DISS. ETH NO. 18475
TIK-Schriftenreihe-Nr. 107

Computational Complexity and Scheduling Algorithms for Wireless Networks

A dissertation submitted to the

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

Olga Goussevskaia

MSc in Computer Science, UFMG, Brazil

born June 7, 1981

citizen of

Russia and Brazil

accepted on the recommendation of

Prof. Dr. Roger Wattenhofer, examiner
Prof. Dr. Nitin Vaidya, co-examiner
Dr. Stephan Eidenbenz, co-examiner

2009

Abstract

The main problem studied in this thesis is that of scheduling communication requests in a wireless network. Given a set of wireless links, comprised by sender and receiver nodes, distributed in an arbitrary manner in space, we want to know how much time it takes until all the links are scheduled successfully. This problem is a fundamental part of the more general problem of determining the throughput capacity of a given wireless network.

One of the most important issues when studying wireless network problems is the choice of the interference model. Traditionally, the algorithmic community has focused on graph-based models. These models 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 local property. The notion of an interference-edge is a useful abstraction and allows for elegant algorithm design and analysis. It is, however, an oversimplification of the continuous and cumulative nature of the radio signal.

In contrast to the algorithmic community, researchers in information and communication theory have worked with fading channel models, such as the signal-to-interference-plus-noise-ratio (SINR) model, also referred to as the physical interference model. This model represents the physical reality more precisely, since the success of a signal reception depends on all concurrently scheduled transmissions. The analysis of algorithms in this model, however, is more challenging. Because of this increased complexity, the theoretical results in this model have been very limited. Most of the work has either consisted of heuristics and simulation-based evaluations of specific protocols, or has focused on theoretical capacity bounds of special-case networks, such as networks with grid topologies or random node distributions.

In this work we would like to gain a deeper understanding of the fundamental communication limits of wireless networks. This thesis covers several aspects of the problem of scheduling wireless requests. We start with the analysis of the problem's complexity and prove that it is NP-hard in the geometric physical interference model. In this model, it is assumed that nodes live in the Euclidean space, and path-loss is determined by the distances between nodes. Since this problem does not admit optimum solutions in polynomial time, unless P = NP, we concentrate on efficient approximation algorithms. In particular, we propose the first scheduling algorithm that computes a feasible solution in the SINR model in polynomial time with worst-case approximation guarantees for arbitrary network topologies. Besides studying the basic scheduling problem, we also address related problems, such as weighted versions of the scheduling problem, distributed algorithms, and scheduling in combination with analog network coding.

Zusammenfassung

Diese Dissertation beschäftigt sich hauptsächlich mit dem Scheduling von Kommunikationsanfragen in drahtlosen Netzwerken. Es wird dabei angenommen, dass eine Menge von drahtlosen Verbindungen, definiert durch beliebig im Raum verteilte Sender- und Empfängerknoten, gegeben ist. Wir wollen nun wissen, wieviel Zeit benötigt wird, bis alle Empfängerknoten alle an sie addressierten Nachrichten empfangen und erfolgreich dekodieren können. Dieses Problem stellt einen fundamentalen Bestandteil eines generelleren Problems dar, in welchem es darum geht, den maximal erreichbaren Datendurchsatz in einem gegebenen drahtlosen Netzwerk zu bestimmen.

Einer der wichtigsten Aspekte bei der Behandlung von Problemen im Bereich drahtloser Netzwerke ist die Wahl des Interferenzmodells. In Algorithmikerkreisen hat man sich dabei traditionellerweise auf graphbasierte Modelle konzentriert. Solche Modelle definieren typischerweise eine Menge von Interferenzkanten, die durch Paare von Knoten welche sich innerhalb einer gewissen Distanz zueinander befinden, gegeben sind. Als Folge dieser Definition wird Interferenz als eine binäre und lokale Eigenschaft modelliert. Diese Auffassung von Interferenz als Kanten ist eine nützliche Abstraktion welche sowohl den eleganten Entwurf wie auch die elegante Analyse von Algorithmen ermöglicht. Sie stellt allerdings auch eine zu starke Vereinfachung der kontinuierlichen und kumulativen Natur elektromagnetischer Signale dar.

Im Gegensatz dazu arbeiten Informations- und Kommunikationstheoretiker oft mit Fading-Channel-Modellen. Zu diesen Modellen gehört unter anderem das Signal-to-Interference-plus-Noise-Ratio (SINR) Modell, welches auch als physikalisches Interferenzmodell bezeichnet wird. Dieses Modell repräsentiert die physikalische Realität genauer, da der erfolgreiche Empfang eines Signals von allen zeitgleich übermittelten Signalen abhängt. Die Analyse von Algorithmen ist in diesem Modell allerdings anspruchsvoller. Aufgrund dieser erhöhten Komplexität sind die bisherigen theoretischen Resultate in diesem Modell sehr limitiert. Der Grossteil der bestehenden Arbeiten beschäftigt sich entweder mit Heuristiken und der simulationsbasierten Evaluation von spezifischen Protokollen, oder konzentriert sich auf theoretische Kapazitätsschranken für Spezialfälle von Netzwerken, wie zum Beispiel Netzwerke basierend auf Gittern oder zufälligen Knotenverteilungen.

Mit dieser Arbeit möchten wir das Verständnis für die fundamentalen Limiten betreffend Kommunikation in drahtlosen Die Dissertation behandelt verschiedene Aspekte rund ums Scheduling von Nachrichten in drahtlosen Netzwerken. Wir beginnen mit der Analyse der Komplexität und zeigen insbesondere, dass das Problem im geometrischen SINR Modell NP-schwierig ist. In diesem Modell wird angenommen, dass die Knoten in einem Euklidischen Raum leben, und dass die paarweise Interferenz einzig durch die

Distanz zwischen den Knoten bestimmt ist. Da es, angenommen P $\neq$ NP, nicht möglich ist, optimiale Lösungen für dieses Problem in polynomieller Zeit zu finden, konzentrieren wir uns auf effiziente Approximationsalgorithmen. Insbesondere schlagen wir den ersten Scheduling Algorithmus für das SINR Modell vor, welcher in polynomieller Zeit eine korrekte Lösung mit Worst-Case Approximationsgarantieren für beliebige Netzwerktopologien berechnet. Ausser dem grundlegenden Scheduling Problem behandeln wir auch verwandte Probleme, wie zum Beispiel gewichtete Varianten des Scheluding Problems, verteilte Algorithmen, und Scheduling in Kombination mit analoger Netzwerk-Codierung.

Acknowledgements

The last four years as a Ph.D student have been a wonderful experience, and I am deeply grateful to all the people who took part of my life during this period. I would like to start by thanking my co-examiners, Stephan Eidenbenz and Nitin Vaidya, for reading my thesis thoroughly and giving me valuable feedback. I would like to thank my advisor, Roger Wattenhofer, for giving me the opportunity to work in the Distributed Computing Group. You have been a fantastic mentor, and your intelligence and ambition will always be an inspiration to me.

I would like to thank all the DCG-members for forming such a friendly working environment and for all the great time we have spent together. Thank you, Michael Kuhn, for trying to map the world of music with me, and for sharing the Russian-speaking office during all these years. Thank you, Yvonne Anne Pignolet, for not leaving me alone wandering between signal and interference. Thank you, Stefan Schmid and Roland Flury, for trying to catch the sunrise at the Ipanema beach in Rio with me. Thank you, Regina O'Dell, for proving that career and a big family can work so well together. Thank you, Thomas Locher, for introducing me to the history of hip-hop. Thank you, Pascal von Rickenbach, Nicolas Burri, and Remo Meier, for showing that the world of business can also be conquered from within the academia. Thank you, Johannes Schneider, for creating such elaborate defense quiz questions. Thank you, Raphael Eidenbenz, for the discussions about philosophy. Thank you, Christoph Lenzen, for making such tasty pancakes. Thank you, Roland Mathis, for saving my hard disk. Thank you, Philipp Sommer and Benjamin Sigg, for showing how to build real sensor networks. Thank you, Jasmin Smula, for not letting DCG become a boy-only group. And many thanks to all the former DCG members: Keno Albrecht, Thomas Moscibroda, Fabian Kuhn, Aaron Zollinger, Mirijam Wattenhofer, and Yves Weber. Moreover, I am very grateful to Magnús Halldórsson, for his collaboration, and to all the students with whom I have had the pleasure to work with.

Most of all I am grateful to my parents, Nikolai and Elena, for their love and support. You will always be my role models. I would like to thank my sister Anna, for being a great friend. I would like to thank my friends and family members, for being there for me. And, finally, I would like to thank my dear husband Bruno, for always encouraging me and sharing this journey with me.

Contents

Chapter 1

Introduction

Wireless telecommunication networks have been increasingly penetrating people's everyday lives all around the world. The idea of using radio for communication purposes dates as far back as wireless telegraphy, in the early 1800s, when it was discovered that radio waves could be used to send telegraph messages. Ever since radio technology has been steadily evolving into more efficient and scalable communication systems. One of the key advances towards efficient channel utilization was due to packet radio, first implemented by the Aloha network, in the 1970s. Another breakthrough was the idea to use the fact that the power of a transmitted signal decays with distance to enable spatial reuse of the frequency spectrum, giving rise to cellular systems. It was not until the 1990s, however, that digital cellular systems transformed wireless networks into the fastest growing segment of communications industry. There are estimates that there are currently over four billion cell phone subscribers worldwide[1]. In addition to cellular networks, wireless local area networks (WLANs) have been supplementing and replacing wired networks and providing wireless access to the Internet in many residences, offices, airports, universities, and even entire city districts.

The ultimate goal of wireless communication systems is probably to provide universal support for information exchange between people and between devices and people. However, the characteristics of the wireless channel make it tricky to achieve such an ambitious objective. On the one hand, the radio spectrum is a scarce resource, which has to be allocated to various systems and applications. Therefore, efficiently exploring its full capacity is essential. On the other hand, the propagation of the radio signal through the wireless medium is subject to several fluctuations caused by obstacles and movement, which make the wireless channel highly unstable and unreliable. Moreover, wireless transmissions interfere intensely with each other unless

[1] Source: United Nations International Telecommunications Union (ITU)

they are separated in space, frequency, or time.

There are numerous technical challenges that have to be overcome before all the envisioned applications become available with acceptable quality of service. Bandwidth-intensive and delay-sensitive applications, such as video teleconferencing, large file transfers, or global multimedia access, are still too constrained by the limitations of current wireless systems. Highly distributed applications, such as distributed control systems with remote devices like sensors are even farther away from becoming a reality. Besides the difficulties originated by the unreliable wireless medium, these applications also demand for flexibility and robustness that only ad hoc networks without central infrastructure can offer. In order for ad hoc networks, such as wireless sensor networks, to leave the research laboratories and become part of everyday applications, we need better distributed protocols, smarter energy-constrained devices, and a better understanding of the algorithmic complexity of wireless networks.

This thesis studies the fundamental problem of determining the communication limits of a given wireless network. We will try to determine to which extent it is possible to answer questions like "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, arbitrarily 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 challenge is to achieve efficient spatial reuse, considering wireless interference among concurrently transmitting nodes. We hope that looking into these questions will help understanding wireless networks, giving insights into interference, spatial reuse, and media access control protocols.

The answers to the questions stated above depend, among other factors, 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 [45]. 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 [84]. 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 thesis 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.

Actually, in this thesis we focus on one specific part of the problem of determining the throughput capacity of a wireless network. We study the problem of scheduling *one-hop* communication requests *without power control*. This problem plays a fundamental role in determining the capacity of the network, however it does not consider neither the routing nor the power control problems, which are also part of the more general problem of determining the throughput capacity of a given wireless network.

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, usually define a set of interference-edges, connecting pairs of nodes, depending on whether they fulfill some criterion, such as, for example, the distance to each other. Interference is therefore a binary and, often, 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.

The research community has tried to approach wireless network problems from different perspectives. Some people have focussed on building protocols for real networks and validated their heuristic solutions by running experiments on small real network instances or through simulations of larger networks. Unfortunately, these results can hardly be extended to arbitrary network topologies, given that they are usually, from the start, designed for a restricted pool of network parameters. Other people have looked at wireless network problems from the information-theoretic perspective, and have derived capacity bounds for networks that obey some random node distribution. These solutions again do not provide enough insight into the understanding of arbitrary wireless networks, since hardly any real network instance fits a particular random distribution. Yet another line of research has modeled wireless systems by means of graphs, and used graph-theoretic techniques to derive elegant algorithms with worst-case analysis and guarantees. Unfortunately, once again, these results are not good enough for us, because graph-based interference models are an extreme oversimplification of the physical properties of the wireless signal.

In this thesis we intend to gain deeper understanding of the communication limits of an arbitrary wireless network. We set ourselves the goal to bridge the gap between heuristics and special-case analysis and theoretical analysis of arbitrary instances in a realistic network model.

In the following section we describe the basic structure of this thesis.

1.1 Thesis Overview

This thesis is organized in nine main chapters. The distribution of the content among these chapters is described in detail in the remainder of this Introduction Chapter.

In Chapter 2 we discuss the most relevant related work to this thesis. We cover a wide range of topics, among them: complexity and hardness results; graph-based scheduling algorithms; scheduling and power control in the physical interference model; capacity of randomly deployed networks; distributed scheduling algorithms; and network coding. We conclude this chapter with a discussion of some of the most recent developments and still on-going work in the area.

In Chapter 3 we introduce some preliminary concepts used throughout this thesis, such as definitions from the computational complexity theory and metric spaces. In this same chapter we describe the physical interference model, or SINR model, which is the main model used in this thesis to represent a wireless network. We conclude this chapter with definitions of three scheduling problems that will play a major role in the posterior chapters.

In Chapter 4 we analyze the complexity of the link scheduling problem in the SINR model. We make a distinction between "geometric SINR" ($SINR_G$) and "abstract SINR" ($SINR_A$), and begin the chapter with a discussion about these models. The geometric version of the model is more restricted than the abstract version, since it explicitly uses the fact that nodes are distributed in the Euclidean plane. We present the first NP-hardness proofs in such a model. In particular, we prove two problems to be NP-hard: Multi-Slot Scheduling and Weighted One-Shot Scheduling (see Sections 3.5 and 3.6 for precise definitions). 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. Before presenting the main results, we illustrate the difference in proof techniques between the $SINR_A$ and $SINR_G$ models by showing that it is NP-hard to approximate the Multi-Slot Scheduling Problem to within a factor of $n^{1-\varepsilon}$, for any constant $\varepsilon > 0$, in the non-geometric $SINR_A$ model.

In Chapter 5 we propose the first algorithms for the scheduling problem in the SINR model, which have an approximation guarantee in arbitrary topologies. The aproximation ratio of the algorithms proposed in this chapter is $O(g(L))$, where $g(L)$ is the so called *diversity* of the network. The diversity depends on the topology of the network and captures the variation in the lengths of the links to be scheduled. The main drawback of this algorithm is that the diversity of a network can be as large as n, the number of links in the network.

In Chapter 6 we improve this result by proposing another algorithm, which to the extent of our knowledge, is 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 possibly non-isotropic signal distortions.

In Chapter 7 we analyze the complexity of local broadcasting in the physical interference model. We present two distributed randomized algorithms: one that assumes that each node knows how many nodes there are in its geographical proximity, and another, which makes no assumptions about the topology of the network. We show that, if the transmission probability of each node meets certain characteristics, the analysis can be decoupled from the global nature of the physical interference model, and each node performs a successful local broadcast in time proportional to the number of neighbors in its physical proximity. We also provide worst-case optimality guarantees for both algorithms and demonstrate their behavior in average scenarios through simulations.

In Chapter 8 we analyze the complexity of scheduling wireless links in the physical interference model with analog network coding capability. We study two models with different definitions of network coding. In one model, we assume that a receiver is able to decode several signals simultaneously, provided that these signals differ in strength significantly. In the second model, we assume that in a two-way relay channel, routers are able to forward the interfering signal of a pair of nodes that wish to exchange a message, and nodes are able to decode the "collided" message by subtracting their own contribution from the interfered signal. For each model, we construct an instance of the scheduling problem in the geometric SINR model, in which nodes are distributed in the Euclidean plane. We present NP-hardness proofs for both scenarios and propose a scheduling algorithm that explores analog network coding opportunities to achieve superior throughput capacity.

Finally, in Chapter 9, we present some conclusions of this thesis and discuss possible future directions.

Chapter 2

Related Work

In this chapter we summarize some of the related work in the area and position the results of this thesis in the respective contexts. In Section 2.1 we describe existing results on the complexity of scheduling problems. In Section 2.2 we discuss what has been done in terms of scheduling algorithms in graph-based models. In Section 2.3 we present an overview of the results involving scheduling and power control in SINR-based models. In Section 2.4 we discuss information-theoretic bounds on the capacity of random networks. In Section 2.5 we cover some results in the area of distributed scheduling algorithms, and in Section 2.6 we discuss related work in the area of network coding. We conclude this chapter with Section 2.7, where we cite some of the most recent developments and still on-going work in the area.

2.1 Scheduling and Complexity

There have been various NP-hardness proofs for the problem of scheduling in wireless networks. Most of these proofs have been designed either for *graph-based* models or for the non-geometric SINR model (see Section 4.1 for a discussion about the non-geometric, or "abstract" SINR model $SINR_A$).

In graph-based models, two different types of interference have been studied in the literature, namely, *primary interference* and *secondary interference*. Primary interference occurs when a node transmits and receives packets at the same time. Secondary interference occurs when a node receives two or more separate transmissions. In [2], Arikan proved that the point-to-point link scheduling problem is NP-complete if both primary and secondary interference are considered in a general graph. In [46], Hajek and Sasaki showed that when secondary conflicts are tolerated, though, the same problem can be solved in polynomial in time. In [26], Ephremides and Truong proved that the problem of scheduling broadcasts without tolerance of secondary conflicts is NP-complete.

In [20], Clark et al. proved that a series of closely related problems to scheduling of wireless links, such as coloring in graphs, independent set, domination, independent domination, and connected domination, are also NP-complete in unit disk graphs (UDG) (please refer to Section 2.2 for a definition of a UDG, also see Figure 2.1). Interestingly enough, finding cliques when a geometric representation (circles in the plane) of a UDG is provided, was shown to be polynomial in time [20].

Another group of graph-based models are the so called K-hop interference models. In these models, no two links within K hops can successfully transmit at the same time. For a given K, in order to find a minimum-length schedule, one has to solve a maximum (possibly weighted) matching problem subject to the K-hop interference constraints.[1] For $K = 1$ the problem can be solved in polynomial time; for $K > 1$, the resulting problems are NP-hard and cannot be approximated within a factor that grows polynomially with the number of nodes. This was proved by Sharma et al. in [102, 103], using a reduction from the 3-CNF-SAT problem. For the case of geometric graphs, such as UDGs, the authors show that the resulting problems admit polynomial time approximation schemes. Note that determining an adequate value for K for a specific network is a challenging task and strongly depends on the characteristics of the physical layer.

Another family of hardness results for the scheduling problem is based on the abstract (non-geometric) SINR model ($SINR_A$) and consist of reductions without a geometric representation. A typical such proof establishes an arbitrary gain matrix between the participating nodes, which results in a standard graph, since the gain between any two nodes can be set to either 1 ("link") or 0 ("no link"). Afterwards, the hardness is proved by a reduction from graph coloring, for example as was done by Björklund et al. in [12]. In Section 4.2 we illustrate this kind of hardness proof by showing that it is NP-hard to approximate the Multi-Slot Scheduling Problem (defined in Section 3.6) to within a factor of $n^{1-\varepsilon}$, for any constant $\varepsilon > 0$, in the $SINR_A$ model.

Several related problems have been shown to be NP-hard in the $SINR_A$ model. 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 was addressed by Kozat et al. in [67]. The authors proved that this problem is NP-hard 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 [77] proved that the problem of maximizing data throughput by adaptive modulation and power control while meeting

[1] Let $d_G(i, j)$ denote the graph (or hop) distance between two vertices in a graph G. For two edges $e_i = i_1 i_2, e_j = j_1 j_2 \in E(G)$, let $d(e_i, e_j) = \min_{u,v \in \{1,2\}} d_G(i_u, j_v)$. A set of edges M is called a K-valid matching if for all $e_1, e_2 \in M$ with $e_1 \neq e_2$, we have $d(e_1, e_2) \geq K$.

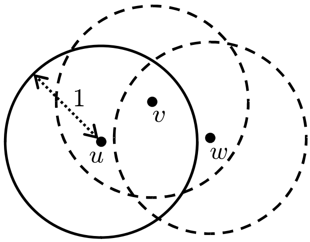


Figure 2.1: Unit disk graph: node u is adjacent to node v ($d_{vu} \leq 1$), but not to node w ($d_{vw} > 1$).

packet error requirements is NP-hard (again under the assumption that the values of the gain matrix are arbitrary). Another related problem, proposed by Chatterjee et al. in [82], is the so called "power constrained discrete rate allocation problem". The authors prove that this problem is NP-hard for CDMA data networks by a reduction from the Knapsack problem, using a gain matrix with gain value 1 for all links.

The complexity of wireless link scheduling with power control in the geometric $SINR_G$ model was examined by Borbash and Ephremides in [14]. A problem called "MAX-SIR-MATCHING" was introduced, and it was shown that if this problem is NP-hard, then computing the minimum length schedule is also NP-hard. However, no proof is given about the NP-hardness of "MAX-SIR-MATCHING", leaving the complexity issue not fully addressed.

To the best of our knowledge, the first NP-hardness proofs for the link scheduling problem in the $SINR_G$ model are the ones presented in Chapter 4 of this thesis and published in [39]. Some follow-up hardness proofs have been later based on the result in [39]. In [30], for instance, Fu et al. prove that a variation of the scheduling problem, called Scheduling Consecutive Transmission Constraints, is NP-hard in the $SINR_G$ model.

2.2 Scheduling Algorithms and Graph-Based Models

As already mentioned in Chapter 1, a popular way to model wireless networks has been by means of *graphs.* A graph model usually consists of a connectivity graph and possibly also of an interference graph. In both graphs, the set of vertices represents the devices, and a successful transmission occurs when the sender-receiver pair is connected in the connectivity graph and no other concurrently scheduled sender-receiver pair inflicts a conflict in the interference graph. As a consequence, graph-based scheduling algorithms usually employ some sort of matching or coloring strategy.

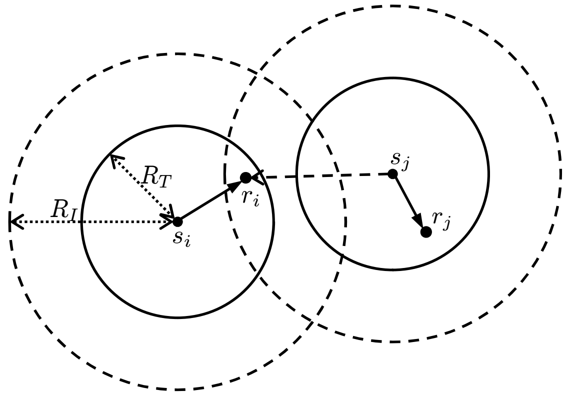


Figure 2.2: Protocol interference model: there are two radii: transmission range R_T and interference range R_I. In this example, node r_i is not able to receive a transmission from node s_i if node s_j concurrently transmits to node r_j–even though r_i is not adjacent to s_j.

Coloring a general graph is not only an NP-complete problem, but is also hard to approximate to within factor of $n^{1-\epsilon}$, for any constant $\epsilon > 0$ [121]. Wireless networks, however, can usually be better modeled by more restricted classes of graphs, such as *geometric graphs*. Geometric graphs are graphs whose vertices are placed in a metric space (usually in a 2-dimensional Euclidean plane), and two vertices are connected if and only if the distance between them is less than or equal to r, for some $r > 0$. When $r = 1$, the geometric graph is commonly called a Unit Disk Graph (UDG) (see Figure 2.1). When r is different for each node and two vertices u and v are connected if and only if the distance between them is less than or equal to $\min(r(u), r(v))$, then the graph is called a *disk graph*. Intuitively, disk graphs are intersection graphs of (possibly equal sized) circles in the plane and have been extensively used to model broadcast networks.

One commonly used graph-based interference model is the *Protocol model* [45]. In this model, a transmission by a node s_i is successfully received by a node r_i iff the intended receiver r_i is sufficiently apart from the sender s_j of any other simultaneous transmission, i.e., $d(s_j, r_i) \geq (1+\rho)d(s_i, r_i), \forall s_j \neq s_i$. The constant $\rho > 0$ models situations, where a guarding region is specified by the protocol to prevent a neighboring node from transmitting (on the same channel) at the same time. This model implicitly assumes that senders use power control to adjust their signals. There are, therefore, two radii: a transmission range R_T and an interference range R_I. A node can successfully transmit to a receiver node in its transmission range only if the receiver is not within the interference range of any other concurrently transmitting node (see Figure 2.2).

The problem of scheduling in graph models has been studied extensively and presents a vast and rich body of literature, only a small fraction of which is going to be covered in this section.

In [46], Hajek and Sasaki propose a polynomial time algorithm for the problem of scheduling links in a general graph, when secondary interference is tolerated. In [26], Ephremides and Truong analyze the complexity and provide heuristic solutions (that can be implemented distributively) for the problem of scheduling broadcasts in general graphs, considering primary and secondary interference. In [94], Ramanathan and Lloyd study both the link scheduling and the broadcast scheduling problems in trees and in general graphs. They prove that tree networks can be scheduled optimally in polynomial time, and that arbitrary networks can be scheduled so that the length of the schedule is bounded by a length that is proportional to a function of the network's *thickness* times the optimum, where *thickness* is defined to be the minimum number of *planar graphs* into which a given graph can be partitioned.

In [54], Hunt et al. present approximation schemes for several graph problems, such as maximum independent set, minimum vertex cover and minimum dominating set, when restricted to geometric graphs, such as UDGs and (r, s)-civilized graphs [2]. The approximation schemes in [54] exhibit the same time versus performance trade-off as approximation schemes for planar graphs. In [70], Krumke et al. build upon the results in [54] and present a series of approximation algorithms for the distance-2 coloring problem, also in geometric graphs, such as (r, s)-civilized graphs.

In [74], Kumar et al. study the joint problem of scheduling and routing. Using LP formulations, the authors propose a "cross-layer" framework to maximize the throughput of the network to within a constant approximation factor in different graph-based interference models, such as the protocol model [45].

In [73], Kumar et al. study decentralized algorithms for the joint problem of routing and scheduling in the distance-2 interference model for various families of disk graphs.

In [111], Wang et al. propose centralized and distributed approximation algorithms (with a constant factor approximation ratio) for the scheduling problem in two graph-based interference models: "fixed power protocol model" and "RTS/CTS model". The "fixed power protocol model" assumes that each node s_i has its own fixed transmission power, or interference range, and any other (receiver) node r_j within this range will be interfered with the signal of s_i. The "RTS/CTS model" reflects the communication under

[2] A graph is called (r, s)-*civilized* if its vertices can be mapped to points on the plane such that the length of each edge is $\leq r$ and distance between any two points is $\geq s$, for some pair of reals $r > 0$ and $s > 0$.

the RTS/CTS based scheme of IEEE 802.11 DCF (Distributed Coordination Function). The exchange of RTS and CTS messages between the sender and the receiver ensures that nodes within one hop of the sender or the receiver cannot participate in a communication, which is equivalent to saying that the chosen set of node pairs must constitute a 2-valid matching. In [102], Sharma et al. analyze formulations of the scheduling problem as weighted matching problem under K-hop interference models. Using the results from [54], they develop a PTAS for the problem for geometric graphs, such as disk graphs and (r, s)-civilized graphs.

In [56], Joo et al. analyze the so called *Greedy Maximal Scheduling* (GMS) scheme to solve the link scheduling problem subject to the K-hop interference constraints in trees and geometric graphs. GMS has low complexity and may be implemented in a distributed manner. Joo et al. show that the full capacity region can be achieved by this scheme in trees, and an efficiency ration between 1/6 and 1/3 can be obtained in geometric graphs.

Although the algorithms discussed in this section present extensive theoretical analysis, they are constrained to the limitations of a model that oversimplifies the nature of wireless communication. In the next section we are going to discuss scheduling algorithm designed for a series of more realistic models, namely the fading channel models.

2.3 Scheduling Algorithms and the SINR Model

In the *physical interference model*, or the SINR model, 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 (see definition in Section 3.3). The inefficiency of graph-based scheduling protocols in the *physical interference* (or SINR) model has been shown both through simulations [7, 42, 43], and through experiments (on mica2 sensor nodes running with TinyOS) [87]. In fact, in [87], Moscibroda et al. show that any protocol which obeys the laws of graph-based models can be broken by a protocol explicitly defined for the physical model. There have been some efforts to model the properties of the physical model using SINR-derived conflict graphs, e.g. [15, 55, 106], however, the obtained bounds are usually too loose, or are only valid in restricted network topologies.

Given that algorithms designed for graph-based models may render infeasible schedules or schedules that poorly explore the network's capacity, the question of how to design protocols specifically designed for the physical communication model has been studied, leading to a large and rich body of literature.

The joint problem of scheduling and power control has been extensively studied in the SINR model. In [41], Grandhi et al. analyze the problem of determining whether a given matching (or set of sender-receiver pairs) is feasible under SINR constraints. In particular, they show that, given a set of k sender-receiver pairs, the maximum SIR that can be achieved simultaneously at all the receivers is the reciprocal of the spectral radius of a $k \times k$ nonnegative irreducible matrix A, in which $A_{ij} = G_{ij}/G_{ii}$ if $i \neq j$ and $A_{ij} = 0$ if $i = j$ and G_{ij} is the gain (or path-loss) between sender s_j and receiver r_i. The obtained SIR is achieved by using the Perron eigenvector as a power vector. This result was extended by incorporating the case of nonzero ambient noise $N > 0$ in [13] by Borbash and Ephremides. However, the problem is that the maximum SINR obtained in this way may be too low to guarantee correct reception at all receivers. A brute force approach for finding the optimal schedule would involve checking, for all 2^n subsets of links, whether the obtained power vector provides a sufficiently high SINR, which incurs an exponentially growing time complexity.

