Complexity in Geometric SINR

Olga Goussevskaia
Computer Engineering and Networks Laboratory
ETH Zurich
8092 Zurich
golga@tik.ee.ethz.ch

Yvonne Anne Oswald
Computer Engineering and Networks Laboratory
ETH Zurich
8092 Zurich
oswald@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich
8092 Zurich
wattenhofer@tik.ee.ethz.ch

ABSTRACT

In this paper we study the problem of scheduling wireless links in the geometric $SINR$ model, which explicitly uses the fact that nodes are distributed in the Euclidean plane. We present the first NP-completeness proofs in such a model. In particular, we prove two problems to be NP-complete: Scheduling and One-Shot Scheduling. The first problem consists in finding a minimum-length schedule for a given set of links. The second problem receives a weighted set of links as input and consists in finding a maximum-weight subset of links to be scheduled simultaneously in one shot. In addition to the complexity proofs, we devise an approximation algorithm for each problem.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems, Geometrical Problems and Computations, Sequencing and Scheduling; G.2.1 [**Discrete Mathematics**]: Combinatorics, Combinatorial Algorithms; G.2.2 [**Discrete Mathematics**]: Graph Theory, Network Problems.

General Terms

Algorithms, Theory.

Keywords

Wireless, Ad-Hoc Networks, SINR, Geometric SINR, Scheduling, Weighted Scheduling, NP-complete, Approximation Algorithms.

1. INTRODUCTION

How long does it take to find an optimal schedule for a given set of communication links in a wireless ad-hoc network? Is this problem difficult – even in a simplified model? What if we do not need to schedule all communication links, but simply want to choose the most "valuable" ones? And how hard is it to produce a result which is not necessarily optimal, but only falls short of an optimal solution by a guaranteed factor? In this paper, we study these questions. In particular, we present NP-completeness results and approximation algorithms for two problems: Scheduling and One-Shot Scheduling.

When studying wireless networks, the choice of the interference model is of fundamental significance. Not only has the selected model to incorporate the nature of real networks, but also to facilitate the development of rigorous reasoning. One model of choice is the "*abstract*" Signal-to-Interference-plus-Noise-Ratio (or short, $SINR_A$) model. In the $SINR_A$ model, a signal is received successfully depending on the ratio of the received signal strength and the sum of the interference caused by nodes sending simultaneously (plus noise).

The wireless networking community usually adheres to a *geometric* $SINR$ (or short, $SINR_G$) model. In the $SINR_G$ model, the nodes live in space, and the gain (or signal attenuation) between two nodes is determined by the distance between the two nodes. In particular, a signal fades with the distance to the power of alpha, alpha being the so-called path-loss parameter.

$SINR_G$ makes some simplifying assumptions, such as perfectly isotropic radios, no obstructions, or a constant ambient noise level. On the other hand, $SINR_A$ is not all that realistic either, as it allows arbitrary values in the gain matrix among the participating nodes of a wireless network. In reality, if a node u is close to a node v, which in turn is close to a node w, then u and w will also be close. So the entries in the gain matrix will be constrained by the other entries. Thus, $SINR_G$ is too optimistic, whereas $SINR_A$ is too pessimistic. Hence, a real network is positioned somewhere between the $SINR_G$ and $SINR_A$ models.

When studying algorithms or protocols, upper bounds should be derived for the pessimistic model, as an algorithm for a strictly[1] more pessimistic model will also work for reality. However, also the converse is true: If one is interested in lower bounds (impossibility results or capacity constraints), one must use the optimistic model. A strictly more optimistic model guarantees that results are applicable in practice.

In this paper we study two optimization problems in wireless networks: Scheduling and One-Shot Scheduling. Apart from presenting approximation algorithms, our main result

[1]Note that models are rarely strictly harder than reality; $SINR_A$ is a typical example, as $SINR_A$ does not include several difficulties of reality, e.g. short-term fading.

MobiHoc'07, September 9–14, 2007, Montreal, Quebec, Canada.

is the proof of hardness of these problems. In particular, we formally prove that Scheduling and One-Shot Scheduling are both NP-complete in the $SINR_G$ model. Since the $SINR_G$ model is weaker than reality, this implies that one cannot compute an optimal schedule of wireless requests in practice, unless $P = NP$.

To the best of our knowledge, these are the first NP-completeness proofs for $SINR_G$. As we will discuss in the related work section in more detail, there have been various NP-completeness proofs for wireless networks model, in particular for so-called unit disk graphs (UDG) or for the $SINR_A$ model. In contrast to our work, these proofs are graph-based. In an orthodox $SINR_A$ proof one establishes an arbitrary gain matrix between the participating nodes of a wireless network, giving $O(n^2)$ degrees of freedom. In particular, this allows to build a graph, as the gain between any two nodes can be set to either 1 ("link") or 0 ("no link"). One ends up with a standard graph, and it trivially follows that e.g. scheduling is as hard as coloring in graphs. A similar argument holds for proofs for the UDG model.[2]

In reality, however, gain cannot be chosen arbitrarily. As we argued before, the triangular inequality makes all the entries in the gain matrix interdependent. If we turn to the $SINR_G$ model, we must choose positions of the nodes in space (e.g. in a plane), which determines the attenuation between two nodes, giving only $O(n)$ degrees of freedom. Arguing that two nodes cannot transmit concurrently in a schedule becomes much harder, since the nodes all influence each other. This is what intuitively makes the problem harder. In $SINR_G$, one must always deal with the complete (weighted) graph; this asks for a different kind of proof.

The paper is structured as follows. In Section 2 we discuss some results in different interference models. In Section 3 we describe the $SINR_G$ model and the problems we address in this paper. In Section 4 we present the NP-completeness proofs for the Scheduling and the One-Shot Scheduling problem. In Section 5 we describe two approximation algorithms for the referred problems. Finally, in Section 6, we discuss our results and suggest directions for future research.

2. RELATED WORK

The problem of scheduling link transmissions in a wireless network in order to optimize one or more of performance objectives (e.g. throughput, delay, fairness or energy) has been a subject of much interest over the past decades.

An issue of prime importance is the complexity of scheduling problems. As has already been argued in the introduction, there have been various NP-completeness proofs for wireless networks. To the best of our knowledge, these proofs are either built for the UDG model [15, 19], or for the abstract $SINR$ model ($SINR_A$), and present reductions without a geometric representation. A typical such proof establishes an arbitrary gain matrix between the participating nodes, which results in a standard graph. The hardness is proved by a reduction from graph coloring, for example [3].

The joint problem of power control and scheduling with the objective of minimizing the total transmit power subject to the end-to-end bandwidth guarantees and the bit error rate constraints of each communication session is addressed by Kozat et al. in [18]. They prove their problem to be NP-complete by using a reduction from integer programming under the assumption that the values of the gain matrix can be chosen arbitrarily. Similarly, Leung and Wang [21] prove that the problem of maximizing data throughput by adaptive modulation and power control while meeting packet error requirements is NP-complete under the assumption that the values of the gain matrix are arbitrary. Another problem is proposed by Chatterjee et al. in [22] as the power constrained discrete rate allocation problem. A solution finds the rates at which the base station must transmit to each user including $SINR$ constraints. They prove that this problem is NP-complete for CDMA data networks by a reduction from the Knapsack problem using a gain matrix with gain value 1 for all links.

