Capacity of Arbitrary Wireless Networks

Olga Goussevskaia
Computer Engineering and
Networks Laboratory
ETH Zurich, Switzerland
Email: golga@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and
Networks Laboratory
ETH Zurich, Switzerland
Email: wattenhofer@tik.ee.ethz.ch

Magnús M. Halldórsson
School of Computer Science
Reykjavik University, Iceland
Email: mmh@ru.is

Emo Welzl
Institute of Theoretical Computer Science
ETH Zurich, Switzerland
Email: emo@inf.ethz.ch

***Abstract*— In this work we study the problem of determining the throughput capacity of a wireless network. We propose a scheduling algorithm to achieve this capacity within an approximation factor. Our analysis is performed in the physical interference model, where nodes are arbitrarily distributed in Euclidean space. We consider the problem separately from the routing problem and the power control problem, i.e., all requests are single-hop, and all nodes transmit at a fixed power level. The existing solutions to this problem have either concentrated on special-case topologies, or presented optimality guarantees which become arbitrarily bad (linear in the number of nodes) depending on the network's topology.**

We propose the first scheduling algorithm with approximation guarantee independent of the topology of the network. The algorithm has a *constant approximation* guarantee for the problem of maximizing the number of links scheduled in one time-slot. Furthermore, we obtain a $O(\log n)$ approximation for the problem of minimizing the number of time slots needed to schedule a given set of requests. Simulation results indicate that our algorithm does not only have an exponentially better approximation ratio in theory, but also achieves superior performance in various practical network scenarios. Furthermore, we prove that the analysis of the algorithm is extendable to higher-dimensional Euclidean spaces, and to more realistic bounded-distortion spaces, induced by non-isotropic signal distortions. Finally, we show that it is NP-hard to approximate the scheduling problem to within $n^{1-\varepsilon}$ factor, for any constant $\varepsilon > 0$, in the non-geometric SINR model, in which path-loss is independent of the Euclidean coordinates of the nodes.

I. Introduction

In this paper we study two fundamental questions in wireless networking: "*What* is the throughput capacity of a given network?", and "*How* can this capacity be achieved?". More precisely, given a set of communication requests between nodes, typically distributed in a metric space, how efficiently can these requests be scheduled? This question can be formulated in several ways. One might want to know the maximum number of requests that can be scheduled simultaneously. Alternatively, one might ask about the minimum number of time slots needed to schedule all requests. The main objective is to achieve efficient spatial reuse, considering wireless interference among concurrently transmitting nodes.

The answers to these questions depend on the topology of the network. One could be interested in networks where nodes are *randomly distributed*, or are positioned on a regular grid, for example. The problem of determining the capacity of such networks has been extensively studied, starting with the pioneering work of Gupta and Kumar [12]. Another special case are *worst-case* networks, i.e., topologies with particularly low capacity. The algorithmic challenges of worst-case networks have also received some attention, e.g., [16], [17]. These studies suggest that there is a significant gap between the capacities of randomly deployed and worst-case networks. When power control is allowed, this gap is polylogarithmic in the number of nodes. Remarkably, when power control is not allowed, this gap becomes exponential, i.e., whereas the capacity of a random network is $\Theta(1/\log n)$, it becomes as bad as $\Theta(1/n)$ in worst-case topologies. This suggests that, in fact, very little is understood about the capacity of networks that fall in the middle of these two extremes.

In this work we generalize this research and ask what the capacity of *any* network (i.e., a network with *arbitrary topology*) is. Whether it is a topology formed by nodes distributed uniformly at random in the plane, or a topology that follows some other probability distribution, whether it is highly clustered, or a worst-case topology, we want to be able to compute the network's throughput capacity. Besides answering the question of *what* the capacity is, we also want to determine *how* this capacity can be achieved.

An important issue when studying scheduling algorithms for wireless networks is how to model interference. The most commonly used interference models can be roughly classified into graph-based models and fading channel models. Graph-based models, such as the protocol model or the UDG (Unit Disk Graph) model, typically define a set of interference-edges, containing pairs of nodes within a certain distance to each other, thus modeling interference as a binary and a local measure. Such models serve as a useful abstraction of wireless networks; they facilitate the process of designing protocols and proving their efficiency, but are subject to several limitations. Although the interference of a single far-away transmitter can be relatively small, the accumulated interference of several such nodes can be sufficiently high to corrupt a transmission. Therefore protocols based on localized interference models that simply ignore interference beyond a certain range are not guaranteed to work in a real scenario.

Fading channel models such as the physical interference model offer a more realistic representation of wireless communication. A signal is received successfully if the SINR—the ratio of the received signal strength to the sum of the interference caused by all other nodes sending simultaneously, plus noise—is above a hardware-defined threshold. This definition of a successful transmission, as opposed to the graph-based definition, accounts also for interference generated by transmitters located far away. Observe that, since the SINR depends on which transmissions are being scheduled concurrently in each time slot, it is not possible to build an interference graph a priori. The notion of an interference edge is not a binary relation anymore, and thus a conflict graph cannot be constructed without knowing the solution beforehand. This makes the analysis of algorithms more challenging than in graph-based models. Many problems such as power assignment and topology control have revealed new perspectives when SINR constraints are considered [1], [17], [18], [19].

The tradeoff between accuracy and efficiency of different interference models has not been rigorously analyzed so far, i.e., it is not clear exactly how big the error is in one or another model. However, it is easy to construct instances that indicate that the relative error might be as big as $\Theta(n)$, i.e., linear in the number of nodes.[1]

In this work we study the capacity/scheduling problem in the physical interference model, where nodes are arbitrarily distributed in Euclidean space. We consider the problem separately from the routing problem and the power control problem, each of which constitute a topic of their own. Therefore, we concentrate our attention on the scheduling problem, where all requests are *single-hop* and all nodes transmit at a fixed power level.[2] This problem was shown to be NP-complete in [9], and the few algorithms with provable approximation guarantee proposed so far yield arbitrarily bad performance, depending on the topology of the network [2], [4], [9] (see Related Work Section).

Our first result is an algorithm that maximizes the number of concurrently scheduled links in one time-slot. The algorithm has a constant approximation guarantee, regardless of the topology of the network (Section IV). This means that, given a set of links, of arbitrary lengths and arbitrarily distributed in a Euclidean space, it returns a subset of links obeying the SINR constraints, of size at most a constant factor smaller than the optimum solution. To the extent of our knowledge, this is the first scheduling algorithm with approximation guarantee independent of the topology of the network. We further use this (maximization) one-slot scheduling algorithm to derive a minimum-length schedule with $O(\log n)$ approximation factor (Section IV-C). Furthermore, we present simulation results (Section V), which indicate that our algorithm, besides having an exponentially better approximation ratio in theory, is also practical. It is easy to implement and achieves superior performance in various network scenarios.