Power assignment strategies typically fall into three groups: uniform power ssignment, "energy metric" or liner power assignment, and link removal algorithms. Uniform power assignment has been widely adopted in practical systems, and works by assigning the same power level to all nodes in the network [44, 50, 104]. Linear power assignment is another frequently used strategy, and works by assigning a power level proportional to a so-called "energy metric", i.e., proportional to the minimum power required to reach the receiver node [5, 83, 113]. In [86], Moscibroda and Wattenhofer show that both of these strategies can lead to very long schedules, when compared to schedules that employ a sophisticated power control mechanism.

Another group of power assignment algorithms is based on the results in [41, 118], where it was shown that finding a power assignment yelding the maximum achievable signal to interference ratio (the result was later extended to include noise e.g. in [13]) can essentialy be reduced to solving an Eigenvalue problem for the link gain matrix, which takes time $O(n^3)$. If the maximum achievable ratio is below the minimum threshold β, a set is "unfeasible". A number of heuristics that remove links from such unfeasible set according to different criteria, such as the amount of interference a link causes or experiences, have been proposed [76, 110, 118, 119]. Other heuristics for the joint problem of scheduling and power control can be found in [21, 24].

In [85], Moscibroda et al. present examples of network topologies for which all the above mentioned heuristics perform very poorly, i.e., generate schedules of length $\Omega(n)$, where n is the number of links in the network. Significant progress in terms of providing worst-case guarantees for a polynomial time power assignment algorithm was achieved in [86], where Moscibroda et al. propose an algorithm that successfully schedules a strongly connected set of links in polylogarithmic number of time slots, even in arbitrary worst-case networks. This result has been extended and applied to different scenarios, such as topology control [31, 88], sensor networks [84], scheduling combined with routing [16], or ultra-wideband [53]. Until the time of writing of this thesis, however, no algorithm with a non trivial approximation guarantee has been presented for the problem of scheduling an arbitrary set of links using power control in the SINR model. As an exception we can maybe cite [30], where Fu et al. derive an algorithm with the same approximation guarantee as the scheduling algorithm without power control, presented in [39]. Some very recent results, discussed in Section 2.7, promise a breakthrough in this respect.

Scheduling algorithms without power control with approximation guarantees in the SINR model have not been very numerous. In [15], Brar et al. present 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. This result, however, holds only under the assumption that nodes are distributed uniformly at random in a square of unit area. In [39] (see Section 5.1 and Section 5.2), we proposed scheduling algorithms with a factor $O(g(L))$ approximation guarantee in arbitrary network topologies, where $g(L)$ is the so called *diversity of the network*. 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 [17], Chafekar et al. proposed an algorithm with approximation guarantee of $O(\log \Delta)$, where Δ is the ratio between the maximum and the minimum distances between nodes. This parameter can be arbitrarily large (note that $g(L) \leq \log \Delta$). Following the result in [39], the undesired dependency on the diversity $g(L)$ of the network has been inherited by a number of scheduling algorithms in the SINR model, e.g. [22, 30].

In contrast to the above mentioned results, the scheduling algorithm that we proposed in [37] (see Section 5.1) has *constant approximation* guarantee for the One-Slot Scheduling Problem (defined in Section 3.4) and an approximation ratio of $O(\log n)$ for the Multi-Slot Scheduling Problem (defined in Section 3.6). These bounds are valid in the physical interference model and arbitrary node distributions. To the extent of our knowledge, these are the first scheduling algorithms with approximation guarantee independent of the topology of the network. We complement our results by looking at the algorithms' performance in metric spaces beyond the two-dimensional Euclidean plane (see Section 6.3). We prove that the analysis is extendable to higher-dimension Euclidean spaces, provided that the path-loss exponent α is strictly higher than the number of dimensions.

2.4 Capacity of Randomly Deployed Networks

Throughput capacity of randomly deployed wireless networks has been intensely studied from the information theory perspective. In their seminal work [45], Gupta and Kumar provide upper and lower bounds on the capacity of networks with two kinds of topology: one where nodes are distributed uniformly at random in a disk of unit area, and one where nodes are "optimally" distributed on a regular grid lattice. In the former case, the authors show that if each node is capable of transmitting W bits per second, the per node capacity of the network with n nodes is $\Theta(W/\sqrt{n \log n})$. In the "optimum" topology and traffic pattern, the capacity is $\Theta(W/\sqrt{n})$. These results hold in both the protocol and the physical interference models and hold a rather pessimistic character, since they essentially state that large networks cannot achieve high throughput.

Using similar techniques, in [68], Kozat and Tassiulas show that this capacity bound can be improved up to $\Theta(W/\log n)$, if the ad hoc network is overlaid with an infrastructure network. To achieve this bound, the number of (also distributed uniformly at random) access points to the infrastructure has to grow linearly with n, and the ad hoc network (excluding the infrastructure nodes) has to form a connected topology. The capacity of multichannel wireless networks with channel switching constraints was studied by Bhandari and Vaidya in [8, 9]. In particular, in [8] it was shown that, in a model where each node is allowed to switch between a pre-assigned random subset of f channels out of c (each having bandwidth W/c), the per-flow capacity is $\Theta(W\sqrt{p_{rnd}/n \log n})$, where $p_{rbd} \geq 1 - e^{-f^2/c}$. The capacity of random networks has been further analyzed in many different contexts, such as MIMO [19], multi-user cooperation [90], use of relays [35], multicast [78], data gathering [36], and cognitive networks [109]. Although these results are important, they do not provide algorithmic tools to determine the capacity of concrete wireless networks. In practice, network topologies hardly ever follow any particular random distribution.

2.5 Distributed Scheduling Algorithms

One of the commonly used communication models to analyze distributed algorithms for wireless networks is the *synchronous message passing model*, which models the network as an undirected graph, in which nodes represent the devices, and edges represent point-to-point communication channels. Time is divided into time-slots, all nodes start the execution of a protocol simultaneously, typically in time slot $t = 0$, and in each time slot each node can reliably exchange one message with each of its neighbors. To simulate the broadcast nature of wireless communication, it is common to assume that nodes cannot send different messages to different neighbors. As already pointed out, scheduling protocols are usually based on graph coloring algorithms. There have been proposed a rich variety of distributed scheduling algorithms for the message passing model in combination with graph-based interference models [26, 56, 73, 98, 111]. Although this model allows algorithms to be rigorously analyzed, it fails in capturing many essential characteristics of wireless communication, such as unreliable communication, collisions, or lack of synchronization.

The above cited algorithms usually establish a TDMA (Time Division Multiple Access) schedule, assigning to each node a sequence of time-slots in which they are allowed to transmit. When it comes to designing a MAC (Medium Access Control) protocol, a "fixed" TDMA schedule might provide low channel utilization when contention is low, since time slots are reserved for particular nodes even when they do not have any data to transmit. Other disadvantages of TDMA-based MAC protocols are their dependency on tight time synchronization constraints and networks information, such as the 2-hop neighborhood of each node. To overcome some of these issues, many MAC protocols combine TDMA scheduling with CSMA (Carrier Sense Multiple Access). In a CSMA scheme, a node checks the channel for concurrent traffic before transmitting, and if the channel is busy, it waits for a random period of time before transmitting again. A multitude of MAC protocols implementing some combination of TDMA and CSMA have been proposed [25, 52, 57, 58, 91, 97, 105, 107, 117]. The evaluation of the performance of these protocols is typically done through simulations or experiments in specific network topologies, providing no analytical guarantees for general cases.

A model that better captures the nature of wireless communication, but still permitting rigorous analysis, is the *radio network model*, extensively studied in the context of the broadcast problem [6, 10, 11, 18, 66]. In this model, time is also divided into time slots, all nodes start the execution of the algorithm (or wake up) simultaneously (i.e., there is access to a global clock), and in each time slot each node can transmit one message to all nodes within its transmission range. A message is received correctly by a node in a time slot only if exactly one of its neighbors transmitted in that time slot. Otherwise a collision occurs. This model has been further extended by dropping the assumption of a global clock and studied in the context of the wake-up problem [28, 34, 59]. In the latter model, each node wakes up asynchronously, upon successful reception of a message. The network is assumed to be single-hop, so as soon as one node transmits successfully, all nodes are woken up. In [59], Jurdzinski and Stachowiak showed that when nodes know the total number n of nodes in the (one-hop) network, the wake up can be done in time $O(\log n \log(1/\epsilon))$ with probability at least $1-\epsilon$. If n is unknown, the authors proved a lower bound of $\Omega(n/\log n)$.

The closest communication model to the one that we use in Chapter 7, is the so called *unstructured radio network model*, proposed by Kuhn et al. in [71, 72]. This model introduces an even harsher definition of asynchrony than the one used to study the wake-up problem. It is assumed that nodes can wake up and join the execution of the protocol asynchronously at any time, i.e., nodes do not have to be woken up by a clear message reception. Upon waking up, nodes do not have any information about their neighborhood, about which nodes are awake, or for how long they have been awake. As opposed to our work, the analysis in [71, 72] still assumes an underlying graph-based interference model.

To the extent of our knowledge, there have not been a lot of work on distributed algorithms in the physical interference model. In [99], Richa et al. present a distributed algorithm for establishing a dominating set in the SINR model. In [75], Lebhar and Lotker propose an algorithm to emulate a UDG-like structure in the SINR model and a network where nodes are distributed uniformly at random on the plane. In [38] (see Chapter 7), we propose efficient distributed algorithms for the problem of local broadcasting in the SINR model. We propose two distributed randomized algorithms. One is a very simple Aloha-like algorithm that is based on the assumption that each node knows the number of its neighbors; the other is more involved and makes no assumptions about topology knowledge. Finally, aloha-based MAC schemes have also been analyzed in the SINR model [4, 29, 112]. In contrast to our work, the analysis presented in these papers is primarily based on the assumptions of homogenous and uniformly random node distributions that do not provide any strong worst-case bounds.

2.6 Scheduling and Network Coding

Network coding is a technique that extends the traditional definition of routing by allowing routers to not just forward copies of received messages, but to mix the bits from different packets before forwarding them. The topic has received a lot of attention in the research community, starting with the pioneering work of Ahlswede et al. [1], where the authors prove that full capacity (i.e. the maximum flow or minimum cut between a source and a receiver) can be achieved in a graph where one source multicasts information to other nodes in a multihop fashion and any node in the network is allowed to encode all its received data before passing it on. Network coding within one such *multicast flow* has been extensively studied ever since. In [79], Li et al. showed that every (solvable) multicast network has a scalar linear solution over a sufficiently large finite field alphabet, i.e., the multicast capacity can be achieved using linear codes. In [101], Sanders et al. provided polynomial time algorithms for constructing coding and decoding schemes for multicasting at the maximal data rate. In [51], Ho et al. showed that multicast capacity can be achieved in a distributed manner, by using random linear codes over a sufficiently large finite field. In [115], Wu at al. showed that the minimum energy-per-bit multicast in a graph-based wireless network model can be solved as a linear program.

Network coding in more general networks, with an arbitrary collection of point-to-point connections and not just one multicast flow, was shown to be a much more difficult problem. In [23], Dougherty et al. prove that linear codes are not sufficient to achieve full capacity in this case, and in [65], Koetter and Medard show that finding linear codes for networks with more than one flow is NP-hard. In [63], Katti et al. integrated network coding into a link layer enhancement scheme for (arbitrary) multi-hop wireless networks. A mixing engine was introduced into the nodes, operating between the MAC and the network layer, in order to identify opportunities to make bitwise XORs of different packets and sending them in a single transmission. In [80], Liu et al. analyzed the theoretical throughput gains of such a scheme in a random network topology and showed that network coding provides no order difference improvement on the capacity bounds obtained by Gupta and Kumar in [45] for networks that do not employ network coding. In [81], Liu et al. analyze the constant factor throughput improvement obtained by network coding in the same network model studied by Gupta and Kumar in [45].

Network coding in the physical layer, or *analog network coding*, is similar in spirit to digital network coding. However, it operates on the raw analog signal, instead of first decoding and then mixing packets in a bitwise manner. Some techniques, such as *cochannel signal separation*, explore differences in the characteristics of interfered signals, such as the signal's strength, to decode several signals simultaneously [48, 49]. Other analog coding techniques exploit the fact that, in a wireless network, often a receiver has prior knowledge about some packets destined to other nodes, by having overheard or forwarded them earlier. This situation has been extensively studied in the context of 2-way relay channel [62, 95, 96, 69]. The capacity bounds of analog network coding in a 2-way relay channel have been thoroughly analyzed from an information theoretic perspective [62, 69, 93, 92, 114, 116]. In [62] Katti et al. show that joint relaying and network coding can double the throughput for certain channel conditions. In [116], Xue et al. analyze the influence of traffic patterns and channel conditions on the gains achieved by network coding.

In [120], Zhang at al. propose an algorithm for separating two physical-layer signals using higher level information. The approach is not directly implementable in practice, though, because of several assumptions that the authors make, e.g., they assume that the interfering signals are synchronized at the symbol boundaries and that both signals have undergone the same attenuation when arriving at the router. These problems are overcome in [61], where Katti et al. make analog network coding more practical. The authors propose a communication scheme, where pairs of nodes that wish to exchange packets through a relay node are encouraged to transmit simultaneously. The relay node, without decoding the collided signal, amplifies and forwards it. The destination nodes then extract the packet destined to them by filtering out their own contribution from the mixed signal. Katti et al. design and implement the approach using software radios and show through experiments that significant gains can be achieved in several network topologies, such as the canonical 2-way relay channel (to which the authors refer as "Alice and Bob" topology), the "X" topology (two flows intersecting at one relay node), and unidirectional flow in a chain topology.

In [40] (see Chapter 8), we study the combined problem of analog network coding and link scheduling in the physical interference model. We make use of two definitions of analog network coding: one definition that uses cochannel signal separation to decode several messages simultaneously, and another definition that is based on the "amplify and forward" approach of [61]. We first analyze whether the ability to decode several signals simultaneously alters the complexity of the scheduling problem (see Section 8.3 and Section 8.4), and then propose a heuristic to schedule an arbitrary set of links using cochannel signal separation (see Section 8.5). As opposed to the related work discussed in this section, which is mostly aimed at deriving information theoretic capacity bounds or implementing working prototype systems, we are more interested in analyzing the impact of analog network coding on our link scheduling problem from an algorithmic perspective. We derive our results in the geometric setting of the physical interference model, and do not assume any underlying graph structure, as is frequently done when analyzing the complexity of network coding problems [65].

2.7 Latest Developments

The problem of scheduling wireless links in the physical interference model has been receiving an increasing attention in the algorithmic community. A lot of work is being carried out at the time of writing of this thesis. We would like to mention some of the results that came to our knowledge, some of which have not been published yet.

An interesting line of research has investigated static properties of the SINR model. In [3], for instance, Avin et al. analyze the shape of reception zones under uniform power assignment, i.e., when all nodes transmit at the same power level. Specifically, based on some algebraic properties of the polynomials defining the reception zones, the authors show that the reception zones are convex and relatively well-rounded. The authors refer to the partition of the plane in such reception zones as "SINR diagrams", and argue that a deeper understanding of this kind of structures might play a key role in the development of algorithms for wireless networks, similar to what Voronoi diagrams represent to proximity queries in computational geometry.

The problem of joint scheduling and power control has been studied by Fanghänel et al. [27]. They define *oblivious power assignment* to be an assignment in which the power value of a link depends only on the distance between the sender and the receiver. They prove that oblivious power assignments cannot achieve approximation ratios better than $\Omega(n)$ for the directed version of the problem. This is the worst possible performance guarantee that could be expected. Interestingly, for bidirectional version of the problem, i.e., when the SINR constraints have to be satisfied at both the receiver and the sender, the authors prove that there exists an algorithm with polylogarithmic approximation guarantee.

The scheduling problem without power control has been further studied by Halldórsson and Wattenhofer in [47]. The authors develop an algorithm with constant approximation guarantee and prove that the problem is in APX.

Chapter 3

Preliminaries, Models, and Definitions

This chapter starts with some preliminary definitions. In Section 3.1 we introduce some concepts from the computational complexity theory, and in Section 3.2 we present the definition of a metric space. The chapter proceeds with the description, in Section 3.3, of the wireless network model used to analyze the algorithms proposed in this thesis, namely the Physical interference model. The chapter is concluded with the definitions of three main problems studied in this work: the One-Slot Scheduling Problem, defined in Section 3.4, the Weighted One-Slot Scheduling Problem, defined in Section 3.5, and the Multi-Slot Scheduling Problem, defined in Section 3.6.

3.1 Computational Complexity Theory

In this thesis we study optimization problems which are not likely to admit polynomial-time algorithms. Therefore, we focus on approximation algorithms for these problems. Below are some important definitions from the *computational complexity theory*, that will be used throughout this work.

The complexity class NP-complete is a class of decision problems[1], which are both NP (verifiable in Nondeterministic Polynomial time) and NP-hard (any problem in NP can be translated into this problem). Consider an NP-complete problem Π. Intuitively, the first property states that candidate solutions to Π can be verified quickly. The second property implies that, if Π can be solved in polynomial time, then so can every problem $\Pi' \in$ NP.

In order to prove the first property, it is enough to show that it is possible to decide whether a candidate solution to Π is correct or not in polynomial

[1] Informally, a decision problem is a question with a yes-or-no answer, depending on the values of some input parameters.

time. In order to prove the second property, a polynomial-time *many-to-one reduction* from an already known NP-hard problem Π' to Π has to be provided. A polynomial-time many-to-one reduction from a problem Π' to Π is a deterministic algorithm, which transforms any instance $\pi' \in \Pi'$ into an instance $\pi \in \Pi$, such that the answer to π is YES if and only if the answer to π' is YES.

If a problem is NP-hard, we are unlikely to find a polynomial-time algorithm to solve it exactly. However, it may still be possible to find near-optimal solutions in polynomial time. An algorithm that returns provably near-optimal solutions is called an *approximation algorithm*. We say that an approximation algorithm has a ratio bound of $\rho(n)$ if, for any input of size n, the cost ALG of the solution produced by the approximation algorithm is within a factor of $\rho(n)$ of the cost OPT of an optimal solution:

$$\max\left(\frac{ALG}{OPT}, \frac{OPT}{ALG}\right) \leq \rho(n). \tag{3.1}$$

This definition applies both for minimization and maximization problems.

When it comes to classifying NP-hard problems according to their *hardness of approximation*, three main classes have been identified [108]. The approximation factors for theses classes of problems are *constant* (> 1), $\Omega(\log n)$, and n^{ε} for a fixed constant $\varepsilon > 0$, if the problem is a minimization problem; and *constant* (< 1), $O(1/\log n)$, and $1/n^{\varepsilon}$ for a fixed $\varepsilon > 0$, if the problem is a maximization problem. Examples of problems are Vertex Cover and Steiner Tree in the first class, Set Cover in the second class, and Clique and Coloring in the third class.

To prove that a problem Π_2 is hard to approximate, it is sufficient to provide a so-called *gap-preserving reduction* from a problem Π_1, already known to be hard to approximate with factor $\rho_1(|\pi_1|) \leq 1$ (assuming P $\neq$ NP), where Π_1 is a minimization problem, and π_1 is an instance of Π_1.

A gap-preserving reduction from Π_1 to Π_2, assuming both are minimization problems[2], comes with four parameters (functions), f_1, ρ_1, f_2, and ρ_2. Given an instance π_1 of Π_1, it computes, in polynomial time, an instance π_2 of Π_2 such that

- $OPT(\pi_1) \leq f_1(\pi_1) \quad \Rightarrow \quad OPT(\pi_2) \leq f_2(\pi_2)$,
- $OPT(\pi_1) > \rho_1(|\pi_1|) f_1(\pi_1) \quad \Rightarrow \quad OPT(\pi_2) > \rho_2(|\pi_2|) f_2(\pi_2)$.

We proceed with some terminology regarding metric spaces.

3.2 Metric Space

A *metric space* is a set V together with a function d (called a metric or *distance function*), which assigns a real number $d(x, y)$ to every pair $x, y \in V$

[2]The cases, where one or both of Π_1 and Π_2 are maximization problems, are similar.

satisfying the following properties (or axioms):

- $d(x, y) \geq 0$ (non-negativity);
- $d(x, y) = 0$ iff $x = y$ (identity of indiscernibles);
- $d(x, y) = d(y, x)$ (symmetry);
- $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality).

The last property is called the *triangle inequality* because (when applied to $\mathbb{R}^2$ with the usual metric) it says that the sum of two sides of a triangle is at least as big as the third side.

In a metric space $M = (V, d)$, the *ball* of radius R around element $x \in V$, denoted by $B_R(x)$, consists of all points that are within distance R of x. Formally,

$$B_R(x) = \{y \in V \mid d(y, x) \leq R\}.$$

In the following section we introduce the physical interference model, used to model a wireless communication network in this thesis.

3.3 The Physical Interference (SINR) Model

As already discussed in Chapter 1, traditionally, wireless multi-hop networks have been modeled by means of *graphs.* A graph model for a wireless network typically consists of two graphs: a *connectivity graph* and an *interference graph* (also referred to as *conflict graph*). In both graphs, the set of vertices represents the devices (senders and receivers). A successful message transmission is said to occur whenever the intended receiver r_i is connected by an edge in the connectivity graph to the sender s_i, *and* no other sender s_j, adjacent to r_i in the interference graph, transmits simultaneously. For a set of simultaneous transmissions to be feasible, therefore, the edges representing them in the conflict graph must not be adjacent. As a natural consequence, solving the scheduling problem in this kind of model usually boils down to finding *independent sets* and *colorings* in graphs.

Although the concept of an edge is a useful abstraction and allows for elegant algorithm design and analysis, it is a stark oversimplification of the physical laws behind the wireless communication medium. As opposed to a binary and local, distance-defined concept of an edge, the wireless signal is of continuous and accumulative nature. Whereas one transmission originated outside the pre-defined interference range of a node x may not disrupt the signal at x, the interference of many, possibly far-away located, transmitters might accumulate and prevent the correct reception at x. This illustrates that a set of transmissions forming an independent set in a graph model might be conflicting in the underlying physical model, and might therefore represent an unfeasible scheduling solution.

Fading channel models, such as the *physical interference model* [45], offer a more realistic representation of wireless communication. The definition of a successful transmission in this model, as opposed to the binary concept of a conflict in a graph-based model, accounts for interference generated by all senders that transmit simultaneously. 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.

In this thesis we study the problem of scheduling wireless communication requests (or simply links) in the *physical interference model.*

We assume that the *input* to the problem is a set $L = \{l_1, \ldots, l_n\}$ of n wireless links, where each link l_i represents a communication request from a sender s_i to a receiver r_i:

$$l_i = (s_i, r_i).$$

The communication devices, such as senders and receivers, are viewed as nodes positioned in a metric space (e.g. the Euclidean 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 a link l_i is referred to by

$$d_{ii} = d(s_i, r_i).$$

We assume that there are no primary 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.

Moreover, we assume that each link has a unit-traffic demand, and model the case of *non-unit traffic demand* by replicating each link k times, where k is the demand on the link.

In the *physical interference* model, the received signal power decays proportionally to the inverse of the distance between the sender and the receiver to the power of a so called *path-loss exponent* α, which is a constant, whose exact value depends on external conditions of the medium (humidity, obstacles, etc.), as well as the exact sender-receiver distance. It is commonly assumed that $\alpha > 2$ [45]. Therefore $d_{ij}^{-\alpha}$ denotes the propagation attenuation or *link gain* between sender s_i and receiver r_i. If $P(s_i)$ is the transmitting power level of a sender s_i, the received power at the receiver r_i is

$$P_{ii} = P_{r_i}(s_i) = \frac{P(s_i)}{d_{ii}^{\alpha}}.$$

The power received from s_i by the receiver r_j of a concurrently scheduled link l_j is referred to as *interference* and denoted by

$$I_{ij} = I_{r_j}(s_i) = P_{r_j}(s_i) = \frac{P(s_i)}{d_{ij}^{\alpha}}.$$

Note that since the power level received at the receiver cannot be higher than the power level at which the sender actually transmitted, there is an implicit assumption here that the minimum distance between any sender-receiver pair is one, i.e.:

$$d(s_i, r_j) \geq 1, \quad \forall s_i, r_j \mid l_i, l_j \in L.$$

Without loss of generality, we assume that the input instance L is always *normalized*, such that the minimum sender-receiver distance is at least one, i.e., the space in which the nodes $(s_1, r_1), \ldots, (s_n, r_n) \in L$ are located is "stretched" by a factor of $\max(1, 1/\min_{l_i, l_j \in L}(d_{ij}))$.

We denote by $\mathcal{S}_t = \{l_1, \ldots, l_m\}$ the set of concurrently scheduled links in time-slot t. As in [45], we assume that transmissions are slotted into *synchronized time-slots* of equal length and in each time-slot t, a node can either transmit or remain silent.

The *total interference* $I_{r_i}(\mathcal{S}_t)$ (sometimes also referred to as $I_{l_i}(\mathcal{S}_t)$, or simply as I_{r_i} or I_{l_i}) experienced by a receiver r_i is the sum of the interference power values created by all nodes in the network transmitting simultaneously in time-slot t (except the intending sender s_i), that is,

$$\begin{aligned} I_{r_i} &= I_{r_i}(\mathcal{S}_t) \\ &= \sum_{\substack{l_j \in \mathcal{S}_t, \\ l_j \neq l_i}} I_{r_i}(s_j). \end{aligned}$$

Let N denote the *ambient noise* power level. We define the *signal-to-interference-plus-noise ratio* of a link l_i, transmitting in time-slot t as

$$\begin{aligned} SINR_{l_i} &= SINR_{r_i}(\mathcal{S}_t) \\ &= \frac{P_{ii}}{I_{r_i} + N} \\ &= \frac{\frac{P(s_i)}{d_{ii}^{\alpha}}}{\sum_{\substack{l_j \in \mathcal{S}_t, \\ l_j \neq l_i}} \frac{P(s_j)}{d_{ji}^{\alpha}} + N}. \end{aligned}$$

Finally, let $\beta \geq 1$ denote a hardware-dependent minimum SINR threshold required for a successful message reception. A *successful transmission* between a sender s_i and a receiver r_i in time-slot t occurs if and only if

$$SINR_{l_i}(\mathcal{S}_t) \geq \beta. \tag{3.2}$$

We say that a *schedule* $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$ is feasible, or *correct*, if the following condition holds:

$$SINR_{l_i}(\mathcal{S}_t) \geq \beta, \quad \forall l_i \in \mathcal{S}_t,\ \forall t \in \{0, \ldots, T-1\}.$$

Throughout this thesis, we deal with the scheduling problem under *uniform power assignment* [45], i.e., we assume that all nodes transmit with the same power level P:

$$P(s_i) = P, \quad \forall l_i \in L.$$

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.

An important aspect of our model is the placement of nodes. We assume that nodes can be placed arbitrarily in the plane, possibly in a worst-case fashion (as opposed to uniform random distribution).

In practice, networks with heterogenous topologies are quite typical, and protocols should be designed such that they are capable of coping well with such heterogeneous topologies. In a sensor network, for instance, the density of sensors is expected to be much higher in areas of interest in order to capture all the desired data, whereas other locations are expected to contain the minimum number of nodes just to maintain connectivity. Note that the worst-case node deployment assumption makes the analysis significantly more involved compared to random, uniform scenarios.

Next we present formal definitions of three variations of the scheduling problem.

3.4 One-Slot Scheduling Problem (OSP)

The One-Slot Scheduling Problem can be formulated as follows. The input to the problem is a set of links $L = \{l_1, \ldots, l_n\}$, where each link l_i represents a communication request from a sender s_i to a receiver r_i. The objective of the One-Slot Scheduling Problem is to *maximize* the number of links scheduled concurrently in one time-slot, such that all messages are received *successfully*. In other words, we attempt to use one slot to its full capacity.

Formally, a set $\mathcal{S} = \{l_1, \ldots, l_m\} \subseteq L$ is a solution to an instance of the One-Slot Scheduling Problem if the following conditions hold:

$$\begin{aligned} \mathcal{S} &= \underset{\mathcal{S}' \subseteq L}{\operatorname{argmax}} |\mathcal{S}'|, \\ SINR_{l_i}(\mathcal{S}') &\geq \beta, \quad \forall l_i \in \mathcal{S}'. \end{aligned} \tag{3.3}$$

3.5 Weighted One-Slot Scheduling Problem (WOSP)

The Weighted One-Slot Scheduling Problem is a "weighted version" of the One-Slot Scheduling Problem. It can be formulated as follows. The input to the problem is a set of links $L = \{l_1, \dots, l_n\}$, where each link l_i is assigned a *weight* $w(l_i)$. The weights might represent, for example, the relative priorities of the communication requests, or the revenue values associated to different clients. The objective of the problem is to pick a subset of weighted links, such that the total weight (or value) is maximized and the $SINR$ level is at least β at every scheduled receiver.

A set $\mathcal{S} = \{l_1, \dots, l_m\} \subseteq L$ is a solution to an instance of the Weighted One-Slot Scheduling Problem if the following conditions hold:

$$\begin{aligned} \mathcal{S} &= \operatorname*{argmax}_{\mathcal{S}' \subseteq L} \sum_{l_i \in \mathcal{S}'} w(l_i), \\ SINR_{l_i}(\mathcal{S}') &\geq \beta, \quad \forall l_i \in \mathcal{S}'. \end{aligned} \tag{3.4}$$

3.6 Multi-Slot Scheduling Problem (MSP)

As opposed to the one-slot versions of the scheduling problem, where the objective is to use one time-slot to its full capacity, the objective of the Multi-Slot Scheduling Problem is to schedule all links in as few time-slots as possible, guaranteeing that all messages a delivered successfully according to the SINR condition (3.2).