The problem of scheduling broadcast requests has been studied by Ephremides and Truong [7]. They show that in a generalized, *non-geometric* model, finding an optimal schedule is NP-complete, if no interference is tolerated. Other aspects of scheduling and power control using an arbitrary gain matrix are studied for instance in [3, 4, 6, 28, 29, 30].

One of the very few lower bounds for the $SINR_G$ model is due to Gupta and Kumar [13]. They analyze the overall capacity of ad-hoc networks in the $SINR_G$ model from an *information theoretic* perspective, and prove that a wireless network comprised of n nodes cannot provide a throughput of more than $\Theta(1/\sqrt{n})$. More recently, a study of data aggregation capacity in the $SINR_G$ has been performed. In [9], Giridhar and Kumar show that, in a random network, the maximum rate for computing divisible functions is $\Theta(1/\log n)$. Furthermore, in [23], Moscibroda shows that symmetric functions can be computed at rate $\Omega(1/\log^2 n)$ in every network, even if its nodes are positioned in a worst-case manner.

Of course the design of efficient algorithms for scheduling has been explored as well. In order to compute a time-schedule such that spatial reuse is maximized, most of the proposed schemes are based on traditional graph-theoretic models. They use a graph representation of a wireless network, modeling interference by some (often binary) graph property. For example, a set of "interference-edges" might be defined, containing pairs of nodes within a certain distance to each other, thus modeling interference as a local measure.

Graph-based scheduling algorithms usually employ an implicit or explicit coloring strategy, which neglects the aggregated interference of nodes located farther away. A variety of centralized and decentralized approximation algorithms have been proposed and their quality analyzed for this kind of model [14, 20, 24, 31, 32]. Most recently, Brar et al. [5] present a scheduling method that is based on a greedy assignment of weighted colors. Although these algorithms present extensive theoretical analysis, they are constrained to the limitations of a model that does not reflect the real nature of wireless networks. In particular, such graph-based models ignore the accumulated interference of a large number of distant nodes.

In [1, 10, 11], it is argued that the performance of graph-based algorithms is inferior to algorithms in more realistic $SINR$ models. More recently, Moscibroda et al. [26] show experimentally that the theoretical limits of any protocol, which obeys the laws of graph-based models, can be broken by a protocol explicitly defined for the $SINR_G$ model.

The computation of efficient schedules in the $SINR_G$ model has been studied in a more restricted number of pa-

[2]Not surprisingly as the G in UDG stands for graph.

pers. In [25], an efficient power-assignment algorithm, which schedules a strongly connected set of links in $O(\log^4 n)$ time slots in the $SINR_G$ model, is presented. The work of [2, 3, 16] proposes mathematical programming formulations for deriving optimal schedules. However, the resulting formulations are infeasible from a computational point of view as the running time is exponential in the input.

3. MODEL

In this paper we attend to the problem of scheduling communication requests (links) of nodes positioned in a Euclidean plane. The set of links is denoted by $L = l_1, \ldots, l_n$, where each link l_i represents a communication request from a sender s_i to a receiver r_i, which are determined by points in the plane. The Euclidean distance between two nodes s_i, r_j is denoted by $d_{ij} = d(s_i, r_j)$, so the length of link l_i is referred to by d_{ii}.

Choosing an appropriate interference model is crucial when studying scheduling in wireless networks. We use the standard *Signal-to-Interference-plus-Noise-Ratio* ($SINR$) model [13], where a message can be transmitted successfully depending on the ratio of the received signal strength and the sum of the interference caused by nodes sending simultaneously plus noise level. We assume a transmission can be decoded correctly if this ratio exceeds a hardware-dependent value β. This model captures important aspects of real wireless networks and it is at the same time succinct enough to allow a concise performance analysis.

More formally, the received power $P_{r_i}(s_j)$ of a signal transmitted by sender s_j at receiver r_i is

$$P_{r_i}(s_j) \;=\; \frac{P}{d_{ji}^{\alpha}},$$

where P is the transmission power and $d_{ji}^{-\alpha}$ comprises the propagation attenuation (link gain). The *path-loss exponent* α is a constant, whose exact value depends on external conditions of the medium (humidity, obstacles, etc.), as well as the exact sender-receiver distance. As common, we assume that $\alpha > 2$ [13].

Given a request $l_i = (s_i, r_i)$, we use the notation $I_{r_i}(s_j) = P_{r_i}(s_j)$ for any other sender s_j concurrent to s_i, in order to emphasize that the signal power transmitted by s_j is perceived at r_i as interference. The *total interference* I_{r_i} experienced by a receiver r_i is the sum of the interference power values created by all nodes in the network transmitting simultaneously (except the intending sender s_i), that is, $I_{r_i} := \sum_{s_j \neq s_i} I_{r_i}(s_j)$. Finally, let N denote the ambient noise power level. Then, r_i receives s_i's transmission if and only if

$$\begin{aligned} SINR(r_i) \;&=\; \frac{P_{r_i}(s_i)}{I_{r_i} + N} \\ &=\; \frac{P_{r_i}(s_i)}{\sum_{j \neq i} I_{r_i}(s_j) + N} \\ &=\; \frac{\frac{P}{d_{ii}^{\alpha}}}{\sum_{j \neq i} \frac{P}{d_{ji}^{\alpha}} + N} \;\geq\; \beta, \end{aligned} \tag{1}$$

where β is the minimum $SINR$ required for a successful message reception. In the sequel we assume $\beta \geq 1$.

In this work we assume that all nodes transmit with the same power level. This assumption is also referred to as *uniform power assignment scheme* [12]. This kind of power assignment has been widely adopted in practical systems and has been studied in depth in [33].

For the sake of simplicity, in the following analysis sections, we set $N = 0$ and ignore the influence of noise in the calculation of $SINR$. However, this has no significant effect on the results.

3.1 Scheduling Problem

The aim of an algorithm for the Scheduling problem is to generate a short sequence of link sets, such that the $SINR$ level is above a threshold β at every intended receiver in each link set and all links are scheduled successfully at least once.

More precisely, let L be a set of communication requests. A *schedule* is represented by $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_T)$, where $\mathcal{S}_t$ denotes a subset of links of L, designated to time slot t. As in [13], it is assumed without loss of generality that transmissions are slotted into synchronized slots of equal length and in each time slot t, a node can either transmit or remain silent.

The task of a scheduling algorithm is to schedule a set of communication requests L such that all messages are *successfully* received.

DEFINITION 3.1. *Consider a time slot t. The request $l_i = (s_i, r_i)$ is* successfully scheduled *in time slot t if r_i can decode message from s_i correctly according to the $SINR$ inequality* (1).