We complement our results by looking at the algorithm's performance in metric spaces beyond the two-dimensional Euclidean plane (Section VII). We prove that the analysis is extendable to higher-dimension Euclidean spaces, provided that the path-loss exponent is strictly higher than the number of dimensions. Moreover, we show that our algorithm is also valid in more realistic bounded-distortion spaces, such as spaces induced by non-isotropic signal distortions.

Finally, we present a non-approximability result for the scheduling problem in the non-geometric SINR model (Section VI). More specifically, we show that in the SINR model where path-loss is set arbitrarily (i.e. not determined by the Euclidean coordinates of the nodes), it is NP-hard to approximate the scheduling problem to within $n^{1-\varepsilon}$ factor, for any constant $\varepsilon > 0$. The paper starts with an overview of some related work (Section II) and the description of the model used throughout the analysis (Section III).

II. Related Work

Throughput capacity of randomly deployed wireless networks has been intensely studied from the information theory perspective [8], [12], [14]. Although these results are important, they do not provide algorithmic tools to determine the capacity of concrete wireless networks. In practice, networks with heterogenous topologies may be more common than randomly-deployed networks.

Scheduling algorithms in graph-based models usually employ some sort of matching or coloring, and have been widely studied [13], [15]. Although these algorithms present extensive theoretical analysis, they are constrained to the limitations of a model that ultimately abstracts away the accumulative nature of wireless interference. The inefficiency of graph-based scheduling protocols in the SINR model is well documented and has been shown theoretically and experimentally [11], [18].

There is a large amount of work addressing the problem of joint scheduling and power control in the SINR model. For instance, in [1], [5], [6] optimization models and heuristics for this problem are proposed. In [17], a power-assignment algorithm, which schedules a strongly connected set of links in poly-logarithmic time is presented. In [3], similar techniques are used to solve the combined problem of routing and power

[1]Consider, for instance, the UDG model, which states that there is a conflict between any two nodes within distance less than one of each other. Now imagine n very short links distributed inside such a unit-radius disk in a grid-like fashion. The sender-receiver distance is chosen small enough such that the SINR constraints are satisfied at every receiver even when all links are scheduled simultaneously. The UDG model in this case is overly pessimistic, allowing only one link to be schedule at a time, as opposed to n concurrent transmissions in the SINR model. This is an overly simplistic example, and certainly more sophisticated graph models have been proposed to mimic SINR constraints (e.g. [2]), however it serves as a first motivation for the problem.

[2]Our analysis also holds in case nodes transmit at different power levels, provided that either the ratio P_{max}/P_{min} of maximum and minimum power levels is bounded by a constant, or there are only a constant number of possible power levels.

control. Topology control with SINR constraints has been addressed in [7], [19].

The scheduling problem without power control in the SINR model, where nodes are arbitrarily distributed in Euclidean space, has been shown to be NP-complete in [9]. A variation of the problem, where analog network coding is allowed, has also been shown to be NP-complete in [10]. As already stated in the introduction, to the extent of our knowledge, the few algorithms for this problem that offer an approximation guarantee have arbitrarily bad performance, depending on the topology of the network. In [2], a greedy scheduling algorithm with approximation ratio of $O(n^{1-2/(\psi(\alpha)+\epsilon)}(\log n)^{2/(\psi(\alpha)+\epsilon)})$, where $\psi(\alpha)$ is a constant that depends on the path-loss exponent α, is proposed. This result, however, holds only under the assumption that nodes are distributed uniformly at random in a square of unit area. In [9], an algorithm with a factor $O(g(L))$ approximation guarantee in arbitrary topologies, where $g(L)$ is the so called *diversity* of the network, is proposed. The diversity depends on the topology of the network and captures the variation in the lengths of the links to be scheduled. The problem is that the diversity of a network can be as large as n. In [4], an algorithm with approximation guarantee of $O(\log \Delta)$ was proposed, where Δ is the ratio between the maximum and the minimum distances between nodes. This parameter can be arbitrarily large (note that $g(L) \leq \log \Delta$). In contrast to all the above mentioned approaches, the scheduling algorithm proposed in this work has an approximation ratio of $O(\log n)$, independent of the topology of the network.

III. Model

The scheduling problem can be formulated as follows. Given a set of links $L = \{l_1, \ldots, l_n\}$, where each link l_v represents a communication request from a sender s_v to a receiver r_v, two objectives can be defined: (1) maximize the number of links scheduled concurrently in one time-slot, and (2) schedule all the requests in as few time-slots as possible.

We assume that each link has a unit-traffic demand, and model the case of *non-unit* traffic demand by replicating each link x times, where x is the demand on the link. All nodes are positioned in Euclidean space. The distance between two nodes s_v, r_w is denoted by $d_{vw} = d(s_v, r_w)$. The length of link l_v is denoted by d_{vv}.

We adopt the *physical interference* model. In this model, a node r_v successfully receives a message from a sender s_v if and only if the following condition holds:

$$\frac{\frac{P}{d_{vv}^{\alpha}}}{\sum_{l_w \in \mathcal{S}_t \setminus \{l_v\}} \frac{P}{d_{wv}^{\alpha}} + N} \geq \beta, \qquad (1)$$

where P is the power level of the transmission, $\alpha > 2$ is the path-loss exponent, $\beta > 1$ denotes the minimum SINR required for a message to be successfully received, N is the ambient noise, and $\mathcal{S}_t$ is the set of concurrently scheduled links in slot t.

In this work we assume that all nodes transmit with the same power level P. Nevertheless, our analysis holds in case nodes transmit with different but fixed power levels, provided that either the ratio P_{max}/P_{min} between the maximum and the minimum power levels is bounded by a constant, or there are only a constant number of possible power levels.

We use the notation $P_{vv} = P/d_{vv}^{\alpha}$ to denote the power received by receiver r_v from its intended sender s_v, and $I_{vw} = P/d_{vw}^{\alpha}$ to denote the interference received by receiver r_w from a concurrently scheduled sender s_v.

IV. Scheduling Algorithm

In order to solve the minimum-length scheduling problem, we use a "master-slave" approximation strategy, where the "slave" problem is the *one-slot scheduling*. The one-slot scheduling problem is a maximization problem that, given an input set of links L, has the objective to maximize the number of links to be scheduled successfully in a single time-slot. Firstly we show that our one-slot scheduling algorithm has constant approximation guarantee. Thereafter we show that by iteratively computing constant approximations of maximum one-slot schedules, we obtain a factor $O(\log n)$ for the overall minimum-length scheduling problem.