More precisely, let $L = \{l_1, \dots, l_n\}$ be the input set of communication requests. A *schedule* is represented by $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{T(\mathcal{S})})$, where $T(\mathcal{S})$ denotes the length of the schedule and $\mathcal{S}_t = \{l_1, \dots, l_m\} \subseteq L$ is a subset of links scheduled in time-slot t.

A schedule $\mathcal{S}$ is a solution to an instance of the Multi-Slot Scheduling Problem if the following conditions hold:

$$\begin{aligned} \mathcal{S} &= \operatorname*{argmin}_{\mathcal{S}' = (\mathcal{S}'_1, \mathcal{S}'_2, \dots, \mathcal{S}'_{T(\mathcal{S}')})} T(\mathcal{S}'), \\ \bigcup_{t=1}^{T(\mathcal{S}')} \mathcal{S}'_t &= L, \\ SINR_{l_i}(\mathcal{S}'_t) &\geq \beta, \quad \forall l_i \in \mathcal{S}'_t \subseteq \mathcal{S}', \quad t \in \{0, \dots, T(\mathcal{S}') - 1\}. \end{aligned} \tag{3.5}$$

The above presented problem definitions will be the focus of the following Chapters 4, 5, and 6.

Chapter 4

Complexity in Geometric SINR

As has been discussed in Section 2.1 of the Related Work Chapter, there are relatively few results on the hardness of network problems in a geometric setting. However, insights on the complexity are very important for the design of efficient algorithms.

In this chapter we analyze the complexity of two problems defined in Chapter 3: the Multi-Slot Scheduling Problem and the Weighted One-Slot Scheduling Problem. We prove both problems to be NP-hard in the so-called geometric SINR model ($SINR_G$).

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).

We begin this chapter with a discussion about the $SINR_A$ and the $SINR_G$ models in Section 4.1. We proceed by illustrating how a typical hardness proof in the $SINR_A$ model works in Section 4.2, where we provide such a proof for the Multi-Slot Scheduling Problem. Our main complexity results are presented in Sections 4.3 and 4.4, where we show that the Multi-Slot Scheduling Problem and the Weighted One-Slot Scheduling Problem are both NP-hard in the $SINR_G$ model.

4.1 Geometric x Abstract SINR ($SINR_G$ x $SINR_A$)

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 α, α 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 chapter, we formally prove that the Multi-Slot Scheduling Problem and the Weighted One-Slot Scheduling Problem are both NP-hard 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-hardness proofs for $SINR_G$. As we have discussed in Section 2.1 of the Related Work Chapter, there have been various NP-hardness proofs for wireless network models, in particular for the 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. This is not surprising, as the G in UDG stands for graph.

In reality, however, gain cannot be chosen arbitrarily. As we argued

[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.

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 determine 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.

Before presenting the main results of this chapter, in the next section (Section 4.2) we illustrate a typical hardness proof in the $SINR_A$ model.

4.2 Multi-Slot Scheduling Problem in $SINR_A$

In this section we show that it is NP-hard to approximate the Multi-Slot Scheduling Problem in the "abstract SINR" model ($SINR_A$) to within a factor of $n^{1-\varepsilon}$, for any constant $\varepsilon > 0$.

Theorem 4.1. *There is no $n^{1-\varepsilon}$ factor approximation algorithm for the scheduling problem in the $SINR_A$ model, for any constant $\varepsilon > 0$, assuming* P $\neq$ NP.

Proof. We will prove the result by presenting a *gap-preserving* reduction from the graph coloring problem. In [121] 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 π_C of the graph coloring problem defined for an undirected graph $G = (V, E)$ on n vertices. We construct (in polynomial time) an instance π_S of scheduling, such that

$$OPT(\pi_C) \leq k \Rightarrow OPT(\pi_S) \leq k, \quad (4.1)$$

$$OPT(\pi_C) > n^{1-\varepsilon}k \Rightarrow OPT(\pi_S) > n^{1-\varepsilon}k. \quad (4.2)$$

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 $n \times n$ path-loss matrix A is defined as follows:

- $(v, w) \in E \Rightarrow A_{wv} = A_{vw} = 1$,
- $(v, w) \notin E \Rightarrow A_{wv} = A_{vw} = n$,
- $v = w \Rightarrow A_{wv} = A_{vw} = 1$.

To see that (4.1) holds, assume that we can color π_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 can be lower bounded by

$$\begin{aligned} SINR_{l_v}(V(c_i)) &= \frac{\frac{P}{1}}{\sum_{\substack{w \in V(c_i),\\ w \neq v}} \frac{P}{n}} \\ &\geq \frac{n}{n-1} \\ &> 1 \\ &= \beta, \quad \forall l_v,\ v \in V(c_i),\ i \in \{1, \ldots, k\}. \end{aligned}$$

To see that (4.2) holds, assume we cannot color π_C with $\leq n^{1-\varepsilon}k$ colors. We have to show that π_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 at least one node $v \in V$. If s is the color of v, i.e., $v \in V(c_s)$., the SINR at the link l_v associated with this node is:

$$\begin{aligned} SINR_{l_v}(V(c_s)) &\leq \frac{\frac{P}{1}}{\frac{P}{1} + \sum_{\substack{w \in V(c_s),\\ w \neq v, w \neq x}} \frac{P}{n}} \\ &< 1 \\ &= \beta, \quad l_v \mid v, x \in V(c_s),\ s \in \{1, \ldots, k\}. \end{aligned}$$

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

4.3 Multi-Slot Scheduling Problem in $SINR_G$

Proving a 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 giving a polynomial time reduction from the Partition Problem, an NP-complete special case of the well known Subset Sum problem, to the Multi-Slot Scheduling. If the solution to an instance of the Multi-Slot Scheduling problem implies a solution to any instance of the Partition Problem, then Multi-Slot Scheduling must be at least as hard as Partition.

Theorem 4.2. *Multi-Slot Scheduling is NP-hard.*

Proof. We show that the Partition Problem is reducible to the Multi-Slot Scheduling Problem in polynomial time. The Partition Problem (proved to be NP-complete by Karp in his seminal work [60]) can be formulated as follows: Given a set $\mathcal{I}$ of integers, is it possible to divide this set into two

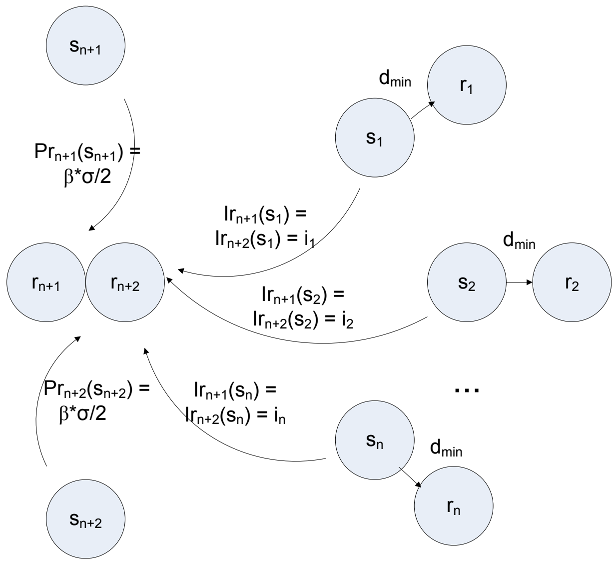


Figure 4.1: Reduction from Partition: link l_{n+1} (or l_{n+2}) can be scheduled if and only if the interference caused by simultaneously scheduled links $s_j, j \in \{1 \dots n\}$ is less our equal to $\sigma/2$.

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 $\mathcal{I}$.

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

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

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 Multi-Slot Scheduling Problem. Then, we argue that the instance of the Multi-Slot 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[2]. We construct the following Multi-Slot Scheduling Problem instance with $n+2$ links $L = \{l_1, \ldots, l_{n+2}\}$ (see Figure 4.1). 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 $(P/i_j)^{1/\alpha}$,

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

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

$$d_{\min} = P^{\frac{1}{\alpha}} \cdot \frac{\left(\frac{1}{(i_{\max}-1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \right)}{\left(1 + (n\beta)^{\frac{1}{\alpha}}\right)},^{3} \tag{4.3}$$

[2]Note that the assumption that the integers in the Partition instance are distinct is not essential for the reduction to work, and we make it merely for simplicity's sake.

[3]Note that this implies that some sender-receiver distances are less than one and the received power $P_{r_i} = P/d_{\min}^{\alpha}$ can be larger than the transmitting power P. As has been stated in Section 3.3, to overcome this issue, we assume that the problem instance is normalized such that the minimum distance between any sender-receiver pair is at least one. The power level P is normalized accordingly, such that it is high enough for the

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

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

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.,

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

Having defined the geometric instance of the Multi-Slot 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 $L_i = \{l_j \mid 1 \leq j \leq n+1 \text{ and } i \neq j\}$. 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,*

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

Proof. 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.

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

longest link in the input set to transmit successfully in the presence of ambient noise. For the sake of simplicity, we do not change the notation to reflect this normalization. Power P therefore denotes an already normalized constant. Note that the exact value of P does not affect the reduction, since P is still uniform (fixed and equal for all nodes) and can be determined for any instance of Partition.

nodes s_j, s_k.

$$\begin{aligned} d(s_j, s_k) &= |d(s_j, r_{n+1}) - d(s_k, r_{n+1})| \\ &= \left| \left(\frac{P}{i_j}\right)^{\frac{1}{\alpha}} - \left(\frac{P}{i_k}\right)^{\frac{1}{\alpha}} \right| \\ &\geq P^{\frac{1}{\alpha}} \left(\frac{1}{(i_{\max} - 1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \right). \end{aligned} \tag{4.4}$$

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 ($d_{\min}$ is defined in (4.3)). All the other sender nodes (including s_{n+1}) are farther away. This suffices to show a lower bound for $SINR_{r_i}(L_i)$.

$$\begin{aligned} SINR_{r_i}(L_i) &> \frac{\frac{1}{d_{\min}^{\alpha}}}{\frac{n}{(d(s_j,s_i)-d_{\min})^{\alpha}}} \\ &\geq \frac{1}{n} \left(\left(1 + (n\beta)^{\frac{1}{\alpha}}\right) - 1 \right)^{\alpha} \\ &= \beta. \end{aligned} \tag{4.5}$$

□

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 Theorem 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 $L \setminus \{l_{n+1}, l_{n+2}\}$ 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}. □

4.4 Weighted One-Slot Scheduling Problem in $SINR_G$

In this section we prove that the decision version of the Weighted One-Slot Scheduling Problem, under uniform power assignment scheme, is also NP-hard in the $SINR_G$ model.

Theorem 4.4. *Weighted One-Slot Scheduling Problem is NP-hard.*

Proof. We prove that the Knapsack Problem is reducible to the Weighted One-Slot Scheduling Problem in polynomial time.

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:

$$\max \sum_{j=1}^{n} p_j x_j, \qquad \text{s.t.} \tag{4.6}$$

$$\sum_{j=1}^{n} w_j x_j \quad \leq \quad W, \tag{4.7}$$

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

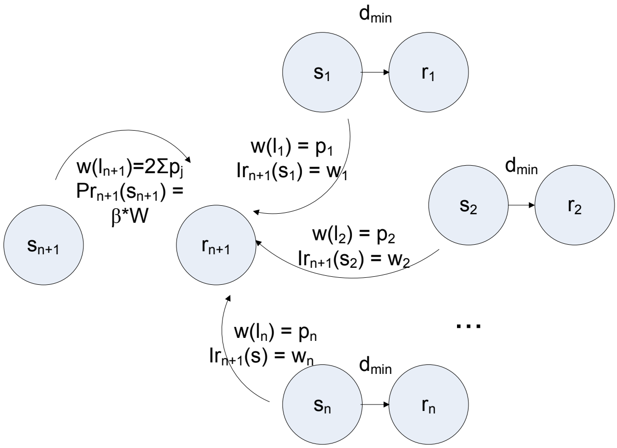


Figure 4.2: Reduction from Knapsack: the weight of simultaneously scheduled links is maximized if and only if the sum of the values p_j assigned to them is maximized and the knapsack capacity W is not violated.

Without loss of generality, we assume that there are only items of distinct integer weights. As in the proof for the Multi-Slot Scheduling Problem, we start by defining a any-to-one reduction from any instance of the Knapsack Problem to a geometric instance of the Weighted One-Slot 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 (see Figure 4.2). We position a sender node s_i in the plane for each item x_i, such that the received power from s_i at (0,0) is exactly the item weight 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.

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

$$d_{\min} = P^{\frac{1}{\alpha}} \cdot \frac{\left(\frac{1}{(w_{\max}-1)^{1/\alpha}} - \frac{1}{w_{\max}^{1/\alpha}} \right)}{\left(1 + (n\beta)^{\frac{1}{\alpha}} \right)},[4]$$

[4] As has been done in Section 4.3, we assume that the problem instance is normalized

and $w_{\max}$ is the largest item 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:

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

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 Weighted One-Slot 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 Weighted One-Slot 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 (4.6) and (4.7), which fulfills the condition of the Knapsack Problem. Because $\mathcal{S}_{OPT}$ maximizes the

such that the minimum distance between any sender-receiver pair is at least one, and the power level P is high enough for the longest link in the input set to transmit successfully in the presence of ambient noise.

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-hard problem. □

4.5 Outlook

In this chapter we have established that the Multi-Slot Scheduling Problem and the Weighted One-Slot Scheduling Problem are both NP-hard in the "geometric $SINR$" ($SINR_G$) model. As we discussed in Section 4.1, 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.

In order to prove that the problems discussed in this chapter are also NP-complete, we have to prove that they are in the complexity class NP. It turns out that, for some operations on integers, it is not yet clear whether they can be computed efficiently by a Turing machine. E.g., it is not known how a sum of square roots of integers can be compared quickly to an integer [89]. Since our model requires the computation of roots of integers, we do not know whether scheduling and related problems are in NP. If we assume the Real RAM model (often used in computational geometry), all our computations can be implemented efficiently.

Note that some problems still remain open in this context, e.g., whether the One-Slot Scheduling Problem is also NP-hard. Moreover, given that the Partition and the Knapsack problems are only weakly NP-hard, the hardness results presented in this chapter are also weak, in the sense that they do not establish strong NP-hardness.[5]

Since the problems that we defined in Chapter 3 are unlikely to admit polynomial-time optimal solutions, in the following chapters we will turn our attention to designing efficient approximation algorithms. In particular, in Chapters 5 and 6, we propose scheduling algorithms that compute feasible solutions in the $SINR_G$ model in polynomial time with worst-case approximation guarantees for arbitrary network topologies.

[5] A problem is said to be NP-hard in the strong sense if it remains so even when all of its numerical parameters are bounded by a polynomial in the length of the input.

Chapter 5

Diversity Scheduling

Solving problems in the $SINR$ model is very difficult, as is documented by the vast amount of literature with heuristics on this subject (see Section 2.3 of the Related Work Chapter). In this chapter we present the first scheduling algorithms with a proven approximation guarantee in the physical interference model.

We propose two scheduling algorithms. In Section 5.1 we present an algorithm for the Multi-Slot Scheduling Problem, and in Section 5.2 we present an algorithm for the Weighted One-Slot Scheduling Problem.

These algorithms represent our initial efforts to solve the link scheduling problem in the SINR model, and algorithms with significantly improved approximation guarantees are going to be presented in Chapter 6 of this thesis.

Before describing the algorithms, let us introduce the notion of *link length diversity* $g(L)$, namely the number of magnitudes of distances between senders and receivers in the network. Formally, $g(L)$ is defined as

$$g(L) := |\{m | \exists l_i, l_j \in L : \lfloor \log(d_{ii}/d_{jj}) \rfloor = m\}|. \tag{5.1}$$

For our problem, $g(L)$ denotes the number of non-empty length classes of the set of links to be scheduled. Note that, in realistic scenarios, the diversity $g(L)$ can usually be regarded as a constant. In theory, however, $g(L)$ can be as large as n, the number of links in the network.

Both algorithms in this chapter 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 $O(g(L))$ Approximation Algorithm for MSP

The algorithm presented in this section is inspired by the heuristic proposed by Moscibroda et al. in [86], which schedules a strongly connected set of links in the SINR model using linear power assignment. Although similar in spirit, the algorithm in [86] is not designed to schedule an arbitrary set of links and does not provide an approximation guarantee for the obtained solution.

Algorithm 1 Approximation Algorithm for the Multi-Slot Scheduling Problem.

1: **input:** Set of links $L = \{l_1, \ldots, l_n\}$;
2: **output:** Schedule $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$;
3: Let $G = \{G_0, \ldots, G_{\lfloor \log(\max d_{ii}) \rfloor}\}$ such that G_k is the set of links l_i of length $2^k \leq d_{ii} < 2^{k+1}$;
4: Set μ according to (5.2);
5: $t := 0$;
6: **for all** $G_k \neq \emptyset$ **do**
7:     Partition the plane into squares of width $\mu \cdot 2^k$;
8:     4-color the squares such that no two adjacent squares have the same color;
9:     **for** $j = 1$ **to** 4 **do**
10:         **repeat**
11:             **for all** squares A_j^k of width $\mu \cdot 2^k$ and color j **do**
12:                 Pick one not yet scheduled link $l_i \in G_k$ with receiver r_i in A_j^k; (if there is any such l_i left unscheduled)
13:                 $L_j^k := L_j^k \cup l_i$; (schedule l_i in time-slot t)
14:             **end for**
15:             $t := t + 1$;
16:             $\mathcal{S}_t := L_j^k$;
17:         **until** all links with receivers in any square A_j^k have been scheduled
18:     **end for**
19: **end for**
20: **return** $\mathcal{S}$;

The algorithm (for a description in pseudo-code see Algorithm 1) starts by partitioning the input set of links L into $\lceil \log(\max d_{ii}) \rceil$ (where $\max d_{ii}$ is the length of the longest link $l_i \in L$) possibly empty length classes. Each length class G_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

$$\mu = 4\left(8\beta \cdot \frac{(\alpha-1)}{(\alpha-2)}\right)^{\frac{1}{\alpha}}, \tag{5.2}$$

and then the grid cells are colored regularly with 4 colors (see Figure 5.1). Links whose receivers belong to different squares of the same color are scheduled simultaneously. Note that the inner *repeat* loop (lines 10-17) constructs a schedule of length $\Delta_j^k = \max_{A_j^k \in G_k}(|A_j^k|)$, which is the maxi-

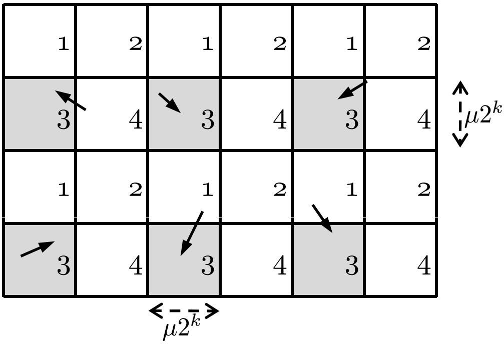


Figure 5.1: In line 11 of Algorithm 1, the algorithm picks all squares colored with color j. The example shows an inner loop iteration for length class G_k and $j = 3$. The algorithm schedules one unscheduled link from each selected square (if there exists one).

mum number of links in length class G_k, whose receivers are in the same grid cell of color j. Given that there are 4 colors and $g(L)$ non-empty length classes, all links are scheduled in $4 \cdot \Delta \cdot g(L)$ time slots, where $\Delta = \max_{A_j^k \in \left\{G_0, \ldots, G_{\lfloor \log(\max d_{ii}) \rfloor}\right\}} (|A_j^k|)$.[1]

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 G_k. Every link $l_i \in G_k$ satisfies $d_{ii} < 2^{k+1}$, thus the perceived power at r_i from s_i is at least

$$P_{r_i}(s_i) \geq \frac{P}{2^{\alpha(k+1)}}. \quad (5.3)$$

[1] Here we overload the term A_j^k to denote the set of receivers $r_i \mid l_i \in G_k$, located inside the grid cell A_j^k; and the term G_k to denote the grid comprised by cells of width $\mu \cdot 2^k$.

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 $d(r_i, s_j) \geq \mu 2^k - 2^{k+1} = 2^k(\mu - 2)$ to r_i (see Figure 5.1). Consequently, the sum of their interference experienced by r_i is less than

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

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

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

We continue aggregating the interference from nodes s_j at distance range

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

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

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

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

$$\begin{aligned}
I_{r_i} &\leq \sum_{l=1}^{\infty} \frac{8P \cdot l}{(2^k((2l-1)\mu-2))^\alpha} \\
&\leq \frac{8P}{2^{k\alpha}} \sum_{l=1}^{\infty} \frac{l}{(\frac{1}{2}(2l-1)\mu)^\alpha} && (5.4) \\
&\leq \frac{8P}{2^{(k-1)\alpha}\mu^\alpha} \sum_{l=1}^{\infty} \frac{l}{(2l-l)^\alpha} \\
&\leq \frac{8P}{2^{(k-1)\alpha}\mu^\alpha} \sum_{l=1}^{\infty} \frac{1}{l^{\alpha-1}} \\
&\leq \frac{8P}{2^{(k-1)\alpha}\mu^\alpha} \frac{(\alpha-1)}{(\alpha-2)}, && (5.5)
\end{aligned}$$

where (5.4) follows because $x - 2 > x/2, \ \forall x > 4$ and $\mu > 4$, given that $\beta \geq 1$ and $\alpha > 2$; and (5.5) follows from a bound on Riemann's zeta function. Using

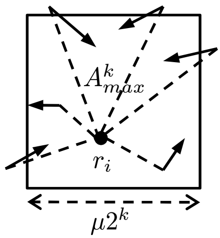


Figure 5.2: Lower Bound: an optimum algorithm could schedule at most q links with receivers in A^k_{max} in length class G_k in a single time slot.

(5.3), (5.5), and plugging in the value of μ, defined in (5.2), 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 (5.1).*

Proof. The proof relies on the choice of a so called *critical grid cell*[2]

$$A^k_{max} = \underset{A^k_j \in \left\{G_0,\ldots,G_{\lfloor \log(\max d_{ii})\rfloor}\right\}}{\operatorname{argmax}} |A^k_j|, \tag{5.6}$$

i.e., we choose the cell with the highest density $\Delta = |A^k_{max}|$ over all $g(L)$ generated grids (see Figure 5.2). Note that Δ is the number of links l_i whose receiver is located in cell A^k_{max} and whose length class is G_k, i.e.,

[2]Here we overload the term A^k_j to denote the set of receivers $r_i \mid l_i \in G_k$, located inside the grid cell A^k_j; and the term G_k to denote the grid comprised by cells of width $\mu \cdot 2^k$.

$2^k \leq d_{ii} < 2^{k+1}$. We proceed by showing that an optimum algorithm OPT can schedule all Δ in at least $T_{OPT} \geq \lceil \Delta/q \rceil$ time slots, where q is a constant dependent on parameters α and β (μ is defined in (5.2)):

$$q = \frac{\left(2(\sqrt{2}\mu + 1)\right)^{\alpha}}{\beta}. \tag{5.7}$$

Assume, by contradiction, that OPT schedules all links in less than T_{OPT} time slots. Therefore, there must exist a time slot $t', 1 \leq t' \leq T_{OPT}$, 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:

$$\begin{aligned} SINR_{r_i \in A^k_{max}} &\leq \frac{\frac{P}{d_{ii}^{\alpha}}}{P \cdot \sum_{j=0}^{q} d(s_j, r_i)^{-\alpha}} \\ &< \frac{\frac{P}{2^{k\alpha}}}{P \cdot q \cdot (2\sqrt{2}\mu 2^k + 2^{k+1})^{-\alpha}} \qquad (5.8) \\ &= \beta, \qquad (5.9) \end{aligned}$$

where (5.8) follows from the fact that $d_{ii} \geq 2^k$, $d_{jj} < 2^{k+1}$ and $d(r_i, r_j) \leq 2\sqrt{2}\mu 2^k$; and (5.9) follows from definition (5.7) of q.

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

$$T_{OPT} \geq \left\lceil \frac{\Delta}{q} \right\rceil. \tag{5.10}$$

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

$$T_{ALG1} \leq 4 \cdot \Delta \cdot g(L). \tag{5.11}$$

The approximation ratio follows from (5.10) and (5.11):

$$\begin{aligned} \frac{T_{ALG1}}{T_{OPT}} &\leq 4q \cdot g(L) \\ &= O(g(L)). \qquad (5.12) \end{aligned}$$

□

5.2 $O(g(L))$ Approximation Algorithm for WOSP

Algorithm 1 can be adapted to solve the Weighted One-Slot Scheduling Problem described in Section 3.5 of Chapter 3 (see pseudo code of the adapted version in Algorithm 2). As before, the input set L is partitioned into $\lceil \log(\max d_{ii}) \rceil$ (possibly empty) length classes, and grids with cell size $\mu \cdot 2^k, k \in \{0 \ldots \lfloor \log(\max d_{ii}) \rfloor\}$ are colored with 4 colors $j \in \{1 \ldots 4\}$. Then,

Algorithm 2 Approximation Algorithm for the Weighted One-Slot Scheduling Problem.

input: Set of links $L = \{l_1, \ldots, l_n\}$;
output: One-slot schedule $\mathcal{S}$;
Let $G = G_0, \ldots, G_{\lfloor \log(\max d_{ii}) \rfloor}$ such that G_k is the set of links l_i of length $2^k \leq d_{ii} < 2^{k+1}$;
Set μ according to (5.2);
for all $G_k \neq \emptyset$ **do**
  Partition the plane into squares of width $\mu \cdot 2^k$;
  4-color the cells such that no two adjacent cells have the same color.
  for $j = 1$ **to** 4 **do**
    for all squares A_j^k of width $\mu \cdot 2^k$ and color j **do**
      Pick the *heaviest* link $l_i \in G_k$ with receiver r_i in A_j^k; (if there is any l_i whose receiver is in A_j^k)
      $L_j^k := L_j^k \cup l_i$;
    end for
  end for
end for
return $\mathcal{S} := \text{argmax}_{L_j^k} \sum_{l_i \in L_j^k} w(l_i)$;

$4 \cdot g(L)$ feasible one-slot schedules L_j^k are generated by greedily picking the *heaviest* link in all squares A_j^k of the same color. In the end, the heaviest one-slot schedule L_j^k among all colors and all link classes is chosen.

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 (5.1)).*

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

$$w(OPT) = \sum_{k=0}^{\lfloor \log(\max d_{ii}) \rfloor} w(OPT_k). \tag{5.13}$$

In Theorem 5.2 we showed that an optimum algorithm could schedule at most q (defined in (5.7)) links in each cell A_j^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:

$$w(L_j^k) \geq \frac{1}{4q} \cdot w(OPT_k), \quad \forall j \in \{1 \dots 4\},\ k \in \{0 \dots \lfloor \log(\max d_{ii}) \rfloor\}. \tag{5.14}$$

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

$$\begin{aligned} w(\mathcal{S}) &\geq \frac{1}{4 \cdot g(L)} \cdot \sum_{k=0}^{\lfloor \log(\max d_{ii}) \rfloor} w(L_j^k) \\ &\underset{(5.14)}{\geq} \frac{1}{16q \cdot g(L)} \cdot \sum_{k=0}^{\lfloor \log(\max d_{ii}) \rfloor} w(OPT_k) \\ &\underset{(5.13)}{=} \frac{w(OPT)}{16q \cdot g(L)} \Rightarrow \\ \frac{w(OPT)}{w(ALG2)} &\leq 16q \cdot g(L) \\ &= O(g(L)). \end{aligned} \tag{5.15}$$

□

5.3 Outlook

The approximation ratio of the algorithms presented in this chapter is $O(g(L))$. Although this is the first result to provide any approximation guarantee for the link scheduling problem in the SINR model, it leaves a lot of space for improvement, given that, depending of the topology of the network, this guarantee becomes extremely bad ($\Omega(n)$), i.e., not better than the guarantees offered by the most naive solutions to the problem.

This undesired dependency on the diversity $g(L)$ of the network, however, has been inherited by a number of scheduling algorithms in the SINR model, e.g. [22, 30].

In the next chapter we are going to present improved scheduling algorithms, whose approximation guarantee no longer depends on the the diversity $g(L)$ or any other topological characteristic of the network.

Chapter 6

Approximative Scheduling

The first result presented in this chapter (Section 6.1) is an algorithm that maximizes the number of concurrently scheduled links in one time-slot, i.e., it solves the One-Slot Scheduling Problem. We prove that the algorithm has a constant approximation guarantee. As opposed to the algorithms presented in Chapter 5, this result holds regardless of the topology of the network. This means that, given a set of links, with arbitrary length diversity $g(L)$ and arbitrarily distributed in the Euclidean space, it returns a subset of links obeying the SINR constraints, of size at most a constant factor smaller than the maximum possible. To the extent of our knowledge, this is the first scheduling algorithm with approximation guarantee independent of the topology of the network.

In Section 6.2, we further use this (maximization) one-slot scheduling algorithm to derive a minimum-length schedule with $O(\log n)$ approximation factor. In Section 6.3, we complement our results by looking at the algorithm's performance in metric spaces beyond the two-dimensional Euclidean plane. 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, in Section 6.4 we present simulation results, 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.

6.1 Constant Approximation Algorithm for OSP

In order to solve the (minimization) Multi-Slot Scheduling Problem, we use a "master-slave" approximation strategy, where the "slave" problem is the (maximization) One-Slot Scheduling Problem (defined in Section 3.4). 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

$$RI_v(u) = \frac{I_{uv}}{P_{vv}}.$$

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

$$\begin{aligned} a_{l_v}(S) &= \beta\left(\frac{N}{P_{vv}} + \sum_{l_u \in S} RI_v(u)\right) \\ &= \beta \cdot \frac{\sum_{l_u \in S} I_{uv} + N}{P_{vv}}. \end{aligned} \tag{6.1}$$

Observe that a solution $\mathcal{S}$ is *valid*, or feasible, iff the affectedness (by the other nodes in $\mathcal{S}$) of each link in $\mathcal{S}$ is at most 1:

$$a_{l_v}(\mathcal{S}) \leq 1, \quad \forall l_v \in \mathcal{S}. \tag{6.2}$$

Algorithm 3 Approximation Algorithm for the One-Slot Scheduling Problem.

