Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master Thesis

Interference and Topology Control in Ad-Hoc Networks

Pascal von Rickenbach

Prof. Roger Wattenhofer
Aaron Zollinger

Distributed Computing Group
Institute for Pervasive Computing
ETH Zürich

8th September 2003 - 7th March 2004

Contents

Abstract

Topology control in ad-hoc networks tries to lower node energy consumption by reducing transmission power and by confining interference, collisions, and consequently retransmissions. In contrast to most of the related work, we assume two intuitive definitions of interference, outgoing and incoming interference, respectively. In the field of outgoing interference characteristics of minimum interference topologies are studied and a local algorithm is proposed constructing an interference-optimal spanner of a given network.

Incoming interference is considered by means of one-dimensional networks. For a simple topology, referred to as *exponential node chain*, a scanline algorithm is presented. In addition, we propose a greedy algorithm for a more general network model.

In a third part, we consider incoming interference in cellular networks, which is formalized introducing the *Minimum Membership Set Cover* optimization problem. We prove that in polynomial time the optimal solution of the problem cannot be approximated more closely than with a factor $\ln n$. On the other hand we present an algorithm exploiting linear programming relaxation techniques which asymptotically matches the lower bound with high probability.

Acknowledgements

I am deeply indebted to my advisors Prof. Roger Wattenhofer and Aaron Zollinger, whose constant support has provided me with inspiration and motivation throughout the thesis. It has been a great pleasure to learn from their profound scientific knowledge and to experience their enthusiasm. Furthermore, my thanks go to all members of the *Distributed Computing Group* and particularly to my fellow master student Thomas Moscibroda for contributing to the friendly and pleasant ambiance within the group.

Chapter 1

Introduction

In mobile wireless ad-hoc networks—formed by autonomous devices communicating by radio—energy is one of the most critical resources. The main goal of *topology control* is to reduce node power consumption in order to extend network lifetime. Since the energy required to transmit a message increases at least quadratically with distance, it makes sense to replace a long link by a sequence of short links. On the one hand, energy can therefore be conserved by abandoning energy-expensive long-range connections, thereby allowing the nodes to reduce their transmission power levels. On the other hand, reducing transmission power also confines interference, which in turn lowers node energy consumption by reducing the number of collisions and consequently packet retransmissions on the media access layer. Dropping communication links however clearly takes place at the cost of network connectivity: If too many edges are abandoned, connecting paths can grow unacceptably long or the network can even become completely disconnected. Topology control can therefore be considered a trade-off between energy conservation and interference reduction on the one hand and connectivity on the other hand.

In contrast to most of the related work done in the field of topology control algorithms—where the interference issue is seemingly solved by sparseness arguments of the resulting topologies—, we assume an explicit notion of interference. We thereby focus on two concepts of interference stated in [4]. In the first part of the thesis we focus on a definition of interference, referred to as outgoing interference, that is based on the natural question, how many nodes are affected by communication over a certain link. By prohibiting specific network edges, the potential for communication over high-interference links can then be confined.

We employ the outgoing interference definition to formulate the trade-off between energy conservation and network connectivity. In particular we state certain requirements that need to be met by the resulting topology. Among these requirements are connectivity (if two nodes are—possibly indirectly—connected in the given network, they should also be connected in the resulting topology) and the spanner property (the shortest path between any pair of nodes on the resulting topology should be longer at most by a constant factor than the shortest path connecting the same pair of nodes in the given network). After stating such requirements, an optimization problem can be formulated to find the topology meeting the given requirements with minimum outgoing interference.

For the requirement that the resulting topology should retain connectivity of the given network, we show that all currently proposed topology control algorithms—already by having every node connect to its nearest neighbor—commit a substantial mistake: Although certain proposed topologies are guaranteed to have low degree yielding a sparse graph, outgoing interference becomes asymptotically incomparable with the interference-minimal topology. We also show that there exist graphs for which no local algorithm can approximate the optimum. With respect to the sometimes desirable requirement that the resulting topology should be planar, we show that planarity can increase outgoing interference.

Furthermore we propose a distributed local algorithm (LocaLISE) that computes a provably interference-optimal topology, if we require the resulting topology to be a spanner with a given stretch factor.

Our results are not confined to worst-case considerations; we also show by simulation that on average-case graphs traditional topology control algorithms—in particular the Gabriel Graph and the Relative Neighborhood Graph—fail to effectively reduce interference. Moreover these constructions are shown to be outperformed by the LocaLISE algorithm, which therefore proves to be average-case effective in addition to its worst-case optimality.

We then switch to the second notion of interference defined in [4], referred to as incoming interference, that is based on the question, how many nodes affect a particular node by transmitting to their farthest neighbors. Again, due to the prohibition of specific network edges, the potential for highly interfered nodes can be confined. Consequently, an optimization problem can be formulated to find the topology with minimum incoming interference for the requirement that the resulting topology should retain connectivity of the given network.

Different from the outgoing interference model, incoming interference—as shown in [4]—is not of such friendly nature. We therefore turn our attention to one-dimensional network instances since already such instances can yield outgoing interference $\Omega(n)$.

We first investigate interference-optimal topologies in an ideal one-dimensional network, referred to as *exponential node chain*. It is shown that incoming interference can be lower-bounded to $\sqrt{n}$ in such instances. Furthermore we propose an algorithm (LION) following the scan-line principle, that asymptotically matches this lower bound. Then a more general model, referred to as highway model, is assumed, where nodes are arbitrarily distributed in one dimension. An attempt to transfer algorithm LION to this model is presented. However, an example shows that such efforts do not appear to be successful. A presentation of a greedy algorithm (GLOW) that appears to be a good heuristic for interference reduction for instances in the highway model, since it is asymptotically optimal in the case of exponential node chains, concludes the analysis of incoming interference in ad-hoc networks.

Based on the insights derived from this investigation, the last part of the thesis focuses on incoming interference in cellular networks. More precisely, the interference at the clients caused by the base stations of a cellular network.

1.1 Related Work

In this section we discuss related work in the field of topology control with special focus on the issue of interference.

1.1.1 Topology Control

The assumption that nodes are distributed randomly in the plane according to a uniform probability distribution formed the basis of pioneering work in the field of topology control [10, 31].

Later proposals adopted constructions originally studied in computational geometry, such as the Delaunay Triangulation [11], the minimum spanning tree [28], the Relative Neighborhood Graph [16], or the Gabriel Graph [29]. Most of these contributions mainly considered energy-efficiency of paths preserved by the resulting topology, whereas others exploited the planarity property of the proposed constructions for geometric routing [3, 19].

The Delaunay Triangulation and the minimum spanning tree not being computable locally and thus not being practicable, a next generation of topology control algorithms emphasized locality. The CBTC algorithm [34] was the first construction to focus on several desired properties, in particular being an energy spanner with bounded degree. This process of developing local algorithms with more and more properties was continued partly based on CBTC, partly based on local versions of classic geometric constructions such as the Delaunay Triangulation [21] or the minimum spanning tree [20]. One of the most recent such results is a locally computable planar distance (and energy) spanner with constant-bounded node degree [33]. Another thread of research takes up the average-graph perspective of early work in the field; [2] for instance shows that the simple algorithm choosing the k nearest neighbors works surprisingly well on such graphs.

Yet another aspect of topology control is considered by algorithms trying to form clusters of nodes. Most of these proposals are based on (connected) dominating sets [1, 13] and focus on locality and provable properties, such as [18], which achieves a non-trivial approximation of the minimum dominating set in constant time. Cluster-based constructions are commonly regarded a variant of topology control in the sense that energy-consuming tasks can be shared among the members of a cluster.

Topology control having so far mainly been of interest to theoreticians, first promising steps are being made towards exploiting the benefit of such techniques also in practical networks [17].

1.1.2 Interference

As mentioned earlier, reducing interference—and its energy-saving effects on the medium access layer—is one of the main goals of topology control besides direct energy conservation by restriction of transmission power. Astonishingly however, all the above topology control algorithms at the most implicitly try to reduce interference. Where interference is mentioned as an issue at all, it is maintained to be confined at a low level as a consequence to sparseness or low degree of the resulting topology graph.

A notable exception to this is [23] defining an explicit notion of interference. Based on this interference model between edges, a time-step routing model and a concept of congestion is introduced. It is shown that there are inevitable trade-offs between congestion, power consumption and dilation. For some node sets, congestion and energy are even shown to be incompatible.

The interference model proposed in [23] is based on current network traffic. The amount and nature of network traffic however is highly dependent on the chosen application. A layered networking architecture—where topology control would take place at a low layer—would therefore be broken by topology control taking into account traffic information to reduce interference. Since usually no a priori information about the traffic in a network is available, a static model of interference depending solely on a node set is therefore desirable.

That is where [4] enters the scene, which provides a basis of this thesis. It discusses in-depth various possible interference definitions depending only on a node set. Furthermore, a classification of different models is given and relations among these models are studied. One of the main differences among the models considered in [4] is whether they focus on outgoing or incoming interference. In addition, an interference-optimal algorithm (GLIT) is proposed in the outgoing interference model for the requirement that the resulting topology should retain connectivity of the given network.

Chapter 2

Modeling Interference

Mobile ad-hoc networks are commonly modeled by graphs. A graph $G = (V, E)$ consists of a set of nodes $V \subset \mathbb{R}^2$ in the Euclidean plane and a set of edges $E \subseteq V^2$. Nodes represent mobile hosts, whereas edges represent links between nodes. In order to prevent already basic communication between directly neighboring nodes from becoming unacceptably cumbersome [26], it is required that a message sent over a link can be acknowledged by sending a corresponding message over the same link in the opposite direction. In other words, only *undirected* edges are considered.

We assume that a node can adjust its transmission power to any value between zero and its maximum power level. The maximum power levels are not assumed to be equal for all nodes. An edge (u, v) may exist only if both incident nodes are capable of sending a message over (u, v), in particular if the maximum transmission radius of both u and v is at least $|u, v|$, their Euclidean distance. A pair of nodes u, v is considered *connectable in the given network* if there exists a path connecting u and v provided that all transmission radii are set to their respective maximum values. The task of a *topology control* algorithm is then to compute a subgraph of the given network graph with certain properties, reducing the transmission power levels and thereby attempting to reduce interference and energy consumption.

In [4] several interference models for this kind of graphs are discussed in detail. In the following two of them are briefly introduced, since they form the basis for the next two chapters.

With a chosen transmission radius—for instance to reach a node v—a node u affects at least all nodes located within the circle centered at u and with radius $|u, v|$. $D(u, r)$ denoting the disk centered at node u with radius r and requiring edge symmetry, the *coverage* of an (undirected) edge $e = (u, v)$ is consequently defined to be the cardinality of the set of nodes covered by the disks induced by u and v:

$$\begin{aligned} Cov(e) :=& \big|\{w \in V | w \text{ is covered by } D(u, |u, v|)\} \cup \\ & \{w \in V | w \text{ is covered by } D(v, |v, u|)\}\big|. \end{aligned}$$

In other words the coverage $Cov(e)$ represents the number of network nodes affected by nodes u and v communicating with their transmission powers chosen such that they exactly reach each other (cf. Figure 2.1). This is also referred to as the environment of e.

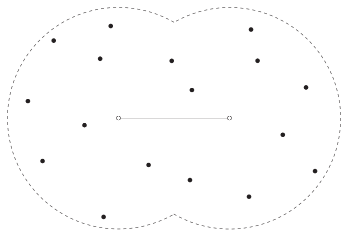

Figure 2.1: Nodes covered by a communication link.

The edge level interference defined so far, also referred to as outgoing interference in [4], since interference is counted at the causing edge, is now extended to a graph interference measure as the maximum coverage occurring in a graph:

Definition 1. *The outgoing interference of a graph G=(V,E) is defined as*

$$I_{out}(G) := \max_{e \in E} Cov(e).$$

On the other hand, each node u features a value r_u defined as the distance from u to its farthest neighbor. Based on this, [4] introduces an alternative interference measure also referred to as incoming interference since interference is counted at the interfered node:

Definition 2. *The incoming interference of a graph G=(V,E) is defined as*

$$I_{in}(G) := \max_{v \in V} |\{u | v \in D(u, r_u)\}|.$$

In other words the interference I_{in} represents the maximum number of disks induced by the maximum transition ranges of the nodes covering a particular network node. I_{in} of a node is analogously defined as the number of disks covering that node.

Since interference reduction per se would be senseless (if all nodes simply set their transmission power to zero, interference will be reduced to a minimum), the formulation of additional requirements to be met by a resulting topology is necessary. A resulting topology can for instance be required

- to maintain connectivity of the given communication graph (if a pair of nodes is connectable in the given network, it should also be connected in the resulting topology graph),

- to be a spanner of the underlying graph (the shortest path connecting a pair of nodes u, v on the resulting topology is longer by a constant factor only than the shortest path between u and v on the given network), or
- to be planar (no two edges in the resulting graph intersect).

Finding a resulting topology which meets one or a combination of such requirements with minimum interference constitutes an optimization problem.

Chapter 3

Outgoing Interference

In this chapter we consider the outgoing interference model I_{out} defined in Chapter 2. For this model [4] describes an algorithm, referred to as GLIT, that results in an optimum interference topology given the requirement for maintaining connectivity of the given graph. In the following we first discuss some properties of such an interference-optimal topology in the I_{out} model. Afterwards two algorithms are described that yield interference-optimal topologies with the additional requirement of being a spanner of the given network.

