

Limitations of Current Wireless Scheduling Algorithms

Magnús M. Halldórsson, Christian Konrad, and Tigran Tonoyan^(✉)

ICE-TCS, Reykjavik University, Reykjavik, Iceland
{mmh,christiank,tigran}@ru.is

Abstract. We consider the problem of scheduling wireless links in the physical model, where we seek a partition of a given a set of wireless links into the minimum number of subsets satisfying the signal-to-interference-and-noise-ratio (SINR) constraints. We consider the two families of approximation algorithms that are known to guarantee $O(\log n)$ approximation for the scheduling problem, where n is the number of links. We present network constructions showing that the approximation ratios of those algorithms are no better than logarithmic, both in n and in Δ , where Δ is a geometric parameter – the ratio of the maximum and minimum link lengths.

Keywords: Wireless scheduling · Oblivious power · Lower bound

1 Introduction

The task of the MAC layer in TDMA-based (time-division multiple access) wireless networks is to determine which nodes can communicate in which time-frequency slot. A scheduler aims to optimize criteria involving throughput and fairness. This requires obtaining effective spatial reuse while satisfying the interference constraints. We treat the fundamental *scheduling* problem of partitioning a given set of communication links into the fewest possible feasible sets.

We adopt the *SINR model* of communication, where signal decays as it travels and a transmission is successful if its strength at the receiver exceeds the accumulated signal strength of interfering transmission by a sufficient (technology determined) factor. Considerable progress has been made in recent years in elucidating essential algorithmic properties of the SINR model (e.g., [1, 8, 9, 13, 24, 27, 29, 31, 34]). Early work on the scheduling problem includes [5, 6, 10]. Gupta and Kumar [16] proposed the geometric version of SINR and initiated average-case analysis of network capacity known as scaling laws. NP-completeness has been shown for scheduling with different forms of power control: none [14], limited [28], and unbounded [32]. Moscibroda and Wattenhofer [34] initiated worst-case analysis in the SINR model.

Although the standard analytic assumption that signal decays polynomially with the distance traveled is far from realistic [33, 36], it has been shown that

results obtained with that assumption can be translated to the setting of arbitrary measured signal decay [3, 15], as well as to Rayleigh fading model [7].

The scheduling problem has been considered both for fixed *oblivious* power assignments (where the power chosen for a link depends only on the link itself) and for arbitrary power control. This paper addresses algorithms dealing with both scenarios.

Finding constant-factor approximation to the scheduling problem has proved challenging. However, there are several approaches giving logarithmic approximation.

- I. One way of achieving $O(\log n)$ approximation is by *greedy or first-fit* algorithms, where the links are processed in a non-decreasing order of length and are assigned to the first slot where they “fit” [22, 26, 29, 30]. The approximation ratio of such algorithms is usually obtained by arguing that such algorithms provide constant factor approximation to the *capacity problem*, where the goal is to select a large subset of links that can successfully transmit in the same time slot. It has also been shown that such algorithms perform well for some randomly generated network instances [2].
- II. Another approach giving $O(\log n)$ -approximation (for fixed power assignments) is to use randomization: links try to transmit in each time slot with certain transmission probabilities, and each link is assigned to the slot where its transmission succeeded [21, 31]. It is known that appropriate choice of the probabilities guarantees an $O(\log n)$ approximation (w.h.p).
- III. Yet another approach is to divide the links into groups of nearly equal length and schedule each group separately. Following this approach, numerous $O(\log \Delta)$ -approximation results have been argued [12, 14, 18], where Δ is the ratio between the longest and shortest link length.

The only known algorithms to achieve less than $O(n)$ approximation for scheduling are from the first two families. The only known constant-factor approximation algorithms for scheduling are obtained in the case of the *linear power scheme* [23, 37]. In a recent paper, we presented an $O(\log^* \Delta)$ -approximation algorithm for scheduling with power control [25], but here too, the approximation factor is only $O(n)$ in terms of n .

The optimum number of slots for scheduling has also been approximated through *interference measures* [19, 21, 31]. However, no measure is known to give better than $O(\log n)$ approximation. It is also not evident how to efficiently compute such measures.

Many variants of the scheduling problem are known to be NP-hard but we are not aware of any significant inapproximability result.

Our Results. While for the third family of algorithms it is easy to find examples attaining the approximation ratio $\Omega(\log \Delta)$, we are not aware of constructions showing that the approximation ratio of $O(\log n)$ of the first and second families cannot be improved. For the second family of algorithms, a construction has been presented in [21] for which the output of the algorithm is a $\Theta(\log n)$ -approximation, but their construction does not exclude that $\Theta(\log n)$ is only

additive, and it is asked in [21] whether another analysis of the algorithm could give a smaller approximation ratio.

We show that algorithms of families I and II achieve (including the algorithms for both fixed power assignments and optimized powers) no better than $\Omega(\frac{\log \Delta}{\log \log \Delta})$ (in terms of only Δ) or $\Omega(\frac{\log n}{\log \log n})$ (in terms of only n) approximation even in one dimensional networks. Our constructions are obtained by modeling sets of links after certain graphs that are “hard” instances for graph coloring algorithms similar to the families I and II.

These results suggest that new methods are needed for obtaining better than logarithmic approximation for scheduling. Note, however, that our constructions do not work for the uniform power assignment, where all links use equal power.

