To achieve this goal, the topological properties studied under the SINR model [31] could be utilized. Our recent distributed algorithms for the local broadcasting problem [46] and the coloring problem [45] could also be borrowed to design a distributed algorithm for the MLSDAN problem.

Acknowledgments

The authors would like to thank Prof. Francis Chi-Moon Lau for helpful discussions and suggestions for improving the presentation of this paper. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61103186, 61073174, 61033001, 61061130540, the Hi-Tech research and Development Program of China Grant 2006AA10Z216, and Hong Kong RGC-GRF grants 714009E and 714311.

References

1. M. Andrews and M. Dinitz. Maximizing Capacity in Arbitrary Wireless Networks in the SINR Model: Complexity and Game Theory. In Proc. *INFOCOM*, 2009.
2. A. Behzad and I. Rubin. Optimum Integrated Link Scheduling and Power Control for Multihop Wireless Networks. *IEEE Transactions on Vehicular Technology*, 56(1):194-205,2007.
3. S. A. Borbash and A. Ephremides. Wireless Link Scheduling With Power Control and SINR Constraints. *IEEE Transactions on Information Theory*, 52(11):5106-5111, 2006.
4. G. S. Brar, D. M. Blough and P. Santi. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In Proc. 12th *MOBICOM*, Los Angeles, CA, USA, 2006.
5. D. Chafekar, V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan. Cross-Layer Latency Minimization in Wireless Networks with SINR Constraints. In Proc. 8th *MOBIHOC*, Montreal, Canada, 2007.
6. D. Chafekar, V. S. Anil Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan. Approximation Algorithms for Computing Capacity of Wireless Networks with SINR Constraints. In Proc. 27th *INFOCOM*, Phoenix, AZ, USA, 2008.
7. X. Chen, X. Hu, and J. Zhu. Minimum Data Aggregation Time Problem in Wireless Sensor Networks. In Proc. 1st *MSN*, Wuhan, China, 2005.
8. A. K. Das, R. J. Marks II, P. Arabshahi and A. Gray. Power Controlled Minimum Frame Length Scheduling in TDMA Wireless Networks with Sectored Antennas. In Proc. 24th *INFOCOM*, Miami, USA, 2005.

9. M. Dinitz. Distributed Algorithms for Approximating Wireless Network Capacity. In Proc. *INFOCOM*, 2010.
10. A. Fanghänel, T. Kesselheim and B. Vöcking. Improved Algorithms for Latency Minimization in Wireless Networks. In Proc. 36th *ICALP*, Rhodes, Greece, 2009.
11. A. Fanghänel, T. Kesselheim, H. Räcke and B. Vöcking. Oblivious Interference Scheduling. In Proc. 28th *PODC*, Calgary, Alberta, Canada, 2009.
12. L. Fu, S. Liew and J. Huang. Power Controlled Scheduling with Consecutive Transmission Constraints: Complexity Analysis and Algorithm Design. In Proc. 28th *INFOCOM*, Rio de Janeiro, Brazil, 2009.
13. O. Goussevskaia, M. Halldorsson, R. Wattenhofer and E. Welzl. Capacity of Arbitrary Wireless Networks. In Proc. 28th *INFOCOM*, Rio de Janeiro, Brazil, 2009.
14. O. Goussevskaia, Y. A. Pignolet and R. Wattenhofer. Efficiency of Wireless Networks: Approximation Algorithms for the Physical Interference Model. *Foundations and Trends in Networking* 4(3): 313-420, 2010.
15. O. Goussevskaia, Y. A. Oswald and R. Wattenhofer. Complexity in Geometric SINR. In Proc. 8th *MOBIHOC*, Montreal, Canda, 2007.
16. Z. Gu, G. Wang, Q.-S. Hua and Y. Wang. Improved Minimum Latency Aggregation Scheduling in Wireless Sensor Networks under the SINR Model. *International Journal of Sensor Networks*, to appear.
17. P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory* 46(2): 388-404, 2000.
18. M. Halldorsson. Wireless Scheduling with Power Control. In Proc. 17th *ESA*, Copenhagen, Denmark, 2009.
19. M. Halldorsson and R. Wattenhofer. Wireless Communication is in APX. In Proc. 36th *ICALP*, Rhodes, Greece, 2009.
20. M.M. Halldorsson and P. Mitra. Wireless Capacity with Oblivious Power in General Metrics. In Proc. *SODA*, 2011.
21. M.M. Halldorsson and P. Mitra. Wireless Connectivity and Capacity. In Proc. *SODA*, 2012.
22. R.A. Horn and C.R. Johnson. Matrix Analysis, New York: Cambridge Univ. Press, 1991.
23. Q.-S. Hua and F.C.M. Lau. The Scheduling and Energy Complexity of Strong Connectivity in Ultra-Wideband Networks. In Proc. 9th *MSWiM*, Torremolinos, Malaga, Spain, 2006.
24. Q.-S. Hua and F.C.M. Lau. Exact and Approximate Link Scheduling Algorithms under the Physical Interference Model. In Proc. 5th DialM-POMC, Toronto, Canada, 2008.
25. Q.-S. Hua and F.C.M. Lau. Joint Link Scheduling and Topology Control for Wireless Sensor Networks with SINR Constraints. In:*Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice*, IGI Global, pages 184-208, 2010.
26. S. C.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao. Nearly Constant Approximation for Data Aggregation Scheduling in Wireless Sensor Networks. In Proc. 26th *INFOCOM*, Anchorage, Alaska, USA, 2007.
27. B. Katz, M. Völker and D. Wagner. Energy Efficient Scheduling with Power Control for Wireless Nnetworks. In Proc. *WiOpt*, 2010.