1: **input:** Set of links $L = \{l_1, \cdots, l_n\}$;
2: **output:** One-slot schedule $\mathcal{S}$;
3: Set c according to (6.3);
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 $a_{l_w}(\mathcal{S}) \geq 2/3$;
8: **until** $L = \emptyset$
9: **return** $\mathcal{S}$;

The one-slot scheduling algorithm (for a description in pseudo-code see Algorithm 3) 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. Firstly (line 6), 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 (6.3)). Secondly (line 7), all links l_w, whose affectedness $a_{l_w}(\mathcal{S})$ 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.

6.1.1 Correctness

In this section we prove that the solution $\mathcal{S}$ obtained in Algorithm 3 is correct, i.e., all selected links can be scheduled concurrently without collisions.

Lemma 6.1. *Algorithm 3 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 $a_{l_v}(S_v^-) < 2/3$. It remains to show that S_v^+ affects l_v by at most 1/3.

Our first observation is that disks D_w of radius $d_{vv}(c-1)/2$ around senders in S_v^+ do not intersect. Consider two senders $s_w, s_z \in S_v^+$. We will first consider the case when l_w was added to S_v^+ before l_z, i.e. $d_{ww} \leq d_{zz}$; and then the case when $d_{ww} > d_{zz}$. In the first case, we know that $d(s_z, r_w) \geq c \cdot d_{ww}$. Therefore, by triangular inequality, we have that $d(s_z, s_w) \geq d(s_z, r_w) - d_{ww} \geq c \cdot d_{ww} - d_{ww} = d_{ww}(c-1) \geq d_{vv}(c-1)$ (the last inequality follows from the fact that $s_w \in S_v^+$). In the second case, when $d_{ww} > d_{zz}$, the reasoning is the same: $d(s_w, r_z) \geq c \cdot d_{zz}$, since l_z was added first, and $d(s_w, s_z) \geq d_{zz}(c-1) \geq d_{vv}(c-1)$. Therefore, disks 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 disks 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 disks 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

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

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

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

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

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

$$c = \max\left(2, \left(2^5 3^2 \beta \frac{\alpha-1}{\alpha-2}\right)^{\frac{1}{\alpha}}\right). \tag{6.3}$$

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

6.1.2 Approximation Ratio

To analyze the performance of Algorithm 3, 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 6.2. *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 6.3. *$OPT_1 \leq \rho_1 \cdot ALG$, where $\rho_1 = (2c+1)^\alpha$ and constant c as defined in (6.3).*

Proof. Consider the set X_v from OPT_1 eliminated in line 6 of Algorithm 3, 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 $2c \cdot d_{vv} \leq 2c \cdot d_{ww}$ from s_w. By Lemma 6.2, 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 3, we will need the following two definitions and a combinatorial lemma, to which we refer as the *blue-dominant centers lemma*. Informally, if we are given two sets of points, let's call them red and blue points, we say that a blue point is *blue-dominant* if it is "shadowed", or "protected", by other blue points from the red points in all directions. We call the set of blue points that "protect" the blue-dominant point from the red points a *guarding set.*

Definition 6.4. *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,*

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

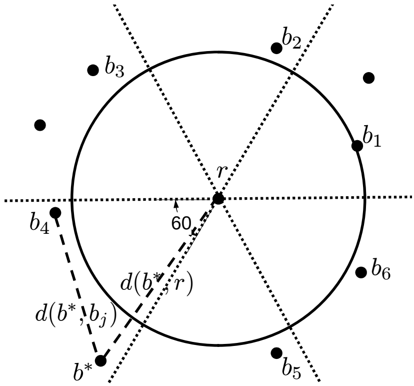


Figure 6.1: Constructing a q-guarding set $G_q(r), q = 1$ of size at most $6 \cdot q = 6$ for the red point r ($G_q(r) = \{b_1, \ldots, b_6\}$).

Definition 6.5. *Let $\mathcal{R}$ and $\mathcal{B}$ be defined as above. Let $r \in \mathcal{R}$ be a red point and $G(r) \subseteq \mathcal{B}$ be a set of blue points. We say that $G(r)$ is guarding r if for all $b^* \in \mathcal{B} \setminus G(r)$, we have that $B_{d(b^*,r)}(b^*) \cap G(r) \neq \emptyset$. Furthermore, we say that $G_q(r)$ is q-guarding r if for all $b^* \in \mathcal{B} \setminus G_q(r)$, we have that $B_{d(b^*,r)}(b^*) \cap G_q(r) \geq q$.*

Lemma 6.6. *(Blue-dominant centers lemma in 2D) Let $\mathcal{R}$ and $\mathcal{B}$ be two disjoint sets of red and blue points in the 2-dimensional Euclidean space, and q be a positive integer. If $|\mathcal{B}| > 6q|\mathcal{R}|$ 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 a q-guarding set $G_q(r) \subseteq \mathcal{B}'$ (guarding r relative to the current $\mathcal{B}'$) and remove $G_q(r)$ from $\mathcal{B}'$.

We claim that it is possible to construct a guarding set $G_q(r)$ of size at most $6q$. The procedure to construct $G_q(r)$ is as follows (see Figure 6.1). Consider a red point r. Draw 6 sectors of $60°$ originating at r. For each of these 6 sectors sec_j, include the closest q blue points $b_j \in sec_j$ in $G_q(r)$ (if sec_j has less than q blue points from $\mathcal{B}'$, pick all the blue points in this sector). Now $G_q(r)$ has size at most $6q$, and we claim that it is guarding r. Suppose it is not. Then there is a point $b^* \in \mathcal{B}' \setminus G_q(r)$ with $B_{d(b^*,r)}(r) \cap G_q(r) < q$. Suppose b^* is located in sec_j and we selected q blue points b_j from sec_j into $G_q(r)$. This means that $d(b^*, b_j) > d(b^*, r)$ for some $b_j \in sec_j$, which implies that the sector angle is larger than $60°$. (Note that if $G_q(r)$ contains less than q blue points 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 $b^* \in \mathcal{B}'$ is now q-blue-dominant. This holds since (1) all $G_q(r)$'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. 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 6.6, we are now able to bound OPT_2, the number of links deleted in the second elimination step (line 7) of the algorithm.

Lemma 6.7. *$OPT_2 \leq \rho_2 \cdot ALG$, where $\rho_2 = 6 \cdot 3^{\alpha+1}$.*

Proof. Suppose otherwise. Consider the set of senders from $ALG \cup OPT_2$. Label those from OPT_2 as blue ($\mathcal{B} = \{s_b \mid l_b \in OPT_2\}$) and those from ALG as red ($\mathcal{R} = \{s_r \mid l_r \in ALG\}$). By Lemma 6.6, 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 q-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 6.2, we have that $d(s^*, s_r) \geq 2d(s^*, r^*)$. By the triangular inequality, it then follows that $d(s_r, r^*) \geq d(s^*, s_r) - d(s^*, r^*) \geq (1/2)d(s^*, s_r)$ and for each $s_b \in G^*(s_r)$, $d(s_b, r^*) \leq d(s^*, s_b) + d(s^*, r^*) \leq (3/2)d(s^*, s_r)$. The relative interference of the red sender s_r on r^* is then bounded by

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

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

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

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

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

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

Theorem 6.8. *The approximation ratio of Algorithm 3 is $O(1)$.*

Proof. The result follows by adding the bounds of Lemmas 6.3 and 6.7, which results in $OPT \leq OPT' + ALG \leq ALG(\rho_1 + \rho_2 + 1)$, where $\rho_1 = (2c+1)^\alpha$, $\rho_2 = 6 \cdot 3^{\alpha+1}$, and $c = \max(2, (2^5 3^2 \beta(\alpha-1)/(\alpha-2))^{1/\alpha})$. □

6.2 $O(\log n)$ Approximation Algorithm for MSP

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 4) consists in iteratively computing a one-slot schedule using Algorithm 3. Each one-slot solution is scheduled in a separate slot, and the remaining links are repeatedly used as input to Algorithm 3. The procedure continues until all links in L have been scheduled.

Algorithm 4 Approximation Algorithm for the Multi-Slot Scheduling Problem.

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 3)
  $L := L \setminus \mathcal{S}_t$;
  $t := t + 1$;
until $L = \emptyset$
return $\mathcal{S}$;

The correctness of the obtained schedule has been proved in Lemma 6.1, and in the following theorem we show that the overall approximation ratio of Algorithm 4 is $O(\log n)$.

Theorem 6.9. *The approximation ratio of Algorithm 4 is* $O(\log n)$.

Proof. For each iteration t of Algorithm 4, 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 4, 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 3 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

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

□

6.3 Going Beyond Two Dimensions

In this section we look into the issue of whether the analysis of Algorithm 3 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 6.1 and 6.3), we compute the value of the constant c (see Def. 6.3) for three dimensions. For higher dimensions D, c_D can be computed analogously, by working with volumes of n-spheres, instead of disk areas as for 2D. For the second part of the proof (Lemmas 6.6 and 6.7), 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. 6.11.

Lemma 6.10. *Algorithm 3 produces a valid schedule in a three-dimensional Euclidean space, if* $\alpha > 3$ *and* $c = c_{3D}$,

$$c_{3D} = \max\left(2, \left(2^5 3^3 7 \beta \frac{\alpha-2}{\alpha-3}\right)^{\frac{1}{\alpha}}\right). \tag{6.4}$$

Proof. The proof proceeds along the lines of Lemma 6.1, replacing disk areas for ball volumes, which renders:

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

which, by plugging in the value of c_{3D} results in affectedness of any scheduled link l_v by longer links of $a_{l_v}(S_v^+) \leq 1/3$. This, together with the affectedness by shorter links of $a_{l_v}(S_v^-) < 2/3$, 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 6.3, 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 (6.4).

For the second part of the proof, we investigate a little bit further into the *blue-dominant centers lemma* (6.6). 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

$$|\mathcal{B}| > \kappa \cdot |\mathcal{R}|, \tag{6.5}$$

then at least one of the blue points is (1-)blue-dominant (Def. 6.4).

Definition 6.11. *Let $(\mathcal{V}, d)$ be a metric space and let $v \in \mathcal{V}$. A set $I \subseteq \mathcal{V}\backslash\{v\}$ 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 6.12. *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. 6.5).*

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 6.13. *(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 6.6, only replacing 6 by γ and applying Lemma 6.12 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 6.13) to bound the number of links OPT_2 eliminated by Algorithm 3 in its second elimination step (line 7). As in Lemma 6.7, 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 6.13 we deduced that the constant κ in (6.5) can be chosen as the so-called independence-dimension, defined in Definition 6.11. 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.

6.4 Simulation Results

In this section we present some simulation results to better illustrate the practical appeal of the scheduling algorithms proposed in Chapters 5 and 6 for the Multi-Slot Scheduling Problem. We refer to Algorithm 1, introduced in Section 5.1, as ApproxDiversity and to the Algorithm 4, introduced in Section 6.2, as ApproxLogN. We compare the performance of ApproxDiversity and ApproxLogN to the performance of the scheduling algorithms proposed in [15], to which we refer as GreedyPhysical. As ours, GreedyPhysical is a polynomial-time algorithm, designed to schedule an arbitrary set of links in the SINR model. To the extent of our knowledge, at the time of writing of this thesis, this was the only algorithm in the literature to fulfill these criteria (see Related Work Chapter).

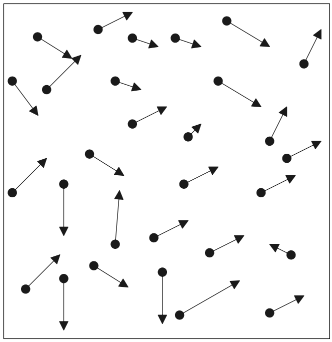
We generated two kinds of topologies: *random* and *clustered* (see Figures 6.2(a) and 6.2(b)). In the random topology, n receiver nodes were distributed uniformly at random on a plane field of size 1000x1000 units, and n senders were positioned uniformly at random inside disks of radius l_{max} around 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 disks 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.

In all experiments, the number of simulations was chosen large enough to obtain sufficiently small confidence intervals.

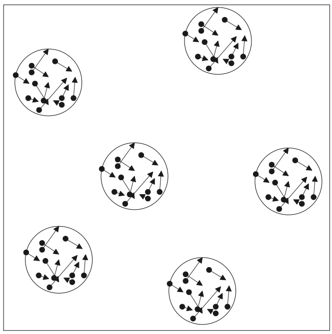
Firstly, we analyze the lengths of the schedules as a function of the number of nodes ($n \in \{100 \cdot 2^0, 100 \cdot 2^1, \cdots, 100 \cdot 2^8\}$). In Figures 6.3(a) and 6.3(b) the results for random topology are shown. Since this scenario is not very challenging, all three algorithms have good 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 6.4(a) and 6.4(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.

In Figures 6.5(a) and 6.5(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

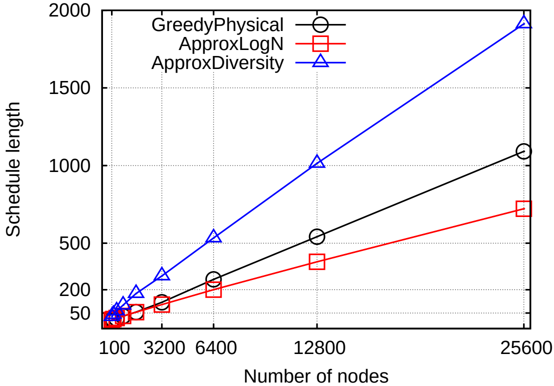


(a) Random.



(b) Clustered.

Figure 6.2: Simulated topologies: 1Kx1K field, $\alpha = 3$, $\beta = 1.2$, $N = 0$



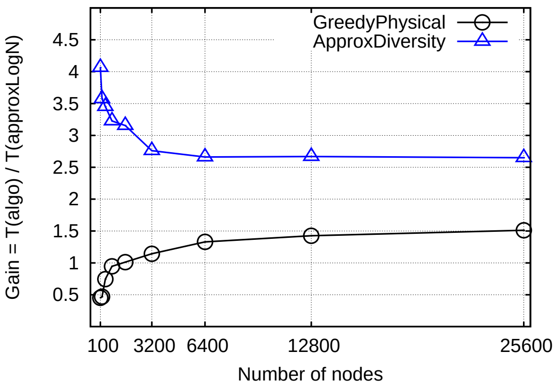
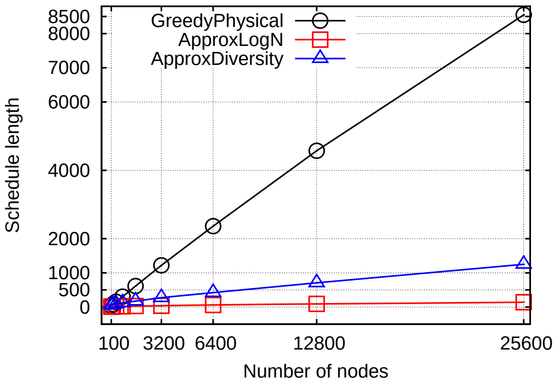


Figure 6.3: Random Topology: $l_{max} = 20$.



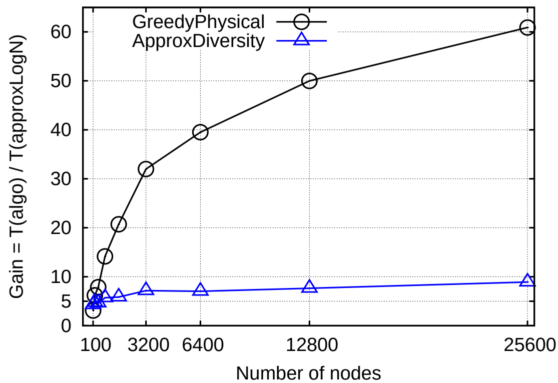


Figure 6.4: Clustered Topology: $n_C = n/10$, $r_C = 10$.

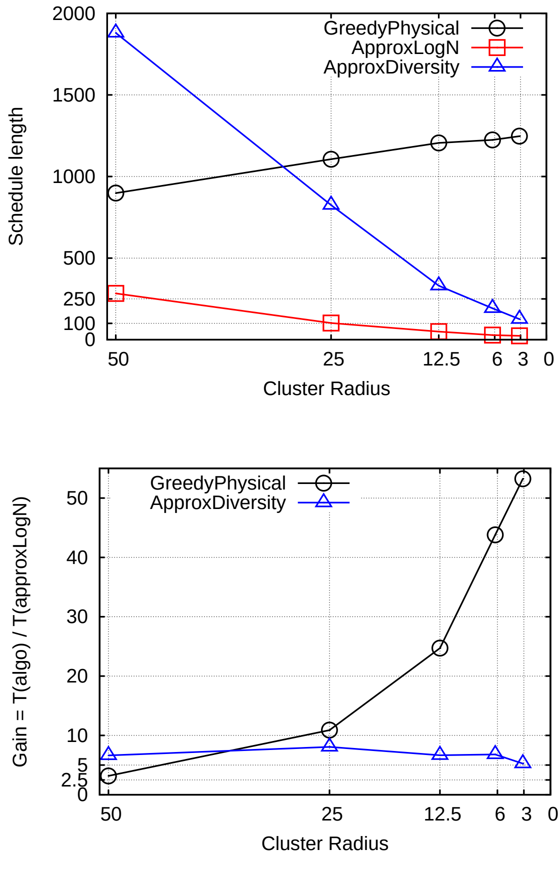


Figure 6.5: Clustered Topology: $n = 3.2K$, $n_C = n/10$.

to take advantage of this possibility. Among all three algorithms, ApproxLogN presents the best performance in all cases. Note that for large cluster radius ($r_C > 25$), the ApproxDiversity approach presents an extremely poor performance. This is due to the fact that, in this scenario, links are relatively very long, and the grid structure built by the algorithm uses very large cells, which forces it to schedule very few links in parallel.

Next we analyze the influence of the path-loss exponent α in both random (Figures 6.6(a) and 6.6(b)) and clustered (Figures 6.7(a), and 6.7(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.

6.5 Outlook

As we have already pointed out, all the solutions to the link scheduling problem in the SINR model have either considered special-case topologies, or presented optimality guarantees that become arbitrarily bad depending on the topology of the network. In this chapter 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 [45], 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

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

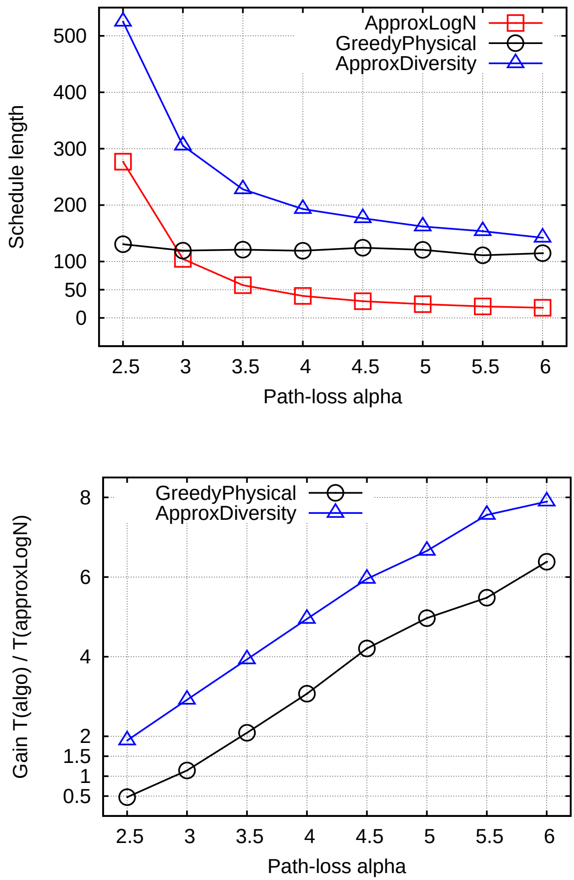


Figure 6.6: Random topology: $n = 3.2K$, $l_{max} = 20$.

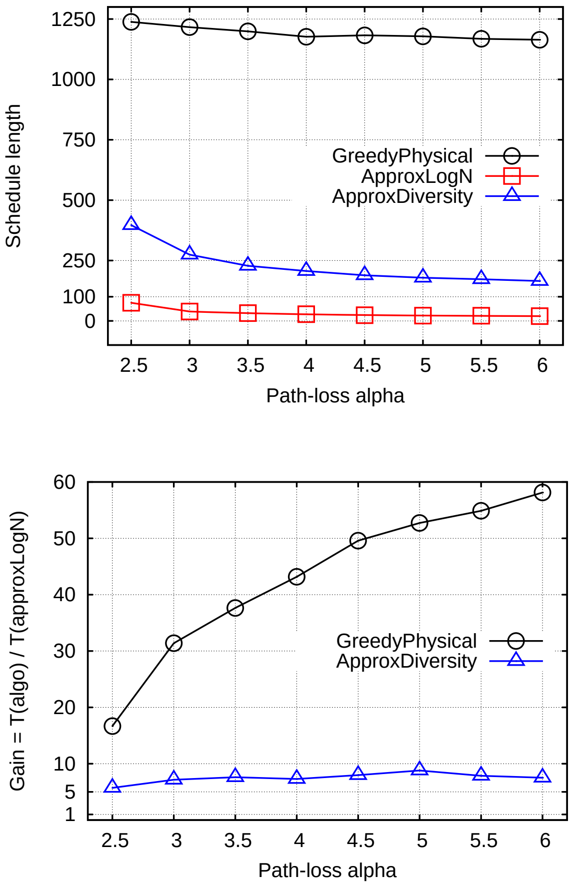


Figure 6.7: Clustered topology: $n = 3.2K$, $n_C = n/10$, $r_C = 10$.

where T is the size of the schedule returned by Algorithm 4.

Chapter 7

Local Broadcasting

In this chapter we study a problem of both theoretical and practical interest: the *local broadcasting problem*. Local broadcasting is an operation used as a building block for many higher-layer protocols (such as routing, synchronization, or coordination protocols) in wireless ad-hoc and sensor networks. As a consequence, the time required to successfully transmit a message to all neighbors in the physical proximity of a node frequently lower bounds and often dominates the overall performance of such critical higher-layer protocols.

We analyze the local broadcasting problem in a particularly harsh communication model, which we describe in Section 7.1. This communication model is based on the so-called *unstructured radio network model* [72], and one of its key characteristics is that there is no pre-defined global start time of the algorithm, meaning that nodes wake up asynchronously and join the network at any time during the execution of the protocol. As opposed to previous work, which used graph-based definitions of interference, we make our analysis in the *physical interference model*.

We present two distributed randomized algorithms. To begin with, in Section 7.2, we study a very simple Aloha-like algorithm that is based on the assumption that each node knows the number of its neighbors, i.e., the number of nodes in geographical proximity. In Section 7.3, we present a second algorithm, whose analysis is significantly more involved. This algorithm makes no assumptions about topology knowledge, and provably achieves close to optimal performance. We present upper bounds on the execution time of the algorithms in Sections 7.2 and 7.3, and we present lower bounds for both algorithms in Section 7.4. Therefore, our analysis establishes approximation guarantees for both algorithms, showing that their performance is close to optimal even in worst-case situations. Finally, in Section 7.5, we look into the average-case behavior of the proposed algorithms through simulations.

7.1 Communication Model

The most intuitive communication model in distributed computing is probably the *synchronous message passing model*. It models the network as an undirected graph, in which vertices represent computing devices and edges represent bidirectional communication channels between pairs of nodes. Time is assumed to be divided into globally synchronized time-slots, all nodes start the execution (of a distributed algorithm) simultaneously, and are allowed to reliably exchange at most one message (of unlimited size) with each of their neighbors in each time slot. Between consecutive time slots, nodes can perform "unlimited" local computation. The efficiency of a distributed algorithm is then measured in terms of the maximum number of time slots it takes for a node to complete its task and/or in terms of the total number of messages sent by all nodes during the execution of the task.

When it comes to modeling wireless networks, the message passing model might abstract away too many important characteristics, such as, for example, the fact that a wireless channel is not always reliable, or the fact that having access to a global clock in a distributed system is usually not possible. In this work we use a communication model, which is based on the so called *unstructured radio network model*, introduced by Kuhn et al. in [72, 71]. This model can be characterized by the following properties:

1. Time is divided into synchronous time slots, but no global clock is assumed to exist.

2. A message is received correctly iff exactly one neighbor transmits. Note that an underlying graph structure is assumed, which establishes which nodes are neighbors and which are not.

3. There is no collision detection at the nodes, i.e., they cannot distinguish between a clear channel and a channel with two collided messages.

4. Nodes *wake up asynchronously* at any time, i.e., new nodes can join at any time during the execution of the protocol, and upon waking up, nodes do not have any information about which nodes are awake or for how long they have been awake.

5. Nodes have no information about how many neighbors they have.

6. Nodes have an estimate on the total number of nodes n in the network.

7. Nodes have unique IDs, which however do not have to be in the range $\{1 \dots n\}$.

This model represents a particularly harsh network scenario, especially because of property (4).

In this work we combine this communication model with the physical interference model. The main difference to our model is in property (2). Since we assume no underlying communication graph in our model, we cannot use definition (2) of a successful transmission. Instead, we use the SINR definition (3.2) of a successful transmission, presented in Chapter 3. As is going to be described later in the section, we are going to analyze two scenarios that differ in property (5).

Note that assumption (1), that time is divided into time-slots, is basically for the sake of the analysis. Our algorithms do not rely on synchronized time-slots in any way. This would be too unrealistic an assumption, given that nodes do not have access to a global clock and synchronizing time-slots is an expensive task. Assuming a slotted channel in the analysis is justified due to the analysis of slotted vs. unslotted Aloha [100], where it was shown that the two scenarios differ only by a factor of 2.

Next we define the problem studied in this chapter.

Definition 7.1. *The problem of Local Broadcasting can be formulated as follows. Given a set of nodes V, such that each one wishes to locally broadcast a message to all its neighbors within a certain broadcasting range, the objective is to schedule all these requests in as few time-slots as possible.*

In order to reason about our algorithms, we now introduce several new definitions and notation. We define terms *broadcasting range*, *proximity range*, and *transmission range* of a node, all of which are important in the context of our work.

Definition 7.2. *The local broadcasting range R_B of a node x is the distance up to which x intends to broadcast its messages. We refer to the region within this range as broadcasting region B_x and to the number of nodes in it as Δ_x^B. A local broadcast is complete if every node x in the network has transmitted a message to every node in B_x.*

Definition 7.3. *The transmission range R_T of a node x is the maximum distance from which it can receive a clear transmission ($SINR_x \geq \beta$), assuming no other transmission occurs simultaneously in the network. We refer to the region within this range as T_x and to the number of nodes in it as Δ_x^T. Given fixed power level P and ambient noise N, and assuming zero interference in Equation 3.2, the transmission range R_T is*

$$R_T \leq \left(\frac{P}{\beta \cdot N} \right)^{1/\alpha}$$

.

In addition to these two definitions, we will make use of the novel notion of a *proximity range* R_A, which is a range between the broadcast and transmission range of a node. Intuitively, it describes the distance within which nodes responsible for the most significant part of interference experienced by x are located. The exact definition of the proximity range is determined by parameters α and β of the SINR model, and changes for each of the algorithms (see Equations (7.2) and (7.4) for the precise definitions), but in all cases, it is at least twice as big as the broadcasting range ($R_A \geq 2R_B$). We call the region covered by this radius *proximity region* A_x and refer to the number of nodes in it as Δ_x^A.

Finally, we define a *successful local broadcast.*

Definition 7.4. *Consider a transmitter x and a power level P. We define a successful local broadcast to be a transmission of a message, such that it is successfully received by all receivers y located in the local broadcasting region B_x. The successful reception condition is defined as in (3.2), Chapter 3.*

The ideas behind the proximity and transmission ranges are reminiscent to those in the *protocol* interference model, where an interference (or carrier sensing) range (maximum distance up to which a node sensing the channel detects an ongoing transmission) and a transmission range (maximum distance up to which a packet can be received) are defined. The proximity range R_A can be viewed as a separator of the deployment area into a "close-in" region (from where the most significant share of interference comes from) and a "far away" region (from where the incoming interference is still significant, but can often be treated as a constant).

In the analysis we show that when the proximity range R_A is carefully chosen, a node can perform a successful local broadcast with high probability whenever it is the only transmitting node in its proximity range. Therefore, in spite of the global nature of the SINR interference model, concurrent local broadcasts are possible when enough spatial separation exists, i.e., the local broadcasting range R_B is sufficiently smaller than the proximity range R_A.

We analyze two topology awareness scenarios:

- *Known competition*: The nodes know the number Δ_x^A of nodes in their proximity range A_x.

- *Unknown competition*: In this more realistic scenario, nodes are clueless about the current number of nodes in close proximity with which they have to compete for the shared medium. However, we assume that all nodes have the same estimate on the total number of nodes in the network $\hat{n} = |V|$.[1] In other words, each node may have between 0 and n nodes in its proximity range, but it does not know how many.

We conclude the section with some useful facts and remarks.

[1] Notice that without this minimal assumption and in absence of a global counter, every algorithm requires at least time $\Omega(n/\log n)$ until a single successful broadcast is achieved, even in a single hop network [59].