2 Model and Definitions

Communication Links. Consider a set L of n links, numbered from 1 to n . Each link i represents a unit-demand communication request from a sender s_i to a receiver r_i - point-size wireless transmitter/receivers located in a metric space with distance function d . We denote $d_{ij} = d(s_i, r_j)$ the distance from the sender of link i to the receiver of link j , $l_i = d(s_i, r_i)$ the length of link i and $d(i, j)$ the minimum distance between $\{s_i, r_i\}$ and $\{s_j, r_j\}$. We let $\Delta(L)$ denote the ratio between longest and shortest link lengths in L , and drop L when clear from context.

SINR Feasibility and the Scheduling Problem. In the *physical model* (or *SINR model*) of communication [35], a transmission of a link i is successful iff

$$\mathcal{S}_i > \beta \cdot \left(\sum_{j \in S \setminus \{i\}} \mathcal{I}_{ji} + N \right), \quad (1)$$

where \mathcal{S}_i denotes the received signal power of link i , \mathcal{I}_{ji} denotes the interference power on link i caused by link j , $N \geq 0$ is a constant denoting the ambient noise, $\beta \geq 1$ is the minimum SINR (Signal to Interference and Noise Ratio) required for a message to be successfully received and S is the set of links transmitting concurrently with link i . If $P : L \rightarrow \mathbb{R}_+$ is the power assignment of links - $P(i)$ defines the transmission power of the sender node s_i , then $\mathcal{S}_i = \frac{P(i)}{l_i^\alpha}$ and $\mathcal{I}_{ji} = \frac{P(j)}{d_{ji}^\alpha}$, where $\alpha \in (2, 6)$ is the path-loss exponent.

A set L of links is called *P-feasible* if (1) holds for each link $i \in L$ when using power assignment P . A collection of sets is *P-feasible* if each set in the collection is *P-feasible*. When transmission power is also subject to optimization, we call a set of links (collection) simply *feasible* if there is a power assignment P such that the set (collection) is *P-feasible*.

The *scheduling problem* with a fixed power scheme P is to partition a given set L into the minimum number of *P-feasible* subsets (or *slots*).

The *scheduling problem with power control* is to partition a given set L into the minimum number of feasible subsets.

For simplicity of constructions, we assume henceforth that $N = 0$. The bounds can be adapted for the case of positive noise by scaling power levels.

Affectance: Fixed Power Assignments and Power Schemes. Following [26], we define the *affectance* $a_P(i, j)$ of link j by link i under power assignment P by

$$a_P(i, j) = \frac{\mathcal{I}_{ij}}{S_j} = \frac{P(i)l_j^\alpha}{P(j)d_{ij}^\alpha}.$$

We let $a_P(j, j) = 0$ and extend a_P additively over sets: $a_P(S, i) = \sum_{j \in S} a_P(j, i)$ and $a_P(i, S) = \sum_{j \in S} a_P(i, j)$. It is readily verified (recall that we assumed $N = 0$) that a set of links S is P -feasible if and only if $a_P(S, i) \leq 1/\beta$ for all $i \in S$.

We will be particularly interested in *power schemes* P_δ of the form $P_\delta(i) = C \cdot l_i^{\delta\alpha}$, where C is constant for the given network instance. These are called *oblivious power assignments* because the power level of each link depends only on a local information - the link length. Examples of such power schemes are *uniform* power scheme (P_0), *linear* power scheme (P_1) and *mean* power scheme ($P_{1/2}$) [11].

It will be useful to note that for any power scheme P_δ , $a_{P_\delta}(i, j) = \left(\frac{l_i^\delta l_j^{(1-\delta)\alpha}}{d_{ij}} \right)^\alpha$.

Affectance: Power Control. For scheduling with power control, the following definition of affectance is used [29]: $a(i, j) = \left(\frac{l_i}{d(i, j)} \right)^\alpha$.

Similarly as before, we define $a(i, i) = 0$ and extend a additively to sets. As shown in [29], if the following condition holds for any link $i \in S$ with a sufficiently small constant γ (depending on α and β), then set S is feasible: $a(S_i^-, i) < \gamma$, where S_i^- denotes the subset of links in S that are no longer than link i : $S_i^- = \{j \in S : l_j \leq l_i\}$.

3 Lower Bounds for First-Fit Algorithms

3.1 Scheduling with Fixed Power Schemes

The *first-fit* algorithm considered in [26] was originally used for the uniform power scheme, but applies also to other oblivious power schemes [22]. The algorithm is a simple greedy procedure, where one starts with empty slots in a fixed order, then the links are processed in *non-decreasing order by length* and a link i is assigned to the first slot S such that $a_P(S, i) + a_P(i, S) < \gamma$ for a given constant γ . One may also generalize the acceptance condition with $f[a_P(S, i), a_P(i, S)] < \gamma$, where $f : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is a decreasing function of both arguments that goes to 0 when both arguments go to 0. Below, NDFirstFit will refer to such an algorithm.

The family of hard network instances for the first-fit algorithm is inspired by a well known tree construction called *binomial trees* (in relation to binomial heaps),

that has been used to obtain lower bounds for first-fit algorithms for graph coloring [4, 17]. For any given power scheme P_δ with $\delta \in (0, 1)$, we construct a family L_R of network instances on the real line using binomial trees as a model, where no pair of links corresponding to adjacent nodes in the tree can be in the same P_δ -feasible set together, but are otherwise spatially well separated. We show that while L_R can be scheduled in constant number of slots using P_δ , NDFirstFit gives only an $\Omega(\log n)$ approximation in terms of n and $\Omega(\frac{\log \Delta}{\log \log \Delta})$ approximation in terms of Δ .