28. T. Kesselheim. A Constant-Factor Approximation for Wireless Capacity Maximization with Power Control in the SINR Model. In Proc. *SODA*, 2011.
29. H. Li, Q.-S. Hua, C. Wu and F.C.M. Lau. Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks under Physical Interference Model. In Proc. *MSWiM*, 2010.
30. H. Li, Q.-S. Hua, C. Wu and F.C.M. Lau. Latency-Minimizing Data Aggregation in Wireless Sensor Networks under Physical Interference Model. *Ad Hoc Networks*, to appear.
31. E. Kantor, Z. Lotker, M. Parter and D. Peleg. The Topology of Wireless Communication. In Proc. *STOC*, 2011.
32. X.-Y. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao and Y. Qi. Efficient Data Aggregation in Multi-hop Wireless Sensor Networks under Physical Interference Model. In Proc. *MASS*, 2009.
33. T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks, In Proc. 25th *INFOCOM*, Barcelona, Spain, 2006.
34. T. Moscibroda, R. Wattenhofer and A. Zollinger. Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies. In Proc. 7th *MOBIHOC*, Florence, Italy, 2006.
35. T. Moscibroda. The Worst-Case Capacity of Wireless Sensor Networks. In Proc. 6th *IPSN*, Cambridge, Massachusetts, USA, 2007.
36. T. Moscibroda, Y. A. Oswald and R. Wattenhofer. How Optimal are Wireless Scheduling Protocols? In Proc. 26th *INFOCOM*, Anchorage, Alaska, USA, 2007.
37. V.Y. Pan and Z. Q. Chen. The Complexity of the Matrix Eigenproblem. In Proc. 31st *STOC*, Atlanta, Georgia, USA, 1999.
38. S. U. Pillai, T. Suel, and S. Cha. The Perron-Frobenius Theorem and Some of its Applications. *IEEE Signal Processing Magazine*, 22(2):62-75, 2005.
39. J. Tang, G. Xue, C. Chandler and W. Zhang: Link Scheduling with Power Control for Throughput Enhancement in Multihop Wireless Networks. *IEEE Transactions on Vehicular Technology*, 55(3):733-742, 2006.
40. P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan and X. Jia. Minimum-Latency Aggregation Scheduling in Multihop Wireless Networks. In Proc. 10th *MobiHoc*, New Orleans, LA, USA, 2009.
41. P.-J. Wan, X. Jia and F. F. Yao. Maximum Independent Set of Links under Physical Interference Model. In Proc. *WASA*, 2009.
42. X.-H. Xu, S. Tang, P.-J. Wan. Maximum Weighted Independent Set of Links under Physical Interference Model. In Proc. *WASA*, 2010.
43. X. Xu, S. Wang, X. Mao, S. Tang, and X. Li. An Improved Approximation Algorithm for Data Aggregation in Multi-hop Wireless Sensor Networks. In Proc. 2nd *FOWANC*, New Orleans, LA, USA, 2009.
44. B. Yu, J. Li and Y. Li. Distributed Data Aggregation Scheduling in Wireless Sensor Networks. In Proc. 28th *INFOCOM*, Rio de Janeiro, Brazil, 2009.
45. D. Yu, Y. Wang, Q.-S. Hua and F.C.M. Lau. Distributed (Δ+1)-Coloring in the Physical Model. In Proc. *ALGOSENSORS*, 2011.
46. D. Yu, Y. Wang, Q.-S. Hua and F.C.M. Lau. Distributed Local Broadcasting Algorithms in the Physical Interference Model. In Proc. *DCOSS*, 2011.