3.1 Interference-Optimal Topologies

In [4] It is shown that an optimum interference topology does not always contain the *Minimum Spanning Tree* (MST) of the given network. Moreover a worst-case example was presented that yields interference $I_{out} \in \Omega(n)$, where n denotes the number of nodes in the networks, in case of the MST, whereas an interference-optimal topology results in constant interference. Thus, a topology containing the MST is not always optimal. Using the same example as the one introduced in [4] (see Figure 3.1) we are however able to derive a much stronger conclusion than the one stated above.

To the best of our knowledge, all currently known topology control algorithms as described in Section 1.1 have in common that every node establishes a (symmetric) connection to at least its nearest neighbor. In other words all these topologies contain the Nearest Neighbor Forest constructed on the given network. In the following we show that by including the Nearest Neighbor Forest as a subgraph, the interference of a resulting topology can become incomparably bad with respect to a topology with optimum interference.

Theorem 1. *No currently proposed topology control algorithm—required to maintain connectivity of the given network—is guaranteed to yield a non-trivial interference approximation of the optimum solution. In particular,* *interference of any proposed topology is $\Omega(n)$ times larger than the interference of the optimum connected topology, where n is the total number of network nodes.*

Proof. Figure 3.1 depicts an extension of the exponential node chain (see [4]. In addition to a horizontal exponential node chain, each of these nodes h_i has a corresponding node v_i vertically displaced by a little more than h_i's distance to its left neighbor. Denoting this vertical distance d_i, $d_i > 2^{i-1}$ holds. These additional nodes form a second (diagonal) exponential line. Between two of these diagonal nodes v_{i-1} and v_i, an additional helper node t_i is placed such that $|h_i, t_i| > |h_i, v_i|$.

The Nearest Neighbor Forest for this given network (with the additional assumption that each node's transmission radius can be chosen sufficiently large) is shown in Figure 3.2. Roughly one third of all nodes being part of the horizontally connected exponential chain, interference of any topology containing the Nearest Neighbor Forest amounts to at least $\Omega(n)$. An interference-optimal topology, however, would connect the nodes as depicted in Figure 3.3 with constant interference. □

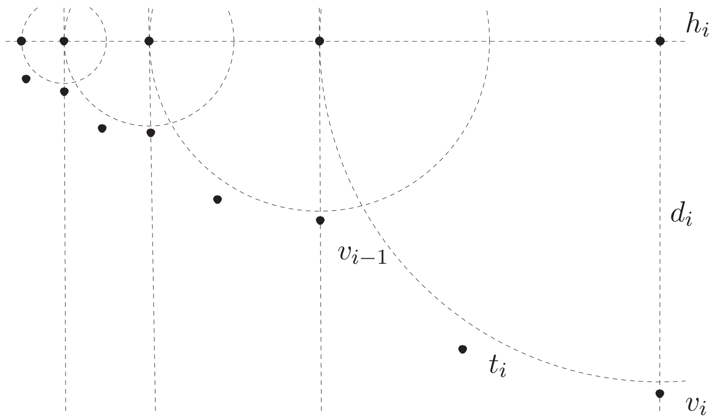

Figure 3.1: Two exponential node chains.

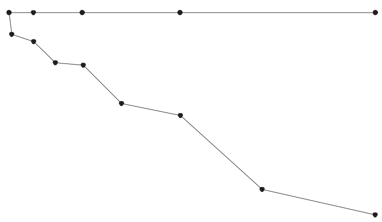

Figure 3.2: The Nearest Neighbor Forest yields interference $\Omega(n)$.

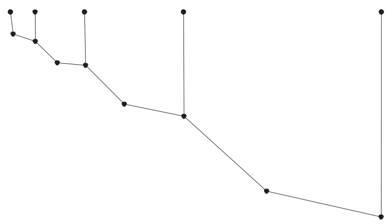

Figure 3.3: Optimal tree with constant interference.

In other words, already by having each node connect to the nearest neighbor, a topology control algorithm makes an "irrevocable" error. Moreover, it commits an asymptotically worst possible error, since the interference in any network cannot become larger than n.

Since roughly one third of all nodes are part of the horizontal exponential node chain in Figure 3.1, the observation stated in Theorem 1 would also hold for an average interference measure, averaging interference over all edges.

The following theorem even shows that for connectivity-preserving topologies no local algorithm can approximate optimum interference for every given network.

Theorem 2. *For the requirement of maintaining connectivity of the given network, there exists a class of graphs for which there is no local algorithm that approximates optimum interference.*

Proof. In Figure 3.4 the maximum transmission radius of a node is $|u, v|$. Let n be the number of nodes in the graph. Then the shaded area contains $\Omega(n)$ evenly distributed nodes which can be connected with constant interference. For each such node i the inequalities $|i, v| < |u, v|$ and $|u, i| > |u, v|$ hold. It follows that edge (u, v) has $\Omega(n)$ interference, since it covers all nodes in the shaded area. In addition there is a chain of nodes (dashed path) connecting node u with node v indirectly through the nodes located in the shaded area. The nodes in the chain are located in such a way that it is possible to connect them with constant interference. For such a graph $O(1)$ interference can be achieved by connecting u to the rest of the graph through the chain of nodes and not directly through edge (u, v), which would cause $\Omega(n)$ interference.

A local algorithm at node u has to decide if it can drop edge (u, v) or not. This is only possible if u knows about the existence of an alternative path from u to v in order to maintain connectivity. By elongating the chain sufficiently, the local algorithm can thus be forced to include edge (u, v), pushing up interference to $O(n)$ whereas the optimum is $\Omega(1)$. □

In addition to the properties shown above, we now prove that an optimum interference topology features also bounded degree. This requirement is often desired in order to save resources at the nodes.

Theorem 3. *Algorithm GLIT resulting in an interference-optimal topology for any given network has bounded degree at most 12.*

Proof. Assume a network consisting of three interconnected nodes $\{u, v, w\}$. Since GLIT follows the lines of Kruskal's MST algorithm [6] with attributed edge weight $Cov(e)$ for an edge e, we know that the algorithm discards the edge with maximum coverage. We therefore prove that two adjacent edges in an optimum interference topology enclose an angle greater than $2\pi/13$, from which the theorem follows. Without loss of generality, we assume $|u, v|$

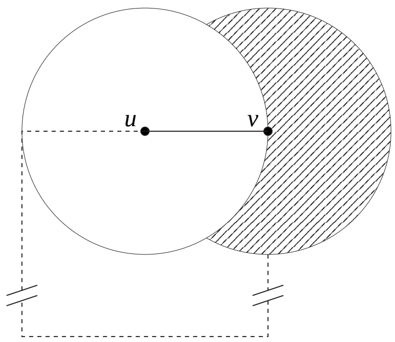

Figure 3.4: Worst case graph for which no local algorithm can approximate optimum interference.

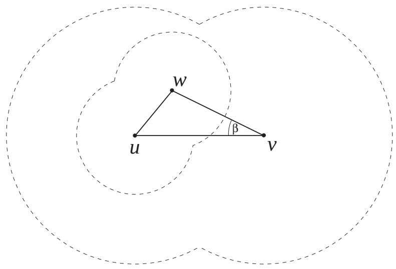

Figure 3.5: The interfering area of (u, w) is within that of (u, v) and thus $Cov(u, w) \leq Cov(u, v)$.

to be greater than $|v, w|$. In order for the edge (u, w) to be discarded in an optimum interference topology, it is required that $Cov(u, w) \geq Cov(u, v)$ holds. Figure 3.5 depicts a case where the environment of (u, w) is entirely inside the one of (u, v) and thus by definition $Cov(u, w)$ cannot be greater than $Cov(u, v)$. Consequently, we can lower-bound the angle β in Figure 3.5 such that the environment of (u, w) is not entirely within that of (u, v). In case of $|u, v| = |v, w|$ it can be seen that $|u, w| \leq |u, v|/2$ is required, in order to make $Cov(u, w) \geq Cov(u, v)$ possible. Setting $|u, w| = |u, v|/2$ it follows that

$$\sin \frac{\beta}{2} = \frac{\frac{|u,w|}{2}}{|u, v|} = \frac{1}{4},$$

and consequently $\beta \geq 2 \cdot \sin^{-1}(1/4) > 2\pi/13$. □

Additionally, it can be shown that the upper bound is tight, since there exist network instances that yield node degree 12 in an interference-optimal topology. As mentioned in Section 2, another popular requirement for topology control algorithms besides bounded degree is planarity of the resulting topology. This is often desired, because numerous well-understood routing algorithms exist that are only applicable to planar graphs. But topology control algorithms enforcing planarity are not optimal in terms of interference:

Theorem 4. *There exist graphs on which interference-optimal topologies—required to maintain connectivity—are not planar.*

Proof. In Figure 3.6 the maximum transmission radius of a node is $|a, b|$. All eligible edges are depicted together with the coverage area for edges whose incident nodes are both in $\{a, b, c, d\}$. The indicated weight of an edge e corresponds to its coverage $Cov(e)$. V and W represent sets of 3 and 4 nodes, respectively. The nodes in set V and W, respectively, can be connected among themselves with interference 3. A topology control algorithm can only reduce interference by removing all edges with maximum interference (here (a, c) and (b, c)) from the graph. Thereafter, no further edge can be removed without breaking connectivity, since the graph without (a, c) and (b, c) is a tree. Thus the resulting tree is interference-optimal and non-planar, since both edges (a, b) and (c, d) must remain in the resulting topology. □

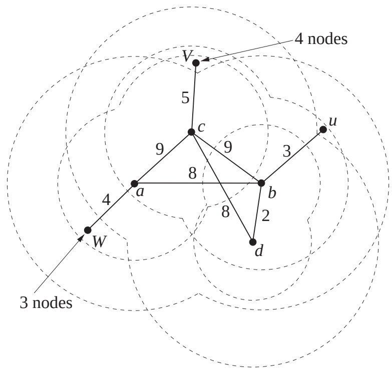

Figure 3.6: Node set whose interference-optimal topology is not planar.

3.2 Low-Interference Topologies

In this section we present two algorithms that explicitly reduce outgoing interference of a given network. They both compute an interference-optimal topology maintaining connectivity of the given network with the additional requirement of being a spanner of the network. Whereas the first spanner algorithm assumes global knowledge of the network, the second can be computed locally.

3.2.1 Low-Interference Spanners

The algorithm GLIT as defined in [4] optimizes interference for the requirement that the resulting topology has to maintain connectivity. In addition to connectivity it is often desired that the resulting topology should be a spanner of the given network. A formal definition of a t-spanner follows:

Definition 3 (t-Spanner). *A t-spanner of a graph $G = (V, E)$ is a subgraph $G' = (V, E')$ such that for each pair (u, v) of nodes $|p^*_{G'}(u, v)| \leq t \cdot |p^*_G(u, v)|$, where $|p^*_{G'}(u, v)|$ and $|p^*_G(u, v)|$ denote the length of the shortest path between u and v in G' and G, respectively.*

In this section we consider Euclidean spanners, that is, the length of a path is defined as the sum of the Euclidean lengths of all its edges. With slight modifications our results are however also extendable to hop spanners, where the length of a path corresponds to the number of its edges.

Algorithm LISE is a topology control algorithm that constructs a t-spanner with optimum interference. LISE starts with a graph $G_{LISE} = (V, E_{LISE})$ where E_{LISE} is initially the empty set. It processes all eligible edges of the given network $G = (V, E)$ in descending order of their coverage. For each edge $(u, v) \in E$ not already in E_{LISE}, LISE computes a shortest path from u to v in G_{LISE} provided that the Euclidean length of this path is less than or equal to $t\,|u, v|$. As long as no such path exists, the algorithm keeps inserting all unprocessed eligible edges with minimum coverage into E_{LISE}.

To prove the interference optimality of G_{LISE}, we introduce an additional lemma, which shows that G_{LISE} contains all eligible edges whose coverage is less than $I(G_{LISE})$.

Lemma 5. *The graph $G_{LISE} = (V, E_{LISE})$ constructed by LISE from a given network $G = (V, E)$ contains all edges e in E whose coverage $Cov(e)$ is less than $I(G_{LISE})$.*

Proof. We assume for the sake of contradiction that there exists an edge e in E with $Cov(e) < I(G_{LISE})$ which is not contained in E_{LISE}. Consequently, LISE never takes an edge with coverage $Cov(e)$ in line 7, since the algorithm would insert all edges with $Cov(e)$ into E_{LISE} in line 8 instantly (thus also e). There exists however an edge f in E_{LISE} with $Cov(f) = I(G_{LISE})$ eventually taken in line 7. Therefore the inequality $Cov(e) < Cov(f)$ holds. At the time the algorithm takes f in line 7, all edges taken in line 5 must have had coverage greater than or equal to $Cov(f)$, since the maximum of an ordered set can only be greater than or equal to the minimum of the same set. Hence e has never been taken in line 5 and therefore has never been removed from E in line 10. Consequently, e is still in E when f is taken as the edge with minimum coverage in E. Thus it holds that $Cov(f) \leq Cov(e)$ which leads to a contradiction. □

With Lemma 5 we are ready to prove that the resulting topology constructed by LISE is an interference-optimal t-spanner.