Theorem 1. *Let $\delta \in (0, 1)$. For each $n_0 > 0$, there is a set of $n > n_0$ links L_R on the real line such that NDFirstFit achieves no better than $\Omega(\log n) = \Omega(\frac{\log \Delta}{\log \log \Delta})$ approximation for scheduling with P_δ .*

Binomial Trees. We use the following family of trees, known as *binomial trees*, as a model for our construction. The rooted tree T_R , ($R \geq 0$), is constructed recursively, as follows. T_0 consists of a single root vertex. For $R \geq 0$, the tree T_{R+1} is obtained from T_R by adding a new child vertex to the root, then adding a copy of T_R , by identifying its root node with the new child. For example, T_1 consists of two nodes connected by an edge and T_2 consists of a root vertex that has two children and one “grandchild”. Note that the number of vertices in T_R is $n = 2^R$. Let us call the set of leaves of T_R *layer R* . For $t = R - 1, R - 2, \dots, 1$, *layer t* denotes the set of leaves of the tree that remains after removing layers $R, R - 1, \dots, t + 1$. Thus, T_R has $R + 1$ layers and the root is in layer 0. Further, each layer t contains 2^{t-1} vertices, except layer 0, which contains one vertex (the root). Note also that each layer t node has exactly one child from each of layers $R, R - 1, \dots, t + 1$.

Remark. “Layer” should not be confused with “level”, i.e. the set of vertices at a given distance from the root. Note that a layer contains links from many levels.

The Network Instance. Let us fix an integer $R > 0$ and $\delta \in (0, 1)$. For simplicity of the argument, we assume that $\beta = 1$. We model the set L_R of links after the tree T_R . Each link $i \in L_R$ corresponds to a vertex v_i of the tree and all the links are arranged on the real line. See Fig. 1 for an example. The links

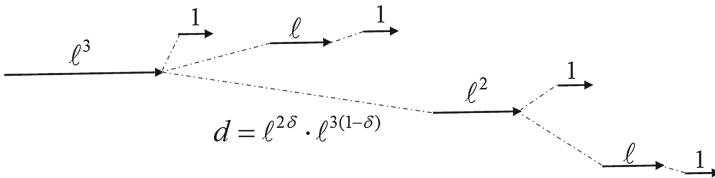


Fig. 1. The set L_R with $R = 3$.

corresponding to layer t vertices have length ℓ^{R-t} , where $\ell = cR^{1/\gamma} = c \log^{1/\gamma} n$, $\gamma = \alpha \cdot \min\{\delta, 1 - \delta\}$ and $c > 1$ is a large enough constant to be determined below. For instance, the “root link” has length ℓ^R and the “leaves” have length 1. Note also that $\Delta(L_R) = \ell^R$. Links are numbered arbitrarily, from 1 to n . The link corresponding to the root (leaves) of the tree is called *root link* (*leaf links*). Link j is called the *parent* of link i (and i is a *child* of link j) if v_j is the parent of vertex of v_i . Link j is called a *left sibling* of a link i (and i is called a *right sibling* of j) if $l_j < l_i$ and v_i and v_j have the same parent in the tree. We will place the parent and all left siblings of each link i to the left from link i (no two links occupy intersecting intervals in our construction). The *descendants* of link i are the links corresponding to the set of nodes of the subtree rooted at v_i .

The root is placed with its sender on the origin and the receiver at the coordinate ℓ^R . Assume links i and j are so that the node corresponding to i is the parent of the node of j in the tree (hence, $l_i > l_j$). Moreover, assume that the placement of link i has already been determined. Then we place link j so that $s_j = r_i + d_{ji} = r_i + l_i^{1-\delta} l_j^\delta$ and $r_j = s_j + l_j$. Such placement guarantees that any two links corresponding to adjacent tree nodes cannot be in the same P_δ -feasible set (recall that $\beta = 1$).

Analysis. The goal of the analysis below is to show that the set L_R can be scheduled in two sets, each corresponding to a color class in a proper 2-coloring of tree T_R . This is achieved by taking the constant c large enough. Having this, it will follow immediately that NDFirstFit schedules L_R into R slots, by allocating a separate slot for each layer of links (as in the increasing order of length, links come layer-by-layer), and, by the construction, no two layers can be in the same slot. This yields an approximation lower bound $\Omega(R) = \Omega(\log n) = \Omega(\log \Delta / \log \log \Delta)$.

Let us give a taste of the analysis below, by considering the affectance between a chain of three links; it will also give an idea why the construction cannot be implemented for the uniform power scheme (i.e. when $\delta = 0$). Let i, j, k be such that i is the parent of j and j is the parent of k . The placement of the links is such that the distance between e.g. j and k is $d_{kj} = l_j^{1-\delta} l_k^\delta$, meaning that $a_{P_\delta}(k, j) = \left(\frac{l_j^{1-\delta} l_k^\delta}{d_{kj}}\right)^\alpha = 1$, so j and k cannot coexist in the same feasible set, nor can i and j . On the other hand, the distance between i and k is at least $d_{ki} > d_{ji} = l_i^{1-\delta} l_j^\delta$. Hence, $a_{P_\delta}(k, j) < \left(\frac{l_i^{1-\delta} l_k^\delta}{l_i^{1-\delta} l_j^\delta}\right)^\alpha = \left(\frac{l_k}{l_j}\right)^{\alpha\delta} = \ell^{-\delta\alpha}$, which can be made arbitrarily small by taking large constant c (note that this property does not hold in the case of uniform power scheme, as $\delta = 0$).

Analysis: Link Placement. First we show that if the constant c is large enough then all the left siblings of a link i appear to the left of i . Note that if a link i has length ℓ^s , then its descendants constitute an instance L_s (i.e. correspond to the tree T_s). We start by computing the diameter $d(L_s)$ of L_s for any $s > 0$, i.e. the distance from the left-most node to the rightmost node of L_s .