Fact 7.5. *Given a set of probabilities $p_1 ... p_n$ with $\forall i : p_i \in [0, \frac{1}{2}]$, the following inequalities hold (proof in [59]):*

$$(1/4)^{\sum_{k=1}^{n} p_k} \leq \prod_{k=1}^{n} (1 - p_k) \leq (1/e)^{\sum_{k=1}^{n} p_k}.$$

Fact 7.6. *For all n, t, such that $n \geq 1$ and $|t| \leq n$,*

$$e^t \left(1 - \frac{t^2}{n}\right) \leq \left(1 + \frac{t}{n}\right)^n \leq e^t.$$

Fact 7.7. *Consider two disks D_1 and D_2 of radii R_1 and R_2, $R_1 > R_2$, we define χ^{R_1,R_2} to be the smallest number of disks D_2 needed to cover the larger disk D_1. Because the limit of the ratio of the area of D_1 to the total area of smaller disks D_2 when $R_2 \to 0$ is $2\pi/3\sqrt{3}$ [64], and because all small disks D_2 intersecting D_1 are completely inside the area of radius $R' = R_1 + 2R_2$, it holds that*

$$\chi^{R_1,R_2} \leq \frac{2\pi}{3\sqrt{3}} \cdot \frac{(R_1 + 2R_2)^2}{R_2^2}.$$

Remark 7.8. *We assume that the ambient noise level N is upper bounded by a fraction of the maximum tolerable interference level for a successful broadcast ($(I_y + N) \ll P/\beta(R_B)^\alpha$), such that spatial reuse is achievable by concurrent local broadcasts:*

$$N \leq \frac{P}{2\beta(2R_B)^\alpha}. \tag{7.1}$$

Note that the exact value of the maximum ambient noise level does not influence our analysis in any significant way, the upper bound in (7.1) is set for the sake of simplicity.

7.2 Known Competition

We start the technical part of this chapter by analyzing the performance of a simple algorithm, which we call *Multi-Hop Aloha*. *Multi-Hop Aloha* assumes that each node knows the number of nodes Δ_x^A in its proximity range R_A. Then, after waking up, each node x simply transmits with probability $p = 1/\Delta_x^A$ and remains silent with probability $1 - p$.

Our goal is to show that, although the SINR model is intrinsically global and the interferences of distant nodes can accumulate and cause collisions, it is possible to guarantee efficient medium access (in particular, local broadcasts) using this simple and completely distributed algorithm. Specifically, in the analysis we prove that with high probability, every node x performs at least one successful local broadcast after $O(\Delta_x^A \log n)$ time-slots.

7.2.1 Analysis

For the purpose of our analysis, we introduce the concept of *probabilistic interference*, which is the expected value of total interference experienced by a node.

Definition 7.9. *Consider a node $x \in V$. The probabilistic interference at x, ψ_x^V, is defined as the expected value of interference experienced by x in a certain time-slot.*

$$\psi_x^V = P \sum_{v \in V \setminus \{x\}} \frac{p_v}{d_{vx}^\alpha},$$

where P is the transmission power, p_v is the sending probability of node v in time-slot t, and d_{vx} is the distance between x and the interfering node v.

In the following lemma we show that, given an upper bound on the sum of sending probabilities inside each broadcasting region $B_v, v \in V$, the probabilistic interference caused by nodes located outside the proximity region A_x of a node x can be bounded by a constant. Given an upper bound on the expected interference coming outside the region A_x, it becomes possible, in a way, to abstract away this interference and to reason mainly about the interference caused by nodes within the proximity range R_A. The analysis in the *physical* interference model then becomes similar to the analysis used in the *protocol* interference model.

Lemma 7.10. *Consider a node x and its proximity region A_x, of radius R_A. If in a time-slot t, the sum of transmission probabilities inside all broadcasting regions can be bounded by a constant, i.e., if $\sum_{w \in B_v} p_w \leq c, \forall v \in V$, then the probabilistic interference experienced by x, caused by nodes outside region A_x, can be bounded by*

$$\begin{aligned}
\psi_x^{v \notin A_x} &= P \sum_{v \notin A_x} \frac{p_v}{d_{vx}^\alpha} \\
&\leq c \cdot P \left(\frac{\alpha - 1}{\alpha - 2}\right) 3^3 2^{(\alpha-2)} R_A^{(2-\alpha)} R_B^{-2}.
\end{aligned}$$

Proof. Consider rings $Ring^l$ of width R_A around x, containing all nodes v, for which $lR_A \leq d_{vx} \leq (l+1)R_A$. The first such layer $Ring^0$ is the proximity region A_x. Consider all nodes $v \in Ring^l$ for some integer $l > 0$. All corresponding broadcasting regions B_v must be located entirely in an extended ring $Ring_+^l$ of area

$$\begin{aligned}
A(Ring_+^l) &= \left[((l+1)R_A + R_B)^2 - (lR_A - R_B)^2\right]\pi \\
&= (2l+1)(R_A^2 + 2R_A R_B)\pi \\
&< (2l+1)(R_A^2 + 2R_A R_B + R_B^2)\pi \\
&= (2l+1)(R_A + R_B)^2\pi \\
&\leq (2l+1)(3/2R_A)^2\pi.
\end{aligned}$$

Each transmitter v in $Ring^l, l \geq 1$ has distance at least $l \cdot R_A$ from x, each transmitter $w \in B_v$ has distance $d(w,x) \geq (lR_A - R_B)$ from x. Since $R_B \leq 1/2R_A$ and $l \geq 1$, $d(w,x) \geq l \cdot R_A/2$. By applying a geometric argument[2], we can bound the probabilistic interference $\psi_x^{Ring^l}$ incurred by nodes located in ring $Ring^l, l \geq 1$ as

$$\begin{aligned}
\psi_x^{Ring^l} &= \sum_{v \in Ring^l} \psi_x^v \\
&\leq \frac{A(Ring_+^l)}{A(B_v)} \cdot P \sum_{\substack{w \in B_v,\\ v \in Ring^l}} \frac{p_w}{(lR_A/2)^\alpha} \\
&\leq \frac{(2l+1)}{l^\alpha} \cdot P \cdot c \cdot 3^2 2^{\alpha-2} R_A^{(2-\alpha)} R_B^{-2} \\
&\leq \frac{1}{l^{(\alpha-1)}} \cdot P \cdot c \cdot 3^3 2^{\alpha-2} R_A^{(2-\alpha)} R_B^{-2}.
\end{aligned}$$

Summing up the interferences over all rings yields

$$\begin{aligned}
\psi_x^{v \notin A_x} &< \sum_{l=1}^{\infty} \psi_x^{Ring^l} \\
&\leq c \cdot P \cdot \sum_{l=1}^{\infty} \frac{1}{l^{\alpha-1}} \cdot 3^3 2^{\alpha-2} R_A^{(2-\alpha)} R_B^{-2} \\
&< c \cdot P \cdot \frac{\alpha-1}{\alpha-2} 3^3 2^{\alpha-2} R_A^{(2-\alpha)} R_B^{-2},
\end{aligned}$$

which concludes the proof of the lemma. □

In the following theorem we prove that the algorithm is correct and efficient.

Theorem 7.11. *After $O(\Delta_x^A \log n)$ time-slots, each node x performs a local broadcast successfully, with probability at least $1 - 1/n^2$. The claim also holds for all nodes with probability at least $1 - 1/n$.*

Proof. Given the user-defined broadcasting range R_B, we define the proximity range R_A of a node x to be a function of R_B, α and β:

$$R_A = R_B \left(3^3 2^\alpha \beta \cdot \left(\frac{\alpha-1}{\alpha-2}\right)\right)^{\frac{1}{(\alpha-2)}}. \tag{7.2}$$

[2] Here the argument is similar to the one used in Lemma 6.1, only instead of using a bound on the minimum distance between senders (located in the ring area), we use a bound on the sum of sending probabilities inside each broadcasting region B_v. Note that when two such regions intersect, the sending probabilities within the intersected area is counted twice. Therefore, by "geometric argument", we mean that there can be at most $A(Ring)/A(B_v)$ such disjoint broadcasting regions B_v in each ring.

Note that $R_A > 2R_B$, since $\beta \geq 1$ and $2 < \alpha \leq 6$. It follows that if a node y is located inside the broadcasting region of x, then

$$\begin{aligned} B_x \subset A_{y \in B_x} \quad &\Rightarrow \quad \Delta_x^B \leq \Delta_y^A \Rightarrow \\ p_y = \frac{1}{\Delta_y^A} \leq \frac{1}{\Delta_x^B} \quad &\Rightarrow \quad \sum_{y \in B_x} p_y \leq 1. \end{aligned} \tag{7.3}$$

The main goal is to bound the expected $SINR_{y \in B_x}$ of the intended receiver of x. Consider the proximity region A_y of the receiver y. Using (7.2), (7.3) and Lemma 7.10 (note that constant c in the Lemma is equal to 1 due to (7.3), i.e., $c = 1$), we can bound the probabilistic interference experienced by y caused by nodes located *outside* A_y:

$$\begin{aligned} \psi_y^{v \notin A_y} \quad &< \quad 1 \cdot P \cdot \frac{\alpha - 1}{\alpha - 2} 3^3 2^{\alpha - 2} R_A^{(2-\alpha)} R_B^{-2} \\ &= \quad \frac{P}{4\beta R_B^\alpha}. \end{aligned}$$

Given the expected value of interference at the intended receiver y, caused by transmissions outside A_y, we can use *Markov inequality* to claim that the probability that the interference at y caused by transmissions outside its proximity region exceeds $2 \cdot \psi_y^{v \notin A_y}$ is less than 1/2. Consequently, provided that x is the only node transmitting in A_y, with probability $P_{SINR \geq \beta} \geq 1/2$, the $SINR$ at the intended receiver $y \in B_x$ can be lower bounded by

$$SINR_{y \in B_x} \quad \geq \quad \frac{\frac{P}{d_{xy}^\alpha}}{2 \cdot \psi_y^{v \notin A_y} + N} > \beta,$$

which holds since $d_{xy} \leq R_B$ and ambient noise N is upper bounded by (7.1).

The probability $P_{none}^{A_y}$ that no node attempts to transmit in the proximity region A_y of y is at least

$$\begin{aligned} P_{none}^{A_y} \quad &\geq \quad \prod_{w \in A_y} (1 - p_w) \\ &\underset{Fact\ 7.5}{\geq} \quad \left(\frac{1}{4}\right)^{\sum_{w \in A_y} p_w} \geq \left(\frac{1}{4}\right)^{\sum_{v \in A_y} \sum_{w \in B_v} p_w} \\ &\underset{Eq.(7.3)}{\geq} \quad \left(\frac{1}{4}\right)^{\sum_{v \in A_y}} \geq \left(\frac{1}{4}\right)^{\chi^{R_A, R_B}}. \end{aligned}$$

Putting everything together, we define the probability that node x performs a local broadcast successfully at a time-slot as

$$\begin{aligned} P_{\substack{send \\ success}} \quad &\geq \quad P_{SINR \geq \beta} \cdot P_{none}^{A_y} \\ &\geq \quad \left(\frac{1}{2}\right)\left(\frac{1}{4}\right)^{\chi^{R_A, R_B}}. \end{aligned}$$

Since at each time slot each node locally broadcasts successfully with constant probability, the probability P_{fail} that a node does not transmit successfully after $\lambda\lceil\log n\rceil$ time-slots, where $\lambda = 4\Delta_x^A \cdot 4^{\chi^{R_A,R_B}}$, is

$$P_{\substack{fail \\ send}} \leq \left(1 - \frac{1}{2\Delta_x^A}\left(\frac{1}{4}\right)^{\chi^{R_A,R_B}}\right)^{\lambda\lceil\log n\rceil} < \frac{1}{n^2}.$$

Because there are n nodes to be scheduled, the probability that the claim holds for all nodes is at least

$$P_{all} \geq \left(1 - \frac{1}{n^2}\right)^n \geq \left(1 - \frac{1}{n}\right).$$

□

Note that Theorem 7.11 proves that *Multi-Hop Aloha* is not only efficient and provides fast media access, but is also fair, given that each node's schedule depends only on the local parameter Δ_x^A, allowing fast scheduling in low-density areas, regardless of the existence of highly dense regions somewhere else in the network.

7.3 Unknown Competition

The simple protocol in the previous section crucially depends on nodes knowing the number of neighbors in their proximity. If nodes do not have this information, designing an efficient algorithm becomes substantially more difficult, because nodes do not know at what probability they should transmit. In this section we describe and analyze the *SSMA* (*Slow-Start Media Access*) protocol. Since nodes do not know with how many nodes they have to compete for the medium, we use a technique that allows each node to start with a very low sending probability and exponentially increase it until they make an attempt to transmit or hear a successful broadcast on the channel. The idea is to eliminate conflicts through randomization, but still guarantee fast medium access for all nodes. The only assumption here is that each node has a rough estimate $\hat{n}$ of the total number of nodes in the network. From now on, we will refer to the estimate $\hat{n}$ as n.[3]

The *SSMA* protocol (Algorithm 5) works in rounds, each of which contains $\delta\lceil\log n\rceil$ time-slots. In every time-slot, a node sends with probability p. Starting from a very small value, this sending probability p is doubled in the beginning of every round. For the algorithm to work properly, we must prevent the noise floor (i.e., the sum of sending probabilities) from reaching too high values. Otherwise, too many collisions will occur. Hence, upon making

[3]Notice that the algorithm's running time depends only poly-logarithmically on the estimate of n. Hence, it degrades only marginally even if the estimate is very inaccurate.

an attempt to send or upon receiving a message (i.e., when $SINR_x \geq \beta$), a node x resets the value of p and starts the incrementing process again. Once a node makes an attempt to broadcast (without knowing whether it was successful or not), it increments a counter. After a node has made $\lambda\lceil\log n\rceil$ attempts, it stops executing the algorithm.

Consider the broadcasting region B_x of a node x. Let t be a time-slot in which a message is sent by a node $y \in B_x$ and received (without collision) by all other nodes $z \in B_x, z \neq y$. We say that a *Drastic Interference Reduction* (DIR) occurs in the broadcasting region B_x in time-slot t, since all nodes decide to reset their sending probability.

Algorithm 5 *SSMA: Slow-Start Media Access*

```
count := 0;
λ := 4 · 4^((3/2)χ^(R_A,R_B));
δ := 12 · 4^((3/2)(1+χ^(R_A,R_B)));
loop
  p := 1/(4n);
  for i := 0 to ⌈log n⌉ do
    p := 2p;
    for j := 0 to δ⌈log n⌉ do
      if (SINR_x ≥ β) then
        goto line 5; (reset)
      end if
      s := { 1  with probability p
             0  with probability (1 − p)
      if (s = 1) then
        transmit();
        count := count + 1;
        goto line 5; (reset)
      end if
    end for
  end for
  if (count > λ⌈log n⌉) then halt; fi
end loop
```

The parameters δ and λ are chosen as to optimize the results and guarantee that all claims hold with high probability. Parameter δ is chosen large enough to ensure that, with high probability, there is a round in which a DIR occurs. Parameter λ is chosen large enough to ensure that each node performs a local broadcast successfully in at least one round with high probability.

7.3.1 Analysis

We begin the analysis by defining the proximity range R_A:

$$R_A = R_B \left(3^4 2^{(2\alpha-1)} \beta \left(\frac{\alpha-1}{\alpha-2} \right) \right)^{\frac{1}{(\alpha-2)}} \tag{7.4}$$

We proceed in the following manner. In Lemma 7.12, we prove that, during the entire execution of the algorithm, the sum of sending probabilities in every broadcasting region B_x is bounded by a constant. In Lemma 7.13, we show that every node x makes $(\lambda \log n)$ attempts to transmit and stops executing the algorithm after $O(\Delta_x^T \log^3 n)$ time-slots, where Δ_x^T is the number of nodes in its transmission region (Def. 7.3). Finally, in Theorem 7.14, we prove that Algorithm 5 is correct and efficient, i. e., after $O(\Delta_x^T \log^3 n)$ time-slots, every node is scheduled successfully, i.e, every node performs a successful local broadcast. All claims hold with high probability.

Lemma 7.12. *Consider the execution of Algorithm 5. The sum of sending probabilities of nodes in any broadcasting region $B_x, x \in V$ at any time-slot t is upper bounded by*

$$\sum_{y \in B_x} p_y \le \frac{3}{2}, \tag{7.5}$$

with probability at least $(1 - 1/n)$.

Proof. The claim holds in the beginning of execution, since all nodes start with sending probability $1/4n$. Consider a time slot t_1, in which for the first time the sum of sending probabilities exceeds $1/2$ in one of the broadcasting regions, say B_x. We now consider the time interval $\tau = [t_1 \ldots t_1 + \delta\lceil \log n \rceil]$. We first claim that the sum of sending probabilities in the considered interval is at most $3/2$. The claim holds since (1) by choice of t_1, at the beginning of the interval the sum of sending probabilities is at most $1/2$; (2) by definition of Algorithm 5, during the specified interval each node can at most double its sending probability; and (3) there can be only less than n newly awaken nodes, which in $\delta\lceil \log n \rceil$ time slots can achieve sending probability at most $1/2n$ each, yielding

$$\sum_{v \in B_x} p_v^t \le 2 \cdot \frac{1}{2} + n \cdot \frac{1}{2n} \le \frac{3}{2}, \quad \forall t \in \tau.$$

Therefore, the following bounds hold for the entire time interval τ:

$$\frac{1}{2} \le \sum_{v \in B_x} p_v^t \le \frac{3}{2} \quad \forall t \in \tau \tag{7.6}$$

$$0 \le \sum_{\substack{v \in B_y \\ y \ne x}} p_v^t \le \frac{3}{2} \quad \forall t \in \tau. \tag{7.7}$$

The second inequality holds because t_1 is the very first time slot in which the sum of sending probabilities exceeds 1/2. Hence, in each $B_y, y \neq x$, the sum of sending probabilities is at most 3/2 in the considered time interval. (Otherwise, one of B_y would have reached 1/2 before B_x and t_1 would not be the first time slot considered).

The proof proceeds by showing that, before the claimed bound is surpassed, the sum of sending probabilities in B_x falls back to less than 1/2, since, with high probability, a DIR occurs in B_x in the considered interval. Record that a *Drastic Interference Reduction* (DIR) occurs in the broadcasting region B_x in time-slot t when all nodes $y \in B_x$ decide to reset their sending probability, which happens if every node $y \in B_x$ either makes an attempt to transmit or receives a clear message ($SINR_y \geq \beta$).

We proceed by bounding the *probabilistic interference* experienced by a node $z \in B_x$, caused by nodes located outside its proximity region A_z, in interval τ. Using (7.4), (7.7), and Lemma 7.10 ($c = 3/2$), we have

$$\begin{aligned} \psi_z^{w \notin A_z} &< \frac{3}{2} \cdot P \left(\frac{\alpha - 1}{\alpha - 2} \right) 3^3 2^{(\alpha-2)} R_A^{(2-\alpha)} R_B^{-2} \\ &= \frac{P}{4\beta(2R_B)^\alpha}. \end{aligned}$$

By *Markov inequality*, the probability that the interference at $z \in B_x$, caused by transmissions outside its proximity range, exceeds $2 \cdot \psi_z^{w \notin A_z}$ is less than 1/2. Therefore, with probability $P_{SINR \geq \beta} \geq 1/2$, the signal received by z from transmitter $v \in B_x$ can be lower bounded by

$$SINR_{z \in B_x} > \frac{\frac{P}{(d_{vz})^\alpha}}{2 \cdot \psi_z^{w \notin A_z} + N} > \beta,$$

which holds since $d_{vz} \leq 2R_B$ and ambient noise N is upper bounded by (7.1).

We proceed by calculating the probability that exactly one transmission $(v, z) \in B_x$ occurs:

$$\begin{aligned} P_{one}^{B_x} &\geq \sum_{v \in B_x} \left(p_v \cdot \prod_{\substack{w \in B_x \\ w \neq v}} (1 - p_w) \right) \\ &\geq \sum_{v \in B_x} p_v \cdot \prod_{w \in B_x} (1 - p_w) \\ &\underset{Fact\ 7.5}{\geq} \sum_{v \in B_x} p_v \cdot \left(\frac{1}{4} \right)^{\sum_{w \in B_x} p_w} \\ &\underset{Eq.(7.6)}{\geq} \frac{1}{2} \left(\frac{1}{4} \right)^{\frac{3}{2}}. \end{aligned}$$

Furthermore, we define the probability that no other node transmits in A_z:

$$\begin{aligned}
P_{none}^{A_z} &\geq \prod_{\substack{w\in A_z\\ w\neq z}} \prod_{k\in B_w} (1-p_k) \\
&\underset{Fact\ 7.5}{\geq} \prod_{\substack{w\in A_z\\ w\neq z}} \left(\frac{1}{4}\right)^{\sum_{k\in B_w} p_k} \\
&\underset{Eq.(7.7)}{\geq} \prod_{\substack{w\in B_x\\ w\neq v}} \left(\frac{1}{4}\right)^{\frac{3}{2}} \\
&\underset{Fact\ 7.7}{\geq} \left(\frac{1}{4}\right)^{\frac{3}{2}\chi^{R_A,R_B}}. \qquad (7.8)
\end{aligned}$$

Hence, the probability that a DIR occurs in one time slot is

$$\begin{aligned}
P_{DIR} &\geq P_{one}^{B_x} \cdot P_{none}^{A_z} \cdot P_{SINR\geq\beta} \\
&\geq \frac{1}{2}\cdot\frac{1}{2}\left(\frac{1}{4}\right)^{\frac{3}{2}(1+\chi^{R_A,R_B})}.
\end{aligned}$$

The probability that a DIR does not occur in the whole interval τ is

$$\begin{aligned}
\overline{P_{DIR}} &\leq \left(1-\frac{1}{4}\left(\frac{1}{4}\right)^{\frac{3}{2}(1+\chi^{R_A,R_B})}\right)^{\delta\log n} \\
&< \frac{1}{n^3},
\end{aligned}$$

where $\delta = 12\cdot 4^{3/2(1+\chi^{R_A,R_B})}$.

The argument that a DIR occurs with probability $(1-n^{-3})$ in the critical interval τ is not sufficient, since the number of such intervals could be infinitely large. However, we can bound the total number of intervals using the fact that each node maintains a counter and makes at most $(\lambda\log n)$ attempts to transmit, stopping the execution of the algorithm afterwards. Since there are n nodes, there can be at most $(n\cdot\lambda\log n)$ critical intervals τ during the entire execution of the algorithm. The probability that a DIR occurs in all such intervals is therefore

$$\begin{aligned}
P_{DIR}(\text{all } \tau\text{'s}) &\geq \left(1-\frac{1}{n^3}\right)^{n\lambda\log n} \\
&\geq \left(1-\frac{1}{n}\right).
\end{aligned}$$

□

In the following lemma we prove that the sending probability, although bounded from above as shown in Lemma 7.12, grows quickly enough, allowing each node x to make $\lambda\lceil\log n\rceil$ transmission attempts in time $O(\Delta_x^T \log^3 n)$.

Lemma 7.13. *Given the number of nodes Δ_x^T in the transmission region T_x of a node x, every node x makes $\lambda\lceil\log n\rceil$ attempts to transmit and stops executing Algorithm 5 after $O(\Delta_x^T \log^3 n)$ time-slots.*

Proof. The first observation is that, since a node x can only reset its sending probability upon reception of a clear transmission ($SINR_x \geq \beta$), the reset can only be caused by nodes within its *transmission range* R_T. Given that there are at most $(\Delta_x^T - 1)$ nodes in the transmission region T_x and that each of these nodes makes at most $\lambda\lceil\log n\rceil$ attempts to transmit, node x can reset its sending probability at most $(\Delta_x^T - 1)\lambda\lceil\log n\rceil$ times.

On the other hand, according to the definition of Algorithm 5, every node starts with sending probability $p_0 = 1/(4n)$ and doubles its sending probability after $\delta\lceil\log n\rceil$ consecutive time-slots without resets. Assuming that x does not reset its sending probability, after $\delta\lceil\log n\rceil(\lceil\log n\rceil + 2)$ time slots, x transmits with probability $p = 1$.

Putting everything together, after at most $(\Delta_x^T - 1)\lambda\delta\lceil\log n\rceil^2(\lceil\log n\rceil + 1) + \delta\lceil\log n\rceil(\lceil\log n\rceil + 2) = O(\Delta_x^T \log^3 n)$ time slots, every node makes $\lambda\lceil\log n\rceil$ attempts to transmit and halts the execution of the algorithm. □

Using Lemmas 7.12 and 7.13, we can now prove that Algorithm 5 is correct and efficient.

Theorem 7.14. *Given the number of nodes Δ_x^T in the transmission region T_x of a node x, every node x performs a local broadcast successfully after $O(\Delta_x^T \log^3 n)$ time-slots with probability at least $1 - 1/n^2$. The bound holds for all nodes with probability at least $1 - 1/n$.*

Proof. The high probability result is based on the fact that each attempt to transmit has a constant probability of success, i.e., once a node x attempts to transmit, all intended receivers $y \in B_x$ in its broadcasting region will receive the message successfully ($SINR_y \geq \beta$) with constant probability. Since each node makes $\lambda\lceil\log n\rceil$ attempts to transmit, setting λ to high enough a value gives the high probability result.

In Lemma 7.12 we proved that the sum of sending probabilities in every broadcasting region B_x is bounded by 3/2 during the entire execution of Algorithm 5 w.h.p. Using this fact we can apply Lemma 7.10 to bound the *probabilistic interference* experienced by a receiver $y \in B_x$, caused by nodes located outside its proximity range by

$$\psi_y^{w \notin A_y} < \frac{P}{4\beta(2R_B)^\alpha}.$$

As argued earlier, with probability $P_{SINR \geq \beta} \geq 1/2$, the $SINR$ at the intended receiver $y \in B_x$ can be lower bounded by

$$\begin{aligned} SINR_{y \in B_x} &\geq \frac{\frac{P}{(d_{xy})^\alpha}}{2 \cdot \psi_y^{w \notin A_y} + N} \\ &> 2^\alpha \beta \\ &> \beta. \end{aligned}$$

Using the result of Lemma 7.12, the probability that the transmission (x, y) is the only one in the proximity range of y can be calculated in the same way as in (7.8).

Putting everything together, the probability that transmission attempt is successful can be lower bounded by

$$\begin{aligned} P_{\substack{send \\ success}} &\geq P_{SINR \geq \beta} \cdot P_{none}^{A_y} \\ &\geq \frac{1}{2} \left(\frac{1}{4} \right)^{\frac{3}{2} \chi^{R_A, R_B}}. \end{aligned}$$

Applying Lemma 7.13, which states that after time $O(\Delta_x^T \log^3 n)$ node x makes $\lambda \log n$ attempts to transmit and the fact that each attempt has constant probability of success, the probability that node x does not broadcast successfully during the entire execution of Algorithm 5 is

$$\begin{aligned} P_{\substack{fail \\ send}} &\leq \left(1 - \frac{1}{2} \left(\frac{1}{4} \right)^{\frac{3}{2} \chi^{R_A, R_B}} \right)^{\lambda \lceil \log n \rceil} \\ &< \frac{1}{n^2}, \end{aligned}$$

where $\lambda = 4 \cdot 4^{(3/2) \chi^{R_A, R_B}}$. Because there are at most n nodes, the probability that the claim holds for all nodes is at least

$$\begin{aligned} P_{\substack{all \\ success}} &\geq \left(1 - \frac{1}{n^2} \right)^n \\ &\geq \left(1 - \frac{1}{n} \right). \end{aligned}$$

□

The upper bound on the execution time of the algorithm is proportional to the number of nodes Δ_x^T in the transmission range R_T of each node, which depends on the transmission power level P. Note that, since nodes aim to broadcast messages only to those receivers located within their broadcasting region B_x, and since high power levels require higher energy spending, the power level P should be chosen somehow proportional to the maximum sender-receiver distance, which is R_B. Therefore, R_T/R_B is typically bounded by a constant, and Δ_x^T remains a local property.

7.4 Lower Bound

The algorithms presented in the previous sections achieve local broadcasts in time $O(\Delta_{max}^A \log n)$ and $O(\Delta_{max}^T \log^3 n)$, respectively. We now show that this is close to optimal.

Theorem 7.15. *Both algorithms schedule all local broadcasts in time at most a poly-logarithmic factor away from the optimum.*

Proof. Consider a broadcasting region B_x and the number of nodes in it Δ_x^B. A successful broadcast corresponds to a local broadcast within radius R_B around a sender x. Since the receivers inside this area can decode the signal of only one sender at a time, the transmission can succeed only if no other node sends within this area simultaneously. This means that disks of radius R_B do not overlap in the optimum. Therefore, the optimum can schedule only one node in each broadcasting region at a time and, therefore, needs at least Δ_{max}^B time-slots to schedule all nodes, $T_{OPT} \geq \Delta_{max}^B$.

Multi-Hop Aloha and *SSMA*, on the other hand, need at most $O(\Delta_{max}^A \log n)$ and $O(\Delta_{max}^T \log^3 n)$ time-slots to schedule all broadcasts successfully with high probability. Given that $\Delta_{max}^A \leq \Delta_{max}^B \cdot \chi^{R_A,R_B}$ and $\Delta_{max}^M \leq \Delta_{max}^B \cdot \chi^{R_T,R_B}$, where χ^{R_A,R_B} and χ^{R_T,R_B} are constants defined in Fact 7.7, we have

$$\begin{aligned} T_{Aloha} &\leq T_{OPT} \cdot \chi^{R_A,R_B} \cdot O(\log n), \quad \text{and} \\ T_{SSMA} &\leq T_{OPT} \cdot \chi^{R_T,R_B} \cdot O(\log^3 n), \end{aligned}$$

i.e., our algorithms are only a poly-logarithmic factor away from the optimum. □

7.5 Simulation results

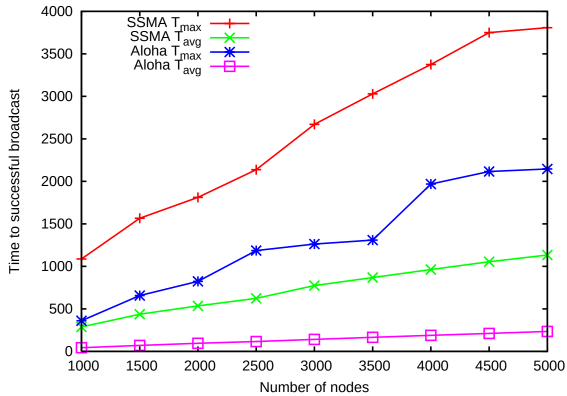
Our analytical studies show that both algorithms for local broadcasting perform provably well in worst-case scenarios. In this section we use simulations to investigate the performance in the average case, when nodes are distributed uniformly at random in the plane. Our simulations are coded in the Sinalgo [4] simulation framework, which is a packet-level wireless network simulator. The Sinalgo framework can be tuned to model a wide variety of wireless communication models, including the *physical* and the *protocol* models.