We aim at ensuring that after as few time slots as possible every link has been transmitted. The *scheduling complexity* defined in [25] is a measure that captures the amount of time required by a scheduling protocol to schedule requests in the physical $SINR$ model.

DEFINITION 3.2. *Let $\mathcal{S}_t$ be the set of all successfully scheduled links in time slot t. The Scheduling problem for L consists in finding a schedule $\mathcal{S}$ of minimal length T such that the union of all successfully transmitted links $\bigcup_{t=1}^{T(\mathcal{S})} \mathcal{S}_t$ equals L. An algorithm's* scheduling complexity *is the length of the schedule generated.*

Evidently, an algorithm's quality is reflected by its scheduling complexity. Ideally, a wireless scheduling protocol should achieve an optimal scheduling complexity in all networks and for arbitrary communication requests.

In the sequel, we assume that there are no conflicts in the transmission setup, i.e., each node is either a sender or a receiver and each receiver is associated with only one sender. These conflicts can be resolved efficiently by introducing additional nodes at the same position such that there is one sender-receiver pair for each link. Therefore we neglect them for simplicity's sake.

3.2 One-Shot Scheduling Problem

In contrast to the Scheduling problem, where we were interested in a schedule for all links, the objective of an algorithm solving the One-Shot Scheduling problem is to pick a subset of weighted links such that the total weight is maximized and the $SINR$ level is at least β at every scheduled receiver. In other words, we attempt to use one slot to its full capacity.

Formally, let L be a set of communication requests, where each link l_i is assigned a weight w_i. A set $\mathcal{S} = (l_1, l_2, \ldots, l_m) \subseteq L$ is a solution to an instance of a One-Shot Scheduling problem if the following two conditions hold:

$$\begin{aligned} \mathcal{S} &= \operatorname*{argmax}_{\mathcal{S}' \subseteq L} \sum_{l_j \in \mathcal{S}'} w_j, \\ SINR(r_j) &\geq \beta, \quad \forall l_j \in \mathcal{S}. \end{aligned}$$

4. COMPLEXITY OF PROBLEMS IN GEOMETRIC SINR

Solving problems in the $SINR$ setting is very difficult. Even finding an algorithm determining a good approximation for every problem instance is hard, as is documented by the vast amount of literature with heuristics on this subject [2, 3, 8, 10, 16, 25, 27].

As mentioned in the introduction and related work section, there are hardly any results on the hardness of problems in a geometric setting. However, insights on the complexity are very important for the design of efficient algorithms. In this section we analyze the Scheduling problem and the One-Shot Scheduling problem and prove them to be NP-complete in the $SINR_G$ model.

4.1 Scheduling Problem

Proving the Scheduling problem to be NP-hard implies that there exists no polynomial time algorithm for determining an optimal schedule, unless $P = NP$. It is widely believed that an NP-hard computational problem is not tractable efficiently.

We proceed by first showing that the decision version of Scheduling problem under uniform power assignment scheme is in the complexity class NP and then give a polynomial time reduction from the Partition problem, an NP-complete special case of the well known Subset Sum problem. If the solution to an instance of the Scheduling problem implies a solution to any instance of the Partition problem, Scheduling must be at least as hard as Partition.

LEMMA 4.1. *Scheduling is in NP.*

PROOF. A decision problem is in NP if one can verify a solution deterministically in polynomial time. To decide whether a schedule of a given size permits the successful transmission of all links, we have to verify, for every link, whether there is a time slot assigned to it and if the $SINR$ exceeds β under the interference of the links scheduled in the same slot. Since computing the $SINR$ level for each receiver in its time slot can be done in $O(n^2)$ time, a schedule is an efficiently verifiable witness for this problem. □

LEMMA 4.2. *The Partition problem is reducible to the Scheduling problem in polynomial time.*

PROOF. The Partition problem (proved to be NP-complete by Karp in his seminal work [17]) can be formulated as follows: Given a set $\mathcal{I}$ of integers, is it possible to divide this set into two subsets $\mathcal{I}_1$ and $\mathcal{I}_2$, such that the sums of the numbers in each subset are equal? The subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ must form a partition in the sense that they are disjoint and they cover $\mathcal{I}$.

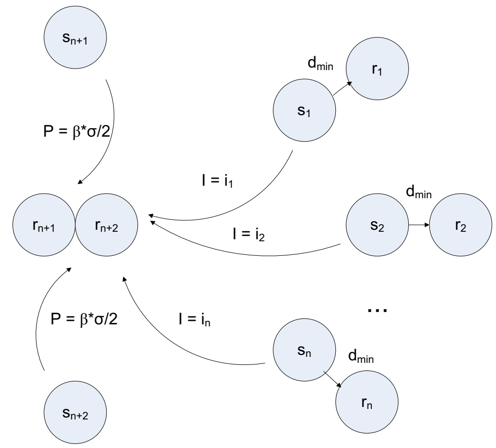

Figure 1: Reduction from Partition: link l_{n+1} (or l_{n+2}) can be scheduled if and only if the interference caused by simultaneously scheduled links $s_j, j \in \{1 \cdots n\}$ is less our equal to $\sigma/2$.

Partition problem: Find $\mathcal{I}_1, \mathcal{I}_2 \subset \mathcal{I} = \{i_1, \ldots, i_n\}$ s.t.:

$$\begin{aligned} \mathcal{I}_1 \cap \mathcal{I}_2 &= \emptyset, \\ \mathcal{I}_1 \cup \mathcal{I}_2 &= \mathcal{I}, \quad \text{and} \\ \sum_{i_j \in \mathcal{I}_1} i_j &= \sum_{i_j \in \mathcal{I}_2} i_j = \frac{1}{2} \sum_{i_j \in \mathcal{I}} i_j. \end{aligned}$$

The proof proceeds as follows. First, we define a many-to-one reduction from any instance of the Partition problem to a geometric instance of the Scheduling problem. Then, we argue that the instance of the Scheduling problem cannot be scheduled in $T \leq 1$ time slots, but can be scheduled in $1 < T \leq 2$ time slots if and only if the instance of the Partition problem is solved.

Let us look at a set $\mathcal{I} = \{i_1, \ldots, i_n\}$ of integers, where the elements of $\mathcal{I}$ add up to σ,

$$\sum_{j=1}^{n} i_j = \sigma.$$

Without loss of generality, we can assume all elements to be distinct and positive. We construct the following Scheduling problem instance with $n+2$ links $L = \{l_1, \ldots, l_{n+2}\}$ (cf. Figure 1). We refer to the sender node belonging to l_j as s_j and the receiver node r_j. We assign each of these nodes a position (X,Y) in the plane. For each integer i_j in I we set the x-axis coordinate of s_j to $(P/i_j)^{1/\alpha}$,

$$pos(s_j) = \left(\left(\frac{P}{i_j} \right)^{1/\alpha}, 0 \right) \; \forall 1 \leq j \leq n.$$

Next, we designate for every $r_i, 1 \leq i \leq n$ its position to be at distance $d_{\min}$ to its sender s_i, where

$$d_{\min} = P^{\frac{1}{\alpha}} \cdot \frac{\left(\frac{1}{(i_{\max}-1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \right)}{\left(1 + (n\beta)^{\frac{1}{\alpha}} \right)} \tag{2}$$

and $i_{\max}$ is the maximal value of the integers in set $\mathcal{I}$. Thus

$$pos(r_i) = pos(s_i) + (d_{\min}, 0).$$

Formeln nicht so wichtig, es geht um die Ideen...