Proposition 1. *If $\ell \geq 2$ then, for any $s > 0$, $\ell^s < d(L_s) < 4\ell^s$.*

Proposition 2. *If link j is a left sibling of link k then link j and its descendants are placed to the left from link k , provided that the constant c is large enough.*

Proof. Let link i be the parent of links j, k and assume w.l.o.g. that $l_i = \ell^p$, $l_j = \ell^t$ and $l_k = \ell^{t+1}$. We want link k to appear to the right of the whole “subtree” of links rooted at link j (i.e. descendants of link j); namely, $d(r_i, s_k) > d(r_i, s_j) + 2\ell \cdot d(L_t)$. Since i is the parent of j and k , we have, by definition, $d(r_i, s_k) = \ell^{p \cdot (1-\delta)} \cdot \ell^{(t+1) \cdot \delta}$ and $d(r_i, s_j) = \ell^{p \cdot (1-\delta)} \cdot \ell^{t \cdot \delta}$. Thus, due to the bound $d(L_t) < 4\ell^t$, it suffices to have: $\ell^{(1-\delta)p + \delta(t+1)} > \ell^{(1-\delta)p + \delta t} + 8\ell^{t+1}$ or

$$\ell^{(1-\delta)p + \delta t} (\ell^\delta - 1) > 8\ell^{t+1}.$$

Recall that $p \geq t + 2$ as link i is strictly longer than its children. Thus, assuming $\ell \geq 2^{1/\delta}$, the requirement above boils down to $\ell^{t-\delta+2} > 8\ell^{t+1}$, and thus to $\ell > 8^{1-\delta}$ which holds if the constant c is large enough. \square

Analysis: Affectance. Note that by the claim above, if links i and j are such that $l_i > l_j$ and j is not a descendant of link i then $d(i, j) > 2l_i$.

Let us fix a link i with $l_i = \ell^p$ and let L_R^i denote the set of all links in L_R except the parent and the children of link i . We show that the affectance of link i by links in L_R^i can be made arbitrarily small if the constant c is large enough.

We split L_R^i into the following subsets: D_i - the descendants of i , A_i - links that are longer than i , B_i - links that are shorter than i , *excluding the descendants of link i* , and E_i - links of length l_i . Recall that $\gamma = \alpha \min\{\delta, 1 - \delta\}$. While the following three claims follow relatively easily from the construction, the last one requires more care.

Proposition 3. $a_{P_\delta}(E_i, i) + a_{P_\delta}(i, E_i) = O(c^{-\alpha})$.

Proposition 4. $a_{P_\delta}(A_i, i) + a_{P_\delta}(i, A_i) = O(c^{-\gamma})$.

Proposition 5. $a_{P_\delta}(D_i, i) + a_{P_\delta}(i, D_i) = O(c^{-\gamma})$.

Proposition 6. $a_{P_\delta}(B_i, i) + a_{P_\delta}(i, B_i) = O(c^{-\gamma})$.

Proof. Let B_i^q denote the set of length ℓ^q links in B_i . We collect the links of B_i^q into disjoint subsets S_1, S_2, \dots by “climbing” from link i towards the root, as follows. Suppose we are at link i . The set S_1 contains the links of B_i^q that appear in the interval between link i and its parent; the distance between each link in S_1 and link i is at least $2l_i = 2\ell^p$, and the number of such links is at most 2^{p-q} . S_2 contains the links of B_i^q that are descendants of the first right sibling of link i ; the distance between each link in S_2 and link i is at least $2\ell^{p+1}$ and the number of such links is at most 2^{p-q+1} . S_3 contains the links that are descendants of the second right sibling of link i ; the links in S_3 are at a distance at least $2\ell^{p+2}$ and the number of links in S_3 is at most $2\ell^{p+3}$. After collecting the descendants of all right siblings of link i , we move to the parent of link i and repeat the same

procedure. This process is carried on until reaching the root. Thus, we get the following bound on the affectance by the links in B_i^q :

$$a_{P_\delta}(B_i^q, i) < \sum_{t \geq 0} 2^{p+t-q} \left(\frac{\ell^{q\delta} l_i^{1-\delta}}{2\ell^{p+t}} \right)^\alpha < 2^{-\alpha} (\ell/2)^{-\delta\alpha(p-q)} \sum_{t \geq 0} (\ell/2)^{-t\alpha}.$$

The last sum is clearly bounded by 2, given that $\ell > 4$. Thus, we have that $a_{P_\delta}(B_i^q, i) < (\ell/2)^{-\delta\alpha} < (c/2)^{-\delta\alpha}/R$. The first part of the claim follows because there are at most R different sets B_i^q for a fixed link i . The second part follows by a symmetric argument. \square

Analysis: Conclusion. Thus, we have that the affectance of each link i from all other links except its parent and children is small. In order to schedule the set L_R into two slots, it is enough to place each link separately from its parent and children. Two slots are sufficient because of the tree topology.