Theorem 6. *The graph $G_{LISE} = (V, E_{LISE})$ constructed by LISE from a given network $G = (V, E)$ is an interference-optimal t-spanner of G.*

Proof. To show that G_{LISE} meets the spanner property, it is sufficient to prove that for each edge $(u, v) \in E$ there exists a path in G_{LISE} with length not greater than $t\,|u, v|$. This holds, since for a shortest path $p^*(u, v)$ in G a path $p'(u, v)$ in G_{LISE} with $|p'| \leq t\,|p|$ can be constructed by substituting each edge on p with the corresponding spanner path in G_{LISE}. For edges in

Low Interference Spanner Establisher (LISE)

Input: V, a set of nodes v, each of which having attributed a maximum transmission radius r_v^{max}

1: E = all eligible edges (u, v) ($r_u^{max} \geq |u, v|$ and $r_v^{max} \geq |u, v|$) (∗ unprocessed edges ∗)
2: $E_{LISE} = \emptyset$
3: $G_{LISE} = (V, E_{LISE})$
4: **while** $E \neq \emptyset$ **do**
5: $e = (u, v) \in E$ with maximum coverage
6: **while** $|p^*(u, v)$ in $G_{LISE}| > t\,|u, v|$ **do**
7: f = edge $\in E$ with minimum coverage
8: move all edges $\in E$ with coverage $Cov(f)$ to E_{LISE}
9: **end while**
10: $E = E \setminus \{e\}$
11: **end while**

Output: Graph G_{LISE}

E which also occur in E_{LISE} the spanner property is trivially true. On the other hand an edge (u, v) can only be in E but not in E_{LISE} if a path from u to v in G_{LISE} with length not greater than $t\,|u, v|$ exists (see if-condition in line 6). Thus G_{LISE} is a t-spanner of G.

Interference optimality of LISE can be proved by contradiction. We therefor assume, that G_{LISE} is not an interference-optimal t-spanner. Let $G^* = (V, E^*)$ be an interference-optimal t-spanner for G. Since G_{LISE} is not optimal, it follows that $I(G_{LISE}) > I(G^*)$. Thus all edges in E^* have coverage strictly less than $I(G_{LISE})$. From Lemma 5 follows that E^* is a nontrivial subset of E_{LISE}. Let T be the set of edges in E_{LISE} with coverage $I(G_{LISE})$ and $\tilde{G} = (V, \tilde{E})$ the graph with $\tilde{E} = E_{LISE} \setminus T$. $\tilde{G}$ is a t-spanner, since E^* is still a subset of $\tilde{E}$, and $I(\tilde{G}) \leq I(G_{LISE}) - 1$ holds. Because T is eventually inserted into E_{LISE} in line 8, there exists an edge $(u, v) \in E$ that was taken in line 5 and for which no path $p(u, v)$ exists in $\tilde{G}$ with $|p| \leq t\,|u, v|$. Thus $\tilde{G}$ is no t-spanner (and therefore also G^*), which contradicts the assumption that G^* is an interference-optimal t-spanner. □

As regards the running time of LISE, it computes for each edge at most one shortest path. This holds, since multiple shortest path computations for the same edge in line 6 cause at least as many edges to be inserted into E_{LISE} in line 8 without computing shortest paths for them. Since finding a shortest alternative path for an edge requires $O(n^2)$ time and as the network contains at most the same amount of edges, the overall running time of LISE is as well polynomial in the number of network nodes.

In contrast to the problem of finding a connected topology with optimum interference, the problem of finding an interference-optimal t-spanner is locally solvable. The reason for this is that finding an interference-optimal path $p(u, v)$ for an edge (u, v) with $|p| \leq t\,|u, v|$ can be restricted to a certain neighborhood of (u, v).

In the following we describe a local algorithm similar to LISE that is executed at all eligible edges of the given network. In reality, algorithm LocaLISE (Local LISE) is executed for each edge by one of its incident nodes (for instance the one with the higher identifier). The description of LocaLISE assumes the point of view of an edge $e = (u, v)$. The algorithm consists of three main steps:

1) Collect $(\frac{t}{2})$-neighborhood,

2) compute minimum interference path for e, and

3) inform all edges on that path to remain in the resulting topology.

In the first step, e gains knowledge of its $(\frac{t}{2})$-neighborhood. For a Euclidean spanner, the k-neighborhood of e is defined as all edges that can be reached (or more precisely at least one of their incident nodes) over a path p starting at u or v, respectively, with $|p| \leq k\,|e|$. Knowledge of the $(\frac{t}{2})$-neighborhood at all edges can be achieved by local flooding.

During the second step a minimum-interference path p from u to v with $|p| \leq t\,|e|$ is computed. LocaLISE starts with a graph $G_{LL} = (V, E_{LL})$ consisting of all nodes in the $(\frac{t}{2})$-neighborhood and an initially empty edge set. It inserts edges consecutively into E_{LL}—in ascending order according to their coverage—, until a shortest path $p^*(u, v)$ is found in G_{LL} with $|p^*| \leq t\,|e|$.

In the third step, e informs all edges on the path found in the second step to remain in the resulting topology. The resulting topology then consists of all edges receiving a corresponding message. In the following we show that it is sufficient for e to limit the search for an interference-optimal path $p(u, v)$ meeting the spanner property to the $(\frac{t}{2})$-neighborhood of e.

Lemma 7. *Given an edge $e = (u, v)$, no path p from u to v with $|p| \leq t\,|e|$ contains an edge which is not in the $(\frac{t}{2})$-neighborhood of e.*

Proof. For the sake of contradiction we assume that a path p from u to v with $|p| \leq t\,|e|$ containing at least one edge (w, x) not in the $(\frac{t}{2})$-neighborhood of e. Without loss of generality we further assume that, traversing p from u to v, we visit w before x. Since (w, x) is not in the $(\frac{t}{2})$-neighborhood, by definition, no path from u to w with length less than or equal to $(\frac{t}{2})|e|$ exists (the same holds for any path from v to x). Consequently, the inequality $|p| > t\,|e| + |(w, x)|$ holds, which contradicts the assumption that $|p| \leq t\,|e|$. □

With Lemma 7 we are now able to prove that the topology constructed by LocaLISE is a t-spanner with optimum interference.

LocaLISE

1: collect $(\frac{t}{2})$-neighborhood $G_N = (V_N, E_N)$ of $G = (V, E)$
2: $E = \emptyset$
3: $G' = (V_N, E')$
4: **repeat**
5: $\quad f$ = edge $\in E_N$ with minimum coverage
6: $\quad$ move all edges $\in E_N$ with coverage $Cov(f)$ to E'
7: $\quad p$ = shortestPath$(u - v)$ in G'
8: **until** $|p| \leq t\,|u, v|$
9: inform all edges on p to remain in the resulting topology.

Note: $G_{LL} = (V, E_{LL})$ consists of all edges eventually informed to remain in the resulting topology.

Theorem 8. *The graph $G_{LL} = (V, E_{LL})$ constructed by LocaLISE from a given network $G = (V, E)$ is an interference-optimal t-spanner of G.*

Proof. The spanner property of LocaLISE can be proven similar to the first part of the proof of Theorem 6, where LISE is shown to be a t-spanner.

To show interference optimality, it is sufficient to prove that the spanner path constructed for any edge $e = (u, v) \in G$ by LocaLISE is interference-optimal, where interference of a path is defined as the maximum interference of an edge on that path. The reason for this is that only edges that lie on one of these paths remain in the resulting topology; non-optimality of G_{LL} would therefore imply non-optimality of at least one of these spanner paths. In the following we look at the algorithm executed by $e = (u, v)$. In line 6 edges in E are consecutively inserted into E', starting with $E' = \emptyset$, until a spanner path p from u to v is found in line 8. Since LocaLISE inserts the edges into E' in ascending order according to their coverage and p is the first path meeting the spanner property, p is an interference-optimal t-spanner path from u to v in the $(\frac{t}{2})$-neighborhood. From Lemma 7 we know that the $(\frac{t}{2})$-neighborhood of e contains all spanner paths from u to v and therefore also the interference-optimal one. Thus it is not possible that LocaLISE does not see the global interference-optimal t-spanner path due to its local knowledge about G. Consequently, p is the global interference-optimal t-spanner path of e. □

3.3 Average-Case Interference

In this section we consider interference of topology control algorithms on average-case graphs, that is on graphs with randomly placed nodes.

In particular networks were constructed by placing nodes randomly and uniformly on a square field of size 20 by 20 units and subsequently computing for each node set the Unit Disk Graph—defined such that an edge exists if and only if its Euclidean length is at most one unit. The resulting Unit Disk Graphs were then employed as input networks for topology control. Since node density is a fundamental property of networks with randomly placed nodes, the networks were generated over a spectrum of node densities.

3.3.1 Connectivity-Preserving Topologies

To evaluate connectivity-preserving topologies on average-case graphs, two well-known topology control algorithms are considered, in particular the Gabriel Graph [9] and the Relative Neighborhood Graph [32]. The interference-reducing effect of these two constructions is considered by comparison with the interference value of the given Unit Disk Graph network on the one hand and with the interference-optimal connectivity-preserving topology on the other hand. The interference-optimal topology was constructed by means of the GLIT algorithm presented in [4].

Figure 3.7 shows the interference mean values over 1000 networks for each simulated network density. While the resulting interference curves behave similarly for very low network densities, they fall into three groups with increasing density: At a density of roughly 5 network nodes per unit disk the interference-optimal curve stagnates and remains at a value of approximately 11.5. On the other hand the interference curve of the Unit Disk Graph without topology control rises almost linearly. Between these two extremes the Gabriel Graph and Relative Neighborhood Graph values increase clearly more slowly than the Unit Disk Graph curve, but show significantly higher values than the interference-optimal topology.

The simulation results show that the edge reduction performed by the Gabriel Graph and Relative Neighborhood Graph constructions reduce interference of the given network; this effect is clearer with the Relative Neighborhood Graph due to its stricter edge inclusion criterion and consequently its being a subgraph of the Gabriel Graph. However, the interference values of these two constructions are considerably higher than the results of the interference-optimal connectivity-preserving topology. Furthermore, although (unless in special cases) the Relative Neighborhood Graph has degree at most 6, it is not even clear whether with increasing network density the respective interference curve remains around the maximum value found so far or whether it would increase further for densities beyond the simulated spectrum. It can therefore be concluded that also for average-case graphs sparseness does not imply low interference.

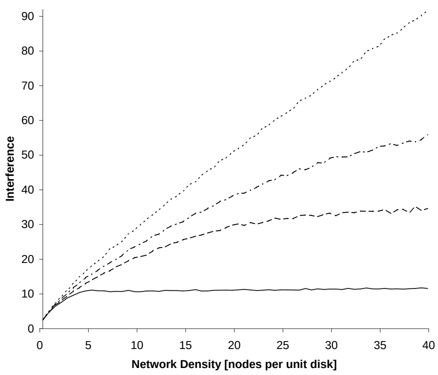

Figure 3.7: Interference values of the Unit Disk Graph without topology control (dotted), the Gabriel Graph (dash-dotted), the Relative Neighborhood Graph (dashed), and the interference-optimal connectivity-preserving topology (solid).

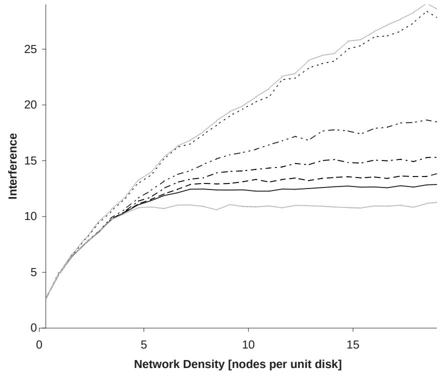

Figure 3.8: Interference values of LISE for stretch factors 2 (dotted), 4 (dash-dot-dotted), 6 (dash-dotted), 8 (dashed), and 10 (solid). Interference values of the Relative Neighborhood Graph (upper gray) and interference-optimal connectivity-preserving topology (lower gray) are plotted for reference.

3.3.2 Low Interference Spanners

Going beyond connectivity-preserving topologies, we consider in this section spanners, that is topologies guaranteeing that the shortest paths on the resulting topology are only by a constant factor longer than on the given network (cf. Section 3.2.1).

Figure 3.8 depicts simulation results—in particular the mean interference values over 100 networks at each simulated network density—of the topology constructed by the LISE algorithm introduced in Section 3.2.1 for different stretch factors t. The simulation results show that by increasing the requested stretch factor it is possible to achieve interference values close to the optimum interference values caused by connectivity-preserving topologies as described in the previous section. Moreover, even with a low stretch factor of 2, LISE does not perform worse than the Relative Neighborhood Graph, which is *not* a spanner. In summary, the simulation results show that the LocaLISE algorithm performs well with respect to interference also on average-case graphs. An illustration of the simulation graphs is provided in Figure 3.9.