Finally, we place r_{n+1} and r_{n+2} at the center $(0,0)$ and their senders s_{n+1}, s_{n+2} perpendicular to the x-axis, at distance $(2P/\beta\sigma)^{1/\alpha}$, i.e.,

$$\begin{aligned} pos(r_{n+1}) &= pos(r_{n+2}) = (0,0), \\ pos(s_{n+1}) &= \left(0, \left(\frac{2P}{\beta\cdot\sigma}\right)^{1/\alpha}\right), \\ pos(s_{n+2}) &= \left(0, -\left(\frac{2P}{\beta\cdot\sigma}\right)^{1/\alpha}\right). \end{aligned}$$

Having defined the geometric instance of the Scheduling problem for any instance of the Partition problem, we proceed by showing that in order to find a schedule of length $1 < T \leq 2$, a solution to the Partition problem is required. Clearly, it is not possible to schedule all links in one slot, since the receivers r_{n+1} and r_{n+2} are at the same position and we assume $\beta \geq 1$.

In order to transmit successfully, the $SINR$ constraint at the intended receiver has to be satisfied. In the following lemma we prove that the receivers $r_1, \ldots, r_n$ are close enough to their respective senders to guarantee successful transmission, regardless of the number of other links scheduled simultaneously.

LEMMA 4.3. *Let $L_i = \{l_j | 1 \leq j \leq n+1$ and $i \neq j\}$. It holds for all $i \leq n$ that the $SINR$ exceeds β when the link l_i is scheduled concurrently with the set L_i,*

$$SINR(r_i) = \frac{\frac{P}{d_{ii}^{\alpha}}}{\sum_{l_j \in L_i} \frac{P}{d_{ji}^{\alpha}}} > \beta.$$

We are not considering l_{n+2}, since l_{n+1} and l_{n+2} can never be scheduled simultaneously and the distance between s_{n+2} and any other node is the same as the distance between s_{n+1} and this node.

PROOF. Since the positions of the sender nodes $s_1, \ldots, s_n$ depend on the values of $i_1, \ldots, i_n$, we can determine the minimum distance between two sender nodes s_j, s_k.

$$\begin{aligned} d(s_j, s_k) &= |d(s_j, r_{n+1}) - d(s_k, r_{n+1})| \\ &= \left| \left(\frac{P}{i_j}\right)^{\frac{1}{\alpha}} - \left(\frac{P}{i_k}\right)^{\frac{1}{\alpha}} \right| \\ &\geq P^{\frac{1}{\alpha}} \left(\frac{1}{(i_{\max}-1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \right). \qquad (3) \end{aligned}$$

Thus, one can deduce that the sender s_j closest to r_i, $i \neq j$ is located at least at distance $d(s_j, s_i) - d_{\min}$ from r_i. All the other sender nodes (including s_{n+1}) are farther away. This suffices to show a lower bound for $SINR(r_i)$.

$$\begin{aligned} SINR(r_i) &> \frac{\frac{1}{d_{\min}^{\alpha}}}{\frac{n}{(d(s_j,s_i)-d_{\min})^{\alpha}}} \\ &\geq \frac{1}{n}\left(\left(1 + (n\beta)^{1/\alpha}\right) - 1\right)^{\alpha} \\ &= \beta. \qquad (4) \end{aligned}$$

□

Having proved that successful transmission is guaranteed for links $l_1, \ldots l_n$, no matter how many other links are scheduled concurrently, we now return to the proof of Lemma 4.2.

We claim that there exists a solution to the Partition problem if and only if there exists a 2-slot schedule for L. For the first part of the claim, assume we know two subsets $\mathcal{I}_1, \mathcal{I}_2 \subset \mathcal{I}$, whose elements sum up to $\sigma/2$. To construct a 2-slot schedule, $\forall i_j \in \mathcal{I}_1$, we assign the link l_j to the first time slot, along with l_{n+1}, and assign the remaining links to the second time slot. Due to Lemma 4.3 we can focus our analysis on the receivers r_{n+1} and r_{n+2}. The situation is the same for both receivers, so it suffices to examine r_{n+1}. The signal power r_{n+1} receives from its sender node s_{n+1} is

$$P_{r_{n+1}}(s_{n+1}) = \frac{P}{\left(\left(\frac{2P}{\beta\sigma}\right)^{\frac{1}{\alpha}}\right)^{\alpha}} = \frac{\beta\sigma}{2}.$$

The interference r_{n+1} experiences from each sender s_j is

$$I_{r_{n+1}}(s_j) = \frac{P}{\left(\left(\frac{P}{i_j}\right)^{\frac{1}{\alpha}}\right)^{\alpha}} = i_j,$$

which results in total interference of

$$I_{r_{n+1}} = \sum_{i_j \in \mathcal{I}_1} i_j = \frac{\sigma}{2}.$$

This allows to lower bound the $SINR$ at r_{n+1}

$$SINR(r_{n+1}) \geq \frac{P_{r_{n+1}}(s_{n+1})}{I_{r_{n+1}}} = \frac{\beta\sigma/2}{\sigma/2} = \beta,$$

which, in combination with Lemma 4.3, proves that our schedule guarantees successful transmission for all links.

For the second part of the claim, we need to show that if no solution to the Partition problem exists, we cannot find a 2-slot schedule for L. No solution to the Partition problem implies that for every partition of $\mathcal{I}$ into two subsets, the sum of one set is greater than $\sigma/2$. Assume we could still find a schedule with only two slots. Since the receivers r_{n+1} and r_{n+2} are at the same position, they have to be assigned to different slots to permit a successful transmission. Because we have to split $L \backslash \{l_{n+1}, l_{n+2}\}$ into two sets and the received power from $s_j, j = 1, \ldots, n$ at (0,0) is i_j, we end up with a total interference at (0,0) greater than $\sigma/2$ for one slot, which prevents the correct reception of the signal from s_{n+1} or s_{n+2}.

We can now state a theorem on the complexity of the Scheduling problem.

THEOREM 4.4. *The Scheduling problem in $SINR_G$ is NP-complete.*

PROOF. By Lemma 4.1, Scheduling is NP. By Lemma 4.2, Partition is reducible to Scheduling. Therefore, Scheduling is NP-complete. □

4.2 One-Shot Scheduling problem

In this section we prove that the decision version of the weighted One-Shot version of the Scheduling problem, under uniform power assignment scheme, is also NP-complete in the $SINR_G$ model. We proceed by first showing in Lemma 4.5 that the One-Shot Scheduling problem is in the complexity class NP and then give a polynomial time reduction for the Knapsack problem in Lemma 4.6.