[4] http://dcg.ethz.ch/projects/sinalgo

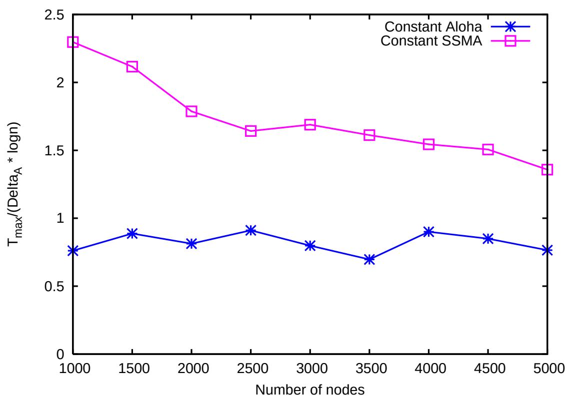
For our purposes, we used a communication model that accurately captures SINR-based signal propagation in a wireless communication environment, modeling the reception of packets according to definition (3.2). The simulations were set up on a square of area 1000×1000; the number of simulations was chosen in order to reduce the confidence interval to a meaningful value.

In Figures 7.1(a) and 7.1(b), we evaluate the average and maximum time needed for all nodes to perform a successful local broadcast. The broadcasting range was set to $R_B = 25$, and the total number of nodes was varied from $n = 1000$ to $n = 5000$. The average number of neighbors in a broadcasting region B_x ranged from $\Delta_x^B = 2$ (for $n = 1000$) to $\Delta_x^B = 10$ (for $n = 5000$). The SINR parameters used in the simulations were $\alpha = 6$ and $\beta = 1$, but as we show in Figures 7.2(a) and 7.2(b), SSMA is robust to changes in these parameters. In Figure 7.1(a), it can be seen that the number of time slots needed for a successful broadcast increases with increasing density. In Figure 7.1(b), we compare the average execution time to the asymptotic bounds presented in the analysis sections. Recall that *Multi-hop Aloha* and *SSMA* have time complexity $O(\Delta^A \log n)$ and $O(\Delta^T \log^3 n)$, respectively. The plotted lines show the hidden constants in the asymptotic bounds, i.e., the ratio of the maximum execution time T_{max} and $\Delta^A \cdot \log n$ (in the simulation of *SSMA*, the transmission range is equal to the proximity range ($\Delta^T = \Delta^A$)). The simulations suggest that, when nodes are distributed uniformly on the plane, the hidden constants are actually very small. Moreover, *SSMA* has similar performance to *Multi-hop Aloha*, even though it uses no information about network topology. Interestingly, the performance of SSMA approaches that of the simple Multi-hop Aloha more closely as the number of nodes in the system (and hence the density) increases.

In Figures 7.2(a) and 7.2(b), we analyze the influence of SINR parameters α and β on the average broadcasting time. In Figure 7.2(a), we use $\beta = 1$ and $\alpha \in \{3, 4, 5, 6\}$. In Figure 7.2(b), we use $\alpha = 6$ and $\beta \in \{1, 1.5, 2, 2.5, 3\}$. The simulations were performed on $n = 1000$ nodes, and the broadcasting range was set to $R_B = 25$. In Figure 7.2(a), it can be seen that the performance of *Multi-hop Aloha* strongly depends on the path-loss exponent α. This is due to the fact that the transmission probability is inversely proportional to the number of nodes within proximity range R_A, which decreases with higher path-loss (see Eq. 7.2). *SSMA*, on the other hand, is less sensitive to the path-loss, given that its transmission probability is not dependent on the topology of the network. In Figure 7.2(b), it can be seen that, due to the dependency of *Aloha*'s sending probability on β (see Eq. 7.2), the execution time slightly increases with increasing β. Once again, the influence of β on the performance of *SSMA* is less explicit. Overall, on average, the performance of the *SSMA* protocol was comparable to the performance of *Multi-hop Aloha*, even though the former operates without having topology knowledge.

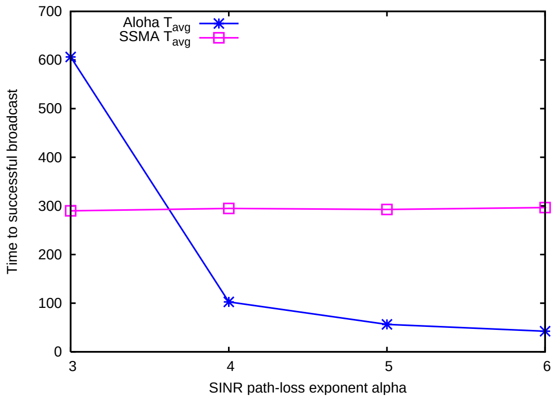


(a) Time until all nodes broadcast successfully.

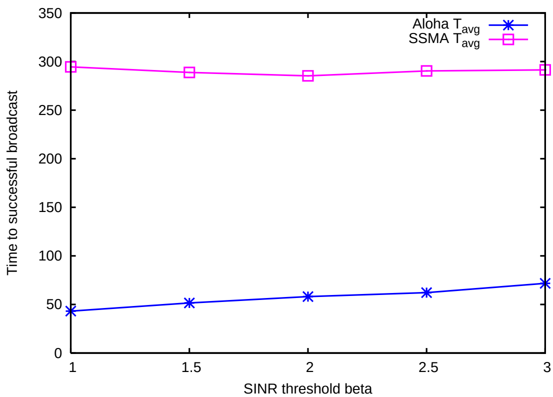


(b) Constants hidden in the Big O notation.

Figure 7.1: Simulation Results.



(a) Influence of the path-loss exponent α.



(b) Influence of the SINR threshold β.

Figure 7.2: Influence of SINR parameters.

7.6 Outlook

In this chapter we analyzed the complexity of a wireless communication primitive such as local broadcasting in the physical interference model. We looked into two distributed randomized algorithms and found out that, even when only limited knowledge about the topology is provided, close to optimum performance can be achieved in a global interference model, such as the physical model.

Our analysis reveals some important insight into the structural relationship between the protocol and physical model. In particular, we prove that if the transmission probability of each node meets certain characteristics, the performance of our algorithms can be decoupled from the global nature of the physical interference model, and each node is capable of performing a successful local broadcast in time proportional to the number of neighbors in its physical proximity. This holds regardless of the density distribution of the nodes in the network.

Chapter 8

Scheduling and Analog Network Coding

One of the key concepts on which wireless interference models rely is the definition of a successful transmission. So far, in this thesis, we have assumed that a receiver successfully decodes one, and only one, message at a time. The message that is decoded successfully is the one which was sent by the transmitter whose signal strength is the highest compared to the sum of signals of concurrently scheduled transmissions at that moment.

Analog network coding brings some revision into the assumption that wireless interference is harmful and that a receiver can decode only the strongest signal at a time. Cochannel separation techniques allow the receiver to decode several signals simultaneously under the assumption that these signals differ significantly in their strength [48, 49]. Analog network coding in a 2-way relay topology makes it possible to simultaneously decode two signals of similar strength, under the assumption that the receiver knows one of the interfered signals by having overheard or forwarded it earlier [61].

As already pointed out in the Related Work Chapter, most of the results in network coding have concentrated on capacity improvements from an information theoretic perspective [62, 69, 81, 96], and sometimes on feasibility and gains of practical solutions in restricted network topologies [61, 63, 116], but have not addressed the issue of complexity of scheduling an arbitrary set of links in the physical interference model. Does the fact that a receiver is able to decode more than one signal simultaneously make the problem easier? Does the problem remain NP-hard? Does it open possibilities for better approximation algorithms?

In this chapter we make some initial steps into the study of these issues. We analyze two models with different definitions of analog network coding. In one model (described in Section 8.1), we assume that a receiver is able to decode several signals simultaneously, provided that these signals differ in strength significantly. In the second model (described in Section 8.2), we assume that routers are able to forward the superposition of two interfering signals of nodes that wish to exchange a message, and nodes are able to decode the "collided" message by subtracting their own contribution from the interfered signal. For each network coding definition, we construct an instance of the scheduling problem in the geometric physical interference model, in which nodes are distributed in the Euclidean plane, and present NP-hardness proofs for both scenarios (see Sections 8.3 and 8.4). Moreover, in Section 8.5 we present a scheduling algorithm that explores the first definition of analog network coding. We prove that the algorithm builds a correct schedule in the physical interference model, where nodes are arbitrarily distributed in Euclidean space. Finally, in Section 8.6, we analyze the throughput gain of the algorithm in different network topologies through simulations.

Next, we define two variations of the link scheduling problem, which make use of two different definitions of a successful transmission: *Scheduling with Analog Coding by Filtering* ($SACF$) and *Scheduling with Analog Coding by Signal Mixing* ($SACSM$).

8.1 Analog Coding by Filtering ($SACF$)

In this model, we assume that a receiver r is able to decode several signals simultaneously, provided that these signals differ in strength significantly. This kind of model has been studied in the context of cochannel signal separation [48, 49].

Consider a set of concurrently scheduled links $\mathcal{S}_t$, and a subset of k signals sorted in decreasing order of power received at a node r: $\Upsilon = \{P_r(s_1), P_r(s_2), \ldots, P_r(s_k)\}$. We assume that the receiver r is able to decode all k signals in Υ if and only if the following condition holds $\forall x \in \{1, \ldots, k\}$:

$$\frac{P_r(s_x)}{\sum_{\substack{P_r(s_y)\in\Upsilon,\\ P_r(s_y)<P_r(s_x)}} P_r(s_y) + \sum_{\substack{s_z\in\mathcal{S}_t,\\ P_r(s_z)\notin\Upsilon}} P_r(s_z) + N} \geq \beta, \tag{8.1}$$

where the first component of the denominator is the accumulated interference caused by transmissions in Υ, which have weaker power level than $P_r(s_x)$; the second component of the denominator is the accumulated interference of all other concurrent transmissions in the network, which are not in Υ; N is the ambient noise; and β is the minimum SINR threshold.

The idea is that, one by one, each signal $P_r(s_x) \in \Upsilon$ can be "filtered out" from the accumulated interference, provided that the SINR between this signal and the remaining interference is above the threshold β. The key point here is that a receiver r is able to decode not only the strongest signal, as in the traditional physical interference model, but also a relatively weak signal, provided that each of the stronger signals has been filtered out. Therefore, a signal $P_r(s_x)$ can be correctly decoded if and only of all concurrently

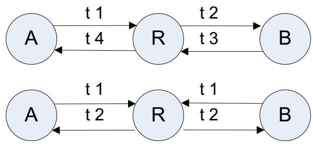


Figure 8.1: Analog network coding by signal mixing.

scheduled stronger signals ($P_r(s_y)$) obey the following constraints:

$$\frac{P_r(s_y)}{P_r(s_x) + \sum_{\substack{s_z \in \mathcal{S}_t, \\ P_r(s_z) < P_r(s_y)}} P_r(s_z) + N} \geq \beta,$$

$$\forall s_y \in \mathcal{S}_t, \text{ where } P_r(s_y) > P_r(s_x), \quad \text{and} \tag{8.2}$$

$$\frac{P_r(s_x)}{\sum_{\substack{s_z \in \mathcal{S}_t, \\ P_r(s_z) < P_r(s_x)}} P_r(s_z) + N} \geq \beta. \tag{8.3}$$

8.2 Analog Coding by Signal Mixing ($SACSM$)

The second definition of analog network coding that we analyze was introduced in [61, 120]. This model explores the fact that in a wireless network, when two packets collide, nodes often know one of the colliding packets due to having forwarded it earlier or having overheard it. Consider a situation where two nodes A and B wish to send a message to each other (see Fig. 8.1). Due to the interference of concurrent transmissions or due to the ambient noise, A and B cannot communicate directly, but only through a relay node R. Instead of scheduling 4 sequential transmissions $A \to R, R \to B, B \to R, R \to A$, as in the traditional approach, by using analog network coding, A and B can transmit simultaneously, allowing their transmissions to interfere at R. The router, not being able to decode the collided packets, can simply amplify and forward the interfered signal. It has been shown in [61] that A (as well as B) is able to decode B's packet by subtracting the contribution of its own packet from the interfered signal, even if the two transmissions are not fully synchronized and the wireless channel distorts the signals. As a result, only two time-slots are sufficient to schedule these requests.

In order for such a signal mixing to result in two successful transmissions, the following SINR conditions must hold in two time slots $t_i, t_j, j > i$:

$$\frac{P_R(A)}{\sum_{\substack{s_y \in \mathcal{S}_{t_i}, \\ s_y \notin \{A,B\}}} I_R(s_y) + N} \geq \beta, \tag{8.4}$$

$$\frac{P_R(B)}{\sum_{\substack{s_y \in \mathcal{S}_{t_i}, \\ s_y \notin \{A,B\}}} I_R(s_y) + N} \geq \beta, \tag{8.5}$$

$$\frac{P_A(R)}{\sum_{\substack{s_y \in \mathcal{S}_{t_j}, \\ s_y \neq R}} I_A(s_y) + N} \geq \beta, \tag{8.6}$$

$$\frac{P_B(R)}{\sum_{\substack{s_y \in \mathcal{S}_{t_j}, \\ s_y \neq R}} I_B(s_y) + N} \geq \beta. \tag{8.7}$$

This means that in order for A (and B) to be able to decode the mixed signal $(P_R(A) + P_R(B))$ amplified and forwarded by R in time-slot t_j, the signals received by R in time-slot t_i from both B and A must, individually, obey $SINR_{l_{AR}}(S_{t_i} \setminus \{B\}) \geq \beta$ and $SINR_{l_{BR}}(S_{t_i} \setminus \{A\}) \geq \beta$. Note that the relative signal strength of A and B does not have to exceed any threshold. In fact, it has been shown in [61] that even when $P_R(A) = P_R(B)$, the signals can still be correctly decoded by their receivers. However, note that the mixed signal sent by R must still have $SINR_{l_{RA}}(\mathcal{S}_{t_j}) \geq \beta$ and $SINR_{l_{RB}}(\mathcal{S}_{t_j}) \geq \beta$ at both receivers A and B in time-slot t_j.

For those transmissions that occur without employing signal mixing, we define a successful transmission as in standard physical interference model (see Eq. 3.2).

For the sake of simplicity, in sections 8.3 and 8.4 we set $N = 0$ and ignore the influence of noise in the calculation of $SINR$, given that this has no significant effect on the results. In Section 8.5, where we present an algorithm to solve the SACF problem, however, we do consider the case of non zero ambient noise N.

8.3 Complexity of SACSM

In this section we prove that scheduling with analog coding by signal mixing is NP-hard in the physical interference model, where nodes live in a Euclidean space (geometric SINR model).

The hardness proof is by reduction from the well known NP-complete *numerical matching with target sums* problem (NMTS) [33], which can be formulated as follows: Given 3 sets $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ of positive integers, is it possible to match each element $i \in \mathcal{A}$ to a distinct element $j \in \mathcal{B}$, such that their sum $(i + j)$ equals to each of the elements $k \in \mathcal{C}$? The triples (i, j, k) must form a partition in the sense that they are disjoint and cover $\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$.

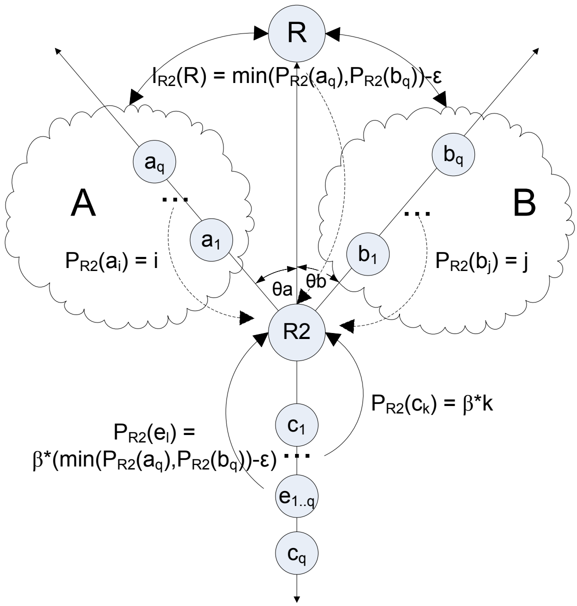


Figure 8.2: Reduction from NMTS: all $4q$ links can be scheduled successfully in $2q$ time-slots if and only if senders $a_1, \ldots, a_q, b_1, \ldots, b_q, c_1, \ldots, c_q$, are partitioned into q triples (a_i, b_j, c_k) such that $(i + j) = k$.

NMTS problem: Find q triples $triple_l = (i, j, k), i \in \mathcal{A} = \{i_1, \ldots, i_q\}, j \in \mathcal{B} = \{j_1, \ldots, j_q\}, k \in \mathcal{C} = \{k_1, \ldots, k_q\}$, such that:

$$\begin{aligned} triple_1 \cap triple_2 \cap \cdots \cap triple_q &= \emptyset, \\ triple_1 \cup triple_2 \cup \cdots \cup triple_q &= \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}, \\ (i + j) = k, \quad \forall (i, j, k) \in triple_l, & \quad \forall l \in \{1, \ldots, q\}. \end{aligned} \tag{8.8}$$

Note that, for a solution to exist, we need:

$$\sum_{i \in \mathcal{A}} i + \sum_{j \in \mathcal{B}} j = \sum_{k \in \mathcal{C}} k. \tag{8.9}$$

Theorem 8.1. *SACSM is NP-hard.*

Proof. We prove that NMTS is reducible to SACSM in polynomial time. First, we define a many-to-one reduction from any instance of NMTS to an instance of SACSM. Then, we argue that the instance of SACSM cannot be scheduled in $T \leq 2q$ time-slots, but can be scheduled in $T = 2q$ time-slots if and only if there is a solution to the NMTS problem instance.

Consider any instance of NMTS defined by $\mathcal{A} = \{i_1, \ldots, i_q\}, \mathcal{B} = \{j_1, \ldots, j_q\}, \mathcal{C} = \{k_1, \ldots, k_q\}$. The instance of SACSM is constructed by placing $(4q+2)$ nodes in the plane in the following way (see Figure 8.2). First, two nodes R and R_2 are placed at positions $(0, r(R))$ and $(0,0)$, respectively. Thereafter, q nodes, corresponding to integers $j \in \mathcal{B}$ are placed on a straight line originating at R_2 at angle $(\pi/2 - \theta_b)$; q nodes, corresponding to integers $i \in \mathcal{A}$ are placed on a line originating at R_2 at angle $(\pi/2 + \theta_a)$; and q nodes, corresponding to integers $k \in \mathcal{C}$ are placed on a line originating at R_2 at angle $-\pi/2$. The polar coordinates of each of these $3q+2$ nodes are:

$$\begin{aligned} r(R) &= \max(a_{\max}, b_{\max}) + \epsilon, \quad \theta(R) = \pi/2, \\ r(R_2) &= 0, \qquad \theta(R_2) = 0, \\ r(a_i) &= \left(\frac{1}{i}\right)^{1/\alpha}, \quad \theta(a_i) = \pi/2 + \theta_a, \ \forall i \in \mathcal{A}, \\ r(b_j) &= \left(\frac{1}{j}\right)^{1/\alpha}, \quad \theta(b_j) = \pi/2 - \theta_b, \ \forall j \in \mathcal{B}, \\ r(c_k) &= \left(\frac{1}{\beta k}\right)^{1/\alpha}, \quad \theta(c_k) = -\pi/2, \ \forall k \in \mathcal{C}, \end{aligned} \tag{8.10}$$

where ϵ is a small positive constant, and angles are defined as follows:

$$\begin{aligned} \theta_a &= \min(\theta_a^1, \theta_a^2), \quad \theta_b = \min(\theta_b^1, \theta_b^2), \text{ where} \\ \theta_a^1 &= \arccos\left(\frac{\beta^{\frac{2}{\alpha}}(a_{\min}^2 + r(R)^2) - a_{\min}^2 - r(e_l)^2}{2\beta^{\frac{2}{\alpha}} a_{\min} r(R) + 2a_{\min} r(e_l)}\right) \\ \theta_a^2 &= \arccos\left(\frac{a_{\min}^2 + r(R)^2 - \left(\frac{c_{\min} + r(R)}{\beta^{1/\alpha}}\right)^2}{2a_{\min} r(R)}\right), \\ \theta_b^1 &= \arccos\left(\frac{\beta^{\frac{2}{\alpha}}(b_{\min}^2 + r(R)^2) - b_{\min}^2 - r(e_l)^2}{2\beta^{\frac{2}{\alpha}} b_{\min} r(R) + 2b_{\min} r(e_l)}\right) \\ \theta_b^2 &= \arccos\left(\frac{b_{\min}^2 + r(R)^2 - \left(\frac{c_{\min} + r(R)}{\beta^{1/\alpha}}\right)^2}{2b_{\min} r(R)}\right), \end{aligned} \tag{8.11}$$

where $a_{\min} = (1/i_{\max})^{1/\alpha}, i_{\max} = \max_{i \in \mathcal{A}}(i)$, $b_{\min} = (1/j_{\max})^{1/\alpha}, j_{\max} = \max_{j \in \mathcal{B}}(j)$, $c_{\min} = (1/k_{\max})^{1/\alpha}, k_{\max} = \max_{k \in \mathcal{C}}(k)$, $a_{\max} = (1/i_{\min})^{1/\alpha}, i_{\min} = \min_{i \in \mathcal{A}}(i)$, $b_{\max} = (1/j_{\min})^{1/\alpha}, j_{\min} = \min_{j \in \mathcal{B}}(j)$.

Next we position the last q nodes $\{e_1, \ldots, e_q\}$ at the following location:

$$r(e_l) = \frac{r(R)}{\beta^{\frac{1}{\alpha}}}, \qquad \theta(e_l) = -\pi/2, \quad l \in \{1, \ldots, q\}. \tag{8.12}$$

The communication requests are defined as follows: nodes $\{c_1, \ldots, c_q, e_1, \ldots, e_q\}$ all demand to transmit to the same receiver R_2; nodes $\{a_1, \ldots, a_q\}$ and $\{b_1, \ldots, b_q\}$ are grouped into two groups A and B, respectively, and wish to transmit q messages $\{m(a_1), \ldots, m(a_q)\}$ from group A to group B and q messages $\{m(b_1), \ldots, m(b_q)\}$ from group B to group A. The exact recipient of a message $m(a_i)$ is not set, being enough to transmit successfully to any node $b_j \in B$. The same holds for a message $m(b_j)$, originated at node $b_j \in B$, which has to be transmitted to *any* node $a_i \in A$.[1]

Having defined the geometric instance of SACSM for any instance of NMTS, we show that it cannot be scheduled in $T < 2q$ time-slots using signal mixing analog coding.

It is enough to look at the $2q$ transmissions from nodes $\{c_1, \ldots, c_q, e_1, \ldots, e_q\}$ to receiver R_2. Given that signal mixing analog coding allows simultaneous decoding of two signals only when one of the signals is already known by the receiver, and at time $t = 0$ receiver R_2 does not know any of the considered $2q$ signals, it needs at least $2q$ time-slots to receive and successfully decode each of them.

We proceed by showing that the problem instance defined in equations (8.10) through (8.12) can be scheduled in $T = 2q$ time-slots using signal mixing analog coding if and only if there is a solution to the NMTS problem.

($\Rightarrow$) For the first part of the claim, assume we know q triples $(i, j, k), i \in \mathcal{A}, j \in \mathcal{B}, k \in \mathcal{C}$, such that conditions (8.8) through (8.9) are satisfied. To construct a $2q$-slot schedule, we assign transmissions $a_i \rightarrow R, b_j \rightarrow R, c_k \rightarrow R_2, \forall (i+j) = k$ to every odd slot $\{t_1, t_3, \ldots, t_{2q-1}\}$ (we refer to the odd-time-slot schedules as $\mathcal{S}_{2t+1}$). Note that the relay node R receives a collided signal $(P_R(a_i) + P_R(b_j))$. To every even slot $\{t_2, t_4, \ldots, t_{2q}\}$ we assign the transmissions $R \rightarrow \{a_i, b_j\}$ and $e_l \rightarrow R_2$ (we refer to the even-time-slot schedules as $\mathcal{S}_{2t}$). In this way we schedule all $4q$ requests in $2q$ time-slots. Now we prove that the obtained schedule is valid, i.e., all messages are decoded successfully.

[1] Note that the above description of the scheduling instance establishes a set-to-set communication, as opposed to point-to-point communication.

First we look at the *odd* time-slots. The SINR at receiver R_2 is equal to:

$$
\begin{aligned}
SINR_{R_2}(\mathcal{S}_{2t+1}) &= \frac{P_{R_2}(c_k)}{P_{R_2}(a_i) + P_{R_2}(b_j)} \\
&= \frac{\frac{P}{r(c_k)^\alpha}}{\frac{P}{r(a_i)^\alpha} + \frac{P}{r(b_j)^\alpha}} \\
&= \frac{P\beta k}{P(i+j)} = \beta
\end{aligned}
$$

Now we check the conditions (8.4) and (8.5):

$$
\begin{aligned}
\frac{P_R(a_i)}{\sum_{\substack{s_j \neq a_i \\ s_j \neq b_j}} I_R(s_j)} &= \frac{P_R(a_i)}{P_R(c_k)} \\
&= \frac{d(c_k, R)^\alpha}{d(a_i, R)^\alpha} \\
&= \frac{(r(R) + r(c_k))^\alpha}{(r(a_i)^2 + r(R)^2 - 2r(a_i)r(R)\cos\theta_a)^{\frac{\alpha}{2}}} \\
&\geq \frac{(r(R) + c_{\min})^\alpha}{(a_{\min}^2 + r(R)^2 - 2a_{\min}r(R)\cos\theta_a)^{\frac{\alpha}{2}}} \\
&= \beta.
\end{aligned}
\tag{8.13}
$$

The last inequality holds by plugging in the value of θ_a, defined in (8.11) (here we assume that θ_a is acute enough, s.t. $d(R, a_{\max}) > d(R, a_{\max})$.). Condition (8.5) is proved as in (8.13), using b_i instead of a_i and θ_b instead of θ_a.

Now we look at the *even* time-slots. The SINR at receiver R_2 is equal to:

$$
\begin{aligned}
SINR_{R_2}(\mathcal{S}_{2t}) &= \frac{P_{R_2}(e_l)}{P_{R_2}(R)} \\
&= \frac{\frac{P}{r(e_l)^\alpha}}{\frac{P}{r(R)^\alpha}} \\
&= \beta
\end{aligned}
$$

And finally we check the conditions (8.6) and (8.7):

$$
\begin{aligned}
\frac{P_{a_i}(R)}{\sum_{s_j \neq R} I_{a_i}(s_j)} &= \frac{P_{a_i}(R)}{P_{a_i}(e_l)} \\
&= \frac{d(e_l, a_i)^\alpha}{d(R, a_i)^\alpha} \\
&= \frac{\left(r(a_i)^2 + r(e_l)^2 - 2r(a_i)r(e_l)\cos(\pi - \theta_a)\right)^{\frac{\alpha}{2}}}{\left(r(a_i)^2 + r(R)^2 - 2r(a_i)r(R)\cos\theta_a\right)^{\frac{\alpha}{2}}} \\
&\geq \frac{\left(a_{\min}^2 + r(e_l)^2 + 2a_{\min} r(e_l)\cos\theta_a\right)^{\frac{\alpha}{2}}}{\left(a_{\min}^2 + r(R)^2 - 2a_{\min} r(R)\cos\theta_a\right)^{\frac{\alpha}{2}}} \\
&= \beta.
\end{aligned}
$$

Condition (8.7) is proved in the same way, only using b_i instead of a_i and θ_b instead of θ_a.

To sum up, we showed that in every odd time-slot, conditions (8.4) and (8.5) hold for every relay node R participating in signal mixing; in every even time-slot, conditions (8.6) and (8.7) hold for every sender a_i and b_j participating in signal network coding; every mixed packet forwarded by the relay node R can be decoded by at least one node in each group A and B, since exactly one node in every group is the sender of one of the mixed packets; and condition (3.2) holds for every transmission $\{c1, \ldots, c_q, e_1, \ldots, e_q\} \rightarrow R$ not employing network coding. This proves that our schedule guarantees successful decoding for all transmissions scheduled in each time-slot $t \in \{t_1, \ldots, t_{2q}\}$.