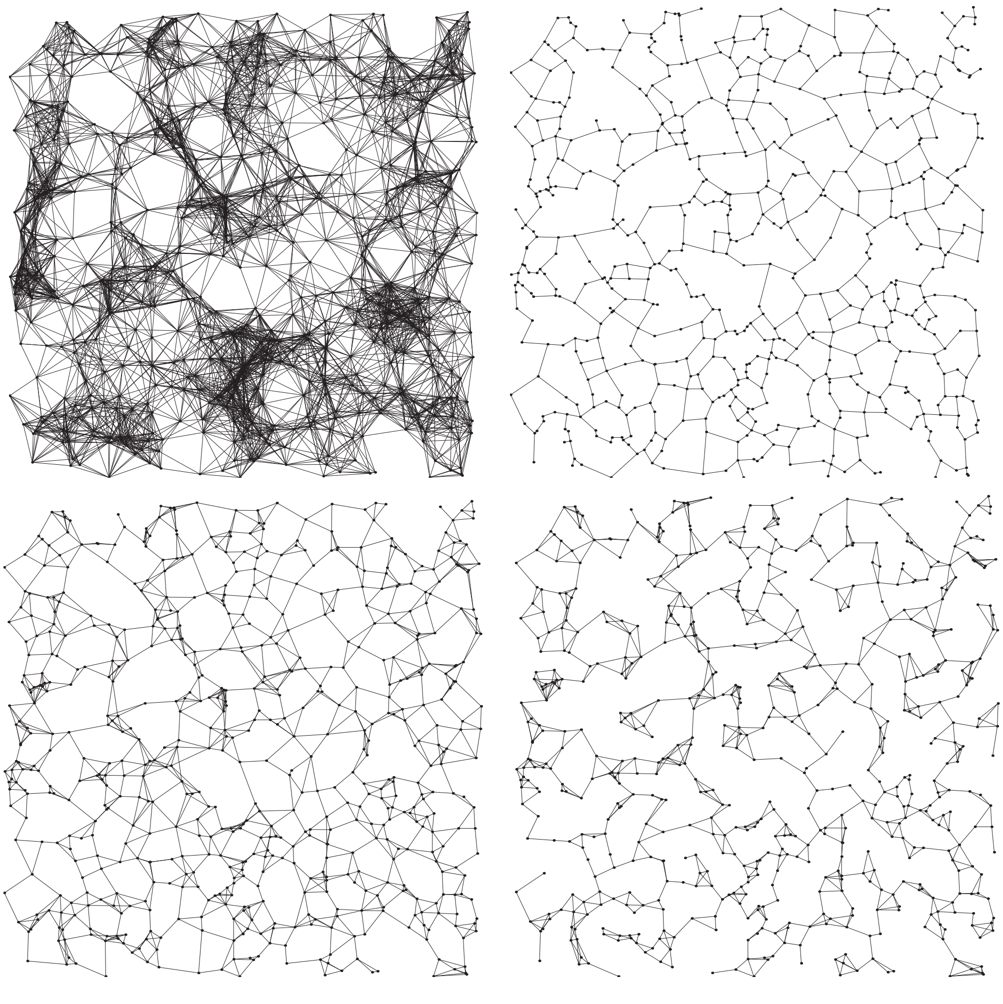

Figure 3.9: The Unit Disk Graph G (top left, interference 50), the Relative Neighborhood Graph of G (top right, interference 25), G_{LL} computed by LocaLISE with stretch factor 2 (bottom left, interference 23) and 10 (bottom right, interference 12) at a network density of 20 nodes per unit disk on a square field of 10 units side length. Note that, for instance in the western region of the graph, LocaLISE—depending on the chosen stretch factor—omits high-interference "bridge" edges if alternative spanning paths exist.

3.4 Conclusion

Based on the work presented in [4], we focus on the characteristics of interference-optimal topologies in the outgoing interference model. In particular we first show that such a topology cannot be computed locally and that even the inclusion of the Nearest Neighbor Forest in the resulting graph results in unnecessary high interference. Further we prove that an optimum interference topology in fact has bounded degree but does not lead to planar graphs.

In addition, we propose provenly interference-minimal connectivity-preserving and spanner constructions. A locally computable version of the interference-minimal spanner construction can even be considered practicable, since it is shown to significantly outperform previously suggested topology control algorithms also on average-case graphs.

Chapter 4

Incoming Interference

In this chapter we consider the incoming interference model introduced in Definition 2 of Chapter 2. In fact, [4] does not present I_{in} to be the only incoming interference model. Also an edge-based incoming interference model is described, referred to as I_{in}^e. Given a graph $G = (V, E)$ the interference I_{in}^e of a node v in V is defined to be the number of edges in E covering v with their environments. This is exactly the inverse definition of I_{out}. In the following we show the relation between I_{in} and I_{in}^e.

We therefor consider a graph with $I_{in}^e = x$. Let u be a node with $I_{in}^e(u) = x$. That is, there are x edges whose environments cover u. The cardinality of the set S of adjacent nodes to these edges is then at most $2x$ because each edge contributes at most two disjoint nodes to S. Since only nodes in S are candidates to contribute to $I_{in}(u)$, we derive $I_{in}(u) \leq 2x$.

On the other hand, we consider a graph with $I_{in} = y$. Let u be a node with $I_{in}(u) = y$. By definition u is covered by y disks. Since we claim y to be the interference of the graph, each node corresponding to one of the y disks has at most y incident edges—[4] shows that the degree of a node is a lower bound for its I_{in} value. Since only edges incident to one of these y nodes need to be considered for I_{in}^e of the graph, we obtain $I_{in}^e \leq y^2$.

Summing up the above presented results we derive the following relation for the I_{in} and the I_{in}^e interference models:

$$\sqrt{I_{in}^e} \leq I_{in} \leq 2I_{in}^e.$$

Thanks to the above relation we do not need to consider both models in this chapter, since results for one of them are also applicable for the other. In the following we therefore restrict ourselves to the I_{in} model. The reason is that I_{in} is the more natural model, since interference is obviously caused by sending nodes and not through imaginary edges.

The I_{in} model is however, other than the I_{out} interference model defined in Chapter 2 and covered in Chapter 3, not of such friendly nature. This can be seen from the *Minimum Interference Broadcast* problem presented in [4], which is optimally solvable in the I_{out} model but turns out to be NP-complete in I_{in}.

Based on the observation in [4] that already one-dimensional network instances yield optimum interference $\Omega(n)$ in the I_{out} model, we turn our attention to topologies in one dimension. Additionally, we again require the resulting topology to maintain connectivity of the given network. A topology graph meeting this requirement can therefore consist of a tree of the given network, since additional edges might unnecessarily increase interference. Thus we focus on trees maintaining connectivity of the given network with least possible interference.

4.1 Exponential Node Chains

Let an *exponential node chain* be a one-dimensional configuration of n nodes where the distance between two nodes v_i and v_{i+1} is 2^i and node v_1 is the leftmost node of the chain. Furthermore, we assume the maximum transmission radius of each node is sufficiently large in order to connect to every other node in the chain. In the I_{out} interference model exponential node chains inherently yield interference $\Omega(n)$ [4]. Figure 4.1 depicts an exponential node chain consisting of 5 *linearly connected* nodes, where "connecting linearly" means that node v_i is connected to node v_{i+1} for all $i = 1, ..., n-1$ in the resulting topology. In addition to the disk $D(v_i, r_{v_i})$ for each node v_i, Figure 4.1 depicts their interference values $I_{in}(v_i)$. Since all but the disk of the rightmost node cover v_1, interference at the latter is in $\Omega(n)$ and thus also I_{in} is in $\Omega(n)$. However, other than in the I_{out} model, linear connection in exponential node chains does not result in an interference-optimal topology in the I_{in} interference model.

Due to the construction of an exponential node chain, only nodes connecting to at least one node to their right increase v_1's interference. Consequently, a hub of an exponential node chain is defined as follows:

Definition 4. *Given a connected topology for an exponential node chain C. A node u is defined to be a hub in C if and only if there exists an edge (u, v) with v being a node to the right of u in C.*

Algorithm LION constructs a topology for an exponential node chain that does not yield interference $\Omega(n)$. The algorithm starts with a graph $G = (V, E_{LION})$, where V is the set of nodes in the chain and E_{LION} is initially the empty set. Following the scan-line principle, it processes all nodes in the order of their occurrences from left to right. Initially, the leftmost node is set to be the current hub h. Then for each node v_i LION inserts an edge (h, v_i) into E_{LION}. This is repeated until I_{in} increases due to the addition of such an edge, node v_i becomes the current hub and subsequent nodes are connected to v_i as long as interference does not increase.

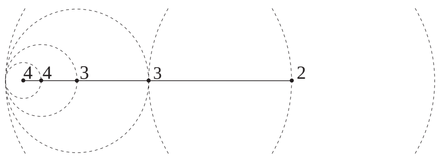

Figure 4.1: Linearly connecting an exponential node chain results in interference $\Theta(n)$ at the leftmost node.

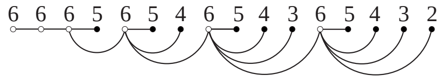

Figure 4.2: Exponential node chain in a logarithmic scale. The topology is obtained by applying LION. Only hubs (hollow points) interfere with the leftmost node.

Figure 4.2 depicts the resulting topology if LION is applied to an exponential node chain. The exponential node chain is thereby depicted in a logarithmic scale[1]. In order to clarify the resulting topology and to prevent overlapping edges, they are depicted as arcs. In addition, Figure 4.2 also shows the individual interference values at each node.

Theorem 9. *Given an exponential node chain consisting of n nodes, applying algorithm LION to this chain results in a connected topology with interference $I_{in} \in \mathrm{O}(\sqrt{n})$.*

Proof. The resulting topology obtained by application of LION shows a clear structure (see Figure 4.2). Each hub, not taking into account the first two, is connected to one more node to its right than its predecessor hub to the left. This follows from the fact that if the current topology leads to interference $I_{in} = I$ at the determination of a new hub, this hub can be connected to $I - 2$ nodes to its right until I_{in} is again increased by one. Therefore the minimum number of nodes n needed in an exponential node chain such that interference $I_{in} = I$ is obtained, when LION is applied, is

$$n = \sum_{i=1}^{I-2} i + 2 = \frac{1}{2}I^2 - \frac{3}{2}I + 3.$$

By solving for interference $I_{in} = I$ and $n \geq 2$ in the above equation, we consequently obtain

$$I_{in} = \left\lfloor \frac{\sqrt{8n-15}+3}{2} \right\rfloor \in \mathrm{O}\big(\sqrt{n}\big).$$

□

This is an intriguing result since it can be shown that $\sqrt{n}$ is a lower bound for I_{in} in exponential node chains.

[1] Another way to look at it is as if the exponential node chain was viewed through a pair of glasses with logarithmic cut.

Low Interference on Exponential No Chains (LION)

Input: V, a set of nodes vs forming an exponential node chain

$E_{LION} = \emptyset$
$G_{LION} = (V, E_{LION})$
$h = v_1$ (∗ current hub ∗)
$I_{cur} = 1$ (∗ current interference ∗)
for $i = 2$ to n **do**
    $E_{LION} = E_{LION} \cup \{(h, v_i)\}$
    if $I_{in} > I_{cur}$ **then**
        $h = v_i$
        $I_{cur} = I_{in}$
    end if
end for

Output: Graph G_{LION}

Theorem 10. *$\sqrt{n}$ is a lower bound for the incoming interference I_{in} in an exponential node chain consisting of n nodes.*

Proof. In order to prove the theorem, we state two properties for I_{in} in an exponential node chain C. First, it holds that I_{in} is at least the number of hubs in C, since the leftmost node is interfered by exactly all hubs (property 1). On the other hand, I_{in} is greater than the maximum degree of the resulting topology, since [4] shows that the maximum degree of a graph is a lower bound for I_{in} (property 2). We assume for the sake of contradiction that there exists a connected graph that yields interference less than $\sqrt{n}$. In other words, the degree of any node is required to be at most $\sqrt{n} - 2$, and the number of hubs must not exceed $\sqrt{n} - 1$. Let H denote the set of hubs in the graph and S the nodes in the graph that are not hubs. By definition, each node in the graph is either in H or in S and therefore $|H| + |S| = n$ holds. Due to property 1, it follows that $|H| \leq \sqrt{n} - 1$. Without loss of generality we assume that the hubs are linearly connected among themselves in order to guarantee connectivity of the graph. Consequently, with property 2, each hub can connect to at most $\sqrt{n} - 4$ nodes in S (the leftmost and the rightmost hub, respectively, to $\sqrt{n} - 3$). By the definition of a hub, nodes in S are only connected to hubs and not among themselves. Therefore we obtain

$$|S| \leq \left(\sqrt{n} - 1\right)\left(\sqrt{n} - 4\right) + 2.$$

Consequently, $|H| + |S|$ results in $n - 4\sqrt{n} + 5$, which is less than n for $n \geq 2$ and thus leads to a contradiction. □

From Theorem 9 and 10 it follows that algorithm LION is asymptotically optimal in terms of interference in exponential node chains.

4.2 Highway

In this section we assume a more general network model than in Section 4.1. We still consider a one-dimensional scenario, but now the n nodes are arbitrarily distributed. This model is also referred to as the highway model because network instances can be seen as a bird's-eye view of a highway with nodes representing cars. Figure 4.3 depicts an example network in the model presented above with linearly connected nodes.

Figure 4.3: Example of a network in the highway model, where nodes are connected linearly.

4.2.1 Searching for Chains

Based on the results in Section 4.1, we propose a generalization of algorithm LION for the highway model. If we assume the nodes of a highway instance to be linearly connected, high interference at a node u requires many nodes to cover u. However, with increasing distance to u the nodes also need increasing distances to their next neighbors in the highway instance in order to interfere with u. This leads to an exponential characteristic of these nodes, since the edges that account for the interference at u form a fragmented exponential node chain.

Definition 5. *Let u be a node of an instance of a highway and let all nodes be connected linearly. Then $\Gamma_l(u)$ is the set of edges to the left of u that cause one of the incident nodes to account for $I_{in}(u)$. For edges to the right of u, $\Gamma_r(u)$ is defined accordingly.*

Figure 4.4 depicts an example of a highway. Edges in $\Gamma_l(u)$, in $\Gamma_r(u)$ respectively, are depicted by dashed lines and the corresponding nodes interfering u by hollow points. One can see that $\Gamma_r(u)$ defines an exponential node chain, if each section not in $\Gamma_r(u)$ (e.g. the edge (v_l, v_r)) was contracted to one virtual node (e.g. v'). Consequently, we can replace all edges in $\Gamma_r(u)$ by edges obtained by applying algorithm LION to the virtual exponential node chain in order to reduce $I_{in}(u)$. But the edges between virtual nodes obtained by the algorithm need to be translated into edges between real nodes. Therefore all edges (v', w') where v' and w' are not direct neighbors in the exponential node chain must be replaced by (v_l, w_r) (this also applies to $\Gamma_l(u)$ with interchanged indices).