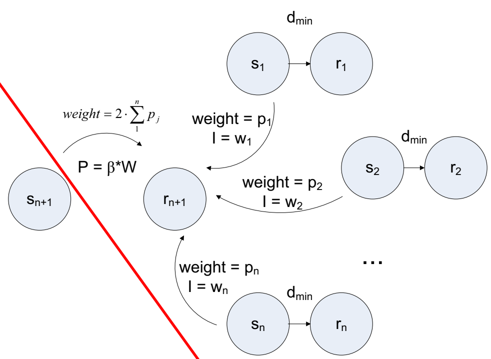

Figure 2: Reduction from Knapsack: the weight of simultaneously scheduled links is maximized if and only if the sum of the values p_j assigned to them is maximized and the knapsack capacity W is not violated.

Lemma 4.5. *The One-Shot Scheduling problem is in NP.*

Proof. Given a set of links, it is possible to verify whether these links satisfy the $SINR$ constrains and whether the sum of their weights exceeds a certain threshold in time polynomial to the size of the input, analogously to the Scheduling problem (Lemma 4.1). $\square$

Lemma 4.6. *Knapsack is reducible to the One-Shot Scheduling problem in polynomial time.*

Proof. Let us first introduce the Knapsack problem: Consider n kinds of items, x_1 through x_n, where each item x_j has a value p_j and a weight w_j. The maximum weight that we can carry in a bag is W. Our aim is to choose the items we put in the bag such that the sum of the values is maximized. We can formulated this task as an integer program.

Knapsack problem:

$$\max \sum_{j=1}^{n} p_j x_j, \qquad \text{s.t.} \tag{5}$$

$$\sum_{j=1}^{n} w_j x_j \quad \leq \quad W, \tag{6}$$

$$x_j \quad \in \quad \{0,1\}, \quad j = 1, \ldots, n$$

Without loss of generality, we assume that there are only items of distinct integer weights. As in the proof for the Scheduling problem, we start by defining a many-to-one reduction from any instance of the Knapsack problem to a geometric instance of the One-Shot Scheduling problem, and afterwards prove that the latter can be solved if and only if the former is also solved.

We have to dispose links in the plane, such that the rules of the Knapsack problem are enforced (cf. Figure 2). We position a sender node s_i in the plane for each x_i, such that the received power from s_i at (0,0) is w_i, i.e.,

$$pos(s_i) = \left(\left(\frac{P}{w_i} \right)^{\frac{1}{\alpha}}, 0 \right), \quad \forall 1 \leq j \leq n.$$

Now we set r_i close enough to s_i to guarantee successful reception regardless of other links.

$$pos(r_i) = pos(s_i) + (d_{min}, 0), \quad \text{where}$$

$$d_{min} = P^{\frac{1}{\alpha}} \cdot \frac{\left(\frac{1}{(w_{\max}-1)^{1/\alpha}} - \frac{1}{w_{\max}^{1/\alpha}} \right)}{\left(1 + (n\beta)^{\frac{1}{\alpha}} \right)},$$

and $w_{\max}$ is the largest weight in this problem instance.

In the next step we place an additional link l_{n+1}, such that r_{n+1} is at (0,0) and s_{n+1} is in such a distance that the received power at (0,0) is βW.

$$\begin{aligned} pos(r_{n+1}) &= (0,0), \\ pos(s_{n+1}) &= \left(0, \left(\frac{P}{\beta W} \right)^{\frac{1}{\alpha}} \right). \end{aligned}$$

Thereafter, we assign a weight to each link:

$$\begin{aligned} weight(l_i) &= p_i, \quad \forall 1 \leq i \leq n \\ weight(l_{n+1}) &= 2 \cdot \sum_{j=1}^{n} p_j. \end{aligned}$$

Note that $SINR(r_i) > \beta, \forall i = 1 \ldots n$, even if all link transmissions are concurrent, since we can apply Lemma 4.3 (due to the fact that we chose the distance between a sender and a receiver of a link to be $d_{\min}$ in both reductions). If we execute an algorithm solving this One-Shot Scheduling problem, we obtain a solution for the Knapsack problem: Let $\mathcal{S}_{OPT}$ be the set of links of an optimal solution to the One-Shot problem constructed above. The described assignment of weights ensures that l_{n+1} is picked, since without it the maximal sum of weights cannot be reached. We can compute $SINR(r_{n+1})$ as follows

$$\begin{aligned} SINR(r_{n+1}) &= \frac{P_{r_{n+1}}(s_{n+1})}{I_{r_{n+1}}} \\ &= \frac{\frac{P}{\left(\left(\frac{P}{\beta W} \right)^{\frac{1}{\alpha}} \right)^{\alpha}}}{\sum_{l_j \in \mathcal{S}_{OPT}} \frac{P}{\left(\left(\frac{P}{w_j} \right)^{\frac{1}{\alpha}} \right)^{\alpha}}} \\ &= \beta \cdot \frac{W}{\sum_{l_j \in \mathcal{S}_{OPT}} w_j}, \end{aligned}$$

and since a valid solution allows l_{n+1} to be transmitted successfully, we have $SINR(r_{n+1}) > \beta$. Consequently a solution to the One-Shot Scheduling problem satisfies

$$\sum_{l_j \in \mathcal{S}_{OPT}} w_j < W.$$

Hence, each of the selected links l_i stands for x_i in (5) and (6), which fulfills the condition of the Knapsack problem. Because $\mathcal{S}_{OPT}$ maximizes the sum of the weights at the same time, the sum of the values of the items of the Knapsack problem is maximized as well. This implies that no algorithm can solve the One-Shot Scheduling problem without solving an NP-complete problem. $\square$

Theorem 4.7. *One-Shot Scheduling in $SINR_G$ is NP-complete.*

Proof. The proof follows from Lemmas 4.5 and 4.6. $\square$

In contrast to these results on the complexity of scheduling with a *uniform power assignment*, the question whether the Scheduling problem *with* power control is also NP-complete remains open and is an area of active research.

5. APPROXIMATION ALGORITHMS

In this section we propose two approximation algorithms for the Scheduling and the One-Shot Scheduling problems.

Before describing the algorithms, let us introduce the notion of *length diversity*, namely the number of magnitudes of distances. Formally, $g(L)$ is defined as

$$g(L) := |\{m|\exists l_i, l_j \in L : \lfloor \log(d_{ii}/d_{jj}) \rfloor = m\}|. \qquad (7)$$

For our problem, $g(L)$ denotes the number of non-empty length classes of the set of links to be scheduled. In realistic scenarios, the diversity $g(L)$ is usually a small constant.

The algorithms we present consist of two steps: First, the problem instance is partitioned into disjoint link length classes; then, a feasible schedule is constructed for each length class using a greedy strategy.