We start with some definitions. The *relative interference* (RI) of a link l_u on link l_v is the increase caused by l_u in the inverse of the SINR at l_v, namely $RI_u(v) = I_{uv}/P_{vv}$. The *affectedness* of link l_v, caused by a set S of links, is the sum of the relative interferences of the links in S on l_v, as well as the effect of noise, scaled by β, or

$$\begin{aligned} a_S(l_v) &= \beta \left(\frac{N}{P_{vv}} + \sum_{l_u \in S} RI_u(v) \right) \\ &= \beta \cdot \frac{\sum_{l_u \in S} I_{uv} + N}{P_{vv}}. \qquad (2) \end{aligned}$$

Observe that a solution S is valid iff the affectedness (by the other nodes in S) of each link in S is at most 1.

Algorithm 1 One-Slot Scheduling Algorithm

1: **input:** Set of links $L = \{l_1, \cdots, l_n\}$;
2: **output:** One-slot schedule $\mathcal{S}$;
3: Set c according to (3);
4: **repeat**
5: Add the *shortest* link $l_v \in L$ to $\mathcal{S}$;
6: Delete $l_u \in L$, where $d_{uv} = d(s_u, r_v) \leq c \cdot d_{vv}$;
7: Delete $l_w \in L$, where $a_{\mathcal{S}}(l_w) \geq 2/3$;
8: **until** $L = \emptyset$
9: **return** $\mathcal{S}$;

The one-slot scheduling algorithm (for a description in pseudo-code see Algorithm 1) greedily schedules links in increasing order of length, i.e., "strong" links are scheduled first. After a link l_v is added to the solution $\mathcal{S}$, its "safety" is guaranteed in two steps. First (line 6 of Algorithm 1), all links l_u (remaining in L) whose senders are within the radius $c \cdot d_{vv}$ of the receiver r_v are removed from L (c is a constant always bigger than 2, and is defined in (3)). Second (line 7 of Algorithm 1), all links l_w, whose affectedness $a_{\mathcal{S}}(l_w)$ rose to or above a threshold of $2/3$, are removed. This process is repeated until all links in L have been either scheduled or deleted. The strength of this simple algorithm lies in the combination of elimination steps in lines 6 and 7, which ensures that the greedily constructed solution does not lose its feasibility after addition of new links. Next we prove that the obtained schedule is both correct and competitive, i.e., is only a constant factor away from the optimum.

A. Correctness of One-Slot Scheduling

In this section we prove that the solution $\mathcal{S}$ obtained in Algorithm 1 is correct, i.e., all selected links can be scheduled concurrently without collisions.

Lemma 4.1: Algorithm 1 produces a valid solution.

Proof: Let S_v^- be the set of links shorter than l_v, i.e., those added to $\mathcal{S}$ before l_v, and S_v^+ be the set of links longer than l_v, i.e., those added after l_v. When a link l_v is added to the solution, its affectedness is less than $2/3$, since it has not been deleted in the previous step. Therefore, the interference caused on l_v by concurrently scheduled shorter links (plus the ambient noise N) is $a_{S_v^-}(l_v) < 2/3$. It remains to show that S_v^+ affects l_v by at most $1/3$.

Our first observation is that, by the first elimination criterion of the algorithm, discs D_w of radius $c \cdot d_{ww}$ around each receiver $r_w \in S_v^+$ do not contain any sender $s_z \neq s_w$. Using this fact and the triangular inequality, we can lower bound the distance between any two senders $(s_w, s_z) \in S_v^+$ as $d(s_w, s_z) \geq d(r_w, s_z) - d_{ww} \geq c \cdot d_{ww} - d_{ww} = d_{ww}(c-1) \geq d_{vv}(c-1)$. Therefore, discs D_w of radius $d_{vv}(c-1)/2$ around senders in S_v^+ do not intersect.

Next, we partition the sender set in S_v^+ into concentric rings $Ring_k$ of width $c \cdot d_{vv}$ around the receiver r_v. Each ring $Ring_k$ contains all senders $s_w \in S_v^+$, for which $k(c \cdot d_{vv}) \leq d_{wv} \leq (k+1)(c \cdot d_{vv})$. We know that the first ring $Ring_0$ does not contain any sender. Consider all senders $s_w \in Ring_k$ for some integer $k > 0$. All discs of radius $d_{vv}(c-1)/2$ around each s_w must be located entirely in an extended ring $Ring_k$ of area

$$\begin{aligned} A(Ring_k) &= [(d_{vv}(k+1)c + d_{vv}(c-1)/2)^2 - \\ &\quad (d_{vv}kc - d_{vv}(c-1)/2)^2]\pi \\ &= (2k+1)d_{vv}^2 c(2c-1)\pi. \end{aligned}$$

Since discs D_w of area $A(D_w) \geq (d_{vv}(c-1)/2)^2\pi$ around senders in S_v^+ do not intersect, and the minimum distance between r_v and $s_w \in Ring_k, k > 0$ is $k(c \cdot d_{vv})$, we can use an area argument to bound the number of senders inside each ring. The total interference coming from ring $Ring_k, k \geq 1$ is then bounded by

$$\begin{aligned} I_{Ring_k}(l_v) &\leq \sum_{s_w \in Ring_k} I_{s_w}(l_v) \\ &\leq \frac{A(Ring_k)}{A(D_w)} \cdot \frac{P}{(kcd_{vv})^\alpha} \\ &\leq \frac{(2k+1)}{k^\alpha} \cdot \frac{4P}{(cd_{vv})^\alpha} \frac{c(2c-1)}{(c-1)^2} \\ &\leq \frac{1}{k^{(\alpha-1)}} \cdot \frac{P}{d_{vv}^\alpha} \frac{2^5 3}{c^\alpha}, \end{aligned}$$

where the last inequality holds since $k \geq 1 \Rightarrow 2k+1 \leq 3k$ and $c \geq 2 \Rightarrow c-1 \geq c/2$. Summing up the interferences over all rings yields

$$\begin{aligned} I_{S_v^+}(l_v) &< \sum_{k=1}^{\infty} I_{Ring_k}(l_v) \\ &\leq \sum_{k=1}^{\infty} \frac{1}{k^{\alpha-1}} \cdot \frac{P}{d_{vv}^\alpha} \frac{2^5 3}{c^\alpha} \\ &< \frac{\alpha-1}{\alpha-2} \cdot \frac{P}{d_{vv}^\alpha} \frac{2^5 3}{c^\alpha}, \end{aligned}$$

where the last inequality holds since $\alpha > 2$. This results in affectedness

$$a_{S_v^+}(l_v) = \frac{\beta I_{S_v^+}(l_v)}{P_v(v)} < \frac{\alpha-1}{\alpha-2} \cdot \frac{2^5 3\beta}{c^\alpha} \leq 1/3, \text{ where}$$

$$c = \max\left(2, \left(2^5 3^2 \beta \frac{\alpha-1}{\alpha-2}\right)^{\frac{1}{\alpha}}\right). \tag{3}$$

We have shown that $\forall l_v \in \mathcal{S}$, $a_{\mathcal{S}}(l_v) \leq 2/3 + 1/3 = 1$, which means that $SINR(l_v) \geq \beta$ for every scheduled link. This concludes the proof of the lemma. ■

B. Approximation Ratio of One-Slot Scheduling

To analyze the performance of Algorithm 1, we compare the solution ALG to an optimal solution, say OPT. In order to compare the two solutions, we will count the number of links eliminated by the algorithm that could have been scheduled in the optimum., i.e., we bound the size of the set $OPT' = OPT \setminus ALG$. Partition OPT' into OPT_1, consisting of links in OPT' that are deleted in the first elimination step of the algorithm (line 6), and OPT_2, with links deleted in the second elimination step (line 7). Overload these terms to refer also to the sizes of these sets.