Different from Section 4.1 not all nodes are incident to an edge in $\Gamma_l(u) \cup \Gamma_r(u)$, and the application of LION to these chains may yield a negative

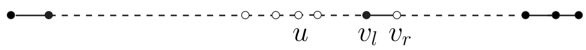

Figure 4.4: $\Gamma_l(u)$ and $\Gamma_r(u)$ (dashed lines) and the corresponding nodes (hollow points) interfering with node u.

impact on them. We consider the fragmented exponential node chain $\Gamma_r(u)$ and a node v. If v is further right than the rightmost node w incident to an edge in $\Gamma_r(u)$, $I_{in}(v)$ increases by at most one due to applying LION to $\Gamma_r(u)$. This follows from the fact that only w is able to interfere with new nodes to its right. On the other hand, if v is inside the fragmented exponential node chain, its interference also increases by at most one. The reason for this is that v is additionally covered only by its closest hub to the left. At last, if v is to the left of u, $I_{in}(v)$ increases by up to $\Omega(\sqrt{n})$, where n denotes the number of nodes in the chain. This follows from the fact that the resulting topology consists of $\Omega(\sqrt{n})$ hubs that interfere with v, because they may establish long-range edges.

An algorithm that constructs a low-interference topology for a given highway instance can be derived by applying the procedure discussed above for a node u to each node in the network. The nodes are thereby processed in descending order according to their initial interference caused by linear connection. But it has to be made sure that edges inserted to decrease interference at a particular node are not removed when dealing with subsequent nodes.

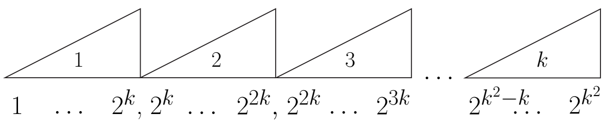

Figure 4.5: Worst-case example, where linear connection yields $I_{in} \in \mathrm{O}(k)$, whereas applying LION to each of the k exponential node chains (triangles) results in interference $\Omega\big(\sqrt{k^3}\big)$.

However, Figure 4.5 depicts a highway instance, where linear connection results in lower interference than applying algorithm LION to all existing exponential node chains. The example consists of k successively arranged exponential node chains diagramed as triangles. Each of these chains is set up by $k + 2$ nodes, where consecutive chains share the leftmost and the rightmost node, respectively. Below each triangle the distance between the first two and the last two nodes of a chain is depicted. If the nodes are connected linearly, we obtain I_{in} in $\mathrm{O}(k)$—and consequently in $\mathrm{O}(\sqrt{n})$, since $n = k(k+1)+1$. This follows from the fact that the maximum interference within one of the exponential node chains is in $\mathrm{O}(k)$ (see Section 4.1) and that the nodes of a chain interfere only with the penultimate node of the exponential node chain to its left, since the latter has exactly the same distance between its last two nodes as the former has between its first two. If LION is applied to the k exponential node chains in Figure 4.5, each individual chain yields interference $\Omega(\sqrt{k})$—and consequently the same number of hubs. Since almost all hubs produced by the algorithm are incident to an edge spanning at least one node, they all interfere with the leftmost node v of the example. As there are k exponential node chains, each of them containing $\Omega(\sqrt{k})$ hubs, v shows interference $\Omega(\sqrt{k^3})$. Consequently, naive approaches that try to reduce overall interference by reducing the interference of individual exponential node chains do not appear to be successful.

4.3 Greedy on Highway

In this section we present a greedy algorithm, referred to as GLOW, that is well suited to minimize I_{in} in the highway model. Before describing the algorithm, we introduce the interference sequence σ of a graph:

Definition 6. *Given a graph $G = (V, E)$, then $\sigma(G)$ is the sequence of interference values $I_{in}(v)$, with $v \in V$, in decreasing order.*

Additionally, comparison operators on two interference sequences are defined in terms of the lexicographic order.

Algorithm GLOW starts with a graph $G_{GLOW} = (V, E_{GLOW})$ where V is a set of nodes in a highway instance and E_{GLOW} an initially empty edge set. In addition all edges of the complete graph induced by V are in the set E[2]. While G_{GLOW} is not connected, GLOW adds an edge in E to E_{GLOW} in each step. Therefore, for each edge $e \in E$ that does not yield cycles in G_{GLOW}, the interference sequence $\sigma(G')$ with $G' = (V, E_{GLOW} \cup \{e\})$ is computed. The algorithm then inserts the edge which yields minimum σ into E_{GLOW}. If there are multiple edges that result in the same interference sequence, the one with minimal Euclidean length is chosen. In other words, GLOW tries to increase the interference values of a node in each step as little as possible.

The running time of the algorithm GLOW is $O(n^5)$. This follows from the fact that the while loop in Line 4 is repeated exactly $n - 1$ times—the resulting topology is a tree—, and that for all of the $O(n^2)$ edges in E the algorithm has to compute the interference of a graph consisting of n nodes, which takes time $O(n^2)$.

Due to local minima, the algorithm does not always lead to an optimal solution. Figure 4.6 depicts two connection strategies for an instance of the highway model. In the upper part the resulting topology is depicted if

[2] If the nodes in V are considered to feature a maximum transmission radius, E consists of all edges (u, v), with u and v in V, respectively, that satisfy $r_u^{max} \geq |u, v|$ and $r_v^{max} \geq |u, v|$.

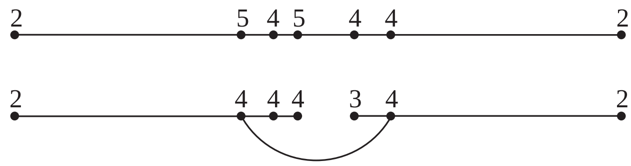

Figure 4.6: Example of a highway, where GLOW results in $I_{in} = 5$ (upper line), whereas an interference-minimal topology yields $I_{in} = 4$ (lower line).

GLOW is applied to the instance. It can be seen that the algorithm connects the nodes linearly, which leads to interference $I_{in} = 5$. On the other hand, in the lower part of Figure 4.6 an interference-optimal topology is depicted that only yields $I_{in} = 4$. However, if GLOW is applied to an exponential node chain presented in Section 4.1, it produces exactly the same topology as algorithm LION, which we have shown to be asymptotically optimal for exponential node chains. GLOW is also applicable to two-dimensional problem instances and appears to result in low-interference topologies for practical networks.

Greedy Low Interference On Highway (GLOW)
Input: V, a set of n nodes distributed in one dimension
E = all eligible edges (u, v) (u and v in V)
$E_{GLOW} = \emptyset$
$G_{GLOW} = (V, E_{GLOW})$
while G_{GLOW} is not connected **do**
  $\sigma_{min} = n, ..., n$ (n times)
  e_{min} = null
  for all $e = (u, v) \in E$ **do**
    if u and v are in the same component of G_{GLOW} **then**
      $E = E \setminus \{e\}$
    else
      $G' = (V, E_{GLOW} \cup \{e\})$
      if $\sigma(G') < \sigma_{min}$ or ($\sigma(G') = \sigma_{min}$ and $|e| < |e_{min}|$) **then**
        $\sigma_{min} = \sigma(G')$
        $e_{min} = e$
      end if
    end if
  end for
  $E_{GLOW} = E_{GLOW} \cup \{e_{min}\}$
  $E = E \setminus \{e_{min}\}$
end while
Output: Graph G_{GLOW}

4.4 Conclusion

The incoming interference model defined in Chapter 2 is studied on the basis of one-dimensional networks. In the first part of the chapter an ideal topology, referred to as *exponential node chain*, is considered. We show that $\sqrt{n}$ is a lower bound for I_{in} in such a network. This lower bound is shown to be asymptotically matched by a scan-line algorithm. In the second part the more general highway model is assumed, where nodes are arbitrarily distributed in one dimension. An attempt to transfer the algorithm from the first part of the chapter is shown. Then an example is presented that shows that such efforts do not appear to be successful. Finally, we propose a greedy algorithm that appears to be a good heuristic for interference reduction for instances in the highway model, since it is asymptotically optimal in the case of exponential node chains.

Besides the presented results within this chapter, there are still open questions to be answered in the field of incoming interference. Continuing problems that surfaced while we were concerned with this field include:

- Is GLIT a $O(\sqrt{n})$-approximation algorithm for I_{in}?
- Is there a network instance yielding optimum interference greater than $O(\sqrt{n})$?
- How well does GLOW approximate I_{in}?
- Is there a local algorithm that approximates optimum interference?
- Are there any algorithms, based on clusters in order to elect hubs, which result in low-interference topologies?

Chapter 5

Minimum Membership Set Cover

Based on the studies in the field of incoming interference introduced in Chapter 2 and tackled in Chapter 4, minimizing I_{in} is considered in another important problem domain in this chapter, namely in the field of cellular networks.

5.1 Introduction

Cellular networks are heterogeneous networks consisting of two different types of nodes: base stations and clients. The base stations—acting as servers—are interconnected by an external fixed backbone network; clients are connected via radio links to base stations. The totality of the base stations forms the infrastructure for distributed applications running on the clients, the most prominent of which probably being mobile telephony. Cellular networks can however more broadly be considered a type of infrastructure for distributed tasks in general.

Since communication over the wireless links takes place in a shared medium, interference can occur at a client if it is within transmission range of more than one base station. In order to prevent such collisions, coordination among the conflicting base stations is required. Commonly this problem is solved by segmenting the available frequency spectrum into channels to be assigned to the base stations in such a way as to prevent interference, in particular such that no two base stations with overlapping transmission range use the same channel.

In this chapter we assume a different approach to interference reduction. The basis of our analysis is formed by the observation that interference effects occurring at a client depend on the number of base stations by whose transmission ranges it is covered. In particular for solutions using frequency division multiplexing as described above, the number of base stations cov-

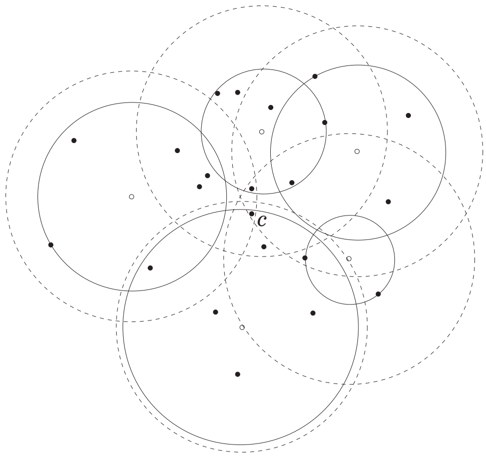

Figure 5.1: If the base stations (hollow points) are assigned identical transmission power levels (dashed circles), client c experiences high interference, since it is covered by all base stations. Interference can be reduced by assigning appropriate power values (solid circles), such that all clients are covered by at most two base stations.

ering a client is a lower bound for the number of channels required to avoid conflicts; a reduction in the required number of channels, in turn, can be exploited to broaden the frequency segments and consequently to increase communication bandwidth. On the other hand, also with systems using code division multiplexing, the coding overhead can be reduced if only a small number of base stations cover a client.

The transmission range of a base station—and consequently the coverage properties of the clients—depends on its position, obstacles hindering the propagation of electromagnetic waves, such as walls, buildings, or mountains, and the base station transmission power. Since due to legal or architectural constraints the former two factors are generally difficult to control, we assume a scenario in which the base station positions are fixed, each base station can however adjust its transmission power. The problem of minimizing interference then consists in assigning every base station a transmission power level such that the number of base stations covering any node is minimal (cf. Figure 5.1). At the same time however, it has to be guaranteed that every client is covered by at least one base station in order to maintain availability of the network.

In our analysis we formalize this task as a combinatorial optimization problem. For this purpose we model the transmission range of a base station having chosen a specific transmission power level as a set containing exactly all clients covered thereby. The totality of transmission ranges selectable by all base stations is consequently modeled as a collection of client sets.

More formally, this yields the *Minimum Membership Set Cover (MMSC)* problem: Given a set of elements U (modeling clients) and a collection S of subsets of U (transmission ranges), choose a solution $S' \subseteq S$ such that every element occurs in at least one set in S' (maintain network availability) and that the *membership* $M(e, S')$ of any element e with respect to S' is minimal, where $M(e, S')$ is defined as the number of sets in S' in which e occurs (interference).

Having defined this formalization, we show in this chapter—by reduction from the related Minimum Set Cover problem—that the MMSC problem is *NP*-complete and that no polynomial time algorithm exists with approximation ratio less than $\ln n$ unless $NP \subset TIME(n^{O(\log \log n)})$. We additionally present a probabilistic algorithm based on linear programming relaxation asymptotically matching this lower bound, particularly yielding an approximation ratio in $O(\log n)$ with high probability. Furthermore we study how the presented algorithm performs on practical network instances.