Now let us see what happens when we run NDFirstFit on the set L_R . Note that when processing the links in a non-decreasing order of length we will process all links of layer $t + 1$ before the first link of layer t . Let S_t be the set of layer- t links. We have, by the results above, that $a_{P_\delta}(S_t, i) + a_{P_\delta}(i, S_t) < \gamma$ for any constant γ and link $i \in S_t$. Thus, NDFirstFit puts the set S_R in the first slot. Each link in S_{R-1} conflicts with a link in S_R (i.e. its child link), so the next slot will consist of the set S_{R-1} . In general, each layer t link conflicts with a link from *each* of layers $t + 1, t + 2, \dots, R$, so by the reasoning above, NDFirstFit schedules each layer in a separate slot (no two layers can occupy the same slot), taking $R + 1$ slots in total for scheduling L_R . Thus, the approximation ratio of NDFirstFit is $\Omega(R) = \Omega(\log n)$. Also, recall that $\Delta(L_R) = \ell^R = cR^{R/\gamma}$, so $R = \Omega\left(\frac{\log \Delta}{\log \log \Delta}\right)$.

Note that the approximation ratio is multiplicative, as we can just multiply each link k times (i.e. replace it with its k identical copies), for any $k = \text{poly}(n)$. Then the optimum number of slots required will be $2k$, while NDFirstFit will schedule the set into $\Omega(k \cdot R) = \Omega(k \cdot \log kn) = \Omega\left(k \cdot \frac{\log \Delta}{\log \log \Delta}\right)$ slots.

3.2 Scheduling with Power Control

In this section, we let NDFirstFitPC denote the first-fit scheduling algorithm (with power control) of [29]. Here also, one starts with empty slots in a fixed order, then the links are processed in non-decreasing order by length and a link i is assigned to the first slot S such that $a(S, i) < \gamma$ for a small constant γ . It is known that NDFirstFitPC achieves $O(\log n)$ approximation [29]. Using similar constructions as in the case of fixed power schemes, we prove the following.

Theorem 2. *For each $n_0 > 0$, there is a set of $n > n_0$ links L_R on the real line such that NDFirstFitPC achieves no better than $\Omega(\log n) = \Omega\left(\frac{\log \Delta}{\log \log \Delta}\right)$ -approximation for scheduling with power control.*

The Construction. For the sake of simplicity, we assume in this section that $\beta > 2^\alpha$: the lower bounds can be straightforwardly adapted for any constant $\beta > 1$. The construction in this case is similar to the one for fixed power schemes and is modeled after trees T_R . Let us fix an integer $R > 0$. Each link i corresponds to a vertex v_i of the tree and all the links are arranged on the real line. The links corresponding to layer t vertices have length ℓ^{R-t} , where $\ell = cR^{1/\alpha} = c \log^{1/\alpha} n$ and $c > 1$ is a large enough constant. The placement of links is similar to the construction for fixed power schemes: each child link j of a link i is placed so that $s_j = r_i + d(i, j) = r_i + l_i$ and $r_j = s_j + l_j$, assuming link i has been placed. It is known that if $\beta > 2^\alpha$, then the minimum distance between any two links that are in the same slot must be greater than the length of the smaller one [25, Theorem 4]; hence, link placement as above guarantees that any two links corresponding to adjacent tree nodes cannot be in the same slot.

Analysis. As before, it can be shown that if the constant c is large enough then all the left siblings of a link i appear to the left of link i .

Proposition 7. *If link j is a left sibling of link k then link j and its descendants are placed to the left from link k at a distance at least $l_k/2$ from link k , provided that $\ell > 6$.*

Note that by the claim above, if links i and j are such that $l_i > l_j$ and j is not a descendant of link i then $d(i, j) > l_i/2$.

For any link i , let \underline{L}_R^i denote the set of links in L_R that are *not longer* than link i , excluding the children of link i . Note that we do not need to consider links longer than link i . The following claim is proved in a similar way as the analogous results for fixed powers. In fact, the bounds here are stronger because we have similar spacing between links as before, while the numerators of affectance are smaller as we consider only the affectance by smaller links.

Proposition 8. $a(\underline{L}_R^i, i) = O(c^{-\alpha})$.

The rest of the analysis is identical to the case of fixed powers.

4 Lower Bounds for Uniform Randomized Algorithms

Next we consider a generalization of the distributed algorithm presented in [31]. In this algorithm, the sender nodes of the links act in synchronous rounds and each sender node transmits with probability p_i or waits with probability $1 - p_i$ in round i , where p_i is the same for all links (but may change across the rounds). Once the transmission succeeds in round i , the node is silent in subsequent rounds.

Our construction in this case is modeled after a complete $\log^b n$ -ary tree with n nodes, where $b > 0$ is a constant, and is loosely based on [20, Theorem 6]. In [20], a similar lower bound is obtained for coloring interval graphs using randomized distributed algorithms. It is worth to mention, however, that the analysis of the algorithm here is done on sparser graphs (trees of cliques) than

in [20] (intersection graphs of laminar sets of intervals), but fortunately the argument can be adapted.

The main challenge is to construct a family of network instances that are structurally similar to $\log^b n$ -ary trees. Namely, links correspond to adjacent nodes in the tree are not P_δ -feasible together, but are otherwise well separated.

Theorem 3. *Let $\delta \in (0, 1)$ and the probabilities $p_i (i = 1, 2, \dots)$ be fixed. For each $n_0 > 0$, there is a set of $n > n_0$ links L on the real line s.t. the randomized algorithm that uses probabilities $p_i (i = 1, 2, \dots)$ yields only a $\Omega(\frac{\log n}{\log \log n}) = \Omega(\frac{\log \Delta}{\log \log \Delta})$ approximation for scheduling with power scheme P_δ , w.h.p.*