Lemma 4.2: Let X be a feasible solution and let l_v be a link in X. The number of senders in X within distance $k \cdot d_{vv}, k \geq 1$ of the receiver r_v is at most k^α. Moreover, the number of senders in X within distance $k \cdot d_{vv}$ of the sender s_v is at most $(k+1)^\alpha$.

Proof: The relative interference of each sender $s_u \in X \setminus \{s_v\}$, where $d_{uv} \leq k \cdot d_{vv}$, on l_v is

$$RI_u(v) = \frac{I_{uv}}{P_{vv}} = \frac{P/d_{uv}^\alpha}{P/d_{vv}^\alpha} = \left(\frac{d_{vv}}{d_{uv}}\right)^\alpha \geq \frac{1}{k^\alpha}.$$

Since the affectedness of l_v is at most one, there can be at most k^α such senders. Moreover, since points within distance $k \cdot d_{vv}$ from r_v are within distance $(k+1)d_{vv}$ from s_v, the number of senders in X within distance $k \cdot d_{vv}$ of the sender s_v is at most $(k+1)^\alpha$. ■

Lemma 4.3: $OPT_1 \leq \rho_1 \cdot ALG$, where $\rho_1 = (2c+1)^\alpha$ and constant c as defined in (3).

Proof: Consider the set X_v from OPT_1 eliminated in in line 6 of Algorithm 1, in the iteration when link l_v was added to the solution. Each link $l_w \in X_v$ is of length at least d_{vv} and has its sender of distance at most $c \cdot d_{vv}$ from receiver r_v. Therefore, all senders in X_v are within distance $2c \cdot d_{vv} \leq$

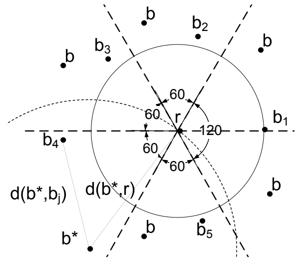


Fig. 1. Constructing a guarding set of size at most 5 for the red point r ($G_i(r) = \{b_1, \cdots, b_5\}$).

$2c{\cdot}d_{ww}$ from s_w. By Lemma 4.2, there can be at most $(2c{+}1)^\alpha$ senders in X_v. ∎

For the second part of the proof, i.e., to bound the number of deleted links in the second elimination step (line 7) of Algorithm 1, we will need the following two definitions and a combinatorial lemma, to which we refer as the *blue-dominant centers lemma*.

Definition 4.1: Let $\mathcal{R}$ and $\mathcal{B}$ be two disjoint sets of points in a metric space $(\mathcal{V}, d)$. Let's call them *red* and *blue* points, respectively. For q a positive integer, a point $b \in \mathcal{B}$ is *q-blue-dominant* if every ball $B_\delta(b)$ around b, comprised by points w such that $d(w, b) \leq \delta$, contains $q \in \mathbb{Z}^+$ times more blue points than red points. Formally,

$$\forall \delta \in \mathrm{R}_0^+ : |B_\delta(b) \cap \mathcal{B}| > q \cdot |B_\delta(b) \cap \mathcal{R}|.$$

Definition 4.2: Let $\mathcal{R}$ and $\mathcal{B}$ be defined as above. Let $r \in \mathcal{R}$ be a red point and $G \subseteq \mathcal{B}$ be a set of blue points. We say that G is *guarding* r if for all $b \in \mathcal{B} \setminus G$, we have that $B_{d(b,r)}(b) \cap G \neq \emptyset$.

Lemma 4.4: (*Blue-dominant centers lemma in 2D*) Let $\mathcal{R}$ and $\mathcal{B}$ be two disjoint sets of red and blue points in a 2-dimensional Euclidean space, and q be a positive integer. If $|\mathcal{B}| > 5q \cdot |\mathcal{R}|$ then there exists at least one q-blue-dominant point in $\mathcal{B}$.

Proof: Process the points in $\mathcal{R}$ in an arbitrary order while maintaining a subset $\mathcal{B}'$ of $\mathcal{B}$ as follows (initially, $\mathcal{B}' = \mathcal{B}$). For $r \in \mathcal{R}$ construct q guarding sets $G_i(r) \subseteq \mathcal{B}', 1 \leq i \leq q$, (guarding r relative to the current $\mathcal{B}'$) and remove $G_i(r)$ from $\mathcal{B}'$ iteratively for every i.

We claim that it is possible to construct a guarding set $G_i(r)$ of size at most 5. The procedure to construct $G_i(r)$ is as follows (see Figure 1). Consider a red point r. Include a closest blue point $b_1 \in \mathcal{B}'$ in $G_i(r)$. Draw 5 sectors originating at r in the following manner. The first sector has 120° and is centered at b_1, the remaining 4 sectors have 60° each and evenly divide the remaining area around r. For each of these 4 sectors sec_j, include the closest blue point $b_j \in sec_j$ in $G_i(r)$ (if sec_j has no blue points from $\mathcal{B}'$, skip this sector). Now $G_i(r)$ has size at most 5 and we claim that it is guarding r. Suppose not. Then there is a point $b^* \in \mathcal{B}' \setminus G_i(r)$ with $B_{d(b^*,r)}(r) \cap G_i(r) = \emptyset$. Suppose b^* is located in sec_j and we selected blue point b_j from sec_j into $G_i(r)$. This means that $d(b^*, b_j) > d(b^*, r)$, which implies that the sector angle is larger than 60°. (Note that if $G_i(r)$ contains no point b_j from sector sec_j, then b^* would have been picked to guard r in that sector, also establishing a contradiction.)

After going through all points in $\mathcal{R}$, the set $\mathcal{B}'$ is still nonempty by the assumption on the relative sizes of $\mathcal{R}$ and $\mathcal{B}$. We claim that every point in $\mathcal{B}'$ is now q-blue-dominant. This holds since (1) all $G_i(r)$s are pairwise disjoint and (2) every ball $B_\delta(b^*), b^* \in \mathcal{B}'$, that contains a red point r, contains also q blue points, one in each $G_i(r), 1 \leq i \leq q$. Hence, for every blue node $b^* \in \mathcal{B}'$, every ball $B_\delta(b^*)$ contains q times more blue points than red points ("more", since the center b^* is also blue). ∎

Using the result of Lemma 4.4, we are now able to bound OPT_2, the number of links deleted in the second elimination step (line 7) of the algorithm.