5.2 Related Work

Interference issues in cellular networks have been studied since the early 1980s in the context of frequency division multiplexing: The available network frequency spectrum is divided into narrow channels assigned to cells in a way to avoid interference conflicts. In particular two types of conflicts can occur, adjacent cells using the same channel (cochannel interference) and insufficient frequency distance between channels used within the same cell (adjacent channel interference). Maximizing the reuse of channels respecting these conflicts was generally studied by means of the combinatorial problem of conflict graph coloring using a minimum number of colors. The settings in which this problem was considered are numerous and include hexagon graphs, geometric intersection graphs (such as unit disk graphs), and planar graphs, but also (non-geometric) general graphs. In addition both static and dynamic (or on-line) approaches were studied [25]. The fact that channel separation constraints can depend on the distance of cells in the conflict graph was studied by means of graph labeling [12]. The problem of frequency assignment is tackled in a different way in [7] exploiting the observation that in every region of an area covered by the communication network it is sufficient that exactly one base station with a unique channel can be heard. As mentioned, all these studied models try to avoid interference conflicts occurring when using frequency division multiplexing. In contrast, the problem described in this chapter assumes a different approach in aiming at interference reduction by having the base stations choose suitable transmission power levels.

The problem of reducing interference is formalized in a combinatorial optimization problem named *Minimum Membership Set Cover*. As suggested by its name, at first sight its formulation resembles closely the long-known and well-studied *Minimum Set Cover (MSC)* problem, where the number of sets chosen to cover the given elements is to be minimized [14]. That the MMSC and the MSC problems are however of different nature can be concluded from the following observation: For any MSC instance consisting of n elements, a greedy algorithm approximates the optimal solution with an approximation ratio at most $H(n) \leq \ln n + 1$ [14], which has later been shown to be tight up to lower order terms unless $NP \subset TIME(n^{O(\log \log n)})$ [8, 22]. For the MMSC problem in contrast, there exist instances where the same greedy algorithm fails to achieve *any* nontrivial approximation of the optimal solution.

5.3 Minimum Membership Set Cover

As described in the introduction, the problem considered in this chapter is to assign to each base station a transmission power level such that interference is minimized while all clients are covered. For our analysis we formalize this problem by introducing a combinatorial optimization problem referred to as *Minimum Membership Set Cover*. In particular, clients are modeled as elements and the transmission range of a base station given a certain power level is represented as the set of thereby covered elements. In the following, we first define the membership of an element given a collection of sets:

Definition 7 (Membership). *Let U be a finite set of elements and S be a collection of subsets of U. Then the membership $M(e, S)$ of an element e is defined as $|\{T \mid e \in T, T \in S\}|$.*

Informally speaking, MMSC is identical to the MSC problem apart from the minimization function. Where MSC minimizes the total number of sets, MMSC tries to minimize element membership. Particularly, MMSC can be defined as follows:

Definition 8 (Minimum Membership Set Cover). *Let U be a finite set of elements with $|U| = n$. Furthermore let $S = \{S_1, \ldots, S_m\}$ be a collection of subsets of U such that $\bigcup_{i=1}^{m} S_i = U$. Then* Minimum Membership Set Cover *(MMSC) is the problem of covering all elements in U with a subset $S' \subseteq S$ such that* $\max_{e \in U} M(e, S')$ *is minimal.*

5.4 Problem Complexity

In this section we address the complexity of the *Minimum Membership Set Cover* problem. We show that MMSC is *NP*-complete and therefore no polynomial time algorithm exists that solves MMSC unless $P = NP$.

Theorem 11. *MMSC is NP-complete.*

Proof. We will prove that MMSC is *NP*-complete by reducing MSC to MMSC. Consider an MSC instance (U, S) consisting of a finite set of elements U and a collection S of subsets of U. The objective is to choose a subset S' with minimum cardinality from S such that the union of the chosen subsets of U contains all elements in U.

We now define a set $\widetilde{U}$ by adding a new element e to U, construct a new collection of sets $\widetilde{S}$ by inserting e into all sets in S, and consider $(\widetilde{U}, \widetilde{S})$ as an instance of MMSC. Since element e is in every set in $\widetilde{S}$, it follows that e is an element with maximum membership in the solution S' of MMSC. Moreover, the membership of e in S' is equal to the number of sets in the solution. Therefore MMSC minimizes the number of sets in the solution by minimizing the membership of e. Consequently we obtain the solution for MSC of the instance (U, S) by solving MMSC for the instance $(\widetilde{U}, \widetilde{S})$ and extracting element e from all sets in the solution.

We have shown a reduction from MSC to MMSC, and therefore the latter is *NP*-hard. Since solutions for the decision problem of MMSC are verifiable in polynomial time, it is in *NP*, and consequently the MMSC decision problem is also *NP*-complete. □

Now that we have proved MMSC to be *NP*-complete and therefore not to be optimally computable within polynomial time unless $P = NP$, the question arises, how closely MMSC can be approximated by a polynomial time algorithm. This is partly answered with the following lower bound.

Theorem 12. *There exists no polynomial time approximation algorithm for the MMSC problem with an approximation ratio less than* $\ln n$ *unless* $NP \subset TIME(n^{O(\log\log n)})$.

Proof. The reduction from MSC to MMSC in the proof of Theorem 11 is approximation-preserving, that is, it implies that any lower bound for MSC also holds for MMSC. In [8] it is shown that $\ln n$ is a lower bound for the approximation ratio of MSC unless $NP \subset TIME(n^{O(\log\log n)})$. Thus, $\ln n$ is also a lower bound for the approximation ratio of MMSC. □

5.5 Approximating MMSC by LP Relaxation

In the previous section a lower bound of $\ln n$ for the approximability of the MMSC problem by means of polynomial time approximation algorithms has been established. In this section we show how to obtain a $O(\log n)$-approximation with high probability[1] using LP relaxation techniques. For an introduction to linear programming see for instance [5].

[1] Throughout the chapter, an event E occurring "with high probability" stands for $Pr[E] = 1 - O(\frac{1}{n})$.

5.5.1 LP Formulation of MMSC

We first derive the integer linear program which describes the MMSC problem and then formulate the linear program that relaxes the integrality constraints.

Let $S' \subseteq S$ denote a subset of the collection S. To each $S_i \in S$ we assign a variable $x_i \in \{0, 1\}$ such that $x_i = 1 \Leftrightarrow S_i \in S'$. For S' to be a set cover, it is required that for each element $u_i \in U$, at least one set S_j with $u_i \in S_j$ is in S'. Therefore, S' is a set cover of U if and only if for all $i = 1, ..., n$ it holds that $\sum_{S_j:u_i \in S_j} x_j \geq 1$. For S' to be minimal in the number of sets that cover a particular element, we need a second set of constraints. Let z be the maximum membership over all elements caused by the sets in S'. Then for all $i = 1, ..., n$ it follows that $\sum_{S_j:u_i \in S_j} x_j \leq z$. The MMSC problem can consequently be formulated as the integer program IP_{MMSC}:

$$
\begin{aligned}
&\text{minimize } z \\
&\text{subject to} \quad \sum_{S_j:u_i \in S_j} x_j \geq 1 \qquad i = 1, ..., n \\
&\qquad\qquad\quad\ \sum_{S_j:u_i \in S_j} x_j \leq z \qquad i = 1, ..., n \\
&\qquad\qquad\qquad\quad x_j \in \{0, 1\} \qquad j = 1, ..., m
\end{aligned}
$$

By relaxing the constraints $x_j \in \{0, 1\}$ to $x'_j \geq 0$, we obtain the following linear program LP_{MMSC}:

$$
\begin{aligned}
&\text{minimize } z \\
&\text{subject to} \quad \sum_{S_j:u_i \in S_j} x'_j \geq 1 \qquad i = 1, ..., n \\
&\qquad\qquad\quad\ \sum_{S_j:u_i \in S_j} x'_j \leq z \qquad i = 1, ..., n \\
&\qquad\qquad\qquad\quad x'_j \geq 0 \qquad j = 1, ..., m
\end{aligned}
$$

The integer program IP_{MMSC} yields the optimal solution z^* for an MMSC problem. The derived linear program LP_{MMSC} therefore obtains a fractional solution z' with $z' \leq z^*$, since we allow the variables x'_j to be in [0,1].

5.5.2 Algorithm and Analysis

We will now present a $\text{O}(\log n)$-approximation algorithm, referred to as $\mathcal{A}_{\text{MMSC}}$, for the MMSC problem. Given an MMSC instance (U, S), the algorithm first solves the linear program LP_{MMSC} corresponding to (U, S). In a second step, $\mathcal{A}_{\text{MMSC}}$ performs randomized rounding (see [27]) on a feasible solution vector $\underline{x}'$ for LP_{MMSC}, in order to derive a vector $\underline{x}$ with $x_i \in \{0,1\}$. Finally it is ensured that $\underline{x}$ is a feasible solution for IP_{MMSC} and consequently a set cover.

Algorithm $\mathcal{A}_{\text{MMSC}}$

Input: an MMSC instance (U, S)

1: compute solution vector $\underline{x}'$ to the linear program LP_{MMSC} corresponding to (U, S)
2: $p_i := \min\{1,\ x'_i \cdot \log n\}$
3: $x_i := \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{otherwise} \end{cases}$
4: **for all** $u_i \in U$ **do**
5:    **if** $\sum_{S_j : u_i \in S_j} x_j = 0$ **then**
6:       set $x_j = 1$ for any j such that $u_i \in S_j$
7:    **end if**
8: **end for**

Output: MMSC solution S' corresponding to $\underline{x}$

For the analysis of $\mathcal{A}_{\text{MMSC}}$ the following two mathematical facts are required. Their proofs are omitted and can be found in mathematical text books.

Fact 1. (Means Inequality) *Let $\mathcal{A} \subset \mathbb{R}^+$ be a set of positive real numbers. The product of the values in $\mathcal{A}$ can be upper-bounded by replacing each factor with the arithmetic mean of the elements of $\mathcal{A}$:*

$$\prod_{x \in \mathcal{A}} x \ \leq\ \left(\frac{\sum_{x \in \mathcal{A}} x}{|\mathcal{A}|} \right)^{|\mathcal{A}|}.$$

Fact 2. *For all n, t, such that $n \geq 1$ and $|t| \leq n$,*

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

We prove $\mathcal{A}_{\text{MMSC}}$ to be a $\text{O}(\log n)$-approximation algorithm for IP_{MMSC} in several steps. We first show that the membership of an element in U after the randomized rounding step of $\mathcal{A}_{\text{MMSC}}$ is bounded with high probability.

Lemma 13. *The membership of an element u_i after Line 3 of $\mathcal{A}_{\text{MMSC}}$ is at most $2e \log n \cdot z^*$ with high probability.*

Proof. The optimal solution of LP_{MMSC} leads to fractional values x'_j and does not admit a straightforward choice of the sets S_j. Using randomized rounding, $\mathcal{A}_{\text{MMSC}}$ converts the fractional solution to an integral solution S'.

In Line 3, a set S_j is chosen to be in S' with probability $x'_j \cdot \log n$. Thus, the expected membership of an element u_i is

$$E[M(u_i, S')] = \sum_{S_j : u_i \in S_j} x'_j \cdot \log n \leq \log n \cdot z'. \tag{5.1}$$

The last inequality follows directly from the second set of constraints of LP_{MMSC}. Since $z' \leq z^*$, it follows that the expected membership for u_i is at most $\log n \cdot z^*$. Now we need to ensure that, with high probability, u_i is not covered too often. Since randomized rounding can be modeled as Poisson trials, we are able to use a Chernoff bound [24]. Let Y_i be a random variable denoting the membership of u_i with expected value $\mu = E[M(u_i, S')]$. Applying the Chernoff bound we derive

$$Pr\left[Y_i \geq (1+\delta)\,\mu\right] < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu.$$

Choosing $\delta \geq 2e - 1$, the right hand side of the inequality simplifies to

$$\left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu \leq \left(\frac{e^\delta}{(2e)^{(1+\delta)}}\right)^\mu < \left(\frac{e^\delta}{(2e)^\delta}\right)^\mu = 2^{-\delta\mu}. \tag{5.2}$$

Since the above Chernoff bound corresponds to the upper tail of the probability distribution of Y_i and as μ is at most $\log n \cdot z^*$, it follows that

$$Pr\left[Y_i \geq (1+\delta) \log n \cdot z^*\right] \leq Pr\left[Y_i \geq (1+\delta)\,\mu\right].$$

But for this inequality to hold, only $(1+\delta)\mu \leq c \log n \cdot z^*$ for some constant c is required. Thus, by setting $(1+\delta)\mu = c \log n \cdot z^*$ and using Inequality (5.1), we obtain

$$\delta\mu \quad \geq \quad (c-1) \log n \cdot z^*. \tag{5.3}$$

Using Inequalities (5.2) and (5.3) we can then bound the probability that the membership of u_i is greater than $c \log n \cdot z^*$ as follows:

$$Pr\left[Y_i \geq c \log n \cdot z^*\right] < 2^{-\delta\mu} \ \leq 2^{-(c-1)\log n \cdot z^*} = \frac{1}{n^{(c-1)z^*}}. \tag{5.4}$$

In order to compute c, we again consider the equation $(1+\delta)\mu = c \log n \cdot z^*$. Solving for δ, we derive

$$\delta = \frac{c \log n \cdot z^*}{\mu} - 1.$$

As a requirement for Inequality (5.2) we demand δ to be greater or equal to $2e - 1$. Furthermore, the right hand side of the inequality is minimal if μ is maximal. Thus, using Inequality (5.1) we obtain

$$\frac{c \log n \cdot z^*}{\log n \cdot z^*} - 1 \quad \geq \quad 2e - 1$$

or $c \geq 2e$. Taking everything together and using $z^* \geq 1$ it follows that

$$Pr\left[Y_i \geq 2e \log n \cdot z^*\right] < \frac{1}{n^{(2e-1)z^*}} \in \mathrm{O}\left(\frac{1}{n^4}\right).$$

□

Now we are ready to show that after randomized rounding all elements have membership at most $2e \log n \cdot z^*$ with high probability.