The Construction. We assume in this section that $\beta = 1$. We start with the description of a preliminary set S of links simulating a rooted complete $\log^b n$ -ary tree over a set of n/M nodes, where $b > 1$ is a constant to be chosen and $M = O(n^\epsilon)$ is a parameter with $\epsilon \in (0, 1)$ a constant. The main construction is a simple extension of S . We will often mix the terminology of links and trees, e.g. by speaking of children of links, hoping it does not cause confusion. We split the tree into *levels*, where the root is at level 0 and the nodes at (tree-) distance t from the root constitute level t . Note that the number of nodes at level t is $\log^{tb} n$; hence, the number of levels is $k = \Theta\left(\frac{\log(n/M)}{\log \log n}\right) = \Theta\left(\frac{\log n}{\log \log n}\right)$. For each $t \geq 0$, the level- t links have uniform length ℓ_t . We set

$$\ell_t = c\ell_{t+1} \log^{d(t+1)} n \tag{2}$$

for large enough constants $c, d > 0$. We describe the placement of links on the real line level by level, starting from level 0, which contains a single link i . We set $s_i = 0, r_i = s_i + \ell_0$, as shown in Fig. 2. The children of link i have length ℓ_1 . We place the $\log^b n$ child links of length ℓ_1 inside the interval occupied by the link i , so that for any children j, k , it holds that (see Fig. 2):

1. $d(j, k) \geq e\ell_1$ for constant $e > 1$,
2. $d(r_j, r_i) \geq \ell_0^{1-\delta} \ell_1^\delta / 2$,
3. $d(s_j, r_i) \leq \ell_0^{1-\delta} \ell_1^\delta$,
4. $d(s_i, s_j) \geq \ell_0 / 2$.

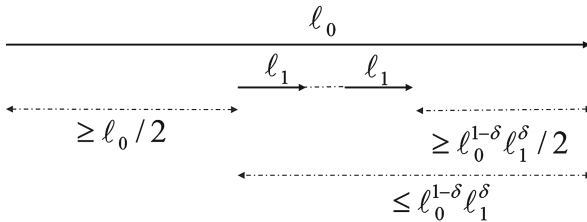


Fig. 2. The first step of the construction of Theorem 3.

This completes the first step of the construction of S . The idea behind the constraints above is to guarantee the following properties: 1. the set of links at the same level is feasible, 2. the children interfere with the parent, 3. the grandchildren do not interfere with their grandparent, 4. the parent does not interfere the children (or their descendants). At the second step, we construct the children of level-1 links in a similar manner, and continue this process until having n/M links. As shown below, the length ratios defined by (2) ensure that the construction is correct and, in particular, that no link can be in the same feasible set as any of its children. On the other hand, we prove that the affectance of any level- t link by all other links, except level- $t - 1$ and level- $t + 1$ links, is small. This implies that the set S can be scheduled in a constant number of slots using the power scheme P_δ .

In order to complete the construction, we replace each link in S with its M identical copies. Let L denote this set of links. Note that $|L| = n$. Note that the optimum number of slots for scheduling L is at least M , as different copies of the same link should be placed in different slots. Using the properties of set S , we show that $\Theta(M)$ slots suffice.

Analysis: Properties of Set S . We start by proving the properties of the set S . The first question to address is: what are the requirements on the lengths of links for satisfying the constraints (1-4)? The first three constraints will hold if $\ell_0^{1-\delta}\ell_1^\delta/2 > 3e\ell_1 \log^b n$, which holds if $\delta < 1$ and the constants c, d in (2) are large enough. The fourth constraint requires: $\ell_0 - \ell_0^{1-\delta}\ell_1^\delta > \ell_0/2$, which holds if $\ell_0 > 2^{1/\delta}\ell_1$. Thus, choosing the constants in (2) large enough guarantees the constraints (1-4).

Now let us show that S can be scheduled in a constant number of slots. Let S_t denote the set of level- t links for $t \geq 0$.

Proposition 9. *If the constants c, d in (2) are large enough, then for any level- t link i ($t \geq 0$), it holds that $a_{P_\delta}(T, i) < 1$, where $T = S \setminus (S_{t-1} \cup S_{t+1})$.*

Proof. First, let us bound the affectance by links from S_t . Recall that those links have equal lengths and their mutual distances are at least $e\ell_t$, by the first constraint of the construction. Thus we have:

$$a_{P_\delta}(S_t, i) < 2 \sum_{r \geq 1} \left(\frac{\ell_t^\delta \ell_t^{1-\delta}}{r e \ell_t} \right)^\alpha = O(e^{-\alpha}),$$

where the factor of 2 accounts for links on both sides of i . The last term can be made arbitrarily small if constant c is large enough, because $\alpha > 1$.

Now, let us fix an $s > t + 1$. The number of level- s links is $|S_s| = \log^{sb} n$. The distance from each level- s link to r_i is at least $\ell_t^{1-\delta}\ell_{t+1}^\delta/2$, by construction. Thus, we have:

$$a_{P_\delta}(S_s, i) \leq |S_s| \frac{\ell_s^{\delta\alpha} \ell_t^{(1-\delta)\alpha}}{(\ell_t^{1-\delta}\ell_{t+1}^\delta/2)^\alpha} = 2^\alpha |S_s| \left(\frac{\ell_s}{\ell_{t+1}} \right)^{\delta\alpha}.$$

Since the number of levels is $O(\log n)$, it is enough to have $a(S_s, i) \frac{e'}{\log n}$ for any constant $e' > 0$, which is provided if

$$\ell_{t+1} > 2^{1/\delta} (e' |S_s| \log n)^{1/(\delta\alpha)} \ell_s = 2^{1/\delta} e'^{1/(\delta\alpha)} \log^{(sb+1)/(\alpha\delta)} n \ell_s.$$

The last inequality holds if we set $d \geq 2b/(\alpha\delta)$ and $c \geq 2^{1/\delta} e'^{1/(\delta\alpha)}$ in (2).