5.1 Scheduling

Algorithm 1 Approximation Algorithm for the Scheduling problem

Require: A set L of links located arbitrarily in the Euclidean plane
Ensure: A schedule $\mathcal{S}$ in which every link can be transmitted successfully
1: Let $R = R_0, \ldots, R_{\log(l_{\max})}$ such that R_k is the set of links l_i of length $2^k \le d_{ii} < 2^{k+1}$;
2: $t = 1$;
3: **for all** $R_k \neq \emptyset$ **do**
4: Partition the plane into squares of width $\mu \cdot 2^k$;
5: 4-color the cells such that no two adjacent cells have the same color.
6: **for** $j = 1$ **to** 4 **do**
7: Select color j;
8: **repeat**
9: For each square A of color j, pick one link $l_i \in R_k$ with receiver r_i in A, assign it to time slot t ($L_j^k = L_j^k \cup l_i$);
10: $t = t + 1$; $\mathcal{S}_t = L_j^k$;
11: **until** all links of R_k in the selected squares are scheduled
12: **end for**
13: **end for**
14: **return** $\mathcal{S}$;

The algorithm (for a description in pseudo-code see Algorithm 1) starts by partitioning the input set of links L into length classes $(R_0, \cdots, R_{g(L)})$. Each subset R_k is scheduled separately. First, the plane is partitioned into square grid cells of side $\mu \cdot 2^k$, where μ is defined as follows

$$\mu = 4\left(8\beta \cdot \frac{(\alpha-1)}{(\alpha-2)}\right)^{\frac{1}{\alpha}}, \qquad (8)$$

and then the cells are colored regularly with 4 colors (cf. Figure 3). Links whose receivers belong to different cells of

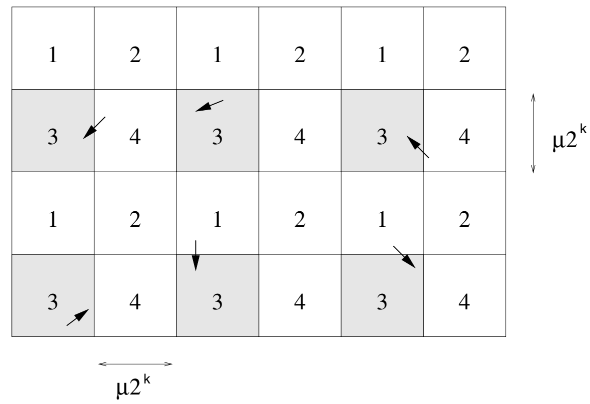

Figure 3: In line 7 of Algorithm 1, the algorithm picks all squares numbered by j. The example shows an inner loop iteration for length class R_k and $j = 3$. The algorithm schedules one unscheduled link from each selected square (if there exists one).

the same color are scheduled simultaneously (added to set L_j^k). Note that the inner *repeat* loop (lines 9-12) constructs a schedule of length $\Delta(A_{max}^k)$, which is the maximum number of links in length class k, whose receivers are in the same grid cell A^k. Given that there are 4 colors and $g(L)$ length classes, all links a scheduled in $4 \cdot \Delta(A_{max}^k) \cdot g(L)$ time slots.

We show now that the schedule obtained by Algorithm 1 is correct, by proving in Theorem 5.1 that all links can be scheduled successfully in their respective time slot.

THEOREM 5.1. *Consider an arbitrary set of links L to be scheduled. For every time slot t, the set $\mathcal{S}_t$ of links output by Algorithm 1 is scheduled successfully, i.e., the $SINR$ at every intended receiver is larger than β.*

PROOF. We demonstrate that all transmissions scheduled in a time slot t are received successfully by the intended receivers, i.e., their $SINR$ is sufficiently high.

Without loss of generality, let us examine links in a length class R_k. Every link $l_i \in R_k$ satisfies $d_{ii} < 2^{k+1}$, thus the perceived power at r_i from s_i is at least

$$P_{r_i}(s_i) \ge \frac{P}{2^{\alpha(k+1)}}. \qquad (9)$$

Since Algorithm 1 schedules at most one link in each cell with the same color concurrently, the closest 8 senders s_j scheduled in the same time slot must be at least at distance $d(r_i, s_j) \ge \mu 2^k - 2^{k+1} = 2^k(\mu - 2)$ to r_i (cf. Figure 3). Consequently, the sum of their interference experienced by r_i is less than

$$\sum_{j=1}^{8} P_{r_i}(s_j) \le \frac{8P}{(2^k(\mu-2))^\alpha}.$$

In the next step, we consider the (at most) 16 senders s_j at distance $3\mu 2^k - 2^{k+1} \le d(r_i, s_j) \le 5\mu 2^k - 2^{k+1}$. They contribute a total interference of

$$\sum_{j=9}^{25} P_{r_i}(s_j) \le \frac{16P}{(2^k(3\mu-2))^\alpha}.$$

We continue aggregating the interference from nodes s_j at distance range

$$(2l-1)\mu 2^k - 2^{k+1} \le d(r_i, s_j) < (2l+1)\mu 2^k - 2^{k+1},$$

$\forall l = 1, 2, \ldots$. Since at most $8l$ links are picked in each interval, the interference caused by them is at most

$$\sum_{\substack{d(r_i,s_j)\ge \\ (2l-1)\mu 2^k - 2^{k+1}}}^{\substack{d(r_i,s_j)< \\ (2l+1)\mu 2^k - 2^{k+1}}} P_{r_i}(s_j) \le \frac{8P \cdot l}{(2^k((2l-1)\mu - 2))^\alpha}.$$

Thus, the total interference at a scheduled receiver r_i can be upper bounded by

$$\begin{aligned} I_{r_i} &\le \sum_{l=1}^{\infty} \frac{8P \cdot l}{(2^k((2l-1)\mu - 2))^\alpha} \\ &\le \frac{8P}{2^{k\alpha}} \sum_{l=1}^{\infty} \frac{l}{(\frac{1}{2}(2l-1)\mu)^\alpha} \qquad (10) \\ &\le \frac{8P}{2^{(k-1)\alpha}\mu^\alpha} \sum_{l=1}^{\infty} \frac{l}{(2l-l)^\alpha} \\ &\le \frac{8P}{2^{(k-1)\alpha}\mu^\alpha} \sum_{l=1}^{\infty} \frac{1}{l^{\alpha-1}} \\ &\le \frac{8P}{2^{(k-1)\alpha}\mu^\alpha} \frac{(\alpha-1)}{(\alpha-2)}, \qquad (11) \end{aligned}$$

where (10) follows because $x - 2 > x/2$, $\forall x > 4$ and $\mu > 4$, given that $\beta \ge 1$ and $\alpha \ge 2$; and (11) follows from a bound on Riemann's zeta function. Using (9), (11), and plugging in the value of μ, defined in (8), the $SINR$ at receiver r_i can be lower bounded by

$$\begin{aligned} SINR(r_i) &= \frac{P_{r_i}(s_i)}{I_{r_i}} \\ &> \frac{\frac{P}{2^{\alpha(k+1)}}}{\frac{8P}{2^{(k-1)\alpha}\mu^\alpha}\frac{(\alpha-1)}{(\alpha-2)}} \\ &= \frac{\mu^\alpha}{4^\alpha \cdot 8 \cdot \frac{(\alpha-1)}{(\alpha-2)}} \\ &= \beta, \end{aligned}$$

□

Now we turn our attention to the efficiency of Algorithm 1. In particular, in Theorem 5.2 we bound its approximation ratio.