($\Leftarrow$) For the second part of the claim, we need to show that if no solution to the NMTS problem exists, we cannot find a $2q$-slot schedule for the SACSM instance. No solution to NMTS implies that for at least one triple $(i, j, k), i \in \mathcal{A}, j \in \mathcal{B}, k \in \mathcal{C}$, it holds that $(i + j) > k$. Assume we could still find a *valid* schedule with only $2q$ slots. As we have already pointed out, transmissions from nodes $\{c_1, \ldots, c_q, e_1, \ldots, e_q\}$ to receiver R_2 have to be scheduled sequentially. So let's assume we have q time-slots, in which senders $\{c_1, \ldots, c_q\}$ are scheduled, and another q time-slots, in which senders $\{e_1, \ldots, e_q\}$ are scheduled. We will show that there is no way to schedule the remaining senders $\{a_1, \ldots, a_q, b_1, \ldots, b_q\}$ in parallel. First we look at time-slots t with an assigned sender e_l. Assume that at least one sender a_i

(or b_j) transmits simultaneously. The SINR at R_2 would be:

$$\begin{aligned} SINR_{R_2}(\mathcal{S}_t) &= \frac{P_{R_2}(e_l)}{P_{R_2}(a_i)} \\ &= \frac{\frac{P}{r(e_l)^\alpha}}{\frac{P}{r(a_i)^\alpha}} \\ &\leq \frac{\frac{P}{\left(\frac{\max(a_{\max}, b_{\max})+\epsilon}{\beta^{\frac{1}{\alpha}}}\right)^\alpha}}{\frac{P}{(a_{\max})^\alpha}} \\ &< \beta. \end{aligned}$$

This means that all $2q$ senders $\{a_1, \ldots, a_q, b_1, \ldots, b_q\}$ have to be scheduled in the remaining q time-slots, together with senders $\{c_1, \ldots, c_q\}$. Since signal mixing analog coding only applies to 2 simultaneous transmissions, one from set A and another from set B, exactly 2 senders $\{a_i, b_j\}$ have to be scheduled in each of these q time-slots. Now consider the time slot t, correspondent to triple $(a_i, b_j, c_k) \mid (i + j > k)$. The SINR at receiver R_2 is:

$$\begin{aligned} SINR_{R_2}(\mathcal{S}_t) &= \frac{P_{R_2}(c_k)}{P_{R_2}(a_i) + P_{R_2}(b_j)} \\ &= \frac{\frac{P}{r(c_k)^\alpha}}{\frac{P}{r(a_i)^\alpha} + \frac{P}{r(b_j)^\alpha}} \\ &= \frac{P\beta k}{P(i+j)} < \beta, \end{aligned}$$

i.e., at least one transmission $c_k \to R_2$ cannot be decoded correctly within $2q$ time-slots if there is no solution to the NMTS problem. This completes the proof. □

8.4 Complexity of SACF

In this section we prove that scheduling with analog coding by filtering is also NP-hard in the geometric SINR model.

We proceed by presenting a polynomial-time reduction from *3-Partition*, a problem closely related to the *subset sum* problem. 3-Partition was proved to be NP-complete by Garey and Johnson in 1975 [32] and can be formulated as follows: Given a set $\mathcal{I}$ of integers, is it possible to partition this set into m subsets $\mathcal{I}_1, \ldots, \mathcal{I}_m$, such that the sum of the numbers in each subset is equal? The subsets $\mathcal{I}_1, \ldots, \mathcal{I}_m$ must form a partition in the sense that they are disjoint and they cover $\mathcal{I}$. Let σ denote the (desired) sum of each subset $\mathcal{I}_i$, or equivalently, let the total sum of the numbers in $\mathcal{I}$ be $m\sigma$. The 3-Partition problem remains NP-complete when every integer in $\mathcal{I}$ is strictly

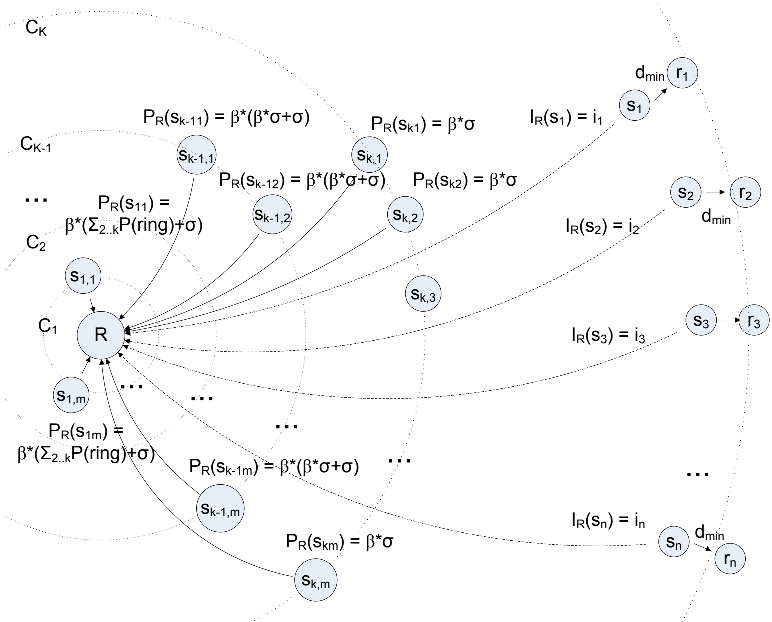


Figure 8.3: Reduction from 3-Partition: all $(K \cdot m + n)$ links can be scheduled successfully in m time-slots if and only if the senders $s_1, \ldots, s_n$, corresponding to the integers $i_1, \ldots, i_n$, are partitioned into m subsets, each summing up to exactly σ.

between $\sigma/4$ and $\sigma/2$, in which case, each subset $\mathcal{I}_i$ is forced to consist of exactly three elements [32].

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

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

Theorem 8.2. *SACF is NP-hard.*

Proof. We prove that 3-Partition is reducible to SACF in polynomial time. First, we define a many-to-one reduction from any instance of 3-Partition to a geometric (Euclidean) instance of SACF. Then, we argue that the instance of SACF cannot be scheduled in $T \leq m$ time-slots, but can be scheduled in $T = m$ time-slots if and only if the instance of 3-Partition is solved.

Consider a set $\mathcal{I} = \{i_1, \ldots, i_n\}$ of positive integers, where

$$\sum_{j=1}^{n} i_j = m \cdot \sigma, \qquad i_j < \frac{\sigma}{2}, \quad \forall i_j \in \mathcal{I}.$$

The instance of SACF is constructed by placing $(K \cdot m + n)$ senders and $(n+1)$ receivers in the plane in the following way (see Figure 8.3). First, one receiver R is placed at position $(0, 0)$. Thereafter, K circles are drawn around R, and m senders are placed on each circle's circumference. The outermost circle C_K has radius $(P/\beta \cdot \sigma)^{1/\alpha}$. Each inner circle's radius is recursively determined as

$$\begin{aligned} r(C_K) &= \left(\frac{1}{\beta \cdot \sigma}\right)^{\frac{1}{\alpha}} \qquad (8.14) \\ r(C_i) &= \beta^{\frac{1}{\alpha}} \cdot \left(\sum_{j=i+1}^{k} \frac{1}{r(C_j)^{\alpha}} + \sigma\right)^{\frac{1}{\alpha}}, \\ &\quad \forall i \in \{K-1, \ldots, 1\}. \end{aligned}$$

The polar coordinates of each of m senders $s_{i,1}, \ldots, s_{i,m}$ placed on circumference C_i are:

$$\begin{aligned} r(s_{i,j}) &= r(C_i), \qquad (8.15) \\ &\quad \forall i \in \{1 \ldots K\}, j \in \{1 \ldots m\}, \\ \theta(s_{i,j}) &\in [0, 2\pi). \end{aligned}$$

All the positioned $m \cdot K$ senders have as intended receiver the receiver R. Now we place the remaining n senders $s_1, \ldots, s_n$ and n receivers $r_1, \ldots, r_n$.

For each integer i_j in $\mathcal{I}$, we set the radial coordinate of s_j to $(P/i_j)^{1/\alpha}$ and leave its angular coordinate free.

$$\begin{aligned} r(s_i) &= \left(\frac{1}{i_j}\right)^{1/\alpha}, \quad \forall i_j \in \mathcal{I}, \\ \theta(s_i) &\in [0, 2\pi). \end{aligned}$$

Next we position the receivers $r_i, 1 \leq i \leq n$ at distance $d_{\min}$ to their corresponding senders s_i:

$$\begin{aligned} r(r_i) &= r(s_i) + d_{\min}, \quad \text{where} \\ d_{\min} &= \frac{\frac{1}{(i_{\max}-1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}}}{1 + ((n+K-1)\beta)^{\frac{1}{\alpha}}},^{2} \qquad (8.16) \\ \theta(r_i) &\in [0, 2\pi), \end{aligned}$$

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

Having defined the geometric instance of SACF for any instance of 3-Partition, we proceed by showing that it cannot be scheduled in $T < m$ time-slots using analog coding by filtering. For that, consider any pair of senders $s_{i,x}, s_{i,y}$ positioned at the same circumference C_i. Since they are equidistant from their intended receiver R, the power perceived at R is the same:

$$\frac{P_R(s_{i,x})}{P_R(s_{i,y})} = 1 < \beta, \quad \forall s_{i,x}, s_{i,y} \in C_i, i \in \{1 \dots K\}.$$

Given that the power levels of any pair of such transmissions do not differ, SINR conditions (8.2) and (8.3) cannot be fulfilled, and R cannot decode them simultaneously. Since this argument applies to any pair of senders belonging to the same circumference, and that there are m senders in each circumference, at least m time-slots are needed to schedule any m-tuple of such requests.

To proceed with the proof, we first need Lemma 8.3, in which we show that each receiver $r_i \in \{r_1, \dots, r_n\}$, corresponding to an integer $i \in \mathcal{I}$, is close enough to its respective sender to guarantee successful transmission, regardless of other links scheduled simultaneously. Since no two senders $s_{i,x}, s_{i,y}$ positioned at the same circumference C_i can be scheduled simultaneously, we assume that at most $(K+n)$ senders are scheduled in the same time-slot as r_i, i.e., one sender in each of K circumferences, plus n senders s_i, corresponding to the n integers in $\mathcal{I}$.

Lemma 8.3. *Consider a time-slot t, in which the schedule $\mathcal{S}_t$ contains $(n+K)$ senders (one sender in each of K circumferences, plus n senders s_i, corresponding to the n integers in $\mathcal{I}$). It holds that for every receiver $r_i \in \{r_1, \dots, r_n\}$, r_i decodes its message successfully, i.e., constraints (8.2) and (8.3) are satisfied.*

Proof. We start by establishing a minimal distance between a receiver $r_i \in \{r_1, \dots, r_n\}$ and any interfering server $s_j, j \neq i$ or $s_{x,y}, x \in \{1, \dots, K\}, y \in \{1, \dots, m\}$.

Since the positions of senders $s_1, \dots, s_n$ depend on the integers $i_1, \dots, i_n$, we can determine the minimum distance between two sender nodes s_i, s_j.

$$\begin{aligned}
d(s_i, s_j) &= |d(s_i, R) - d(s_j, R)| \\
&= \left| \left(\frac{1}{i_i}\right)^{\frac{1}{\alpha}} - \left(\frac{1}{i_j}\right)^{\frac{1}{\alpha}} \right| \\
&\geq \frac{1}{(i_{\max} - 1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \\
&= d_{\min} \left(1 + ((n + K - 1)\beta)^{\frac{1}{\alpha}}\right).
\end{aligned}$$

We proceed by showing that any sender $s_{x,y}$ positioned on a circumference $C_x, x \in \{1, \ldots, K\}$, is even farther away:

$$\begin{aligned} d(s_i, s_{x,y}) &= |d(s_i, R) - d(s_{x,y}, R)| \\ &\geq \frac{1}{i_{\max}^{1/\alpha}} - \frac{1}{(\beta \cdot \sigma)^{1/\alpha}} \qquad (8.17) \\ &\geq \frac{1}{(i_{\max} - 1)^{1/\alpha}} - \frac{1}{i_{\max}^{1/\alpha}} \qquad (8.18) \\ &= \min\left(d(s_i, s_j)\right), \end{aligned}$$

where (8.17) and (8.18) hold because $\sigma > 2 \cdot i_{\max}$, $\beta > 1$, and $i_{\max} \geq 1$. (i.e., $((\sigma \cdot \beta) - i_{\max}) \geq (i_{\max} - (i_{\max} - 1)))$

By triangular inequality, we have:

$$\begin{aligned} d(s_j, r_i) &\geq d(s_i, s_j) - d_{\min} \\ &= d_{\min} \cdot ((n + K - 1)\beta)^{\frac{1}{\alpha}}, \\ &\quad \forall i, j \in \mathcal{I}, i \neq j. \end{aligned}$$

This suffices to show that constraints (8.2) and (8.3) are satisfied for any receiver $r_i, i \in \{1, \ldots, n\}$. Since $d(s_j, r_i) > d_{\min} = d(s_i, r_i)$, the power received at r_i from s_i is stronger than from any other concurrent transmissions. Therefore, constraint (8.2) does not apply, and we only need to show that constraint (8.3) is satisfied:

$$\begin{aligned} \frac{P_{r_i}(s_i)}{\sum_{\substack{s_j \in \mathcal{S}_t, \\ P_{r_i}(s_j) < P_{r_i}(s_i)}} P_{r_i}(s_j)} &\geq \\ \frac{\frac{P}{d_{\min}^{\alpha}}}{(n + K - 1) \cdot \frac{P}{d(s_j, r_i)^{\alpha}}} &\geq \\ \frac{\frac{1}{d_{\min}^{\alpha}}}{\frac{(n+K-1)}{\left(d_{\min} \cdot ((n+K-1)\beta)^{\frac{1}{\alpha}}\right)^{\alpha}}} &= \beta. \end{aligned}$$

□

Having proved that successful transmission is guaranteed for receivers $r_1, \ldots r_n$ under concurrent transmission of K senders $s_{x,y}$ positioned at different circumferences $C_x, x \in \{1, \ldots, K\}$ and any number of senders $s_j, j \in \{1, \ldots, n\}$ corresponding to the integers in the 3-Partition instance, we now return to the proof of Theorem 8.2.

We claim that there exists a solution to the 3-Partition problem if and only if there exists an m-slot schedule for the problem instance defined in equations (8.14) through (8.16).

($\Rightarrow$) For the first part of the claim, assume we know m subsets $\mathcal{I}_1, \ldots, \mathcal{I}_m \subset \mathcal{I}$, whose elements sum up to σ. To construct an m-slot schedule, $\forall i_j \in \mathcal{I}_1$, we assign the corresponding sender s_j to time-slot 1, along with K senders $s_{1,1}, s_{2,1}, \ldots, s_{K,1}$. For every $i_j \in \mathcal{I}_2$, we assign the corresponding sender s_j to time-slot 2, along with K senders $s_{1,2}, \ldots, s_{K,2}$. And so on until senders s_j corresponding to $i_j \in \mathcal{I}_m$ are assigned to time slot m, along with K senders $s_{1,m}, \ldots, s_{K,m}$. In this way we scheduled all $mK + n$ requests in m time-slots. Now we prove that the obtained schedule is valid, i. e., all messages are decoded successfully.

Due to Lemma 8.3, we can assume that all senders $s_i, i \in \{1, \ldots, n\}$ transmit successfully and focus our analysis on the senders $s_{1,t}, \ldots, s_{K,t}, t \in \{1, \ldots, m\}$. Since in each time-slot t only K senders positioned on distinct circumferences are scheduled together, the situation is the same in each t. Therefore, we only look at one time-slot and show that all K transmissions are decoded successfully at receiver R.

The signal power R receives from each sender $s_{i,t}, i \in \{1, \ldots, K\}$ is equal to

$$\begin{aligned} P_R(s_{i,t}) &= \frac{P}{r(s_i)^\alpha} \\ &= \frac{P}{\beta \cdot \left(\sum_{j=i+1}^{K} \frac{1}{r(C_j)^\alpha} + \sigma\right)}. \end{aligned}$$

The interference R experiences from concurrently scheduled senders is

$$\begin{aligned} I_R(s_{i,t}) &= \sum_{j=i+1}^{K} P_R(s_{j,t}) + \sum_{s_j \in \mathcal{I}_t} P_R(s_j) \\ &= \sum_{j=i+1}^{K} \frac{P}{r(s_{j,t})^\alpha} + \sum_{s_j \in \mathcal{I}_t} P \cdot i_j \\ &= P \cdot \left(\sum_{j=i+1}^{K} \frac{1}{r(C_j)^\alpha} + \sigma \right), \end{aligned}$$

Therefore, using the notation introduced in Section 8.1, we show that condition (8.1) holds $\forall s_{x,t} \in \Upsilon = \{s_{1,t}, \ldots, s_{K,t}\}$ and, therefore, all K senders in Υ transmit successfully to receiver R in time slot t:

$$\begin{aligned} \frac{P_R(s_{x,t})}{\sum_{\substack{P_R(s_{y,t}) \in \Upsilon, \\ P_R(s_{y,t}) < P_R(s_{x,t})}} P_R(s_{y,t}) + \sum_{P_R(s_z) \notin \Upsilon} P_R(s_z)} &= \frac{P_R(s_{i,t})}{I_R(s_{i,t})} \\ &= \frac{\frac{P}{\beta \cdot \left(\sum_{j=i+1}^{K} \frac{1}{r(C_j)^\alpha} + \sigma\right)}}{P \cdot \left(\sum_{j=i+1}^{K} \frac{1}{r(C_j)^\alpha} + \sigma\right)} \\ &= \beta, \end{aligned}$$

which, in combination with Lemma 8.3, proves that our schedule guarantees successful decoding for all transmissions scheduled in each time-slot $t \in \{1, \ldots, m\}$.

($\Leftarrow$) For the second part of the claim, we need to show that if no solution to the 3-Partition problem exists, we cannot find an m-slot schedule for our scheduling instance. No solution to 3-Partition implies that for every partition of $\mathcal{I}$ into m subsets, the sum of one set $\mathcal{I}_t$ is greater than σ. Assume we could still find a schedule with only m slots. As we have already pointed out, senders positioned on the same circumference $C_i, i \in \{1, \ldots, K\}$ have to be scheduled separately. Therefore, in each time-slot $t \in \{1, \ldots, m\}$, exactly one sender positioned on each circumference C_i has to be scheduled. We argue that it is not possible to schedule n senders s_j correspondent to the integers $i_j \in \{1, \ldots, n\}$ concurrently. Consider a time-slot t, a sender $s_{K,t}$, positioned on the outermost circumference C_K, and a subset $\mathcal{I}_t$ of integers such that $\sum_{i_j \in \mathcal{I}_t} i_j > \sigma$. To prove that $s_{K,t}$'s transmission cannot be decoded correctly at receiver R, we can ignore the $(K-1)$ senders positioned on inner circumferences and only analyze the senders s_j correspondent to the integers $i_j \in \mathcal{I}_t$. We show that neither condition (8.2) nor (8.3) are satisfied at receiver R. To show that (8.2) does not hold, we observe that the ratio of the power levels of $s_{K,t}$ and s_j is always below β and, therefore, s_j's signal cannot be filtered out at receiver R, $\forall i_j \in \mathcal{I}_t$.

$$\begin{aligned}
\frac{P_R(s_j)}{P_R(s_{K,t})} &\leq \frac{\frac{P}{\left(\frac{1}{i_{\max}^{1/\alpha}}\right)^{\alpha}}}{\frac{P}{\left(\frac{1}{(\beta\sigma)^{1/\alpha}}\right)^{\alpha}}} \\
&= \frac{i_{\max}}{\beta\sigma} \\
&< \beta,
\end{aligned}$$

where the last inequality holds since $i_{\max} < \sigma/2$.

Now we show that (8.3) also does not hold, since the sum of set $\mathcal{I}_t$ is greater than σ.

$$\begin{aligned}
\frac{P_R(s_{K,t})}{\sum_{s_j \in \mathcal{I}_t} P_R(s_j)} &= \frac{\frac{P}{\left(\frac{1}{(\beta\sigma)^{1/\alpha}}\right)^{\alpha}}}{P \sum_{i_j \in \mathcal{I}_t} \frac{1}{\left(1/i_j^{1/\alpha}\right)^{\alpha}}} \\
&< \frac{\beta\sigma}{\sigma} \\
&= \beta,
\end{aligned}$$

Since neither condition (8.2) nor (8.3) are satisfied for link $s_{K,t} \rightarrow R$ when the sum of subset $\mathcal{I}_t$ is greater that σ, the transmission cannot be decoded successfully and the schedule needs more than m time slots. This completes the proof of Theorem 8.2. $\square$

8.5 Algorithm for SACF

In this section we present a scheduling algorithm that explores *analog coding by filtering* in the physical interference model. The algorithm greedily schedules links, checking for coding opportunities at each step. The result is a schedule of length T, where in each time-slot all transmissions can be decoded successfully according to equation (8.1). Note that we do not provide approximation guarantees for this algorithm, i.e., we do not know how well it performs in comparison to an optimal solution to the SACF problem. We compare its performance to scheduling algorithms that do not employ coding techniques through simulations in Section 8.6.

We start by defining a function $SACF(r, \Upsilon, I, \beta')$, which returns *true* iff a receiver r is able to decode all signals in a given set Υ. More precisely, given a set of k signals (sorted in decreasing order of power received by r) $\Upsilon = \{P_r(s_1), P_r(s_2), \ldots, P_r(s_k)\}$, a set of all other concurrent signals I, and an SINR threshold β', $SACF(l, \Upsilon, I, \beta') = true$ iff the following condition holds $\forall x \in \{1, \ldots, k\}$:

$$\frac{P_r(s_x)}{\sum_{\substack{P_r(s_y)\in\Upsilon,\\ P_r(s_y)<P_r(s_x)}} P_r(s_y) + \sum_{P_r(s_z)\in I} P_r(s_z) + N} \geq \beta'.$$

Algorithm 6 starts by setting two constants: $\beta' = 3\beta/2$, a slightly higher SINR threshold than the original β; and c, a constant defined in (8.20). The algorithm schedules links in increasing order of their length. Once a link l_x is selected to be scheduled in time-slot t (line 8), some of the remaining links l_y (those that have not been scheduled yet) are eliminated from the current time-slot in two steps. To do that, the signals which have already been scheduled in this time-slot ($l_i \in \mathcal{S}_t$) are divided into two subsets: Υ, containing signals from senders located within distance d_{yy} of receiver r_y (line 10), and I, containing signals from the remaining senders in $\mathcal{S}_t$ (line 11). In the first elimination step (line 12), all links l_y that do not meet the decoding condition $SACF(r_y, \Upsilon, I, \beta')$ *and* have an $SINR_{r_y}(\mathcal{S}_t)$ (ratio of signal to the interference from senders in $\mathcal{S}_t$, plus noise) lower than β' are removed. In the second elimination step (line 13), all links whose senders are within distance $c \cdot d_{xx}$ from receiver r_x are removed. This process is repeated until all links have been either scheduled in time-slot t or deleted. The whole process is repeated using the deleted links as input, until all links have been scheduled.

In the following theorem we prove that the schedule $\mathcal{S}$ obtained by Algorithm 6 is correct, i.e., all selected links can be scheduled concurrently without collisions using analog coding by filtering.

Algorithm 6 SACF Algorithm

input: Set of links $L = \{l_1, \ldots, l_n\}$;
output: Schedule $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$ of length T, meeting feasibility conditions SACF (8.1);
Set c according to (8.20);
$\beta' := 3\beta/2$;
$t := 0$;
repeat
    repeat
        $\mathcal{S}_t := \mathcal{S}_t \cup \{l_x\}$, where $l_x = \text{argmin}_{l_i \in L \setminus \{\mathcal{S}_t \cup D\}} d_{ii}$;
        for $l_y \in L \setminus \{\mathcal{S}_t \cup D\}$ **do**
            $\Upsilon := \{P_{r_y}(s_i), s_i \in \mathcal{S}_t \mid d(r_y, s_i) \le d_{yy}\}$;
            $I := \{P_{r_y}(s_i), s_i \in \mathcal{S}_t \setminus \Upsilon\}$;
            if $!SACF(r_y, \Upsilon, I, \beta')$ **and** $SINR_{r_y}(\mathcal{S}_t) < \beta'$ **then** $D := D \cup \{l_y\}$;
            else if $d(r_x, s_y) \le c \cdot d_{xx}$ **then** $D := D \cup \{l_y\}$;
        end for
    until $L \setminus \{\mathcal{S}_t \cup D\} = \emptyset$
    $L := L \setminus \mathcal{S}_t$;
    $D := \emptyset$;
    $t := t + 1$;
until $L = \emptyset$
return $\mathcal{S}$;

Theorem 8.4. *Algorithm 6 produces a valid schedule according to SACF feasibility conditions, defined in (8.1).*

Proof. Consider a time-slot t and an arbitrary link l_x scheduled in $\mathcal{S}_t$. Let S_x^- be the set of links shorter than l_x, i.e., those added to $\mathcal{S}_t$ before l_x, and S_x^+ be the set of links longer than l_x, i.e., those added after l_x. When a link l_x is added to the solution, two conditions hold: (1) the signal from the intended sender s_x can be decoded with SINR threshold $\beta' = 3\beta/2$, since l_x either satisfies $SACF(r_x, \Upsilon, S_x^- \setminus \Upsilon, \beta')$ or $SINR_{r_x}(S_x^-) \ge b'$, where $\Upsilon = \{P_{r_x}(s_i) \mid s_i \in \mathcal{S}_t \mid d(s_i, r_x) \le d_{xx}\}$; and (2) senders in S_x^+ are located outside the disk of radius $c \cdot d_{xx}$. It remains to show that the additional interference from S_x^+ is small enough to allow the signal from s_x to be decoded with SINR threshold β. We need to show that either $SACF(r_x, \Upsilon, \{S_x^- \setminus \Upsilon\} \cup S_x^+, \beta)$=*true* or $SINR_{r_x}(S_x^- \cup S_x^+) \ge \beta$, i.e.:

- $SACF(r_x, \Upsilon, \mathcal{S}_t \setminus \Upsilon, \beta)$ *or*
- $SINR_{r_x}(\mathcal{S}_t \setminus s_x) \ge \beta$.

In order to bound the interference from S_x^+ we use the fact that, by the second elimination criterion of the algorithm, disks of radius $c \cdot d_{jj}$ around each receiver $r_j \in S_x^+$ do not contain any sender $s_z \neq s_j$. Using this fact and the triangular inequality, we can lower bound the distance between any two senders $(s_j, s_z) \in S_x^+$ as $d(s_j, s_z) \geq d(r_j, s_z) - d_{jj} \geq c \cdot d_{jj} - d_{jj} = d_{jj}(c-1) \geq d_{xx}(c-1)$. Therefore, disks D_j of radius $d_{xx}(c-1)/2$ around senders in S_x^+ do not intersect.

We partition the space into concentric rings $Ring_k$ of width $c \cdot d_{xx}$ around the receiver r_x. Each ring $Ring_k$ contains all senders $s_j \in S_x^+$, for which $k(c \cdot d_{xx}) \leq d(s_j, r_x) \leq (k+1)(c \cdot d_{xx})$. We know that the first ring $Ring_0$ does not contain any sender. Consider all senders $s_y \in Ring_k$ for some integer $k > 0$. All disks of radius $d_{xx}(c-1)/2$ around each s_j must be located entirely in an extended ring $Ring_k$ of area

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

Since disks of area $A(D_y) \geq (d_{xx}(c-1)/2)^2\pi$ around senders in S_x^+ do not intersect, and the minimum distance between r_x and $s_y \in Ring_k, k > 0$ is $k(c \cdot d_{xx})$, 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

$$\begin{aligned} I_{r_x}(Ring_k) &\leq \sum_{s_y \in Ring_k} I_{r_x}(s_y) \\ &\leq \frac{A(Ring_k)}{A(D_y)} \frac{P}{(kcd_{xx})^\alpha} \\ &\leq \frac{(2k+1)P2^3c^2}{k^\alpha d_{xx}^\alpha c^\alpha (c-1)^2} \\ &\leq \frac{1}{k^{(\alpha-1)}} \frac{P2^5 3}{d_{xx}^\alpha c^{(\alpha)}}, \end{aligned}$$

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

$$\begin{aligned} I_{r_x}(S_x^+) &< \sum_{k=1}^{\infty} I_{r_x}(Ring_k) \\ &\leq \sum_{k=1}^{\infty} \frac{1}{k^{\alpha-1}} \frac{P2^5 3}{d_{xx}^\alpha c^{(\alpha)}} \\ &< \frac{\alpha-1}{\alpha-2} \frac{P2^5 3}{d_{xx}^\alpha c^{(\alpha)}} \\ &\leq \frac{P_{r_x}(s_x)}{3\beta}, \end{aligned} \qquad (8.19)$$

where the last two inequalities hold since $\alpha > 2$ and c is defined as follows

$$c = \max\left(2, \left(2^5 3^2 \beta \frac{\alpha-1}{\alpha-2}\right)^{\frac{1}{\alpha}}\right). \tag{8.20}$$

If we define Υ_i^+ to be the set of signals in Υ coming from senders located closer to r_x than s_i, we know that, since $SACF(r_x, \Upsilon, S_x^-, \beta')$=*true or* $SINR_{r_x}(S_x^-) \geq \beta'$, the following bounds on interference hold:

$$\begin{aligned} I_{r_x}(S_x^- \setminus \Upsilon_i^+) + N &\leq \frac{P_{r_x}(s_i)}{\beta'} \\ &\leq \frac{2P_{r_x}(s_i)}{3\beta}, \ \forall P_{r_x}(s_i) \in \Upsilon, \end{aligned}$$

in case analog coding is used, and

$$\begin{aligned} I_{r_x}(S_x^-) + N &\leq \frac{P_{r_x}(s_x)}{\beta'} \\ &= \frac{2P_{r_x}(s_x)}{3\beta}, \end{aligned} \tag{8.21}$$

in case no coding is performed. In both cases, by using the bound (8.19) on $I_{r_x}(S_x^+)$ (and the fact that $P_{r_x}(s_i) \geq P_{r_x}(s_x), \forall P_{r_x}(s_i) \in \Upsilon$), we obtain

$$\begin{aligned} I_{r_x}(\{S_x^- \setminus \Upsilon_i^+\} \cup S_x^+) + N &\leq \frac{2P_{r_x}(s_i)}{3\beta} + \frac{P_{r_x}(s_x)}{3\beta} \\ &\leq \frac{P_{r_x}(s_i)}{\beta}, \forall P_{r_x}(s_i) \in \Upsilon \\ &\Rightarrow SACF(r_x, \Upsilon, \mathcal{S}_t \setminus \Upsilon, \beta). \end{aligned}$$

for encoded transmissions, and

$$\begin{aligned} I_{r_x}(S_x^- \cup S_x^+) + N &\leq \frac{2P_{r_x}(s_x)}{3\beta} + \frac{P_{r_x}(s_x)}{3\beta} \\ &\leq \frac{P_{r_x}(s_x)}{\beta} \\ &\Rightarrow SINR_{r_x}(\mathcal{S}_t \setminus s_x) \geq \beta. \end{aligned}$$

for non-encoded transmissions. This completes the proof. □

8.6 Simulation Results

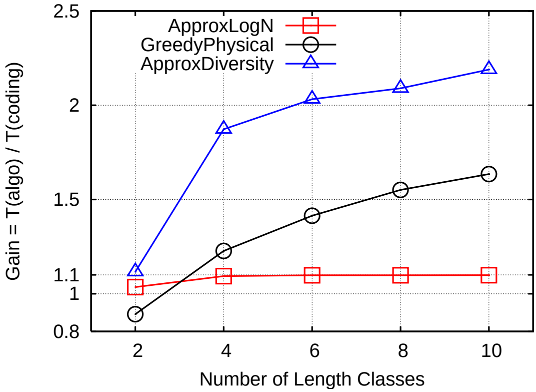
In this section we present some simulation results to illustrate the gain in throughput obtained by using analog network coding by filtering. We generated a topology, where nodes are distributed on a square field of size

$W = 1000$, and links have different levels of variance in length. More precisely, n_C length classes were defined, such that the link length l_k in each class $c_k, 1 \leq k \leq n_C$ is uniformly distributed between $l_{max} = W/2^k$ and $l_{min} = W/2^{k+1} + W/2^{k+2}$. In each length class, n/n_C receiver nodes were distributed uniformly at random in the deployment field, and the respective senders were positioned uniformly at random at distance l_k from their intended receivers. With high diversity topologies we tried to simulate scenarios, where more coding opportunities would arise.