Lemma 14. *The membership of all elements in U after Line 3 of $\mathcal{A}_{\text{MMSC}}$ is at most $2e \log n \cdot z^*$ with high probability.*

Proof. Let E_i be the event that the membership of element u_i after Line 3 of $\mathcal{A}_{\text{MMSC}}$ is greater than $2e \log n \cdot z^*$. Then, the probability that the membership for all elements in U is less than $2e \log n \cdot z^*$ equals

$$Pr[\bigwedge_{i=1}^{n} \overline{E_i}\,].$$

We know from Lemma 13 that the probability $\Pr[E_i]$ is less than $1/n^{(2e-1)z^*}$. Since the events are clearly not independent, we cannot apply the product rule. However, it was shown in [30] that

$$Pr[\bigwedge_{i=1}^{n} \overline{E_i}\,] \geq \prod_{i=1}^{n} Pr[\,\overline{E_i}\,]. \tag{5.5}$$

We can make use of this bound, since IP_{MMSC} features the *positive correlation* property assumed in [30]. Consequently, setting $\alpha = (2e-1)z^*$ and using Inequality (5.5), it follows that

$$\begin{aligned} Pr[\bigwedge_{i=1}^{n} \overline{E_i}\,] \;&\geq\; \left(1 - \frac{1}{n^\alpha}\right)^n \;\geq\; \left(1 - \frac{1}{n^\alpha}\right)^{\frac{n^\alpha - 1}{n^{\alpha-1} - \frac{1}{n}}} \\ &\geq\; e^{-\frac{1}{n^{\alpha-1} - \frac{1}{n}}} \;>\; 1 - \frac{1}{n^{\alpha-1} - \frac{1}{n}}. \end{aligned}$$

For the third inequality we use Fact 2 with $t = -1$, which leads to the inequality

$$e^{-1} \leq (1 - 1/n)^{n-1}.$$

The last inequality is derived through Taylor series expansion of the left hand term. Consequently, using $\alpha = (2e-1)z^*$ and $z^* \geq 1$ we obtain

$$Pr[\bigwedge_{i=1}^{n} \overline{E_i}\,] = 1 - \mathrm{O}\left(\frac{1}{n^3}\right).$$

□

Since $\mathcal{A}_{\text{MMSC}}$ uses randomized rounding, we do not always derive a feasible solution for IP_{MMSC} after Line 3 of the algorithm. That is, there exist elements in U that are not covered by a set in S'. But we can show in the following lemma that each single element is covered with high probability.

Lemma 15. *After Line 3 of $\mathcal{A}_{\text{MMSC}}$, an element u_i in U is covered with high probability.*

Proof. For convenience we define C_i to be the set $\{S_j \mid u_i \in S_j\}$. From LP_{MMSC} we know that $\sum_{S_j \in C_i} x'_j \geq 1$. Thus, it follows that

$$\sum_{S_j \in C_i} p_j \geq \log n. \tag{5.6}$$

Let q_i be the probability that an element u_i is contained in none of the sets in S' obtained by randomized rounding, that is, $q_i = Pr\left[M(u_i, S') = 0\right]$. Consequently, we have

$$\begin{aligned} q_i &= \prod_{S_j \in C_i} (1 - p_j) \leq \left(1 - \frac{\sum_{S_j \in C_i} p_j}{|C_i|}\right)^{|C_i|} \\ &\leq e^{-\sum_{S_j \in C_i} p_j} \leq e^{-\log n} = \frac{1}{n}. \end{aligned}$$

The first inequality follows from Fact 1, the second inequality follows from Fact 2, and the third step is derived from Inequality (5.6).

□

In Lines 4 to 8 of $\mathcal{A}_{\text{MMSC}}$ it is ensured that the final solution S' is a set cover. This is achieved by consecutively including sets in S', until all elements are covered. In the following we show that the additional maximum membership increase caused thereby is bounded with high probability.

Lemma 16. *In Lines 4 to 8 of $\mathcal{A}_{\text{MMSC}}$, the maximum membership in U is increased by at most $\mathrm{O}(\log n)$ with high probability.*

Proof. In order to bound the number of sets added in the considered part of the algorithm, again a Chernoff bound is employed. Let Z be a random variable denoting the number of uncovered elements after Line 3 of $\mathcal{A}_{\text{MMSC}}$. From Lemma 15 we know that an element is uncovered after randomized rounding with probability less than $1/n$. Then, the expected value μ for Z is less than 1. Using a similar analysis as in Lemma 13, we obtain

$$Pr\left[Z \geq c\right] < 2^{-c+1},$$

where $c \geq 2e$ is required. Setting $c = \log n + 2e$, it follows that

$$Pr\left[Z \geq \log n + 2e\right] < \frac{2}{n \cdot 4^e} \in \mathrm{O}\left(\frac{1}{n}\right).$$

The proof is concluded by the observation that each additional set added in the second step of $\mathcal{A}_{\text{MMSC}}$ increases the maximum membership in U by at most one. Since only $\mathrm{O}(\log n)$ elements have to be covered with high probability and as it is sufficient to add one set per element, the lemma follows.[2] □

Now we are ready to prove that $\mathcal{A}_{\text{MMSC}}$ yields a $\mathrm{O}(\log n)$-approximation for IP_{MMSC} and consequently also for MMSC.

Theorem 17. *Given an MMSC instance consisting of m sets and n elements, $\mathcal{A}_{\text{MMSC}}$ computes a $\mathrm{O}(\log n)$-approximation with high probability. The running time of $\mathcal{A}_{\text{MMSC}}$ is polynomial in $m \cdot n$.*

Proof. The approximation factor in the theorem directly follows from Lemmas 14 and 16. The running time result is a consequence to the existence of algorithms solving linear programs in time polynomial in the program size [15] and to the fact that LP_{MMSC} can be described using -1, 0, and 1 as coefficients only. □

5.5.3 Alternative Algorithm

In an alternative version of the algorithm, the values $\underline{x}'$ obtained by solving LP_{MMSC} can be directly employed as probabilities for randomized rounding (without the additional factor of $\log n$). In this case randomized rounding is repeated for all sets containing elements not yet covered until resulting in a set cover. With similar arguments as for $\mathcal{A}_{\text{MMSC}}$, it can be shown that this modified algorithm achieves the same approximation factor and that it terminates after repeating randomized rounding at most $\log n$ times, both with high probability.

5.6 Practical Networks

Whereas the previous section showed that $\mathcal{A}_{\text{MMSC}}$ approximates the optimal solution up to a factor in $\mathrm{O}(\log n)$, this section discusses practical networks. In particular, the algorithms $\mathcal{A}_{\text{MMSC}}$ and $\widetilde{\mathcal{A}}_{\text{MMSC}}$—the alternative algorithm described in Section 5.5.3—are considered. Since the approximation performance of algorithms is studied, we denote by the *membership of a solution* the minimization function value—that is the maximum membership over all clients—of the corresponding MMSC solution.

The studied algorithms were executed on instances generated by placing base stations and clients randomly according to a uniform distribution

[2]Since in the above Chernoff bound μ is at most a constant, a more careful analysis would yield that the maximum membership in U is increased—with high probability—by $\mathrm{O}(\log n/\log\log n)$ only. This improvement has however no impact on the main result of this chapter.

on a square field with side length 5 units. Adaptable transmission power values were modeled by attributing to each base station circles with radii 0.25, 0.5, 0.75, and 1 unit; each such circle then contributes one set containing all covered clients to the problem instance thereafter presented to the algorithms.

As shown in the previous section, the approximation factor of the algorithms depends on the number of clients. For this reason the simulations were carried out over a range of client densities. Since the membership value that is obtained by solving $\mathrm{LP}_{\mathrm{MMSC}}$ lies below the optimal solution and thus the gap between the algorithm result and the solution of the linear program is an upper bound for the obtained approximation ratio, the $\mathrm{LP}_{\mathrm{MMSC}}$ result z' is also considered.

For a base-station density of 2 base stations per unit disk, Figure 5.2(a) shows the mean membership values over 200 networks—for each simulated client density—for the results computed by $\mathcal{A}_{\mathrm{MMSC}}$, $\widetilde{\mathcal{A}}_{\mathrm{MMSC}}$, and the values obtained by solving $\mathrm{LP}_{\mathrm{MMSC}}$. The results depict that for this relatively low base-station density all measured values are comparable and increase with growing client density. In contrast, for a higher base-station density of 5 base stations per unit disk (cf. Figure 5.2(b)), a gap opens between the $\mathcal{A}_{\mathrm{MMSC}}$ and $\mathrm{LP}_{\mathrm{MMSC}}$ results. Whereas the ratio between these two result series rises sharply for low client densities, its increase diminishes for higher client densities, which corresponds to the $\mathrm{O}(\log n)$ approximation factor described in the theoretical analysis. Additionally, it can be observed that $\widetilde{\mathcal{A}}_{\mathrm{MMSC}}$ performs significantly better than $\mathcal{A}_{\mathrm{MMSC}}$. The reason for this effect lies in the fact that $\mathcal{A}_{\mathrm{MMSC}}$ multiplies the $\underline{x}'$ values resulting from $\mathrm{LP}_{\mathrm{MMSC}}$ with the factor $\log n$ to obtain the probabilities employed for randomized rounding, whereas this multiplication is not performed by $\widetilde{\mathcal{A}}_{\mathrm{MMSC}}$. The approximation gap becomes even wider for higher base-station densities, such as 10 base stations per unit disk (Figure 5.2(c)). Our simulations showed however that beyond this base-station density no significant changes in the membership results can be observed.

The increasing gap between the simulated algorithms and the $\mathrm{LP}_{\mathrm{MMSC}}$ solution with growing base-station density can be explained by the following observation: For low base-station densities—where problem instances contain a small number of sets—a relatively large number of clients are covered by only one set, which consequently will have to be chosen in both the $\mathrm{LP}_{\mathrm{MMSC}}$ and the algorithm solutions; for high base-station densities, in contrast, the solution weights $\underline{x}'$ computed by $\mathrm{LP}_{\mathrm{MMSC}}$ can be distributed more evenly among the relatively high number of available sets, and the potential of "committing an error" during randomized rounding increases.

In summary, the simulations show that the considered algorithms approximate the optimal solution well on practical networks. Comparing $\mathcal{A}_{\mathrm{MMSC}}$ and $\widetilde{\mathcal{A}}_{\mathrm{MMSC}}$, it can be observed that, in practice, the latter algorithm performs even better than the former.

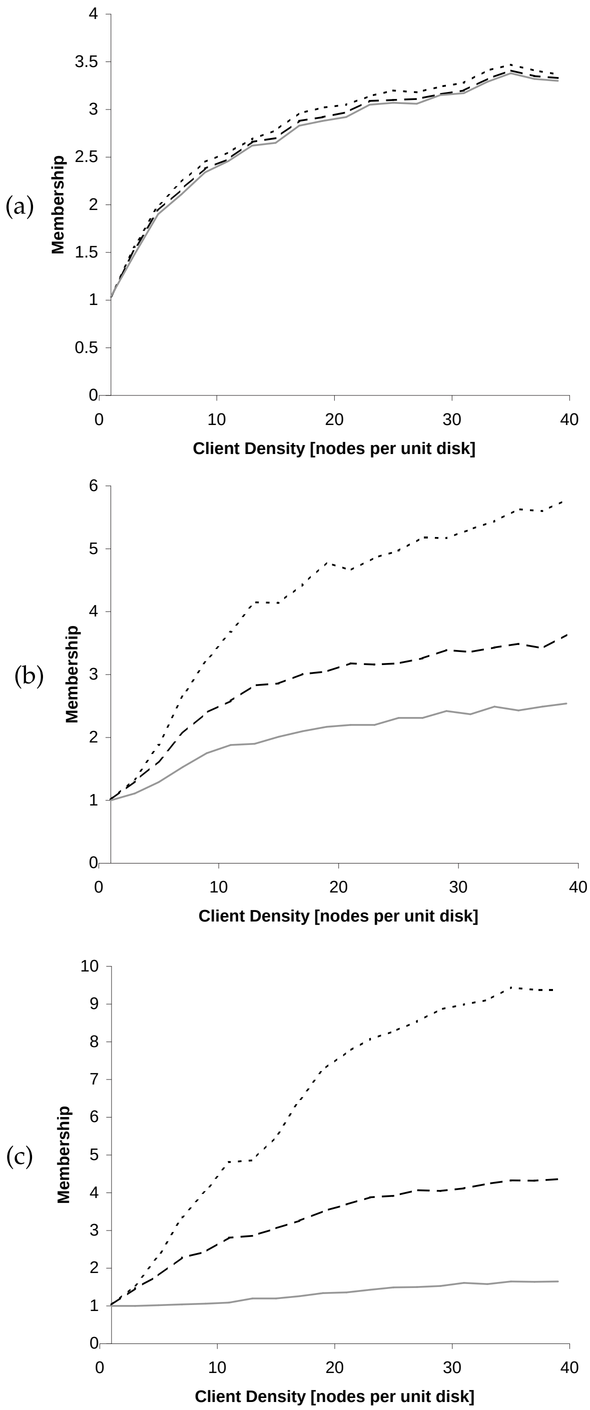

Figure 5.2: Mean values of the membership results obtained by $\mathcal{A}_{\mathrm{MMSC}}$ (dotted), $\widetilde{\mathcal{A}}_{\mathrm{MMSC}}$ (dashed), and the $\mathrm{LP}_{\mathrm{MMSC}}$ solution with 2 (a), 5 (b), and 10 (c) base stations per unit disk.