Lemma 4.5: $OPT_2 \leq \rho_2 \cdot ALG$, where $\rho_2 = 3^{\alpha+1} \cdot 5$.

Proof: Suppose otherwise. Consider the set of senders from $ALG \cup OPT_2$. Label those from OPT_2 as blue ($\mathcal{B} = \{s_b | l_b \in OPT_2\}$) and those from ALG as red ($\mathcal{R} = \{s_r | l_r \in ALG\}$). By Lemma 4.4, there is a q-blue-dominant point (sender) s^* in $\mathcal{B}$, where $q = 3^{\alpha+1}$. We shall argue that the link l^* would have been picked by our algorithm, leading to a contradiction.

Consider any red point $s_r \in \mathcal{R}$. Let $G^*(s_r)$ be the set of points (senders) in s_r's guarding set that are closer to s^* than s^* is to s_r. They are all within radius $d(s^*, s_r)$ from s^*. By the blue-dominant center property, $|G^*(s_r)| \geq q$. By Lemma 4.2, we have that $d(s^*, s_r) \geq 2d(s^*, r^*)$. By the triangular inequality, it then follows that $d(r^*, s_r) \geq d(s^*, s_r) - d(s^*, r^*) \geq (1/2)d(s^*, s_r)$ and for each $s_b \in G^*(s_r)$, $d(r^*, s_b) \leq d(s^*, s_b) - d(s^*, r^*) \leq (3/2)d(s^*, s_r)$. The relative interference of the red sender s_r on r^* is then bounded by

$$RI_{s_r}(l^*) = \frac{d(s^*, r^*)^\alpha}{d(s_r, r^*)^\alpha} \leq 2^\alpha \cdot \frac{d(s^*, r^*)^\alpha}{d(s^*, s_r)^\alpha}.$$

In comparison, the combined relative interference of the blue senders $s_b \in G^*(s_r)$ on r^* is at least

$$\begin{aligned} \sum_{s_b \in G^*(s_r)} RI_{s_b}(l^*) &= \sum_{s_b \in G^*(s_r)} \frac{d(s^*, r^*)^\alpha}{d(r^*, s_b)^\alpha} \\ &\geq q \left(\frac{2}{3}\right)^\alpha \frac{d(s^*, r^*)^\alpha}{d(s^*, s_r)^\alpha} \\ &\geq \left(\frac{q}{3^\alpha}\right) \cdot RI_{s_r}(l^*) \\ &> 2 \cdot RI_{s_r}(l^*). \end{aligned}$$

Since this holds for any $s_r \in \mathcal{R}$, the total interference that r^* receives from blue senders (those in OPT_2) is at least twice as high as the interference it would receive from the red senders (those in ALG). Since l^* is in OPT, it is affected by at most 1 by OPT_2. So we have

$$a_{ALG}(l^*) < \frac{1}{2} \cdot a_{OPT_2}(l^*) \leq \frac{1}{2}.$$

Since the affectedness of l^* is less than $2/3$, it would not have been deleted by Algorithm 1, which establishes the contradiction. ∎

Theorem 4.6: The approximation ratio of Algorithm 1 is $O(1)$.

Proof: The result follows by adding the bounds of Lemmas 4.3 and 4.5, which results in $OPT \leq OPT' + ALG \leq ALG(\rho_1 + \rho_2 + 1)$. ■

C. Approximation Ratio of Minimum-Length Scheduling

In this section we apply our (maximization) one-slot scheduling algorithm to derive a minimum-length schedule. The minimum-length scheduling algorithm (for a description in pseudo-code see Algorithm 2) consists in iteratively computing a one-slot schedule using Algorithm 1. Each one-slot solution is scheduled in a separate slot, and the remaining links are repeatedly used as input to Algorithm 1. The procedure continues until all links in L have been scheduled.

Algorithm 2 Multi-Slot Scheduling Algorithm

input: Set of links $L = \{l_1, \cdots, l_n\}$;
output: Schedule $\mathcal{S} = \{\mathcal{S}_1, \cdots, \mathcal{S}_T\}$;
$t := 0$;
repeat
    $\mathcal{S}_t := OneSlotSchedule(L)$; (Algorithm 1)
    $L := L \setminus \mathcal{S}_t$;
    $t := t + 1$;
until $L = \emptyset$
return $\mathcal{S}$;

The correctness of the obtained schedule has been proved in Lemma 4.1, and in the following theorem we show that the overall approximation ratio of Algorithm 2 is $O(\log n)$.

Theorem 4.7: The approximation ratio of Algorithm 2 is $O(\log n)$.

Proof: For each iteration t of Algorithm 2, define *cost-effectiveness* of a one-slot schedule $\mathcal{S}_t$ to be the average cost at which it schedules new elements, i.e. $1/|\mathcal{S}_t|$, and define the price $p(l_i), l_i \in \mathcal{S}_t$ of a link to be the average cost at which it is scheduled. Note that the total cost of a schedule is $\sum_{i=1}^{n} p(l_i)$. Number the links in input L in the order in which they were scheduled by Algorithm 2, resolving ties arbitrarily. Let $l_1, \cdots, l_n$ be this numbering. In any iteration, the optimum solution can schedule the remaining links at a total cost of at most OPT. Therefore, among all possible one-slot schedules, there must be one having cost-effectiveness of at most $OPT/|n - i + 1|$. Since Algorithm 1 selected $\mathcal{S}_t$ of size at most a constant factor (say ρ) smaller than the best possible, it follows that

$$p(l_i) \leq \rho \cdot \frac{OPT}{(n - i + 1)}.$$

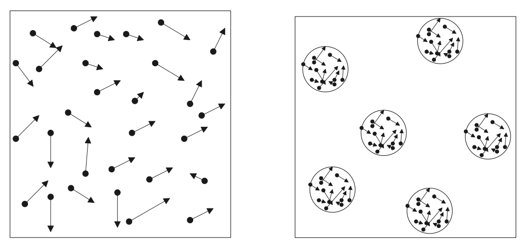
This gives a total cost of

$$\sum_{i=1}^{n} p(l_i) \leq \left(1 + \frac{1}{2} + \cdots + \frac{1}{n}\right) \rho \cdot OPT = O(\log n) OPT.$$

■

V. Simulation Results

In this section we present some simulation results to better illustrate the practical appeal of our scheduling algorithm (to which we refer as ApproxLogN). We generated two kinds of topologies: *random* and *clustered* (see Figures 2(a) and 2(b)). In the random topology, n receiver nodes were distributed uniformly at random on a plane field of size 1000x1000 units, and n senders were positioned uniformly at random inside discs of radius l_{max} around each of the receivers. In the clustered topology, n_C cluster center positions were selected uniformly at random on the plane, and n/n_C sender-receiver pairs were positioned uniformly at random inside discs of radius r_C around each of them. The clustered topology aims to simulate a scenario of heterogeneous density distribution. In practice, networks with heterogenous topologies are more typical. Consider, for example, a sensor network. In some spots of interest the density of sensors is expected to be much higher in order to capture all the desired data, whereas some locations are expected to contain the minimum necessary amount of nodes just to maintain connectivity.