THEOREM 5.2. *The approximation ratio of Algorithm 1 is $O(g(L))$, where $g(L)$ is the length diversity of the input, defined in (7).*

PROOF. The proof relies on the choice of a so called *critical* square $A^k_{max} = \mu 2^k \times \mu 2^k$ (cf. Figure 4), i.e., we choose the cell with the highest density $\Delta(A^k_{max})$ over all $g(L)$ generated grids. Note that $\Delta(A^k_{max})$ is the number of links l_i whose receiver is located in cell A^k_{max} and whose length class is k, i.e., $2^k \le d_{ii} < 2^{k+1}$. We proceed by showing that an optimum algorithm OPT can schedule all $\Delta(A^k_{max})$ in at least $T_{OPT} = \lceil \Delta(A^k_{max})/q \rceil$ time slots, where q is a constant dependent on parameters α and β (μ is defined in (8)):

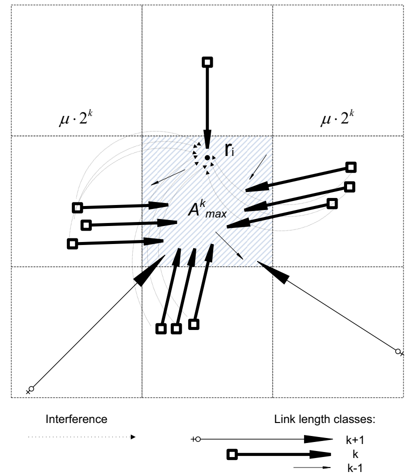

Figure 4: Lower Bound: an optimum algorithm could schedule at most q links with receivers in A^k_{max} in length class k in a single time slot.

$$q = \frac{\left(2(\sqrt{2}\mu + 1)\right)^\alpha}{\beta}. \qquad (12)$$

Assume, by contradiction, that OPT schedules all links in less than T_{OPT} time slots. Therefore, there must exist a time slot $t', 1 \le t' \le T_{OPT}$, such that more than q links in A^k_{max} are scheduled simultaneously. We pick one of the scheduled links $l_i, r_i \in A^k_{max}$ in time slot t' and calculate the resulting $SINR$ level at r_i:

$$\begin{aligned} SINR(r_i \in A^k_{max}) &\le \frac{\frac{P}{d_{ii}^\alpha}}{P \cdot \sum_{j=0}^{q} d(s_j, r_i)^{-\alpha}} \\ &< \frac{\frac{P}{2^{k\alpha}}}{P \cdot q \cdot (2\sqrt{2}\mu 2^k + 2^{k+1})^{-\alpha}} \qquad (13) \\ &= \beta, \qquad (14) \end{aligned}$$

where (13) follows from the fact that $d_{ii} \ge 2^k$, $d_{jj} < 2^{k+1}$ and $d(r_i, r_j) \le 2\sqrt{2}\mu 2^k$; and (14) follows from definition (12) of q.

Hence, to schedule all links in the *critical* square A^k_{max}, OPT needs time

$$T_{OPT} \ge \left\lceil \frac{\Delta(A^k_{max})}{q} \right\rceil. \qquad (15)$$

On the other hand, Algorithm 1 schedules all links in L in time

$$T(Algorithm\ 1) \le 4 \cdot \Delta(A^k_{max}) \cdot g(L). \qquad (16)$$

The approximation ratio follows from (15) and (16):

$$\begin{aligned} \frac{T(Algorithm\ 1)}{T_{OPT}} &\le 4q \cdot g(L) \\ &= O(g(L)). \qquad (17) \end{aligned}$$

□

5.2 One-Shot Scheduling

Algorithm 1 can be adapted to solve the weighted One-Shot Scheduling problem described in Section 3.2 (cf. pseudo code in Algorithm 2). As before, the input set L is partitioned into $g(L)$ length classes, and grids with cell size $\mu \cdot 2^k, k \in \{0 \cdots g(L)\}$ are colored with 4 colors $j \in \{1 \cdots 4\}$. Then, $4 \cdot g(L)$ feasible schedules L_j^k are generated by greedily picking the *heaviest* link in each square A^k of the same color. In the end, the heaviest set of links among all colors and all link classes is chosen.

Algorithm 2 Approximation Algorithm for One-Shot Scheduling

Require: A set L of links located arbitrarily in the Euclidean plane
Ensure: A subset L_j^k in which every link can be transmitted successfully and the total weight $w(L_j^k)$ is maximized

```
Let R = R_0, ..., R_{log(l_max)} such that R_k is the set of links l_i of length 2^k ≤ d_ii < 2^{k+1};
μ = 4(8β(α−1)/(α−2))^{1/α};
for all R_k ≠ ∅ do
    Partition the plane into squares of width μ · 2^k;
    4-color the cells such that no two adjacent cells have the same color.
    for j = 1 to 4 do
        For each square A of color j, pick the heaviest link l_i ∈ R_k with receiver r_i in A, assign it to L_j^k (L_j^k = L_j^k ∪ l_i);
    end for
end for
return argmax_{L_j^k} Σ_{l_i ∈ L_j^k} w(l_i);
```

Since we pick one link per selected square, the feasibility of any schedule L_j^k constructed by Algorithm 2 has been proved in Theorem 5.1. In the next theorem we analyze the approximation ratio of this algorithm.

THEOREM 5.3. *The approximation ratio of Algorithm 2 is $O(g(L))$, where $g(L)$ is the length diversity of the input (defined in (7)).*

PROOF. We start by defining OPT_k to be a subset of the optimum schedule OPT comprised by links that belong to length class k, i.e., $2^k \leq d_{ii} \in OPT_k < 2^{k+1}$. Observe that

$$w(OPT) = \sum_{k=0}^{g(L)} w(OPT_k). \tag{18}$$

In Theorem 5.2 we showed that an optimum algorithm could schedule at most q (defined in (12)) links in each cell A_k at a time. Therefore, given that every feasible schedule L_j^k computed by Algorithm 2 contains the heaviest link in every forth cell, the following bound holds:

$$w(L_j^k) \geq \frac{1}{4q} \cdot w(OPT_k), \quad \forall j \in \{1 \cdots 4\}, k \in \{0 \cdots g(L)\}. \tag{19}$$

Since Algorithm 2 returns the schedule L_j^k of maximum weight over all length classes and colorings (there are at most $4 \cdot g(L)$ schedules L_j^k), the approximation ratio follows:

$$\begin{aligned}
\underset{L_j^k}{\operatorname{argmax}}\, w(L_j^k) &\geq \frac{1}{4 \cdot g(L)} \cdot \sum_{k=0}^{g(L)} w(L_j^k) \\
&\underset{(19)}{\geq} \frac{1}{16q \cdot g(L)} \cdot \sum_{k=0}^{g(L)} w(OPT_k) \\
&\underset{(18)}{=} \frac{w(OPT)}{16q \cdot g(L)} \Rightarrow \\
\frac{w(OPT)}{w(Algorithm\ 2)} &\leq 16q \cdot g(L) \\
&= O(g(L)).
\end{aligned} \tag{20}$$

□

Because of ambient noise, there is usually a maximal distance for a successful transmission in realistic scenarios. Moreover, because of hardware size, a sender and a receiver cannot be arbitrarily close to each other. Hence, one can establish constant minimum and maximum link lengths, which results in a constant number of link length classes $g(L)$. Using this observation, we can state the following corollary.