5.7 Greedy Approaches

For the Minimum Set Cover problem there exists a simple greedy algorithm that approximates the optimal solution with approximation ratio at most $H(n) \leq \ln n + 1$, which asymptotically matches the lower bound for the problem. The question arises if there is also a greedy algorithm for MMSC that yields an $\mathrm{O}(\log n)$ approximation ratio as $\mathcal{A}_{\mathrm{MMSC}}$.

We first consider the greedy criterion used in the well-known greedy algorithm for MSC introduced in [14], referred to as Greedy_{MSC}. That is, we obtain the solution set S' by successively choosing a set S_i in the ith step of the algorithm that maximizes $|S_i \setminus \bigcup_{j=1}^{i-1} S_j|$. In other words, we choose a set in S which covers the biggest number of elements in Unot already covered. Figure 5.3 depicts an instance of the MMSC problem with the points in the upper, the lower line, respectively, representing sets of elements with depicted cardinalities. It is obvious that the top and the bottom disks form an optimal solution for MMSC with $M(u, S') = 2$. Applying Greedy_{MSC} to this instance however yields membership k for element u, where k is the number of disks in the chain. The reason is that Greedy_{MSC} chooses all disks in the chain starting from right to left. This follows from the fact that in each step there exists a disk in the chain that covers exactly one element more than the top and the bottom disk, respectively. As the number of elements n in Figure 5.3is $3 \cdot 2^k + 2$, we derive

$$k \;=\; \log{(n-2)} - \log 3 \;\in\; \Omega(\log n),$$

and consequently we can lower-bound the approximation ratio of Greedy_{MSC} to $\Omega(\log n)$.

The second approach towards a good greedy criterion is inspired by the greedy algorithm presented in Section 4.3. We there make a choice in each step of the algorithm based on the comparison of lexicographically ordered interference values of the nodes. We can apply the same idea to MMSC. For each disk still eligible in the current step—a disk is said to be eligible if there are uncovered elements within the disk area—we compute the membership of all elements based on the assumption that this and all previously chosen disks represent a solution of the problem. We then choose the disk which minimizes the lexicographic order of these memberships. The greedy algorithm using this criterion is referred to as Greedy_{MMSC}. However, this greedy algorithm also yields $M(u, S') \in \Omega(\log n)$ for the instance depicted in Figure 5.3 since it also chooses all disks in the chain instead of the top and the bottom ones. Consequently, Greedy_{MMSC} results in the same lower bound for the approximation ratio as Greedy_{MSC}, namely $\Omega(\log n)$.

It can be seen that the instance in Figure 5.3 is in some way bad for the greedy criteria presented so far because of the element distribution. Let a cell denote an area in the Euclidean plane where all interior points are covered by the same disks (see Figure 5.3). Consequently, the elements represented

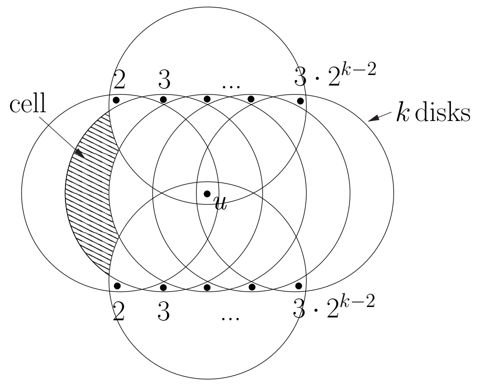

Figure 5.3: Instance of the geometric MMSC problem that yields $M(u, S') \in \Omega(\log n)$ for Greedy$_{MSC}$.

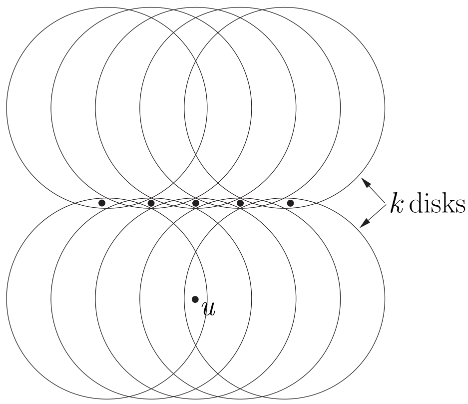

Figure 5.4: Instance of the geometric MMSC problem that yields $M(u, S') \in \Omega(n)$ for Greedy$_{Area}$.

by one of the element sets in Figure 5.3 are all in the same cell. This insight leads to the conjecture that it is useful to derive some kind of normal form for a given instance. An obvious normalization seem to be the replacement of all elements in a cell by only one element, since multiple elements in one cell experience always the same membership. In other words, the normal form only distinguishes occupied cells form unoccupied ones. Applying this normalization step to the instance in Figure 5.3 before executing one of the former two greedy algorithms we obtain an optimal solution for this MMSC instance.

In addition to the general MMSC problem there also exists a geometric version of the problem. In contrast to the general problem, in the geometric version the elements of U are considered to be point in the Euclidean plane and the sets in S are restricted to sets of elements covered by a disk of radius r and centered at a point p. Thereby the radii r of the individual disks do not have to be equal and the points p of the centers are not restricted to be in U. In the following we present three different greedy criteria for the geometric MMSC problem and give lower bounds for their approximation ratios.

The greedy criteria discussed so far do not use any geometric argument and are consequently not only applicable to geometric instances of MMSC. The last criterion we present within the scope of this section takes such a geometric argument into account, namely the size of the uncovered area of a disk. In each step the greedy algorithm, referred to as Greedy$_{Area}$, therefore chooses the disk whose area not already covered by any chosen disk is maximal. Ties are broken arbitrarily. The algorithm comes to a halt when all elements in U are covered by at least one disk. This criterion is in some sense intuitive since the disks successively chosen by the algorithm spread over the whole plane and thus try to reduce the overlap of different disks to a minimum. For good-natured problems Greedy$_{Area}$ actually results in good solutions. However Figure 5.4 depicts a worst-case example since the membership of the solution amounts to $\Omega(n)$ in case of Greedy$_{Area}$, whereas an optimal solution yields only constant membership. This follows from the fact that the algorithm alternatingly chooses disks from the upper and from the lower chain. Consequently, roughly $k/2$ of the disks in the lower chain are in the solution S' in the end, which leads to $M(u, S') \approx k/2$ for Greedy$_{Area}$. However, if we choose all disks from the upper chain and only one from the bottom chain in order to cover node u, we obtain constant interference. Since $k = n - 1$, we can lower-bound the approximation ratio of Greedy$_{Area}$ to $\Omega(n)$.

5.8 Conclusion

Interference reduction in cellular networks is studied in this chapter by means of formalization with the *Minimum Membership Set Cover* problem. To the best of our knowledge this combinatorial optimization problem has not been studied before. We show using approximation-preserving reduction from the Minimum Set Cover problem that MMSC is not only NP-hard, but also that no polynomial-time algorithm can approximate the optimal solution more closely than up to a factor $\ln n$ unless $NP \subset TIME(n^{O(\log \log n)})$. In a second part this lower bound is shown to be asymptotically matched by a randomized algorithm making use of linear programming relaxation techniques. The third part of the chapter discusses the behavior of the algorithm on practical networks. In particular, it shows that the algorithm can be modified to perform well not only in theory but also in practice. Finally, we present two greedy criteria for the general MMSC problem and one for the geometric version of the problem.

Besides the presented results in this chapter, there still exist open questions in the domain of cellular networks in terms of optimizing incoming interference at the clients. Continuing problems that emerged while we were involved with this field include:

- Does a simple greedy algorithm with approximation ratio $O(\log n)$ exist for the general MMSC problem?
- Is there a constant-factor approximation algorithm for the geometric version of the problem?
- Are there any good topology control algorithms for the related *Maximum Set Cover* problem, where each base station should serve approximately the same number of clients?

Chapter 6

Conclusion

Interference reduction in the field of ad-hoc networks is an important aspect of topology control that has been culpably missed out when addressing the issue in the past. Only quite recently [23] and above all [4] addressed the topic, paying it the attention it really deserves.

In this thesis we continue on this path by providing a detailed discussion of minimum interference topologies in the range of the outgoing and the incoming interference model, respectively, introduced in [4]. In the domain of outgoing interference it is shown that currently proposed topology control algorithms do not in the first place focus on reducing interference. We further present an algorithm (LocaLISE) that results in an interference-minimal topology being a spanner of the given network. Additionally, multiple properties of an interference-minimal topology are shown, such as non-planarity, bounded degree, and that it cannot be computed locally.

Incoming interference is considered by means of studying network instances in one dimension. We therefor consider a topology referred to as *exponential node chain*. An algorithm (LION)is proposed, following the scan-line principle, that yields incoming interference $O(\sqrt{n})$ in such a chain and is shown to be asymptotically tight. We then turn our attention towards the more general highway model by describing a generalized version of the above algorithm. However, an instance can be constructed which shows that this does not lead to success. Yet, a greedy algorithm (GLOW) is presented that asymptotically matches the lower bound for exponential node chains and appears to result in low-interference topologies for practical networks.

In the last part of the thesis incoming interference is considered in terms of cellular networks, or strictly speaking, the incoming interference at the clients caused by the base stations of the network. This can be formalized with the *Minimum Membership Set Cover* problem, which is shown to be NP-hard. A randomized algorithm ($\mathcal{A}_{\text{MMSC}}$) is proposed that yields an approximation ratio in $O(\log n)$ with high probability, which is shown to be asymptotically tight. In addition, a modified version of the algorithm is shown to perform well also in practice. Finally, we present three greedy strategies for the MMSC problem and give lower bounds for their approximation ratios.

Bibliography

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks. In *Proc. of the 3rd ACM Int. Symposium on Mobile ad hoc networking & computing (MOBIHOC)*, 2002.

[2] D. Blough, M. Leoncini, G. Resta, and P. Santi. The K-Neigh Protocol for Symmetric Topology in Ad Hoc Networks. In *Proc. of the 4th ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC)*, 2003.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in ad hoc Wireless Networks. In *Proc. of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, pages 48–55, 1999.

[4] M. Burkhart. Analysis of Interference in Ad-Hoc Networks. *Diploma thesis, Distributed Computing Group, Swiss Federal Institute of Technology Zurich*, 2003.

[5] V. Chvátal. *Linear Programming.* W. H. Freeman and Company, 1983.

[6] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*, chapter 24.2. MIT Press, Cambridge, MA, USA, 1990.

[7] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky. Conflict-Free Colorings of Simple Geometric Regions with Applications to Frequency Assignment in Cellular Networks. In 43rd *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[8] U. Feige. A Threshold of ln n for Approximating Set Cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[9] K. Gabriel and R. Sokal. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, 18:259–278, 1969.

[10] T. Hou and V. Li. Transmission Range Control in Multihop Packet Radio Networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.

[11] L. Hu. Topology Control for Multihop Packet Radio Networks. *IEEE Trans. on Communications*, 41(10), 1993.

[12] J. Janssen. Channel Assignment and Graph Labeling. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, chapter 5, pages 95–117. John Wiley & Sons, Inc., 2002.

[13] L. Jia, R. Rajaraman, and R. Suel. An Efficient Distributed Algorithm for Constructing Small Dominating Sets. In *Proc. of the 20th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 33–42, 2001.

[14] D. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[15] N. Karmarkar. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica 4*, pages 373–395, 1984.

[16] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 243–254, 2000.

[17] V. Kawadia and P. Kumar. Power Control and Clustering in Ad Hoc Networks. In *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[18] F. Kuhn and R. Wattenhofer. Constant-Time Distributed Dominating Set Approximation. In *Proc. of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[19] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric Routing: Of Theory and Practice. In *Proc. of the 22nd ACM Symposium on the Principles of Distributed Computing (PODC)*, 2003.

[20] N. Li, C.-J. Hou, and L. Sha. Design and Analysis of an MST-Based Topology Control Algorithm. In *Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

[21] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed Construction of Planar Spanner and Routing for Ad Hoc Wireless Networks. In *Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[22] C. Lund and M. Yannakakis. On the Hardness of Approximating Minimization Problems. *Journal of the ACM*, 41(5):960–981, 1994.

[23] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M. Grünewald. Energy, Congestion and Dilation in Radio Networks. In *Proceedings of the 14th ACM Symposium on Parallel Algorithms and Architectures*, 10 - 13 Aug. 2002.

[24] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[25] L. Narayanan. Channel Assignment and Graph Multicoloring. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, chapter 4, pages 71–94. John Wiley & Sons, Inc., 2002.

[26] R. Prakash. Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks. In *Proc. of the 3^{rd} International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999.

[27] P. Raghavan and C. Thompson. Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica*, 7(4):365–374, 1987.

[28] R. Ramanathan and R. Rosales-Hain. Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment. In *Proc. of the 19^{th} Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000.

[29] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communications*, 17(8), 1999.

[30] A. Srinivasan. Improved Approximations of Packing and Covering Problems. In *Proc. of the 27th ACM Symposium on Theory of Computing*, pages 268–276, 1995.

[31] H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[32] G. Toussaint. The Relative Neighborhood Graph of a Finite Planar Set. *Pattern Recognition*, 12(4):261–268, 1980.

[33] Y. Wang and X.-Y. Li. Localized Construction of Bounded Degree Planar Spanner. In *Proc. of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, 2003.

[34] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *Proc. of the 20^{th} Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1388–1397, 2001.