(a) Random. (b) Clustered.

Fig. 2. Simulated topologies: 1Kx1K field, $\alpha = 3$, $\beta = 1.2$, $N = 0$

We compare the performance of ApproxLogN to the performance of two other scheduling algorithms: GreedyPhysical (proposed in [2]) and ApproxDiversity (proposed in [9]). As ours, both are polynomial-time algorithms, specifically designed for the SINR model (see Related Work Section). In all experiments, the number of simulations was chosen large enough to obtain sufficiently small confidence intervals.

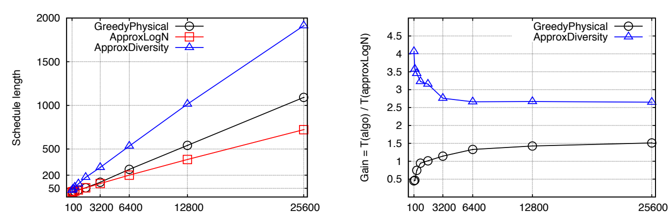


Fig. 3. Random Topology: $l_{max} = 20$.

Firstly, we analyze the lengths of the schedules as a function of the number of nodes ($n \in \{100 \cdot 2^0, 100 \cdot 2^1, \cdots, 100 \cdot 2^8\}$). In Figures 3(a) and 3(b) the results for random topology are shown. Since this scenario is not very challenging, all three algorithms have good performance, computing schedules of comparable sizes. GreedyPhysical presents slightly better performance in very low density scenarios (less than 1600 nodes). As the density increases, however, ApproxLogN presents increasingly better relative performance. In high densities (25.6K nodes) it computes, on average, 50% shorter schedules than GreedyPhysical and 2.5 times shorter schedules than ApproxDiversity.

In Figures 4(a) and 4(b) the results for the clustered topology are shown. As could be expected, the greedy algorithm is not able to deal with this more difficult scenario very efficiently. Even in very sparse topologies (100 nodes), GreedyPhysical computes 3 times longer schedules than ApproxLogN. As the density increases, the relative performance of the greedy algorithm deteriorates. ApproxLogN and ApproxDiversity compute even shorter schedules than in the random case, which indicates that they are able to schedule many clusters in parallel. The performance of ApproxLogN is still superior to that of ApproxDiversity.

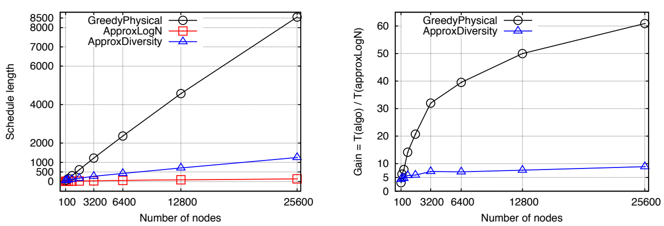


Fig. 4. Clustered Topology: $n_C = n/10$, $r_C = 10$.

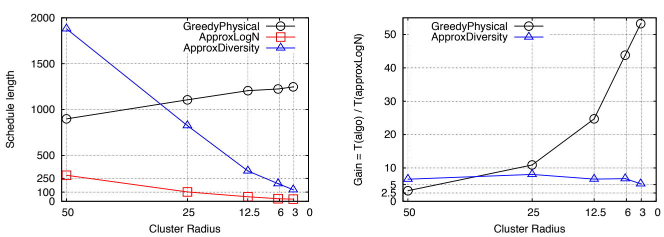


Fig. 5. Clustered Topology: $n = 3.2K$, $n_C = n/10$.

In Figures 5(a) and 5(b) we analyze the influence of the cluster radius. In topologies with smaller clusters, i.e., in scenarios with higher density heterogeneity, the difference in performance becomes more accentuate. Whereas GreedyPhysical's performance slightly decreases with decreasing cluster radius, ApproxLogN and ApproxDiversity are able to compute ever shorter schedules. Smaller cluster radius means more separate clusters, which makes it easier to schedule clusters in parallel. GreedyPhysical, however, is not able to take advantage of this possibility. Among all three algorithms, ApproxLogN presents the best performance in all cases.

Next we analyze the influence of the path-loss exponent α in both random (Figures 6(a) and 6(b)) and clustered

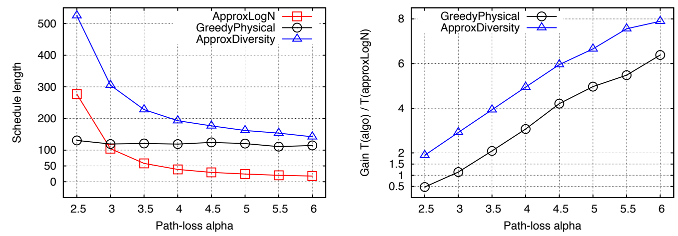


Fig. 6. Random topology: $n = 3.2K$, $l_{max} = 20$.

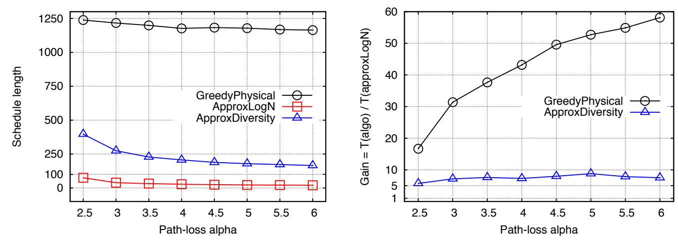


Fig. 7. Clustered topology: $n = 3.2K$, $n_C = n/10$, $r_C = 10$.

(Figures 7(a), and 7(b)) topologies. It can be seen that the performances of ApproxLogN and ApproxDiversity improve with increasing α, whereas GreedyPhysical is more or less invariant to the path loss exponent. For $\alpha < 3$, in the random topology, GreedyPhysical presents a better performance than the other two algorithms. In the clustered topology, however, its performance is very poor even for low α and deteriorates relative to the other two approaches with increasing α in both kinds of topologies. Among all three algorithms, ApproxLogN presents the best performance for all values of α in the clustered topology and for $\alpha \geq 3$ in the random case.