We compare the performance of Algorithm 6 to the performance of three scheduling algorithms without coding: GreedyPhysical (proposed in [15]), ApproxDiversity (proposed in Section 5.1), and ApproxLogN (proposed in Section 6.2). All these algorithms are polynomial in time and are specifically designed for the SINR model. In all experiments, the number of simulations was chosen large enough to obtain sufficiently small confidence intervals.

Firstly, we analyze the size of the obtained schedule as a function of the number of length classes (see Figure 8.4(a)). It can be seen that the more diverse the link lengths, the more coding opportunities exist in the network, and the higher the gain of the coding approach relative to non-coding scheduling algorithms. In Figure 8.4(b) we analyze the influence of the total number of nodes on the relative performance of the algorithms. Since the number of length classes is maintained constant, the number of coding opportunities does not increase significantly. Therefore, the gain in throughput due to network coding does not vary much with varying network density. In Figure 8.5(a) we analyze the impact of the path-loss exponent α. It can be observed that when $\alpha < 3$, the GreedyPhysical algorithm achieves slightly better performance than the coding algorithm. This can be explained by the fact that the second elimination step of Algorithm 6 depends on the constant c, defined in (8.20), which increases when α approaches 2. For higher values of α, however, the coding approach becomes increasingly more efficient. In Figure 8.5(b) we analyze the impact of the SINR threshold β. It can be seen that the value of β does not influence the performances of the algorithms, which is expected, given that β is just a ratio. In Figures 8.4(a) through 8.5(b), it can be observed that the throughput gain of coding is the smallest relative to algorithm ApproxLogN. This is due to the fact that ApproxLogN outperforms the other algorithms, as demonstrated in Section 6.4. Nevertheless, the coding approach achieves gains that vary from 3.5% (when $n_C = 2$) up to 10% (when $n_C = 10$) and 20% (when $n_C = 10$ and $\alpha = 6$).

Overall, the simulation results showed that the gain of the analog coding by filtering approach depends both on the topology of the network and on the SINR parameters. The more coding opportunities a network topology generates, the more explicit the gain of the SACF algorithm over non-coding scheduling algorithms is.



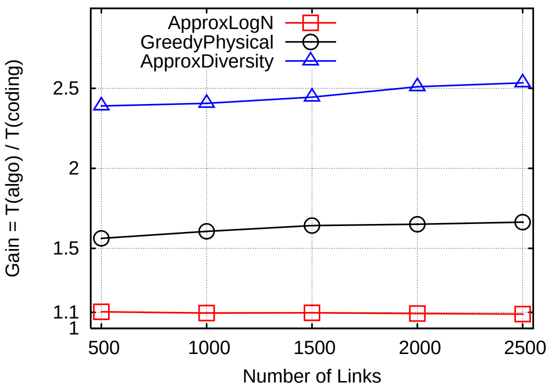


Figure 8.4: Gain obtained with SACF. ($n = 1000$ when variable n_C, $n_C = 10$ when variable n, $\alpha = 5$, $\beta = 1.2$)

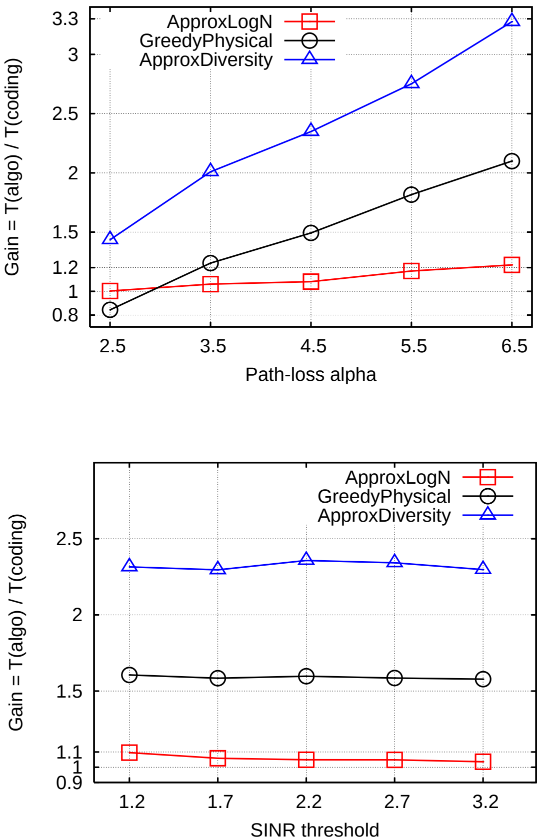


Figure 8.5: Influence of SINR parameters. ($n = 1000$, $n_C = 10$)

8.7 Outlook

Given that network coding changes the definition of a successful transmission, allowing a receiver to decode several messages simultaneously, it is interesting to analyze whether the complexity of the scheduling problem is altered. By showing that the problem remains NP-hard, we can conclude that the basic difficulties of scheduling wireless requests in a global interference model, such as geometric SINR, remain challenging even with coding capability.

Note that the 3-Partition and NMTS problems are strongly NP-hard. Therefore, as opposed to the hardness results presented in Chapter 4, the proofs presented in this chapter are of *strong* NP-hardness, i.e., the problems remain hard even when all their numerical parameters are bounded by a polynomial in the length of the input.

We also proposed a scheduling algorithm that explores analog network coding opportunities in the network. We showed through simulations that better throughput can be obtained in certain network topologies. Finding lower bounds for the scheduling problem with analog network coding, however, remains a subject of future research.

Chapter 9

Conclusions

Although wireless networks are practically omnipresent in our lives nowadays, surprisingly little is known about their algorithmic complexity and efficiency. In order to deploy a wireless network infra-structure, be it a cellular network, a WLAN, or a sensor network, the network engineers still have to rely on their know-how and build each new project from scratch, making daunting measurements and tuning endless parameter lists.

As we have pointed out in the Related Work Chapter, the research community has tried to approach wireless network problems from different perspectives. One widely used strategy has been to model the network as a graph and then apply graph-theoretic techniques to propose algorithms and present extensive theoretical analysis. Another research direction has been to use more realistic models, such as for example the physical interference model. Most of the work using this model has either consisted of heuristics and simulation-based evaluations of specific protocols, or has focused on theoretical capacity bounds of special-case networks, such as grid or random topologies. Unfortunately these results do neither give insights into the computational complexity of the problem, nor do they provide algorithmic tools that could be used to develop new protocols.

In this thesis we intended to gain a somewhat deeper understanding of wireless networks. Our goal was to promote research in this area from heuristics and special-case analysis to theoretical analysis of arbitrary instances in a more realistic network model.

Our starting point was to prove that link scheduling is an NP-hard problem in the geometric physical interference model ($SINR_G$), i.e., in a model where wireless interference is represented by the signal-to-interference-plus-noise ratio at each receiver, and nodes live in the Euclidean space. The $SINR_G$ model assumes that the envionment is unobstructed, and that the radios are perfectly omnidirectionl, which makes it a more optmistic, or weaker model than reality. This implies that one cannot compute an optimal schedule of wireless requests in practice, unless P = NP.

Since we established that the scheduling problem is unlikely to admit polynomial-time optimal solutions, we turned our attention to designing efficient approximation algorithms. In particular, we proposed the first scheduling algorithm that computes a feasible solution in the $SINR_G$ model in polynomial time with worst-case approximation guarantees for arbitrary network topologies.

The approximation ratio of our first scheduling algorithm depended on the topology of the network, and for some problem instances could become extremely bad ($\Omega(n)$), i.e., not better than the guarantees offered by the most naive solutions to the problem. Nevertheless, this undesirable dependency on the topology of the network has been inherited by a number of follow-up results in this area, as we mentioned in the Related Work Chapter. In Chapter 6 of this thesis we overcame this problem, and proposed the first scheduling algorithm with an approximation guarantee independent of the topology of the network.

Besides the basic problem of wireless link scheduling, we also looked into related problems, such as distributed algorithms and analog network coding in the physical interference model. We analyzed distributed randomized algorithms and found out that, even when only limited knowledge about the topology is provided, close to optimum performance can be achieved in a global interference model. We also studied the scheduling problem assuming that analog network coding could be used to allow a receiver to decode several messages simultaneously. We showed that the problem remains NP-hard in the geometric physical interference model, and proposed a scheduling algorithm that achieves superior throughput capacity.

To us this thesis, together with other results, represents a real breakthrough, since it has brought novel analytical tools into the study of wireless network problems. Compared to the state-of-the-art at the starting time of this work, we now have a much deeper and broader understanding of the topic, particularly of the problem of wireless link scheduling, which is a fundamental building block of the general capacity problem in wireless networks.

Many problems, however, remain open in this area, which keeps it more interesting and fervent than ever. It is still unknown, for example, whether the scheduling problem with power control is NP-hard or not. Neither are there any non-trivial lower bounds for this problem. The area of distributed algorithms in an intrinsically global model such as the physical interference model is particularly challenging, and remains wide open for future research. The topic of analog network coding also presents a lot of potential for exploration.

As opposed to the time when we started this thesis, today the problem of scheduling wireless links in the physical interference model is being a target of increasing interest and attention in the algorithmic community. As we pointed out in the last section of the Related Work Chapter, there is a lot of on-going work in this area, and many of the open problems that we have discussed might be solved in the near future. As a conclusion, we are very glad to be part of this progress and are looking forward to what is to come.

Bibliography

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] E. Arikan. Some complexity results about packet radio networks. *IEEE Transactions on Information Theory*, 30(4):681–685, Jul 1984.

[3] C. Avin, Y. Emek, E. Kantor, Z. Lotker, D. Peleg, and L. Roditty. SINR Diagrams: Towards Algorithmically Usable SINR Models of Wireless Networks. *CoRR*, abs/0811.3284, 2008.

[4] F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler. An Aloha protocol for multihop mobile wireless networks. *IEEE Transactions on Information Theory*, 52(2):421–436, 2006.

[5] S. Banerjee and A. Misra. Minimum Energy Paths for Reliable Communication in Multi-Hop Wireless Networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 146–156, 2002.

[6] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the Time-Complexity of Broadcast in Radio Networks: an Exponential Gap between Determinism and Randomization. In *Proceedings 6^{th} ACM Symp. on Principles of Distributed Computing (PODC)*, pages 98–108, 1987.

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

[8] V. Bhandari and N. H. Vaidya. Capacity of multi-channel wireless networks with random (c, f) assignment. In *Proceeding of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 229–238. ACM, 2007.

[9] V. Bhandari and N. H. Vaidya. Connectivity and Capacity of Multi-Channel Wireless Networks with Channel Switching Constraints. In *Proceeding of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 785–793, May 2007.

[10] V. Bhandari and N. H. Vaidya. Reliable Broadcast in Wireless Networks with Probabilistic Failures. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 715–723, May 2007.

[11] V. Bhandari and N. H. Vaidya. Reliable Local Broadcast in a Wireless Network Prone to Byzantine Failures. In *Proceedings of the DIALM-POMC International Workshop on Foundations of Mobile Computing (DIALM-POMC)*, 2007.

[12] 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.

[13] S. A. Borbash and A. Ephremides. The feasibility of matchings in a wireless network. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2749–2755, 2006.

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

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

[16] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Cross-layer Latency Minimization for Wireless Networks using SINR Constraints. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2007.

[17] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Approximation Algorithms for Computing Capacity of Wireless Networks with SINR Constraints. In *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2008.

[18] I. Chlamtac and S. Kutten. On Broadcasting in Radio Networks–Problem Analysis and Protocol Design. *IEEE Transactions on Communications*, 33(12):1240–1246, Dec 1985.

[19] C.-N. Chuah, D. Tse, J. Kahn, and R. Valenzuela. Capacity scaling in MIMO wireless systems under correlated fading . *Information Theory, IEEE Transactions on*, 48(3):637–650, 2002.

[20] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit Disk Graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

[21] R. L. Cruz and A. Santhanam. Optimal Routing, Link Scheduling , and Power Control in Multi-hop Wireless Networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[22] M. Dinitz and M. Andrews. Maximizing Capacity in Arbitrary Wireless Networks in the SINR Model: Complexity and Game Theory. In *Proceeding of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2009.

[23] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of Linear Coding in Network Information Flow. *IEEE Transactions on Information Theory*, 51(8):2745–2759, 2005.

[24] T. A. ElBatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad-hoc Networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[25] A. Ephremides and O. Mowafi. Analysis of a Hybrid Access Scheme for Buffered Users-Probabilistic Time Division. *IEEE Transactions on Software Engineering*, SE-8(1):52–61, Jan. 1982.

[26] A. Ephremides and T. V. Truong. Scheduling Broadcasts in Multihop Radio Networks. *IEEE Trans. Communications*, 38(4):456–460, April 1990.

[27] A. Fanghänel, T. Kesselheim, H. Räcke, and B. Vöcking. Oblivious Interference Scheduling. In *Submitted for publication*, 2009.

[28] M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro. Lower Bounds for Clear Transmissions in Radio Networks. In *Proceedings of the 7th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 447–454, Valdivia, Chile, 2006.

[29] G. Ferrari and O. K. Tonguz. Performance of ad hoc wireless networks with Aloha and PR-CSMA MAC protocol. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 2824–2829, 2003.

[30] L. Fu, S. C. Liew, and J. Huang. Power Controlled Scheduling with Consecutive Transmission Constraints: Complexity Analysis and Algorithm Design. In *Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2009.

[31] Y. Gao, J. C. Hou, and H. Nguyen. Topology control for maintaining network connectivity and maximizing network capacity under the physical model. In *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2008.

[32] M. R. Garey and D. S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, (4):397–411, 1975.

[33] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[34] L. Gasieniec, A. Pelc, and D. Peleg. The Wakeup Problem in Synchronous Broadcast Systems. *SIAM Journal on Discrete Mathematics*, 14(2):207–222, 2001.

[35] M. Gastpar and M. Vetterli. On the capacity of wireless networks: the relay case. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1577–1586, 2002.

[36] A. Giridhar and P. R. Kumar. Computing and Communicating Functions Over Sensor Networks. *IEEE Journal on Selected Areas in Communication*, 23(4), 2005.

[37] O. Goussevskaia, M. Halldórsson, R. Wattenhofer, and E. Welzl. Capacity of Arbitrary Wireless Networks. In *Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2009.

[38] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local Broadcasting in the Physical Interference Model. In *Proceedings of the DIALM-POMC International Workshop on Foundations of Mobile Computing (DIALM-POMC)*, August 2008.

[39] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in Geometric SINR. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), Montreal, Canada*, pages 100–109, September 2007.

[40] O. Goussevskaia and R. Wattenhofer. Complexity of scheduling with analog network coding. In *Proceedings of the ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC)*, May 2008.

[41] S. A. Grandhi, R. Vijayan, D. J. Goodman, and J. Zander. Centralized Power Control in Cellular Radio Systems. *IEEE Transactions on Vehicular Technology*, 42(4):466–468, 1993.

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

[43] J. Grönkvist and A. Hansson. Comparison between graph-based and interference-based STDMA scheduling. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 255–258, 2001.

[44] 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*, pages 547–566, 1998.

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

[46] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *Information Theory, IEEE Transactions on*, 34(5):910–917, 1988.

[47] M. Halldórsson and R. Wattenhofer. Wireless Communication is in APX. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP, Track C)*, 2009.

[48] J. Hamkins. Joint Viterbi algorithm to separate cochannel FM signals. In *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1998.

[49] J. Hamkins. An Analytic Technique to Separate Cochannel FM Signals. *IEEE Transactions on Communications*, 48(4):543–546, 2000.

[50] R. Hekmat and P. van Mieghem. Interference in Wireless Multi-Hop Ad-hoc Networks and Its Effect on Network Capacity. *Wireless Networks*, 10:389–399, 2004.

[51] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A Random Linear Network Coding Approach to Multicast. *IEEE Trans. on Inf. Theory*, 52(10):4413–4430, 2006.

[52] G. Holland, N. H. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-Hop wireless networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking (MOBICOM)*, pages 236–251, New York, NY, USA, 2001. ACM Press.

[53] Q.-S. Hua and F. C. M. Lau. The scheduling and energy complexity of strong connectivity in ultra-wideband networks. In *Proceedings of the 9th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2006.

[54] 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. *Journal of Algorithms*, 26, 1998.

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

[56] C. Joo, X. Lin, and N. Shroff. Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-Hop Wireless Networks. In *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2008.

[57] E.-S. Jung and N. H. Vaidya. An energy efficient MAC protocol for wireless LANs. In *Proceeding of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings (INFOCOM)*, volume 3, 2002.

[58] E.-S. Jung and N. H. Vaidya. A Power Control MAC Protocol for Ad Hoc Networks. *Wireless Networks*, 11(1-2):55–66, January 2005.

[59] T. Jurdzinski and G. Stachowiak. Probabilistic Algorithms for the Wakeup Problem in Single-Hop Radio Networks. In *Proceedings of the 13^{th} Int. Symposium on Algorithms and Computation (ISAAC)*, volume 2518 of *Lecture Notes in Computer Science*, pages 535–549, 2002.

[60] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. 1972.

[61] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: analog network coding. In *Proceedings of the SIGCOMM*, pages 397–408. ACM Press, 2007.

[62] S. Katti, I. Maric, A. Goldsmith, D. Katabi, and M. Medard. Joint Relaying and Network Coding in Wireless Networks. pages 1101–1105, June 2007.

[63] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: practical wireless network coding. In *Proceedings of the SIGCOMM*, pages 243–254, 2006.

[64] R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 62, 1939.

[65] R. Koetter and M. Medard. An Algebraic Approach to Network Coding. *IEEE/ACM Trans. on Networking*, 11(5), October 2003.

[66] C.-Y. Koo, V. Bhandari, J. Katz, and N. H. Vaidya. Reliable broadcast in radio networks: the bounded collision case. In *Proceeding of the 25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 258–264. ACM, 2006.

[67] U. Kozat and I. Koutsopoulos. Cross-layer design for power efficiency and qos provisioning in multi-hop wireless networks. *Wireless Communications, IEEE Transactions on*, 5(11):3306–3315, November 2006.

[68] U. C. Kozat and L. Tassiulas. Throughput capacity of random ad hoc networks with infrastructure support. In *Proceedings of the 9th annual international conference on Mobile computing and networking (MOBICOM)*, pages 55–65, New York, NY, USA, 2003. ACM.

[69] G. Kramer and S. Shamai. Capacity for Classes of Broadcast Channels with Receiver Side Information. In *Proceedings of the IEEE Information Theory Workshop (ITW)*, pages 313–318, 2007.

[70] S. O. Krumke, M. Marathe, , and S. Ravi. Models and Approximation Algorithms for Channel Assignment in Radio Networks. *Wireless Networks*, 6:575–584, 2000.

[71] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing Newly Deployed Ad Hoc and Sensor Networks. In *Proceedings of the 10^{th} Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 260–274, 2004.

[72] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Radio Network Clustering from Scratch. In *Proceeding of the 12nd Annual European Symposium on Algorithms (ESA)*, September 2004.

[73] V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-End Packet-Scheduling in Wireless Ad-hoc Networks. In *Proceedings of the 15th Symposium on Discrete Algorithms (SODA)*, pages 1021–1030, 2004.

[74] V. A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic Aspects of Capacity in Wireless Networks. In *ACM SIGMETRICS Performance Evaluation Review*, volume 33, pages 133–144, New York, NY, USA, 2005. ACM.

[75] E. Lebhar and Z. Lotker. Unit Disk Graph and Physical Interference Model: Putting Pieces Together. In *Proceeding of the IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2009.

[76] T.-H. Lee, J.-C. Lin, and Y. T. Su. Downlink power control algorithms for cellular radio systems. *IEEE Trans. Veh. Technol.*, 44, 1995.

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

[78] S. Li, Y. Liu, and X.-Y. Li. Capacity of large scale wireless networks under Gaussian channel model. In *Proceedings of the 14th ACM international conference on Mobile computing and networking (MOBICOM)*, pages 140–151, New York, NY, USA, 2008. ACM.

[79] S. Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *Information Theory, IEEE Transactions on*, 49(2):371–381, 2003.

[80] J. Liu, D. Goeckel, and D. Towsley. The Throughput Order of Ad Hoc Networks Employing Network Coding and Broadcasting. In *Proceeding of the IEEE Military Communications Conference (MILCOM)*, pages 1–7, Oct. 2006.

[81] J. Liu, D. Goeckel, and D. Towsley. Bounds on the Gain of Network Coding and Broadcasting in Wireless Networks. In *Proceeding of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 724–732, May 2007.

[82] 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.

[83] P. Monks, V. Bharghavan, and W. W. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In *Proceeding of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1567–1576, 2001.

[84] T. Moscibroda. The worst-case capacity of wireless sensor networks. In *Proceeding of the International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–10, 2007.

[85] T. Moscibroda, Y. A. Oswald, and R. Wattenhofer. How Optimal are Wireless Scheduling Protocols? In *Proceeding of the 26th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2007.

[86] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Proceeding of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2006.

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

[88] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology Control meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *Proceeding of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2006.

[89] J. O'Rourke. Advanced problem 6369. *American Mathematical Monthly*, 88(10):769, 1981.

[90] A. Ozgur, O. Leveque, and D. Tse. Hierarchical Cooperation Achieves Optimal Capacity Scaling in Ad Hoc Networks. *Information Theory, IEEE Transactions on*, 53(10):3549–3572, 2007.

[91] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys)*, pages 95–107, New York, NY, USA, 2004. ACM.

[92] P. Popovski and H. Yomo. Physical Network Coding in Two-Way Wireless Relay Channels. In *Proceeding of the IEEE International Conference on Communications (ICC)*, pages 707–712, June 2007.

[93] P. Popovski and H. Yomo. Wireless network coding by amplify-and-forward for bi-directional traffic flows. *IEEE Communications Letters*, 11(1):16–18, Jan. 2007.

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

[95] B. Rankov and A. Wittneben. Achievable Rate Regions for the Two-way Relay Channel. In *Proceeding of the IEEE International Symposium on Information Theory*, pages 1668–1672, July 2006.

[96] B. Rankov and A. Wittneben. Spectral efficient protocols for half-duplex fading relay channels. *IEEE Journal on Selected Areas in Communications*, 25(2):379–389, February 2007.

[97] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks. *IEEE Transactions on Networking*, 16(3):511–524, June 2008.

[98] I. Rhee, A. Warrier, J. Min, and L. Xu. DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. In *Proceeding of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 190–201. ACM Press, 2006.

[99] A. Richa, P. Santi, and C. Scheideler. An O(log n) dominating set protocol for wireless ad-hoc networks under the physical interference model. In *Proceeding of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2008.

[100] L. G. Roberts. ALOHA Packet System with and without Slots and Capture. *Computer Communications Review*, 5(2), 1975.

[101] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms (SPAA)*, pages 286–294, 2003.

[102] G. Sharma, R. R. Mazumdar, and N. B. Shroff. On the complexity of scheduling in wireless networks. In *Proceedings of the 12^{th} International Conference on Mobile Computing and Networking (MOBICOM)*, pages 227–238, 2006.

[103] G. Sharma, N. Shroff, and R. Mazumdar. Maximum weighted matching with interference constraints. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 5–74, March 2006.

[104] 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.

[105] J. So and N. H. Vaidya. Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MOBIHOC)*, pages 222–233, New York, NY, USA, 2004. ACM Press.

[106] K. Sundaresan, W. Wang, and S. Eidenbenz. Algorithmic aspects of communication in ad-hoc networks with smart antennas. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MOBIHOC)*, pages 298–309, New York, NY, USA, 2006. ACM.

[107] T. van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys)*, pages 171–180, New York, NY, USA, 2003. ACM.

[108] V. V. Vazirani. *Approximation Algorithms*. Springer, 2003.

[109] M. Vu, N. Devroye, M. Sharif, and V. Tarokh. Scaling Laws of Cognitive Networks. In *Proceedings of the International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 2–8, 2007.

[110] K. Wang, C. Chiasserini, R. Rao, and J. Proakis. A Joint Solution to Scheduling and Power Control for Multicasting in Wireless Ad Hoc Networks. *EURASIP Journal on Applied Signal Processing*, 2005.

[111] W. Wang, X.-Y. Li, O. Frieder, Y. Wang, and W.-Z. Song. Efficient interference-aware TDMA link scheduling for static wireless networks. In *Proceedings of the 12^{th} International Conference on Mobile Computing and Networking (MOBICOM)*, pages 262–273, 2006.

[112] S. Weber, X. Yang, J. Andrews, and G. de Veciana. Transmission capacity of wireless ad hoc networks with outage constraints. *Inf. Theory, IEEE Tran. on*, 51(12):4091–4102, 2005.

[113] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. *Mobile Networks and Applications (MONET)*, 7(6):481–492, 2002.

[114] Y. Wu. Broadcasting when receivers know some messages a priori. In *IEEE Int. Symp. Inf. Theory*, 2007.

[115] Y. Wu, P. A. Chou, and S.-Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. *IEEE Transactions on Communications*, 53(11):1906–1918, 2005.

[116] F. Xue, C.-H. Liu, and S. Sandhu. MAC-layer and PHY-layer Network Coding for Two-way Relaying: Achievable Regions and Opportunistic Scheduling. In *Proceedings of the 45th Annual Allerton Conference on Communication, Control and Computing*, 2007.

[117] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.

[118] J. Zander. Distributed cochannel interference control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41(3):305–311, 1992.

[119] J. Zander. Performance of optimum transmitter power control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41, 1992.

[120] S. Zhang, S. C. Liew, and P. P. Lam. Hot topic: physical-layer network coding. In *Proceedings of the 12th International Conference on Mobile Computing and Networking (MOBICOM)*, pages 358–365, 2006.

[121] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2006.

Curriculum Vitae

June 7, 1981	Born in Novosibirsk, Russia
1999–2003	B.Sc degree in computer science, Federal University of Minas Gerais, Brazil
2003–2005	M.Sc degree in computer science, Federal University of Minas Gerais, Brazil
2005–2009	Ph.D student and research assistant, Distributed Computing Group, Prof. Dr. Roger Wattenhofer, Swiss Federal Institute of Technology (ETH Zurich), Switzerland
July 2, 2009	Ph.D degree, Distributed Computing Group, Swiss Federal Institute of Technology (ETH Zurich), Switzerland
	Advisor: Prof. Dr. Roger Wattenhofer
	Co-examiners: Prof. Dr. Nitin Vaidya, University of Illinois at Urbana-Champaign, USA Dr. Stephan Eidenbenz, Los Alamos National Laboratory, USA

Publications

In the following, all publications written as a Ph.D student at ETH Zurich are listed.

1. Efficiency of Wireless Networks: Approximation Algorithms for Physical Interference Models. Olga Goussevskaia, Yvonne Anne Pignolet, and Roger Wattenhofer. *Under submission.*

2. Capacity of Arbitrary Wireless Networks. Olga Goussevskaia, Magnús Halldórsson, Roger Wattenhofer, and Emo Welzl. In *Annual IEEE Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, April 2009.

3. From Web to Map: Exploring the World of Music. Olga Goussevskaia, Michael Kuhn, Michael Lorenzi, and Roger Wattenhofer. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Sydney, Australia, December 2008.

4. Exploring Music Collections on Mobile Devices. Olga Goussevskaia, Michael Kuhn, and Roger Wattenhofer. In *International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, Amsterdam, Netherlands, September 2008.

5. Local Broadcasting in the Physical Interference Model. Olga Goussevskaia, Thomas Moscibroda, and Roger Wattenhofer. In *ACM SIGACT-SIGOPT International Workshop on Foundations of Mobile Computing (DialM-POMC)*, Toronto, Canada, August 2008.

6. Complexity of Scheduling with Analog Network Coding. Olga Goussevskaia and Roger Wattenhofer. In *ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC)*, Hong Kong, China, May 2008.

7. Layers and Hierarchies in Real Virtual Networks. Olga Goussevskaia, Michael Kuhn, and Roger Wattenhofer. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Silicon Valley, California, USA, November 2007.

8. Complexity in Geometric SINR. Olga Goussevskaia, Yvonne Anne Oswald, and Roger Wattenhofer. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, Montreal, Canada, September 2007.