Next, let us consider a layer $s < t - 1$ for $t > 0$. Recall that the distance from each link of L_s to r_i is at least $\ell_s/2$, by construction. The affectance by S_s can be bounded as follows: $a_{P_\delta}(S_s, i) \leq |S_s| \frac{\ell_s^{\delta\alpha} \ell_t^{(1-\delta)\alpha}}{(\ell_s/2)^\alpha} < 2^\alpha |S_s| \left(\frac{\ell_t}{\ell_s}\right)^{\delta\alpha}$.

Hence, again, we can easily get $a_{P_\delta}(S_s, i) < \frac{e'}{\log n}$ for any constant $e' > 0$, by tuning the constants c and d in (2). This yields the claim. \square

Thus, the set S can be scheduled into two feasible slots, by taking the union of the odd-numbered levels in one slot and the union of the even-numbered levels in another one. This directly implies that the set L can be scheduled in $2M$ slots.

Analysis: The Lower Bound. It remains to prove that for any sequence $p_i \in (0, 1)$, $i = 1, 2, \dots$, the randomized algorithm using the probabilities p_i will schedule L in $\Omega(kM) = \Omega\left(M \cdot \frac{\log n}{\log \log n}\right) = \Omega\left(M \cdot \frac{\log \Delta}{\log \log \Delta}\right)$ slots, where the last equality holds because $\Delta = \text{poly}(n)$. To that end, it will be more convenient to analyze the algorithm in terms of a *conflict graph* G corresponding to L , rather than the set of links itself. The graph G is constructed by replacing each vertex of a complete $\log^b n$ -ary tree on n/M vertices with a M -clique, where the cliques corresponding to two adjacent vertices form a $2M$ -clique. Obviously, $\chi(G) = 2M$. Level- t vertices in G are the vertices corresponding to level- t vertices in the tree. Let the probabilities p_i , $i = 1, 2, \dots$ be fixed. We consider the following variant of the algorithm with relaxed constraints on transmissions. In round i , each remaining vertex v of G selects itself with probability p_i and is removed from the graph in this round if it selects itself and no neighbor is selected. Lower-bounding the runtime of this algorithm for G implies a similar bound for the original algorithm running on the set L of links, because any feasible set in L corresponds to an independent set in G , i.e. we essentially neglect some part of the interference when dealing with G , which can only make the algorithm use less slots. The argument below is similar to the proof of [20, Theorem 6]. The main idea is to show that whatever the values of p_i are, the algorithm will remove the vertices level by level, starting from the last level vertices. In particular, it will take $\Theta(M)$ steps to start making essential impact on the next level. This gives the desired bound $\Omega(k \cdot M)$.

Let T_t denote the first time step when the size of a level- t M -clique is halved. Let H_t denote the event that for all $s \leq t$, the size of the smallest level- s clique is at least $(1 - 1/\log n)M$ before iteration $T_{t+1} + 1$.

Proposition 10. *Consider $0 \leq t < k$. Suppose that $T_{t+1} < M \log n$. Then*

$$\mathbb{P}[H_t] = 1 - O\left(n^{-\frac{M}{130 \log n} + 1}\right).$$

Observe that given the event H_t , the difference between the times T_{t+1} and T_t is at least $M/4$ if n is large enough. Indeed, H_t implies that in round T_{t+1} , the size of each clique in levels $t, t-1, \dots, 0$ is at least $3M/4$, and in order for a clique of size $3M/4$ to become less than $M/2$, at least $M/4$ rounds must pass. Thus, $\mathbb{P}[T_t - T_{t+1} \geq M/4] \geq \mathbb{P}[H_t] = 1 - O(n^{-\frac{M}{130 \log n} + 1})$ holds for each fixed t . By the union bound, the probability that the event $T_{t+1} - T_t \geq M/4$ is violated for at least one t is at most $O(k \cdot n^{-\frac{M}{130 \log n} + 1}) = O(n^{-\frac{M}{130 \log n} + 2})$. Thus, if $M > 130c \log n$ (recall that $M = \Theta(n^\epsilon)$), then with probability $1 - O(n^{2-c})$, it takes at least $k \cdot M/4 = \Omega(M \log n / \log \log n)$ steps until all the vertices of the graph are removed. This completes the lower bound argument.