To sum up, the simulations show that ApproxLogN, besides having an exponentially better analytical approximation ratio, presents advantages in challenging practical scenarios, such as high-density and heterogeneous-density networks. GreedyPhysical showed to be a reasonable heuristic for low-density uniformly distributed networks, besides having the advantage of being robust to variable path-loss. Its performance, however, rapidly deteriorates in more difficult topologies. ApproxDiversity, although robust to increasing density and heterogeneity of the network, presented performance inferior to that of ApproxLogN in all simulated scenarios.

VI. Non-Approximability in Abstract SINR

In this section we show that it is NP-hard to approximate the scheduling problem in the so called "abstract SINR" model [9] to within a factor of $n^{1-\varepsilon}$, for any constant $\varepsilon > 0$. We distinguish "abstract SINR" ($SINR_A$) from "geometric SINR" ($SINR_G$) model according to the freedom with which the path-loss matrix can be defined. In the $SINR_A$ model, as opposed to the $SINR_G$ model, path-loss between nodes is not constrained by their Euclidean coordinates, but can be set arbitrarily (i.e., triangular inequality must not be preserved when defining the path-loss matrix). Note that $SINR_A$ is more general and therefore a "harder" model than $SINR_G$, which we have been using to derive the results in the previous sections.

Theorem 6.1: There is no $n^{1-\varepsilon}$ factor approximation algorithm for the scheduling problem in the $SINR_A$ model, assuming P $\neq$ NP, for any constant $\varepsilon > 0$.

Proof: We will prove the result by presenting a *gap-preserving* reduction from the graph coloring problem. In [20] it was shown that it is NP-hard to approximate the graph coloring problem to within $n^{1-\varepsilon}$ for all $\varepsilon > 0$.

Consider an instance I_C of the graph coloring problem defined for an undirected graph $G = (V, E)$ on n vertices. We construct (in polynomial time) an instance I_S of scheduling, such that

$$OPT(I_C) \leq k \Rightarrow OPT(I_S) \geq k, \quad (4)$$
$$OPT(I_C) > n^{1-\varepsilon}k \Rightarrow OPT(I_S) > n^{1-\varepsilon}k. \quad (5)$$

For each $v \in V$, we add a link $l_v = (r_v, s_v)$. The SINR parameters are set to $\beta = 1$, $N = 0$, and the path-loss matrix A is defined as follows:

- $(v, w) \in E \Rightarrow a(r_v, s_w) = a(r_w, s_v) = 1$,
- $(v, w) \notin E \Rightarrow a(r_v, s_w) = a(r_w, s_v) = n$,
- $v = w \Rightarrow a(r_v, s_v) = 1$.

To see that (4) holds, assume that we can color I_C with k or less colors. We claim that links associated to nodes with the same color (let's call each such subset $V(c_i), 1 \leq i \leq k$) can be scheduled concurrently, giving a schedule of length k (or less). Since nodes colored with the same color are not adjacent, the SINR at each link $l_v, v \in V(c_i), 1 \leq i \leq k$ can be lower bounded by

$$SINR(l_v) = \frac{\frac{P}{1}}{\sum_{\substack{w \in V(c_i),\\ w \neq v}} \frac{P}{n}} \geq \frac{n}{n-1} > 1 = \beta.$$

To see that (5) holds, assume we cannot color I_C with $\leq n^{1-\varepsilon}k$ colors. We have to show that I_S cannot be scheduled in $n^{1-\varepsilon}k$ time-slots or less. Assume that we could, and consider a schedule of size $n^{1-\varepsilon}k$. Since any coloring of this size must have a violation (an edge to a node x of the same color) at least one node v, we calculate the SINR at the link l_v associated to this node:

$$SINR(l_v) \leq \frac{\frac{P}{1}}{\frac{P}{1} + \sum_{\substack{w \in V(c_i),\\ w \neq v, w \neq x}} \frac{P}{n}} < 1 = \beta.$$

This shows that any schedule of size $n^{1-\varepsilon}k$ or less will have at least one violated node, given the necessary contradiction. ■

VII. Going Beyond Two Dimensions

In this section we look into the issue of whether the analysis of Algorithm 1 could be extended beyond the two-dimensional Euclidean space. We show that, by adjusting the constants, the same techniques work in D-dimensional Euclidean spaces, provided that the path-loss exponent is high enough ($\alpha > D$). For the first part of the analysis (Lemmas 4.1 and 4.3), we compute the value of the constant c (see Def. 3) for three dimensions. For higher dimensions D, c_D can be computed analogously, by working with volumes of n-spheres, instead of disc areas as for 2D. For the second part of the proof (Lemmas 4.4 and 4.5), we show that the *blue-dominant centers lemma* can be extended to more general metric spaces, by applying the concept of *independence-dimension*, which we define in Def. 7.1.

Lemma 7.1: Algorithm 1 produces a valid schedule in a three-dimensional Euclidean space, if $\alpha > 3$ and $c = c_{3D}$,

$$c_{3D} = \max\left(2, \left(2^5 3^3 7 \beta \frac{\alpha-2}{\alpha-3}\right)^{\frac{1}{\alpha}}\right). \quad (6)$$

Proof: The proof proceeds along the lines of Lemma 4.1, replacing disc areas for ball volumes, which renders:

$$\begin{aligned} I_{S_v^+}(l_v) &< \sum_{k=1}^{\infty} I_{Ring_k^{3D}}(l_v) \\ &\leq \sum_{k=1}^{\infty} \frac{1}{k^{\alpha-2}} \cdot \frac{P}{d_{vv}^{\alpha}} \frac{2^5 3^2 7}{c_{3D}^{\alpha}} \\ &< \frac{\alpha-2}{\alpha-3} \cdot \frac{P}{d_{vv}^{\alpha}} \frac{2^5 3^2 7}{c_{3D}^{\alpha}}, \end{aligned}$$

which, by plugging in the value of c_{3D} results in affectedness of any scheduled link l_v by longer links of $a_{S_v^+}(l_v) \leq 1/3$. This, together with the affectedness by shorter links of $a_{S_v^-}(l_v) < 2/3$, guarantees correct reception at all concurrently scheduled links. ■

To prove the approximation ratio, firstly we count the number of links OPT_1 scheduled in the optimal solution, but eliminated by the algorithm in line 6. As in Lemma 4.3, we have $OPT_1 < \rho_1(3D) \cdot ALG$, where $\rho_1(3D) = (2 \cdot c_{3D} + 1)^{\alpha}$ and c_{3D} is defined in (6).

For the second part of the proof, we investigate a little bit further into the *blue-dominant centers lemma* (4.4). More specifically, given a metric space $(\mathcal{V}, d)$, where $\mathcal{V}$ is a set of points and d is a distance function, we want to find out for which metric spaces there is a constant κ with the following property. Whenever we have two disjoint sets of red and blue points, $\mathcal{R} \subset \mathcal{V}$ and $\mathcal{B} \subset \mathcal{V}$, with

$$|\mathcal{B}| > \kappa \cdot |\mathcal{R}|, \quad (7)$$

then at least one of the blue points is (1-)blue-dominant (Def. 4.1).