COROLLARY 5.4. *Assuming a constant maximum and minimum link length, $g(L)$ is constant, and Algorithms 1 and 2 achieve constant approximation ratios.*

6. CONCLUSION

In this work we wanted to gain deeper insights into the complexity of scheduling in wireless ad-hoc networks. To the best of our knowledge, we presented the first NP-completeness proofs for the geometric $SINR$ model. As opposed to other NP-completeness proofs proposed for wireless networks, which rely on a graph structure and an arbitrary gain matrix, our proof explores the geometric nature of such networks – a property, which we consider fundamental. When the distribution of nodes on the Euclidean plane is considered, all the entries in the gain matrix become constrained by the other entries. Therefore, arguing that two nodes cannot transmit concurrently in a schedule becomes much harder. Hence, a different kind of proof is necessary.

Our main contribution is a method of reducing a problem known to be NP-complete by constructing a geometric instance of the scheduling problem. The method consists in disposing nodes in the plane in a way that restricts the number of possible solutions and enforces the constraints of the NP-complete problem. We believe that this method of reduction can be adapted to prove other problems to be hard in the $SINR_G$ model. E.g., an exciting research direction is to analyze the complexity of the joint problem of power control and scheduling.

7. REFERENCES

[1] A. Behzad and I. Rubin. On the Performance of Graph-based Scheduling Algorithms for Packet Radio Networks. In *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2003.

[2] A. Behzad and I. Rubin. Impact of Power Control on the Performance of Ad Hoc Wireless Networks. In *Proc. of the 24^{th} INFOCOM*, 2005.

[3] P. Björklund, P. Värbrand, and D. Yuan. A column generation method for spatial TDMA scheduling in ad hoc networks. *Ad Hoc Networks*, 2(4):405–418, 2004.

[4] A. Borbash, S.A.; Ephremides. Wireless link scheduling with power control and sinr constraints. *Information Theory, IEEE Transactions on*, 52(5):5106–5111, 2006.

[5] G. Brar, D. Blough, and P. Santi. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *Proc. of the 12^{th} International Conference on Mobile Computing and Networking (MOBICOM)*, 2006.

[6] R. Cruz and A. Santhanam. Optimal routing, link scheduling and power control in multi-hop wireless networks, 2003.

[7] A. Ephremides and T. V. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Trans. Communications*, 38(4):456–460, Apr. 1990.

[8] M. Fussen, R. Wattenhofer, and A. Zollinger. Interference Arises at the Receiver. In *International Conference on Wireless Networks, Communications, and Mobile Computing (WIRELESSCOM), Maui, Hawaii, USA*, June 2005.

[9] A. Giridhar and P. R. Kumar. Computing and communicating functions over sensor networks. *IEEE Journal on Selected Areas in Communication*, 23(4), 2005.

[10] J. Grönkvist. *Interference-Based Scheduling in Spatial Reuse TDMA*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2005.

[11] J. Grönkvist and A. Hansson. Comparison Between Graph-Based and Interference-Based STDMA Scheduling. In *Proc. of the 2^{nd} ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC)*, pages 255–258, 2001.

[12] P. Gupta and P. R. Kumar. Critical Power for Asymptotic Connectivity in Wireless Networks. *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W. H. Fleming (March 1998)*, pages 547–566, 1998.

[13] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Trans. Information Theory*, 46(2):388–404, 2000.

[14] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *Information Theory, IEEE Transactions on*, 34(5):910–917.

[15] H. Hunt, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns. NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs. *ALGORITHMS: Journal of Algorithms*, 26, 1998.

[16] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom '03: Proc. of the 9th Annual International Conference on Mobile Computing and Networking*, pages 66–80, New York, NY, USA, 2003. ACM Press.

[17] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. 1972.

[18] I. T. L. Kozat, U.C.; Koutsopoulos. Cross-layer design for power efficiency and qos provisioning in multi-hop wireless networks. *Wireless Communications, IEEE Transactions on*, 5, 2006.

[19] S. O. Krumke, M. Marathe, , and S. Ravi. Models and approximation algorithms for channel assignment in radio networks. *Wireless Networks*, 6:575–584.

[20] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *Proc. of the 15th annual ACM-SIAM symposium on Discrete Algorithms (SODA'04)*, pages 1021–1030, 2004.

[21] K. Leung and L. Wang. Integrated link adaptation and power control for wireless ip networks, 2000.

[22] H. L. Mainak Chatterjee and S. K. Das. Rate allocation and admission control for differentiated services in cdma data networks. *IEEE Transactions on Mobile Computing*, 6(2):179–191, 2007.

[23] T. Moscibroda. The worst-case capacity of wireless sensor networks. In *IPSN*, pages 1–10, 2007.

[24] T. Moscibroda and R. Wattenhofer. Coloring Unstructured Radio Networks. In *Proc. of the 17^{th} Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 39–48, 2005.

[25] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Proc. of the 25^{th} IEEE INFOCOM*, 2006.

[26] T. Moscibroda, R. Wattenhofer, and Y. Weber. Protocol Design Beyond Graph-based Models. In *Proc. of the 5^{th} ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, 2006.

[27] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology Control meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *Proc. of the 7^{th} ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2006.

[28] S. Papavassiliou and L. Tassiulas. Joint optimal channel base station and power assignment for wireless access. *IEEE/ACM Trans. Netw*, 4(6):857–872, 1996.

[29] B. Radunovic and J.-Y. Le Boudec. Optimal Power Control, Scheduling, and Routing in UWB Networks. *Journal on Selected Areas in Communications*, 2(7), 2004.

[30] B. Radunovic and J.-Y. Le Boudec. Rate Performance Objectives of Multi-hop Wireless Networks. In *Proc. 23^{th} IEEE INFOCOM*, 2004.

[31] S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Trans. Netw.*, 1(2):166–177, 1993.

[32] G. Sharma, R. R. Mazumdar, and N. B. Shroff. On the complexity of scheduling in wireless networks. In *MobiCom '06: Proc. of the 12th annual international conference on Mobile computing and networking*, pages 227–238, New York, NY, USA, 2006. ACM Press.

[33] S. Singh and C. S. Raghavendra. PAMAS - Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks. *SIGCOMM Comput. Commun. Rev.*, 28(3):5–26, 1998.