References

1. Avin, C., Emek, Y., Kantor, E., Lotker, Z., Peleg, D., Roditty, L.: SINR diagrams: convexity and its applications in wireless networks. *J. ACM* **59**(4), 18 (2012)
2. Belke, L., Kesselheim, T., Koster, A.M.C.A., Vöcking, B.: Comparative study of approximation algorithms and heuristics for SINR scheduling with power control. In: Bar-Noy, A., Halldórsson, M.M. (eds.) *ALGOSENSORS 2012*. LNCS, vol. 7718, pp. 30–41. Springer, Heidelberg (2013)
3. Bodlaender, M., Halldórsson, M.M.: Beyond geometry: towards fully realistic wireless models. In: *PODC* (2014)
4. Caragiannis, I., Fishkin, A.V., Kaklamani, C., Papaioannou, E.: A tight bound for online colouring of disk graphs. *Theor. Comput. Sci.* **384**(2–3), 152–160 (2007)
5. Chafekar, D., Kumar, V., Marathe, M., Parthasarathy, S., Srinivasan, A.: Cross-layer latency minimization for wireless networks using SINR constraints. In: *Mobihoc* (2007)
6. Cruz, R.L., Santhanam, A.: Optimal routing, link scheduling, and power control in multi-hop wireless networks. In: *INFOCOM* (2003)
7. Dams, J., Hoefer, M., Kesselheim, T.: Scheduling in wireless networks with Rayleigh-fading interference. In: *SPAA* (2012)
8. Daum, S., Gilbert, S., Kuhn, F., Newport, C.: Broadcast in the ad hoc SINR model. In: Afek, Y. (ed.) *DISC 2013*. LNCS, vol. 8205, pp. 358–372. Springer, Heidelberg (2013)
9. Dinitz, M.: Distributed algorithms for approximating wireless network capacity. In: *INFOCOM* (2010)
10. ElBatt, T., Ephremides, A.: Joint scheduling and power control for wireless ad-hoc networks. In: *INFOCOM* (2002)
11. Fanghänel, A., Kesselheim, T., Räcke, H., Vöcking, B.: Oblivious interference scheduling. In: *PODC*, August 2009
12. Fu, L., Liew, S.C., Huang, J.: Power controlled scheduling with consecutive transmission constraints: complexity analysis and algorithm design. In: *INFOCOM*. IEEE (2009)
13. Goussevskaia, O., Halldórsson, M.M., Wattenhofer, R.: Algorithms for wireless capacity. *IEEE/ACM Trans. Netw.* **22**(3), 745–755 (2014)
14. Goussevskaia, O., Oswald, Y.A., Wattenhofer, R.: Complexity in geometric SINR. In: *MobiHoc* (2007)
15. Gudmundsdottir, H., Ásgeirsson, E.I., Bodlaender, M., Foley, J.T., Halldórsson, M.M., Vigfusson, Y.: Measurement based interference models for wireless scheduling algorithms. In: *MSWiM* (2014)

16. Gupta, P., Kumar, P.R.: The capacity of wireless networks. *IEEE Trans. Inf. Theor.* **46**(2), 388–404 (2000)
17. Gyárfás, A., Lehel, J.: On-line and first fit colorings of graphs. *J. Graph Theor.* **12**(2), 217–227 (1988)
18. Halldórsson, M.M.: Wireless scheduling with power control. *ACM Trans. Algorithms* **9**(1), 7 (2012)
19. Halldórsson, M.M., Holzer, S., Mitra, P., Wattenhofer, R.: The power of non-uniform wireless power. In: *SODA*, pp. 1595–1606 (2013)
20. Halldórsson, M.M., Konrad, C.: Distributed algorithms for coloring interval graphs. In: Kuhn, F. (ed.) *DISC 2014. LNCS*, vol. 8784, pp. 454–468. Springer, Heidelberg (2014)
21. Halldórsson, M.M., Mitra, P.: Nearly optimal bounds for distributed wireless scheduling in the SINR model. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) *ICALP 2011, Part II. LNCS*, vol. 6756, pp. 625–636. Springer, Heidelberg (2011)
22. Halldórsson, M.M., Mitra, P.: Wireless capacity with oblivious power in general metrics. In: *SODA* (2011)
23. Halldórsson, M.M., Mitra, P.: Wireless capacity and admission control in cognitive radio. In: *INFOCOM* (2012)
24. Halldórsson, M.M., Mitra, P.: Wireless connectivity and capacity. In: *SODA* (2012)
25. Halldórsson, M.M., Tonoyan, T.: How well can graphs represent wireless interference? In: *STOC* (2015)
26. Halldórsson, M.M., Wattenhofer, R.: Wireless communication is in APX. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikolettseas, S., Thomas, W. (eds.) *ICALP 2009, Part I. LNCS*, vol. 5555, pp. 525–536. Springer, Heidelberg (2009)
27. Jurdzinski, T., Kowalski, D.R., Rozanski, M., Stachowiak, G.: On the impact of geometry on ad hoc communication in wireless networks. In: *PODC* (2014)
28. Katz, B., Volker, M., Wagner, D.: Energy efficient scheduling with power control for wireless networks. In: *WiOpt* (2010)
29. Kesselheim, T.: A constant-factor approximation for wireless capacity maximization with power control in the SINR model. In: *SODA* (2011)
30. Kesselheim, T.: Approximation algorithms for wireless link scheduling with flexible data rates. In: Epstein, L., Ferragina, P. (eds.) *ESA 2012. LNCS*, vol. 7501, pp. 659–670. Springer, Heidelberg (2012)
31. Kesselheim, T., Vöcking, B.: Distributed contention resolution in wireless networks. In: Lynch, N.A., Shvartsman, A.A. (eds.) *DISC 2010. LNCS*, vol. 6343, pp. 163–178. Springer, Heidelberg (2010)
32. Lin, H., Schalekamp, F.: On the complexity of the minimum latency scheduling problem on the Euclidean plane. *arXiv preprint 1203.2725* (2012)
33. Maheshwari, R., Jain, S., Das, S.R.: A measurement study of interference modeling and scheduling in low-power wireless networks. In: *SenSys* (2008)
34. Moscibroda, T., Wattenhofer, R.: The complexity of connectivity in wireless networks. In: *INFOCOM* (2006)
35. Rappaport, T.S.: *Wireless Communications: Principles and Practice*, 2nd edn. Prentice Hall, Upper Saddle River (2002)
36. Son, D., Krishnamachari, B., Heidemann, J.: Experimental study of concurrent transmission in wireless sensor networks. In: *SenSys* (2006)
37. Tonoyan, T.: On some bounds on the optimum schedule length in the SINR model. In: Bar-Noy, A., Halldórsson, M.M. (eds.) *ALGOSENSORS 2012. LNCS*, vol. 7718, pp. 120–131. Springer, Heidelberg (2013)