Definition 7.1: Let $(\mathcal{V}, d)$ be a metric space and let $v \in \mathcal{V}$. A set $I \subseteq \mathcal{V} \setminus \{v\}$ is called *independent* with respect to v if

$$\forall w \in I : B_{d(v,w)}(w) \cap I = \{v, w\}.$$

The maximum cardinality of a set Q of points in $\mathcal{V}$ that is independent relative to some point $v \in \mathcal{V} \setminus Q$ is called the *independence-dimension* of $(\mathcal{V}, d)$, denoted by γ.

Lemma 7.2: Let $\mathcal{R}$ and $\mathcal{B}$ be as usual and let $r \in \mathcal{R}$. There is always a subset G of $\mathcal{B}$ of cardinality at most γ that is guarding r (Def. 4.2).

Proof: Sort the points in $\mathcal{B}$ in order of non-decreasing distance to r. Proceed through the sorted list and add a point to an initially empty set G when the resulting set remains independent w.r.t. r. The claim is that the set G is guarding r in the end (when $G \leq \gamma$ since we kept it independent w.r.t. r). Suppose not. Then there is a point $b \in \mathcal{B} \setminus G$ with $B_{d(b,r)}(b) \cap G = \emptyset$. So the reason we did not add b to G when we encountered it is that there was already a point $b' \in G$ so that $b \in B_{d(b',r)}(b')$. But note that $d(b', r) \leq d(b, r)$, so $b \in B_{d(b',r)}(b')$ ($\Leftrightarrow d(b', b) \leq d(b', r)$) implies $b' \in B_{d(b,r)}(b)$, which is a contradiction. ■

Lemma 7.3: (*Blue-dominant centers lemma*) Let $(\mathcal{V}, d)$ be a metric space with finite independence-dimension γ. If $\mathcal{R}$ and $\mathcal{B}$ are disjoint finite sets of points in $\mathcal{V}$ with $|\mathcal{B}| > q \cdot \gamma \cdot |\mathcal{R}|$ then there is a q-blue-dominant point in $\mathcal{B}$.

Proof: The proof is along the same lines of Lemma 4.4, only replacing 5 by γ, which is the independence-dimension of the two-dimensional Euclidean space, and applying Lemma 7.2 to guarantee that the size of each guarding set $G_i(r)$ is at most γ. ■

We can use the more general version of the *blue-dominant centers lemma* (Lemma 7.3) to bound the number of links OPT_2 eliminated by Algorithm 1 in its second elimination step (line 7). As in Lemma 4.5, we have $OPT_2 < \rho_2(\gamma) \cdot ALG$, where $\rho_2(\gamma) = 3^{\alpha+1} \cdot \gamma$ and γ is the independence-dimension of our metric space (e.g. $\gamma = 12$ in the 3D Euclidean space).

In Lemma 7.3 we deduced that the constant κ in (7) can be chosen as the so-called independence-dimension. This means that our scheduling algorithm can be applied in spaces with bounded-independence property. Consider, for instance, spaces induced by signal distortions. Our $SINR_G$ model makes an overly optimistic assumption that the radios are perfectly isotropic and there are no obstructions. What if the signal is attenuated by a certain factor in one direction but by another factor in another direction? Then we still have a bounded-independence property. This means that, although our algorithm might not be valid in the (overly pessimistic) $SINR_A$ model, it can handle more realistic scenarios than the $SINR_G$ model, where the distortion is such that the independence-dimension of the induced space is bounded.

VIII. Conclusion

Determining the capacity of a wireless network has been a subject of intense research in the past few years. So far, all solutions have either considered special-case topologies, or presented optimality guarantees that become arbitrarily bad depending on the topology of the network. In this work we have proposed the first scheduling algorithm with an approximation guarantee independent of the topology of the network.

If we define network throughput capacity, as in [12], to be the number of bits per second every node can on average transmit to its destination, we can compute it in the following way: Given a set L of n communication requests, such that each node is able to transmit at W bits per second over a common wireless channel (with fixed power level and no routing), the capacity $C(L)$ of a network L lies in the interval

$$\frac{W}{T} \leq C(L) < \frac{W}{T} \cdot O(\log n),$$

where T is the size of the schedule returned by Algorithm 2.

We hope that this is a significant step in understanding the capacity of wireless networks formed by arbitrary topologies. Some problems, however, remain open, most obviously whether a constant minimum-length scheduling approximation is feasible.

IX. Acknowledgments

We would like to thank Xiaohua Xu, Shaojie Tang and Xiang-Yang Li for their helpful remarks and for pointing out some mistakes and proposing corrections in an earlier version of the manuscript.

References

[1] S. A. Borbash and A. Ephremides. Wireless Link Scheduling With Power Control and SINR Constraints. *IEEE Tran. on Inf. Theory*, 52(11):5106–5111, 2006.

[2] G. Brar, D. Blough, and P. Santi. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *Mobicom*, 2006.

[3] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Cross-layer Latency Minimization for Wireless Networks using SINR Constraints. In *Mobihoc*, 2007.

[4] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Approximation Algorithms for Computing Capacity of Wireless Networks with SINR Constraints. In *Infocom*, 2008.

[5] R. L. Cruz and A. Santhanam. Optimal Routing, Link Scheduling , and Power Control in Multi-hop Wireless Networks. In *Infocom*, 2003.

[6] T. A. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. In *Infocom*, 2002.

[7] Y. Gao, J. C. Hou, and H. Nguyen. Topology control for maintaining network connectivity and maximizing network capacity under the physical model. In *Infocom*, 2008.

[8] A. Giridhar and P. R. Kumar. Computing and communicating functions over sensor networks. *IEEE JSAC*, 23(4), 2005.

[9] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in Geometric SINR. In *Mobihoc*, 2007.

[10] O. Goussevskaia and R. Wattenhofer. Complexity of scheduling with analog network coding. In *FOWANC*, 2008.

[11] J. Gronkvist and A. Hansson. Comparison between graph-based and interference-based STDMA scheduling. In *Mobihoc*, 2001.

[12] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Trans. Information Theory*, 2000.

[13] C. Joo, X. Lin, and N. Shroff. Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-Hop Wireless Networks. In *Infocom*, 2008.

[14] U. Kozat and L. Tassiulas. Throughput capacity of random ad hoc networks with infrastructure support. In *Mobicom*, 2003.

[15] V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic Aspects of Capacity in Wireless Networks. In *Sigmetrics*, 2005.

[16] T. Moscibroda. The worst-case capacity of wireless sensor networks. In *IPSN*, 2007.

[17] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Infocom*, 2006.

[18] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-Based Models. In *Hotnets*, November 2006.

[19] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology Control meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *Mobihoc*, 2006.

[20] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *STOC*, 2006.